

Xiaofeng Li · Fan Wang *Editors*

Artificial Intelligence Oceanography

OPEN ACCESS



Springer

Artificial Intelligence Oceanography

Xiaofeng Li · Fan Wang
Editors

Artificial Intelligence Oceanography

 Springer

Editors

Xiaofeng Li
CAS Key Laboratory of Ocean Circulation
and Waves
Institute of Oceanology, Chinese Academy
of Sciences
Qingdao, Shandong, China

Fan Wang
CAS Key Laboratory of Ocean Circulation
and Waves
Institute of Oceanology, Chinese Academy
of Sciences
Qingdao, Shandong, China



ISBN 978-981-19-6374-2

ISBN 978-981-19-6375-9 (eBook)

<https://doi.org/10.1007/978-981-19-6375-9>

© The Editor(s) (if applicable) and The Author(s) 2023. This book is an open access publication.

Open Access This book is licensed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits any noncommercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if you modified the licensed material. You do not have permission under this license to share adapted material derived from this book or parts of it.

The images or other third party material in this book are included in the book's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the book's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

This work is subject to copyright. All commercial rights are reserved by the author(s), whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Regarding these commercial rights a non-exclusive license has been granted to the publisher.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd. The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

The ocean accounts for about 71% of Earth's surface, yet many aspects remain a mystery. Understanding ocean circulation, biogeochemical cycles, and various marine resources directly impact human activities in the 21st century. The enhanced utilization of satellites and autonomous observation platforms has acquired in-situ and remotely sensed data at high spatial and temporal resolutions for the past four decades, entering the Big Ocean Data Era. However, the human capacity to filter, curate, and analyze these data is limited. In the era of big data, efficiently obtaining helpful information from massive data has become a new challenge in oceanographic research.

Artificial intelligence technology has been ubiquitously applied across scientific domains and disciplines and achieved tremendous success. For example, machine learning approaches have been widely used in computer vision, medical, or geophysical fields. Machine learning is an application of artificial intelligence that aims to provide systems to learn from experience without human intervention automatically. With the rapid increase in computing power in recent years, deep learning, a more advanced machine learning technology has begun to show its powers in solving very complex, nonlinear, high-dimensional problems. Promisingly, these artificial intelligence approaches also have enormous potential to improve the quality and extent of ocean research by identifying latent patterns and hidden trends, particularly in large datasets that are intractable using other traditional methods. In addition, the new data-driven and learning-based methodologies may propose novel computationally efficient strategies to improve oceanographic research.

This book brings together state-of-the-art studies on the broad theme of artificial intelligence applications in oceanography, including pure data-driven forecasts, dataset reconstruction, and detection or extraction of oceanic features from remote sensing imagery. The comprehensive contributions clarify the tremendous potential for artificial intelligence technology to contribute to rapid advances in ocean science and may inspire readers of related disciplines. As the editors of this book, we would like to thank all the contributors for their fruitful cooperation and the editorial assistance from Dr. Shuangshang Zhang.

Qingdao, China
December 2021

Xiaofeng Li
Fan Wang

Contents

Artificial Intelligence Foundation of Smart Ocean	1
Xiaofeng Li, Fan Wang, Yuan Zhou, and Keran Chen	
Forecasting Tropical Instability Waves Based on Artificial Intelligence	45
Gang Zheng, Xiaofeng Li, Ronghua Zhang, and Bin Liu	
Sea Surface Height Anomaly Prediction Based on Artificial Intelligence	63
Yuan Zhou, Chang Lu, Keran Chen, and Xiaofeng Li	
Satellite Data-Driven Internal Solitary Wave Forecast Based on Machine Learning Techniques	83
Xudong Zhang, Quanan Zheng, and Xiaofeng Li	
AI-Based Subsurface Thermohaline Structure Retrieval from Remote Sensing Observations	105
Hua Su, Wenfang Lu, An Wang, and Tianyi Zhang	
Ocean Heat Content Retrieval from Remote Sensing Data Based on Machine Learning	125
Wenfang Lu and Hua Su	
Detecting Tropical Cyclogenesis Using Broad Learning System from Satellite Passive Microwave Observations	147
Sheng Wang and Xiaofeng Yang	
Tropical Cyclone Monitoring Based on Geostationary Satellite Imagery	165
Chong Wang, Qing Xu, Xiaofeng Li, Gang Zheng, and Bin Liu	
Reconstruction of $p\text{CO}_2$ Data in the Southern Ocean Based on Feedforward Neural Network	189
Yanjun Wang, Xiaofeng Li, Jinming Song, Xuegang Li, Guorong Zhong, and Bin Zhang	

Detection and Analysis of Mesoscale Eddies Based on Deep Learning 209
Yingjie Liu, Quanan Zheng, and Xiaofeng Li

Deep Convolutional Neural Networks-Based Coastal Inundation Mapping from SAR Imagery: with One Application Case for Bangladesh, a UN-defined Least Developed Country 227
Bin Liu, Xiaofeng Li, and Gang Zheng

Sea Ice Detection from SAR Images Based on Deep Fully Convolutional Networks 253
Yibin Ren, Xiaofeng Li, Xiaofeng Yang, and Huan Xu

Detection and Analysis of Marine Green Algae Based on Artificial Intelligence 277
Le Gao, Xiaofeng Li, Yuan Guo, Fanzhou Kong, and Rencheng Yu

Automatic Waterline Extraction of Large-Scale Tidal Flats from SAR Images Based on Deep Convolutional Neural Networks 287
Shuangshang Zhang, Qing Xu, and Xiaofeng Li

Extracting Ship’s Size from SAR Images by Deep Learning 303
Yibin Ren, Xiaofeng Li, and Huan Xu

Benthic Organism Detection, Quantification and Seamount Biology Detection Based on Deep Learning 323
Yuhai Liu, Yu Xu, Haining Wang, and Xiaofeng Li

Acronyms

AC	Agulhas Current
ACC	Antarctic Circumpolar Current
AE	Anticyclonic Eddy
AHI	Advanced Himawari Imager
AI	Artificial Intelligence
ANN	Artificial Neural Network
AP	Average Precision
ARC	Agulhas Return Current
ARKTOS	Advanced Reasoning Using Knowledge for Typing Of Sea Ice
AVHRR	Advanced Very High Resolution Radiometer
AVISO	Archiving, Validation, and Interpretation of Satellite Oceanographic Data
BCE	Binary Cross-Entropy
Bi-LSTM	Bi-Long Short-Term Memory
BLS	Broad Learning System
BN	Batch Normalization
BP	Back-Propagation
BRNN	Bidirectional Cyclic Neural Network, Bidirectional RNN
CAM	Channel Attention Module
CE	Cyclonic Eddy
CFAR	Constant False Alarm Rate
CNY	Chinese Yuan
CHL	Chlorophyll Concentration
CMA	China Meteorological Administration
CMEMS	Copernicus Marine Environment Monitoring Service
CMW	Cloud Motion Wind
CNN	Convolutional Neural Network
CUDA	Compute Unified Device Architecture
DAU-Net	Dual-Attention U-Net Model
DBN	Deep Belief Networks
DCNN	Deep Convolution Neural Networks

DEM	Digital Elevation Model
DL	Deep Learning
DMSP	Defense Meteorological Satellite Program
DNN	Deep Neural Network
DORS	Deep Ocean Remote Sensing
DT	Dvorak Technique
EAC	East Australia Current
EEI	Earth's Energy Imbalance
EMD	Empirical Mode Decomposition
EMS	Copernicus Emergency Management Service
ENSO	El Niño Southern Oscillation
ENVISAT	Environmental Satellite
EOF	Empirical Orthogonal Function
ESA	European Space Agency
FAI	Floating Algae Index
FAR	False Alarm Ratio
Faster-RCNN	Faster Region-Based Convolutional Network
FCN	Fully Connected Neural
FNN	Feedforward Neural Network
FPN	Feature Pyramid Network
GAN	Generative Adversarial Network
GBDT	Gradient Boosting Decision Tree
GBR	Gradient Boosting Regression
GFO	Geosat Follow On
GLCM	Gray-Level Co-Occurrence Matrix
GMT	Greenwich Mean Time
GPU	Graphic Processing Unit
GRD	Ground Range Detected
GS	Gulf Stream
H-8	Himawari-8
HAB	Harmful Algal Blooms
HED	Holistically-Nested Edge Detection Network
HOG	Histogram of Oriented Gradient
HOV	Human Occupied Vehicle
HR	High-Resolution
HT	Hit Rate
IA	Incident Angle
IBTrACS	International Best Track Archive for Climate Stewardship Dataset
IMS	Ice Mapping System
IoU	Intersection Over Union
IPL	Information Processing Language
IR	Infrared
ISW	Internal Solitary Wave
IW	Interferometric Wide-Swath
JMA	Japan Meteorological Agency

KdV	Korteweg-De Vries Equation
KE	Kuroshio Extension
LBP	Local Binary Pattern
LR	Linear Regression
LR	Low-Resolution
LS	Least-Squares
LSTM	Long Short-Term Memory Neural Network
LWC	Length of Wave Crest
MA	Meridional Average
MAE	Mean Absolute Error
MASIE-NH	Multi-sensor Analyzed Sea Ice Extent-Northern Hemisphere
MBR	Minimum Bounding Rectangle
MIMS	Multi-Year Ice Mapping System
MLD	Depth of The Mixed Layer
MLE	Mean Location Error
MLFN	Multilevel Fusion Network
MLFrnn	Multi-Layer Fusion Recurrent Neural Network
MODIS	Moderate Resolution Imaging Spectroradiometer
MRF	Markov Random Field
MSE	Mean Square Error
MSSE	Mean Scaled Square Error
MSW	Maximum Sustained Wind Speed (2-minute average wind speed)
MYI	Multi-Year Ice
NCC	Normalized Chromaticity Coordinates
NCDC	National Climatic Data Center
NCEI	National Centers for Environmental Information
NDVI	Normalized Difference Vegetation Index
NH	Northern Hemisphere
NIC	National Ice Center
NLP	Natural Language Processing
NN	Neural Network
NOAA	National Oceanic and Atmospheric Administration
NRCS	Normalized Radar Cross Section
NSIDC	National Snow and Ice Data Center
OHC	Ocean Heat Content
OISST	Optimum Interpolation Sea Surface Temperature
OLCI	Ocean and Land Color Instrument
OLR	Outgoing Longwave Radiation
OPEN	Ocean Projection and Extension Neural Network
PAM	Position Attention Module
PCNN	Pulse-Coupled Neural Network
PD	Propagation Direction
POD	Probability Of Detection
PP	Peak to Peak
PS	Propagation Speed

RBB	Rotatable Bounding Box
ReLU	Rectified Linear Unit
RF	Random Forest
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
ROV	Remotely Operated Vehicle
RPN	Regional Proposal Network
RRMSE	Relative Root-Mean-Square Error
RSRs	Radial Sand Ridges
SAM	Southern Annular Mode
SAR	Synthetic Aperture Radar
SARCFMNet	SAR Coastal Flooding Mapping Network
SCS	South China Sea
SH	Southern Hemisphere
SIFT	Scale Invariant Feature Transform
SLA	Sea Level Anomaly
SM	Stripmap
SNAP	Sentinel Application Platform
SOM	Self-Organization Map
SRN	Simple Recurrent Network
SS	Subsurface Salinity
SSA	Subsurface Salinity Anomaly
SSDD	SAR Ship Detection Dataset
SSH	Sea Surface Height
SSHA	Sea Surface Height Anomaly
SSM/I	Special Sensor Microwave Imager
SSS	Sea Surface Salinity
SST	Sea Surface Temperature
SSW	Sea Surface Wind
ST	Subsurface Temperature
STA	Subsurface Temperature Anomaly
SVM	Support Vector Machine
SVR	Support Vector Regression
SWOT	Surface Water and Ocean Topography
TC	Tropical Cyclone
TCC	Tropical Cloud Cluster
TIW	Tropical Instability Wave
TP	Topex/Poseidon
UN	United Nations
WBC	Western Boundary Current
WOA	World Ocean Atlas
XGBoost	Extreme Gradient Boosting
YOLO	You Only Look Once

Artificial Intelligence Foundation of Smart Ocean



Xiaofeng Li, Fan Wang, Yuan Zhou, and Keran Chen

1 The Development of Artificial Intelligence

Artificial intelligence (AI) is the core driver for the fourth technological revolution, following the revolutions in steam technology, electricity technology, and computers and information technology. Since its emergence in the 1950s, AI has fully improved productivity, affected and changed the production structure and production relations. Understanding the history of AI plays an indispensable role in the subsequent research and the development of AI technologies. AI can be divided into three generations, according to the difference in the drive mode. The subsequent subsections introduce each of these three generations of AI.

1.1 The First-Generation AI

Turing proposed the “Turing Test” in 1950 [49]. It states that if a machine can answer a series of questions posed by a human tester within five minutes, and more than 30% of its answers can deceive the tester into thinking that they are answered by a human, then the machine can be considered intelligent. In the same year, Turing predicted the feasibility of intelligent machines. The “Turing test” can be seen as the genesis of AI. Newell and Simon [41] developed the first heuristic program in the world: Logic Theorist. It successfully proved 38 theorems in the book: “Principles of Mathematics”, by simulating human thinking activities. This program successfully

X. Li (✉) · F. Wang

CAS Key Laboratory of Ocean Circulation and Waves, Institute of Oceanology,
Chinese Academy of Sciences, Qingdao 266071, China
e-mail: xiaofeng.li@ieee.org

Y. Zhou · K. Chen

School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China

© The Author(s) 2023

X. Li and F. Wang (eds.), *Artificial Intelligence Oceanography*,
https://doi.org/10.1007/978-981-19-6375-9_1

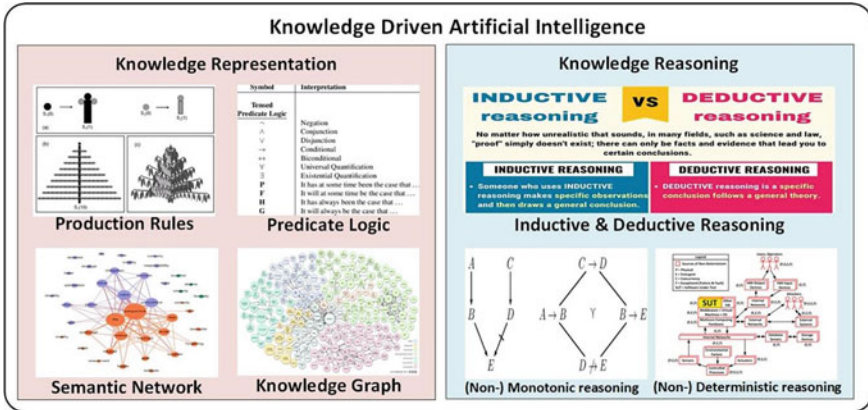


Fig. 1 The research field of the first generation of artificial intelligence. The first generation of AI is knowledge-driven AI, which mainly conducts research on knowledge representation and reasoning. Production rules, predicate logic, semantic network and knowledge graph are common knowledge representations. Inductive reasoning, deductive reasoning, Monotonic reasoning and deterministic reasoning are mainstream reasoning methods

demonstrated the feasibility of the predictions posed by Turing, and it is considered the first successful AI program. In August of the same year, the concept of “artificial intelligence” was first introduced by John McCarthy, Herbert Simon, and a group of scientists from different fields at Dartmouth College. Thus, AI stands on the stage of history as an independent discipline. Newell and Shaw [40] invented the first AI programming language, the information processing language (IPL). It used symbols as basic elements and proposed a reference table structure instead of storing addresses or arrays. McCarthy [36] developed a list processing language based on the IPL, which was widely used in the AI community.

The first generation of AI is known as knowledge-driven AI; these AIs allow machines to learn by imitating the process of human reasoning and thinking. As shown in Fig. 1, the core steps can be divided into two parts, knowledge representation and knowledge reasoning.

Knowledge representation is required to allow machines to achieve intelligent behavior. It represents human-understood knowledge in a certain data structure that allows machines to understand and complete the processing. The methods of knowledge representation include predicate logic, production rules, semantic network representation, and knowledge graphs.

Predicate logic can describe how the human mind works. From the logical system, propositional logic is the simplest logical system, and it is used to describe declarative sentences using truth values. For example, “the sea is blue”; the “true” and “false” of each proposition is called the truth value. Propositions can be divided into atomic propositions and positions. An atomic proposition is a proposition that cannot be split into simpler declarative sentences. Compound propositions are detachable propositions consisting of atomic propositions and connectives. However, they both have

limited expressive power and can only represent established facts. Thus, predicate logic is developed based on propositional logic. It uses connectives and quantifiers to describe objects, and predicates on objects to represent the world. Predicates of objects refer to the properties of objects or the relationships between objects. A constant symbol, a predicate symbol and a function word comprise a predicate. Constant symbols represent objects and predicate symbols represent relationships or attributes. For example, the constant symbol Susan, the predicate symbol mother, and the function word nurse compose the predicate logic “nurse(mother(Susan))”, which indicates that the mother of Susan is a nurse. Predicate logic representation has certain advantages: naturalness, accuracy, rigor, and ease of implementation. However, it cannot represent uncertain knowledge, and when it is used to describe too many things, it becomes inefficient.

Production rules are used to describe the cause-effect relationships between things. For example, if an animal is a mammal and has a long trunk, then the animal is an elephant. The generative system consists of a rule base, comprehensive database, and control system. The rule base is used to represent a set of rules for inferring conclusions from premises. The database is used to store known conditions, intermediate results, and final conclusions. The control system is used to select suitable rules for inference from a rule base.

Semantic network representation is a network graph that represents knowledge through entities and their semantic relationships. It consists of nodes and arcs. Nodes represent entities, which are used to describe various things, concepts, situations, attributes, states, events, actions, etc., and arcs represent semantic relations, such as the instance relations, classification relations, membership relations, attribute relations, inclusion relations, temporal relations, location relations, etc. These basic units are interconnected to form a semantic network.

A knowledge graph is essentially a semantic network that reveals the relationships between entities, allowing a formal description of entities and their interrelationships in the objective world. The study of knowledge graphs originated from a semantic web. Tim Berners Lee [2] proposed the concept of a semantic web at the XML Conference in 2000, expecting to provide services such as information proxy, search proxy, and information filtering by adding semantics to web pages. In 2005, Metaweb was established in the United States to develop an open knowledge base for web semantic services. It extracts entities (people or things) in the real world and relationships between them based on public datasets such as Wikipedia and the United States Securities and Exchange Commission (SEC), and then stores them with a graph structure on a computer. In 2010, Google acquired MetaWeb and acquired its semantic search technology. In 2012, Google formally put forward the concept of a knowledge graph, aiming to improve the capability of the search engine and enhance the search experience of users based on knowledge graphs.

Reasoning is a form of thinking that logically derives new conclusions from known premises. Knowledge reasoning represents the process of using knowledge, which is one of the core issues in AI research. It uses previous knowledge to derive conclusions by reasoning, and solves the corresponding problems. Reasoning can be divided into

deductive reasoning and inductive reasoning, depending on how the conclusion is derived. Where deductive reasoning is a reasoning process from general to special, inductive reasoning is a reasoning process from special to general. Reasoning can also be divided into monotonic and nonmonotonic reasoning, depending on whether the conclusions derived in the reasoning process increase monotonically. Monotonic means that the number of propositions known to be true strictly increases as the reasoning progresses. According to the certainty of reasoning, reasoning can also be divided into deterministic and non-deterministic reasoning, where deterministic means that the knowledge used in reasoning and the conclusions derived are either true or false, whereas non-deterministic means that the knowledge used in reasoning and the conclusions derived are probabilistic.

In the era of knowledge-driven AI, the emergence of expert systems has brought AI into a period of vigorous development. An expert system is an intelligent computer program that introduces the knowledge of a specialized field. Through knowledge representation and reasoning, it can simulate the decision-making process of human experts to solve the problems in this field and provide suggestions for the users. The first expert system was DENDRAL [4], developed by Feigenbaum in 1968. It was used to analyze the molecular structure of organic compounds by mass spectrometry. In the 1970s, the idea of expert systems was gradually accepted. A series of expert systems were developed to solve problems in different fields at that time, such as MYCIN [46] for diagnosis and treatment of blood infection diseases, MACSYMA [35] for symbolic integration and theorem proving, and PROSPECTOR [14] for seismic exploration. Subsequently, the application field of expert systems expanded rapidly, and the difficulty of dealing with problems increased continuously. Several tool systems for building and maintaining expert systems have been developed. In the 1980s, the development of expert systems gradually became specialized, creating huge economic benefits.

Although scientists at that time had great expectations for knowledge-driven AI, there were some fundamental problems in its development. The first problem was the interaction problem. Traditional methods could only simulate the thinking process of human beings, but could not simulate the complex interactions between humans and the environment. The second problem was the expansion problem. Traditional methods were only applicable to the development of expert systems in specific fields, and could not be extended to complex systems with larger scales and wider fields. The third problem was the application problem. Research on traditional methods was detached from the mainstream computing (software and hardware) environment, which seriously hindered the practical application of expert systems. Constrained by the above mentioned problems, the first generation of AI eventually declined.

1.2 The Second-Generation AI

First-generation AI is based on symbols, which believe that sensory information is expressed in a certain encoding way. Second-generation AI establishes a stimulus-response connection in a neural network. This ensures the generation of intelligent behavior through connections. Figure 2 illustrates the development of second-generation AI

In 1958, Rosenblatt established the prototype of an artificial neural network, the perceptron, which followed the idea of connectionism. The perceptron was inspired by two aspects. One was the neuron mathematical model proposed by McCulloch and Pitts in 1943 [37]: the threshold logic circuit, which converted the input of neurons into discrete values. The second was from the Hebb learning rate proposed by D.O.Hebb in 1949 [23], that is, the neurons fired at the same time are connected.

In 1969, Minsky and Papert [38], pointed out that perceptron could only solve linearly separable problems. In addition, it was not practical when the number of hidden layers increased as it lacked an effective learning algorithm. The criticism of the perceptron posed by Minsky proved to be fatal; thus, second-generation AI declined for more than 10 years.

Regarding difficulty, through the joint efforts of many scholars, significant progress has been made in both neural network models and learning algorithms. In addition, mature theories and technologies have gradually formed over the past 30 years.

For example, the gradient descent method was proposed by the French mathematician Cauchy [5]. The method is used to solve the minimum value along the direction of gradient descent or to solve the maximum value along the direction of the gradient rise.

Another example is the back-propagation (BP) algorithm [43]. The algorithm consists of a forward propagation process and a BP process. In the forward propagation process, the input information enters the hidden layer through the input layer. Then, the information is processed layer by layer and passed to the output layer.

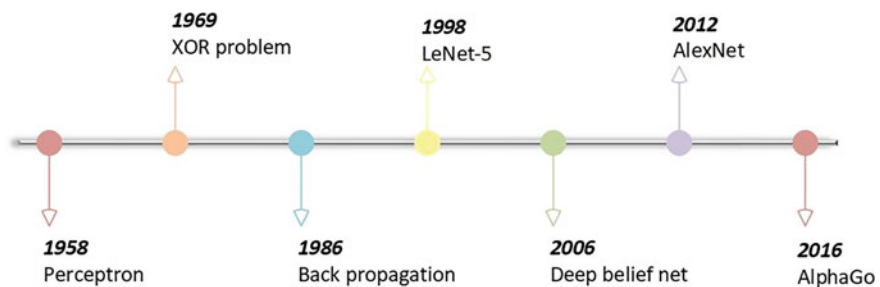


Fig. 2 The development history of the second generation of artificial intelligence. The second-generation AI has developed since the advent of the perceptron in 1958. The birth of AlphaGo in 2016 entered a period of rapid development

If the desired output value cannot be obtained in the output layer, the sum of the squares of the error between the output value and the expected value is taken as the objective function. In the BP process, the partial derivative of the objective function to the weight of each neuron is obtained layer by layer as the basis for modifying the weight. The learning of the network is completed using the weight modification process. When the error reaches the expected value, the network learning ends.

Regarding the loss function, a series of improvements have been made, such as the cross-entropy loss function [28]. Cross entropy is an important concept in Shannon's information theory, and is mainly used to measure the difference in information between two probability distributions. The performance of a language model is typically measured using cross-entropy and complexity. Cross-entropy means the difficulty of text recognition within a model or the average number of bits used to encode each word from a compression viewpoint. Complexity means to use the model to represent the average number of branches in this text. Cross entropy is introduced to the neural network field as a loss function. We used p to represent the distribution of the true markers and q to represent the predicted marker distribution of the trained model. The cross-entropy loss function measures the similarity between p and q .

Algorithm improvements, such as regularization methods, prevent over-fitting [52]. Regularization involves imposing constraints that minimize the empirical error function. Such constraints introduce prior distributions to the parameters and have a guiding effect. When optimizing the error function, they tend to choose the direction that reduces the gradient and satisfies the constraints; hence, the final solution tends to conform to prior knowledge. At the same time, regularization solves the ill-posedness of the inverse problem. The resulting solution exists uniquely and depends on the data. The influence of noise on the ill-posed is weak. If the regularization is appropriate, the solution will not overfit, even if the number of uncorrelated samples in the training set is small.

New network architectures have been developed, such as convolutional neural networks (CNNs) [13], recurrent neural networks (RNNs) [33], long short-term memory neural networks (LSTM) [25], and deep belief networks (DBN) [24].

CNNs are a type of feedforward neural network (FNN) that includes convolution calculations and has a deep structure. A CNN has the abilities of representation learning and can perform the shift-invariant classification of the input information according to its hierarchical structure. A CNN is constructed by imitating the biological visual perception mechanism, which can perform both supervised and unsupervised learning. The convolution kernel parameter sharing in the hidden layer and the sparsity of inter-layer connections enable the CNN to perform smaller calculations to extract features.

An RNN is a type of recursive neural network in which all node cyclic units are connected in a chain. RNNs have applications in natural language processing (NLP) fields, such as speech recognition, language modeling, machine translation, and other fields. An RNN can be combined with a convolution operation to handle computer vision problems.

The LSTM network is a time RNN, which is specifically designed to solve the long-term dependence problem in general RNNs. When receiving new input infor-

mation, the network first forgets all the long-term information that it does not require. Afterward, it learns which part of the new input information has use value and saves them in long-term memory. Finally, the network learns which part of the long-term memory can work immediately.

A DBN is a deep neural network with multiple hidden layers. It is a combination of unsupervised feature learning and supervised parameter adjustment. It performs unsupervised greedy learning using stacked restricted Boltzmann machines to extract high-level and abstract features from the original data. It uses a BP neural network to reversely fine-tune the parameters to realize the supervised learning of data.

Together, these works ushered in a new era of second-generation AI based on deep learning. Owing to the universality of deep neural networks, these networks can approximate any function. Therefore, using deep learning to determine the function of the data has a theoretical guarantee.

In 2014, deep learning was pointed out to be vulnerable to spoofing and attacks. Owing to the uncertainty of observation and measurement data, the acquired data must be incomplete and contain noise. In this case, the choice of the neural network structure is extremely important. If the network is too simple, there is a risk of underfitting; if it is complicated, overfitting occurs. Although the risk of overfitting can be reduced to a certain extent through various regularization methods, it will inevitably lead to a serious decline in the promotion ability if the quality of the data is poor.

1.3 The Third-Generation AI

The third generation of AI needs to solve the shortcomings of the first and second generations of AI. To establish sound AI theory, developed AI technology must be safe, credible, reliable, and scalable. Only when the above conditions are met can a real technological breakthrough be achieved, which will produce new innovative applications of AI. The best current approach is an organic combination of the first-generation knowledge-driven approaches and the second-generation data-driven approaches. The combined use of knowledge, data, algorithms, and arithmetic power results in a more powerful AI.

Research on third-generation AI must move towards making AI capable of powerful knowledge and reasoning. For this purpose, we need to draw on classic examples, such as the Watson conversational system, which was introduced in 2011. The following lessons on knowledge representation and inference methods from this system are worth learning. First, automatic generation of structured knowledge representations from a large amount of unstructured text. Second, a method for representing knowledge uncertainty based on the knowledge quality scoring. Third, an approach based on multiple reasonings to achieve the uncertainty reasoning. The development of third-generation AI requires strong knowledge and reasoning capabilities, in addition to strong perception. There have been some tentative efforts to apply the principle of sparse discharge to the computation of ANN layers [26]. Specific-

cally, the network is trained with simple background images, such as “human,” “car,” “elephant,” and “bird”, as training samples. The neurons representing these “categories” appear in the output layer of the neural network. The network responds to the contours of the human face, car, elephant, and bird. In this way, the semantic information of the “whole object” is extracted, thus making the neural networks perceptive. However, this approach can only extract part of the semantic information, it cannot extract different levels of semantic information; therefore, further research is needed. Furthermore, the third generation of AI also needs to interact with the environment. Reinforcement learning has made good progress in many areas, such as video games [39, 50], board games [47, 48], robot navigation and control [11, 44], and human-computer interaction. In some tasks, the performance of reinforcement learning approaches used in the networks even surpasses that of humans.

Attempts on the third generation of AI have emerged in the academic community. Zhu et al. [56] proposed the use of a triple-space fusion model, that is, a model that fuses both dual-space and single-space approaches. The model has the opportunity to be interpretable and robust. When the model can convert sensory signals such as vision and hearing into symbols, the machine has the opportunity to develop comprehension capabilities, which will help solve the problem of interpretability and robustness of the model. If the symbols in the machine can be generated by the perception of the machine, then the symbols and symbolic reasoning can generate intrinsic semantics, which can hopefully solve the problem of interpretability and robustness of machine behavior at the root. Among the models proposed by Zhu et al. [56], the single-space model is based on deep learning and suffers from being uninterpretable and having poor robustness. The dual-space model mimics the working mechanism of the brain; however, the model has some uncertainties. For example, the machine can establish “intrinsic semantics” through the reinforcement learning of the environment, yet it is not certain whether these semantics are consistent with the “intrinsic semantics” acquired by humans through perception. Therefore, there are many uncertainties associated with this approach. Despite these difficulties, we still believe that machines that take steps in this direction will edge closer to true AI. The single-space model is based on a deep learning algorithm that fully uses the computational power of the computer. In some aspects, it already outperforms humans. However, there are many uncertainties in this approach, and it is still unknown how much progress can be made through algorithmic improvements. Therefore, to achieve the goal of third-generation AI, the best strategy is to simultaneously advance along two lines, namely, the convergence of the three spaces. This will maximize the working mechanism of the brain and make full use of the computing power of the computer. The combination of the two approaches is expected to lead to a more powerful AI.

The third generation of AI is a new form of AI driven by data and knowledge in concert. It fits perfectly with our need to explore the ocean using the ocean and its data. This new form of AI technology will be a powerful tool for people to understand the ocean and further develop it. The combination of AI technology and research in the ocean field will open a new chapter in oceanographic research.

At present, third-generation AI is still in its initial stage, and the AI used in academia and industry is mainly second-generation. Among the second-generation AI technologies, deep neural network-based AI technologies are the most representative. Therefore, the rest of this paper introduces common neural network structures and their applications.

2 The Architecture of Deep Neural Networks

2.1 Deep Feedforward Neural Network

A deep FNN is a typical deep learning model that can be viewed as a mathematical function. It realizes the complex mapping from input to output using a combination of nonlinear functions. The following section introduces the neuron algorithm, BP algorithm, single-layer FNN, and multi-layer FNN.

2.1.1 Neuron

A biological neuron usually contains multiple dendrites and axons. Dendrites are used to receive signals transmitted from other neurons, and the axon has multiple endings for transmitting signals to other connected neurons. Psychologist McCulloch and mathematician Pitts proposed an abstract neuronal model based on biological neurons in 1943 [37].

A complete neuron can be viewed as a computational process of “input, numerical computation, and output.” The “numerical computation” consists of “a linear part and a non-linear part.” Figure 3 shows a typical biological neuron and neuron model.

This neuron contains three inputs, a_1 , a_2 , a_3 , and one output. By assigning the corresponding weights w_1 , w_2 , and w_3 to each input, the output of the linear part can be expressed as

$$z = a_1 * w_1 + a_2 * w_2 + a_3 * w_3 + b \quad (1)$$

where b is the bias term.

The nonlinear part is implemented by the activation function, and the output of the neuron can be expressed as

$$y = f(z) \quad (2)$$

where $f(\cdot)$ is the activation function.

The activation function is a nonlinear mapping relationship and must adhere to the following three conditions:

- (1) It must be a continuous and differentiable (can be non-differentiable at finite points) nonlinear function.

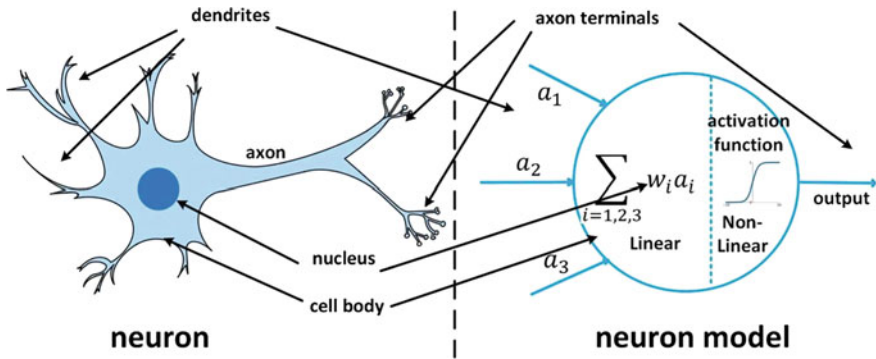


Fig. 3 Comparison of a biological neuron and neuron model. The left side shows a biological neuron. It consists of a nucleus, a cell body, an axon, dendrites and axon terminals. The right side shows a neuron model. It can be viewed as a computational process of “input, numerical computation, and output”

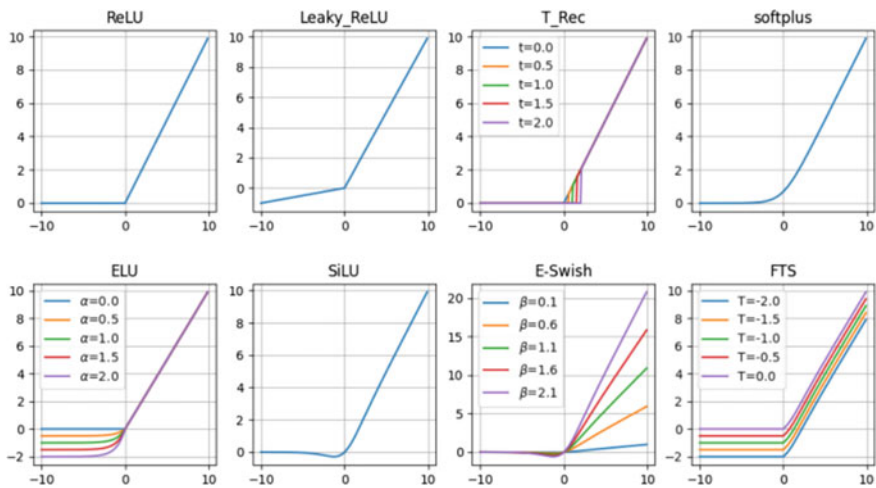


Fig. 4 Function diagrams of some common activation functions

- (2) The activation function and its derivative should be simple; overly complex functions are not conducive for network efficiency.
- (3) The value domain of the derivative should be limited to a suitable interval; this is beneficial for improving network efficiency.

Common activation functions are shown in Fig. 4.

In a neural network, the process of the input going through the neuron to compute the output is called a forward propagation algorithm.

2.1.2 Backpropagation Algorithm

It is not enough to know only the neuron in neural networks because the parameters w and b need to be learned. Therefore, it is necessary to introduce a BP algorithm to update the parameters.

In supervised learning, we have a dataset containing the inputs and the corresponding outputs. We call the correct output a label. The purpose of training a neural network is to learn the correct mappings of inputs to outputs.

First, we assign random values to the parameters w and b , and generate the predicted values using the forward propagation algorithm. We define a cost function $J(w, b)$ to represent the closeness of the predicted output to the label. We use cross-entropy as the cost function in classification problems where the output is a discrete variable, and the mean-squared error is used as the cost function in regression problems where the output is a continuous variable.

The optimization formulas for the cross-entropy cost function and the mean-squared error cost function are expressed as follows:

$$\min J(w, b) = - \sum_{i=1}^K y_i \log p_i \quad (3)$$

$$\min J(w, b) = \frac{1}{K} \sum_{i=1}^K (y_i - p_i)^2 \quad (4)$$

where K is the number of samples, y_i is the label value of the i -th sample, and p_i is the predicted value of the i -th sample.

Thus, the objective is transformed into solving for the parameter values that minimize the cost function $J(w, b)$. The gradient descent algorithm is generally used to solve optimization problems. The gradient indicates that the directional derivative of a function at that point has a maximum value along that direction. The specific process is as follows: it first randomly selects a set of parameter values. Then, it computes the predicted output and the cost function $J(w, b)$, and computes the gradient of the cost function $J(w, b)$ on w and b . Finally, we use the gradient to update w and b so that the cost function takes the minimum value. The modified expressions for the parameters w and b are expressed as:

$$w := w - \alpha \frac{\partial J(w, b)}{\partial w} \quad (5)$$

$$b := b - \alpha \frac{\partial J(w, b)}{\partial b} \quad (6)$$

where α is the learning factor and denotes the step length of each gradient descent.

The process of updating the parameters w and b along the gradient direction to minimize the cost function is called the gradient descent algorithm.

The BP algorithm is a specific implementation of the gradient descent method on deep neural networks. Owing to the deepening of the network layers, the gradient of the parameters of each layer must be computed from backward to forward for the cost function and be continuously updated.

2.1.3 Single-layer Feedforward Neural Network

The FNN is the most common type of neural network in which neurons are arranged in layers. Each layer has several neurons, and each neuron is connected only to the neurons in the previous layer. Neurons in each layer only receive the input signal from the previous layer and output the processed signal to the neurons in the next layer. The first layer is called the input layer and the last layer is called the output layer. The remaining intermediate layers are called hidden layers; these can be one or more layers. The signal enters the network from the input layer and is transmitted layer by layer to the output layer. There is no feedback in the entire network, and the output does not affect the network model or network input. This section depicts a single-layer FNN containing one hidden layer as an example; this will allow us to introduce the basic principles of FNNs.

Figure 5 shows the general structure of a single-layer FNN, where $x_m^{(n)}$ denotes the n -th feature of the m -th sample, and the number of neurons in the input layer is the same as the number of features of the sample. $y_m^{(k)}$ denotes the k -th output of the m -th sample and k is the number of neurons in the output layer, where $k \geq 2$ for the classification problem and $k = 1$ for the regression problem.

The learning process of single-layer FNNs consists of forward propagation and BP; this is illustrated below with a simple network in Fig. 6. The input is propagated forward along the direction of the network structure to the output layer, and then the weights and bias are updated by the BP algorithm.

In the forward propagation, information is propagated from the input layer to the output layer after being processed by the hidden layer, and the state of the neurons in each layer only affects the state of neurons in the next layer. Assuming that the input layer is the 0-th layer, the output of the neurons in the hidden layer and the output layer can be expressed as:

$$h_1^{(1)} = \varphi^{(1)}(x_m^{(1)} * w_1^{11} + x_m^{(2)} * w_1^{21} + x_m^{(3)} * w_1^{31} + b_1^{(1)}) \quad (7)$$

$$h_2^{(1)} = \varphi^{(1)}(x_m^{(1)} * w_1^{12} + x_m^{(2)} * w_1^{22} + x_m^{(3)} * w_1^{32} + b_2^{(1)}) \quad (8)$$

$$y = \varphi^{(2)}(h_1^{(1)} * w_2^{11} + h_2^{(1)} * w_2^{21} + b_1^{(2)}) \quad (9)$$

where w_j^{ij} denotes the connection weight of the i -th neuron in layer $l - 1$ to the j -th neuron in layer l , $b_j^{(l)}$ denotes the bias used to compute the linear weighted summation

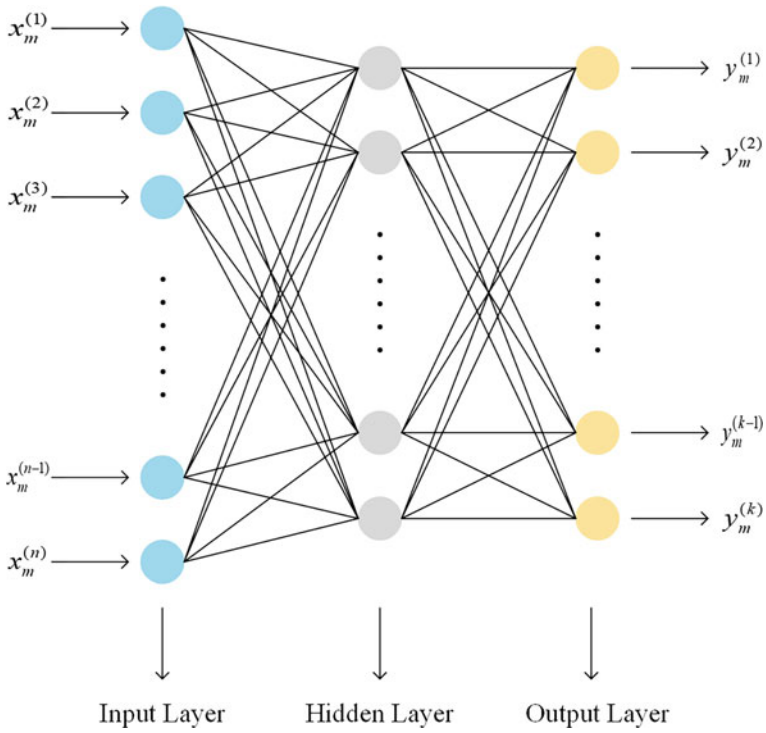


Fig. 5 The general structure of single-layer feedforward neural networks. The blue, gray, and yellow neurons form the input layer, hidden layer, and output layer of the neural network, respectively

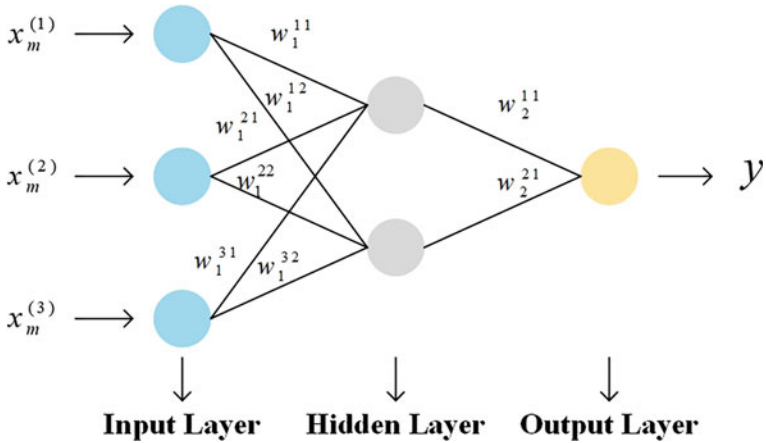


Fig. 6 A simple single-layer feedforward neural network

of the j -th neuron in layer l , $\varphi^{(l)}$ denotes the activation function of the neuron in layer l , $h_j^{(l)}$ denotes the output of the j -th neuron in the hidden layer, and y denotes the output of the neuron in the output layer.

In BP, the gap between the network output and the real value is first calculated; this is called the loss function. Subsequently, the gradient of the parameters, such as the weight and bias term, is calculated, and the parameters are updated by the gradient descent to minimize the loss function. The learning process of the network is continuously iterated to optimize the network model until the loss function is sufficiently small or the maximum number of learning times is reached.

When the structure and weights of a neural network are determined, the network forms a nonlinear mapping from the input to the output. For a single-layer FNN, if the number of neurons in the hidden layer is large enough, it can approximate any continuous function on a bounded region with arbitrary accuracy and can solve complex nonlinear classification tasks well.

2.1.4 Multi-layer Feedforward Neural Network

A network that contains multiple hidden layers between the input and output layers is called a multi-layer FNN, which can also be called a deep FNN. Depth refers to the number of layers in a neural network model. The greater the number of layers designed, the greater the depth and complexity of the model. A deep FNN has better classification and memory ability, as well as a stronger function fitting ability. It can handle more complex data structures or data whose structures are difficult to predefine. Figure 7 shows a neural network containing four hidden layers.

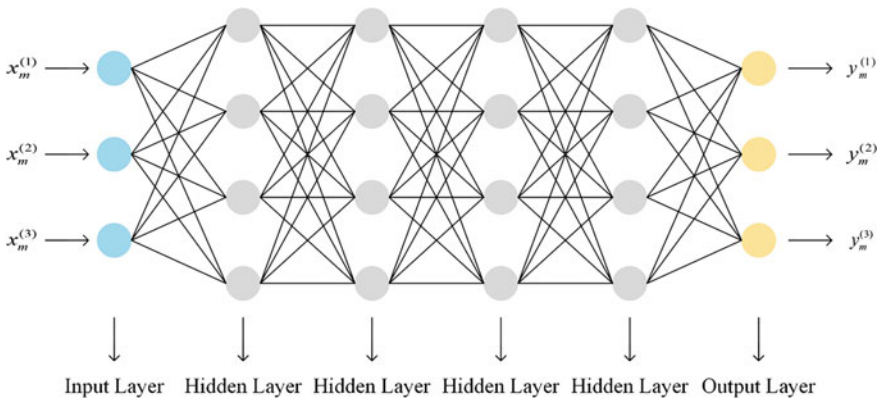


Fig. 7 A multi-layer feedforward neural network containing four hidden layers

The output of the network can be expressed as follows:

$$z^{(l)} = W^{(l)} * a^{(l-1)} + b^{(l)} \quad (10)$$

$$a^{(l)} = f_l(z^{(l)}) \quad (11)$$

where $f_l(\bullet)$ denotes the activation function of neurons in layer l , $W^{(l)}$ and $b^{(l)}$ denote the weight matrix and bias from layer $l - 1$ to layer l , respectively, $z^{(l)}$ and $a^{(l)}$ denote the net input and output of neurons in layer l .

Deep FNN also uses a BP algorithm to optimize the network model. The representation ability of the network is greatly enhanced because of the increase in the number of layers and parameters. However, the network is prone to overfitting; that is, small errors can be obtained from the training data, and large errors are obtained from the test data. Therefore, appropriate regularization techniques need to be applied to improve the generalization ability of the network model.

Deep FNNs are the basis of many AI applications. They can perform complex data processing and pattern recognition tasks. However, they cannot process image data well. The networks require a large number of neurons and parameters when processing the image data; with high computational ability requirements and low computational efficiency, these networks are prone to overfitting.

2.2 *Deep Convolutional Neural Network*

The biggest advantage of CNNs over FNNs is the reduction of parameters. This allows researchers to build and design larger models to solve complex problems. For example, a picture in jpg format with a resolution of 480×480 is represented in the computer as a $480 \times 480 \times 3$ tensor, and the three dimensions correspond to the height, width, and the number of channels of the 3D tensor. If this image data is fed into an FNN, each neuron in the first hidden layer in this network needs to be connected to 691,200 ($480 \times 480 \times 3$) tensor elements. The number of parameters required for one of these neurons is over 600,000. Thus, the number of single-layer network neurons required for FNNs to deal with complex problems is enormous. Therefore, the hidden layer of the FNN requires a large number of parameters to extract the tensor features. This fully connected mechanism of FNNs is inefficient in handling large input data. Compared with the fully connected layer of the FNN, the convolutional layer requires fewer parameters to extract the tensor features. CNNs are widely used in many fields because of their efficient features.

This section contains four subsections. Section 2.2.1 introduces the mechanism of the CNNs and their basic structures. Section 2.2.2 introduces the mechanism of full CNNs. Section 2.2.3 introduces typical CNN structures. Section 2.2.4 introduces the problems and shortcomings of deep convolutional networks.

2.2.1 Mechanism of Convolutional Neural Network

CNNs obtain their names from the mathematical linear operations between the matrices called convolutions. A CNN is a representation learning method with a multilayer structure, which mainly consists of a convolutional layer, pooling layer, and a fully connected layer. The convolutional and fully connected layers contain parameters, whereas the pooling layer does not. As shown in Fig. 8, the image is input to the CNN and then passes through the convolutional and pooling layers alternately, which flattens the image features into a feature vector of dimension one. The CNN finally outputs the result through the fully connected layer. The convolutional and pooling layers are equivalent to feature extraction structures, which are used to extract features from the input tensor. The fully connected layer is equivalent to a classifier, which is used to classify the flattened feature vector.

As the name implies, the convolutional layer is the most important operation in a CNN, and the parameters of the convolutional layer are mainly located in the convolutional kernel. The convolution kernel is usually represented by a small-size tensor that only acts on a local region within the space of the input tensor.

Taking the most widely used two-dimensional convolution as an example, the specific process of the convolution operation is shown in Fig. 9a. The convolution operation selects all local regions in the spatial dimension of the input features that are consistent with the size of the convolution kernel. The calculation is shown in Fig. 9a. The input tensor shares the convolution kernel parameters in the channel dimension when performing the convolution operation. The convolution operation is a three-dimensional inner product operation between the shared convolution kernel parameters along the channel and the features of the same spatially localized region on different channels. The corresponding result is a scalar of the center position of the corresponding spatially localized region. The output corresponding to a single convolution kernel is a two-dimensional feature map, and the number of convolution kernels in the convolution layer is the same as the number of channels in the output tensor.

There is some degree of information overlap between the output results of the convolutional layer and its neighboring outputs in the spatial dimension. This may lead to information redundancy. As shown in Fig. 9b, the input feature map size of the convolution operation is 7×7 . If the selected convolution kernel size is $3 \times$

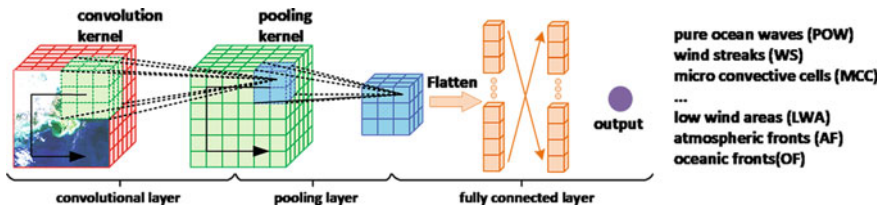


Fig. 8 Overview of the convolutional neural network architecture. The architecture mainly consists of several convolution layers, pooling layers and full connection layers

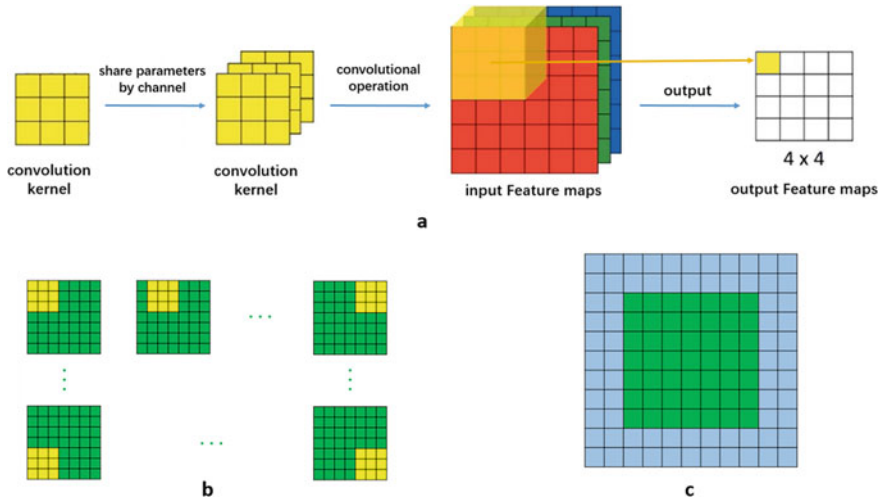


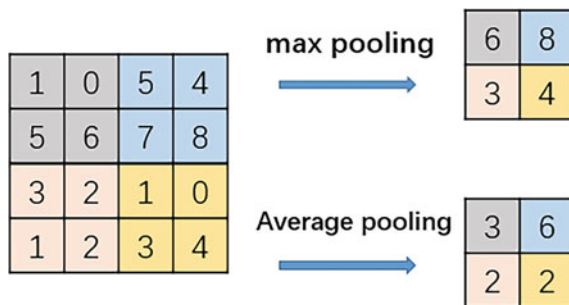
Fig. 9 Illustration of Convolution Implementation Mechanism. **a** Convolution kernel parameters are shared in the input tensor channel dimension. **b** The sliding mechanism of the convolution kernel in the dimension of the input tensor space. **c** Padding operation in the process of convolution implementation

3, then there will be an overlap of six elements between two adjacent convolution operations. If the step size parameter is set to two, there will be an overlap of three elements between two adjacent convolution operations. The size of the output feature map of the convolution operation is 3×3 . A reasonable increase in the step size reduces the overlap between adjacent convolution operations and reduces the size of the output feature.

Convolution operations lose information about edge features, which is an inherent drawback of convolution operations. A simple and effective method is padding. As shown in Fig. 9c, if the size of the convolution kernel is 5×5 and the padding parameter is set to two, the size of the output feature map of the convolution operation can be kept consistent with the input to avoid the loss of the edge information.

Although the convolutional layer can serve to reduce the number of connections between the output neurons and the input features, the number of neurons is not significantly reduced during the execution of the convolutional operation, and the feature dimensionality remains high. To make the model easier to optimize, pooling layers are introduced to the CNN. The pooling layer is often located between two convolutional layers and is designed to perform feature selection and reduce the dimensionality of the feature mapping. There are two mainstream pooling approaches: maximum pooling and average pooling. As shown in Fig. 10, a 2×2 filter is used on the input features with a spatial dimension of 4×4 and slides with a step size of two. The maximum and average feature values in the corresponding position of each filter are determined, and the maximum pooled feature map and the average pooled feature map are the outputs.

Fig. 10 Comparison of two different pooling operations



The fully connected layer in a CNN is composed of the single-layer feedforward network introduced in the previous section. The number of layers of this fully connected layer can be chosen according to the specific needs of the task. Compared with the convolutional layer, the fully connected layer contains a large number of parameters and dense connections. Therefore, more fully connected layers will lead to a more difficult model optimization; thus, the number of fully connected layers in mainstream CNNs does not exceed three.

2.2.2 Mechanism of a Fully Convolutional Neural Network

In contrast to the CNN, the fully CNN does not contain a fully connected layer; its structure is shown in Fig. 11. Each element of the output layer of the convolutional and pooling operations represents the local information of the input features, whereas the fully connected layer depicts the global information of the input features; thus, the convolutional and pooling operations can preserve the spatial dimensional information of the input tensor. A fully CNN composed of all convolutional and pooling operations can output a feature map that retains the spatial location information. Compared to traditional CNNs, fully CNNs are more suitable for tasks such as image segmentation, where both the input and output are images.

2.2.3 Common Convolutional Neural Networks

During the development of CNNs, several representative networks have emerged, such as VGGNet, ResNet, and DenseNet. In this section, we introduce these networks.

VGGNet has two structures of 16 and 19 layers, as shown in Fig. 11. All the convolution kernels in the VGGNet network are of size 3×3 . VGGNet cascades three sets of convolution operations with a kernel size of 3×3 and a step size of one. These three sets of convolution operations are equivalent to one convolution operation with a kernel size of 7×7 . This has two main benefits. First, a deeper network structure will learn more complex nonlinear relationships, which will lead to

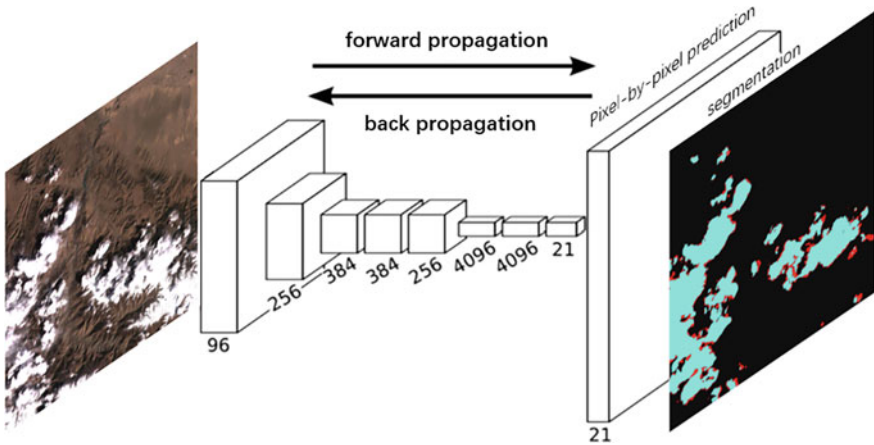


Fig. 11 Overview of the fully convolutional neural network architecture. The input and output of this architecture are pictures

better results for the model. Second, this reduces the number of parameters required to perform the convolutional operations.

In 2015, Kaiming He proposed a 152-layer ResNet to win the 2015 ILSVRC competition with a top-1 error record of 3.6%. The proposed ResNet was revolutionary because the network introduced a residual mechanism. Compared to the traditional convolutional module, the convolutional module in the residual network learns only a small variation in the input features. The output of the residual convolution module is equivalent to the superposition of the input features and the amount of variation in the input features. When BP is performed, the residual structure retains some of the gradient information and alleviates the gradient disappearance problem. Therefore, the ResNet model is deeper and has a better performance.

ResNet uses jump connections to pass gradients directly from the back layer to the front layer. However, features that have been jump-connected and features that have been convolutionally transformed need to be summed before they can be output. This may affect the information propagation in the network. In response, Huang et al. [27] proposed the network structure of DenseNet, which is based on ResNet. Unlike ResNet, which only forms a jump-connected module with the previous layer, DenseNet achieves feature reuse by directly connecting each layer to its preceding layer. Compared to ResNet, DenseNet not only reduces the error rate but also reduces the number of parameters in the network.

2.2.4 The Shortcomings of Convolutional Neural Network

The convolutional operations of CNNs can be divided into one-dimensional, two-dimensional, and three-dimensional convolutions according to the dimensions of the convolutional kernel. One-dimensional convolution is mainly applied to tasks

related to one-dimensional sequence signals, such as EEG signal analysis, speech signal processing, and radio signal classification. Two-dimensional convolution is mainly used in the image field, such as image super-resolution and image denoising; image restoration, such as the image processing field, image recognition, and target detection, and semantic segmentation, such as the computer vision field. Three-dimensional convolution is commonly used in medical CT image segmentation and video motion recognition, etc.

Although CNNs have powerful feature extraction capabilities, they are based on the assumption of mutual independence between consecutive input samples. Therefore, the network is difficult to apply to tasks where there is an inherent logical relationship between successive inputs.

2.3 *Deep Recurrent Neural Network*

2.3.1 Mechanism of Convolutional Neural Network

Jordan [29] proposed Jordan Network in 1986 and designed a memory mechanism that fed back the output of the entire network to the input layer of the network the next moment. One of the foundational works done on RNNs is the simple recurrent networks (SRNs), which was proposed by Elman in 1990 [12]. The SRN was modified on the Jordan Network, and the output of the hidden layer in the network is shown below. Feedback to the input layer occurs at any given moment. The Jordan Network uses the entire network as a loop, whereas the SRN only uses the hidden layer as a loop. Therefore, the SRN is more flexible to use; it also avoids the problem of conversion between network output dimensions and input dimensions.

The structures of the SRN and the widely used RNN are similar. The output value of the RNN at the next moment is jointly determined by multiple past moments. In fact, there is often a problem that the output of the network is affected by the future inputs.

Driven by similar ideas, Schuster and Paliwal [45] improved the traditional RNN by designing a bidirectional cyclic neural network (bidirectional RNN, BRNN). The BRNN is a superposition of two RNNs in opposite directions. Each hidden layer must record two values for both the positive and negative directions. The final output value depends on the RNN calculated in the forward and backward directions (Fig. 12).

Hochreiter and Schmidhuber [25] proposed an LSTM. In RNNs, owing to the uncertain attenuation of information in the cyclic structure, it takes a lot of time to learn to store information over a certain time interval through the BP of the periodic structure. In this problem, the solution used by LSTM is the structure of the input gate, and the output gate is designed in the RNN to control the state and output of the loop unit at any given time. Gers et al. [16] proposed that LSTM did not have a clear prior end mark when processing long sequence inputs. Hence, the original LSTM added a forget gate mechanism, which allowed the LSTM to learn to reproduce at the appropriate time and set itself to release the internally stored information. One year

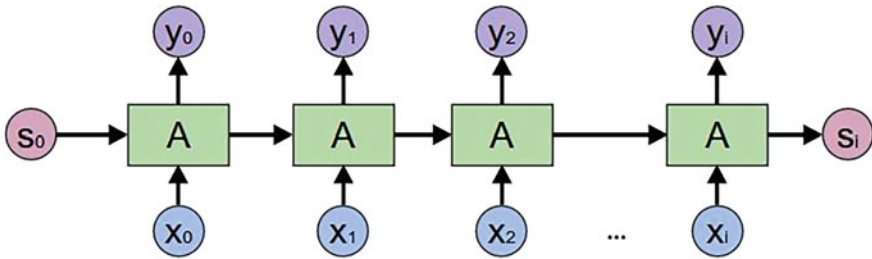


Fig. 12 The structure of the recurrent neural network. The blue neuron represents the time series of sequential input. The purple neuron represents the output of the recurrent network. The pink neurons represent the information transmitted in the middle

later, Gers and Schmidhuber [15], made improvements to the LSTM again, adding a peephole connection; the value memorized in the last moment was also used as the input of the gate structure; thus the structure of LSTM gradually improved to its present-day state.

Graves and Schmidhuber [18] proposed a bidirectional LSTM (BLSTM), which combined the BRNN and LSTM. Graves et al. [19] applied the LSTM to handwritten text recognition tasks, which overcame the difficulty faced by traditional models in segmenting scribbled and overlapping text. The offline text recognition rate was 74.1%, which was the best at that time. Graves et al. [20] combined deep networks and RNNs and proposed a deep recurrent network based on LSTM, which was applied to speech recognition tasks. The error in the TIMIT dataset was only 17.7%, which was the best at that time. In 2014, Cho et al. [7], simplified its gate structure by replacing the input gate, forget gate, and output gate of LSTM with an update gate and reset gate, and proposed a gated recurrent unit for the first time. For traditional statistical machine translation, the encoder-decoder model was proposed.

Owing to the further development of the LSTM, the LSTM has been widely used in various applications of natural language processing and many variants have been developed. Next, we introduce LSTM and ConvLSTM in detail.

2.3.2 Mechanism of LSTM

Recurrent connections can improve the performance of neural networks by leveraging their ability to understand sequential dependencies. However, the memory produced from recurrent connections can be severely limited by the algorithms employed for training the RNNs. To date, all models have been affected by the explosion or disappearance of gradients in the training phase, which renders the network unable to learn the long-term order dependence in the data. The LSTM was specifically designed to solve this problem.

LSTM is one of the most commonly used and effective methods for reducing the vanishing gradient and exploding gradient. It changes the structure of the hidden

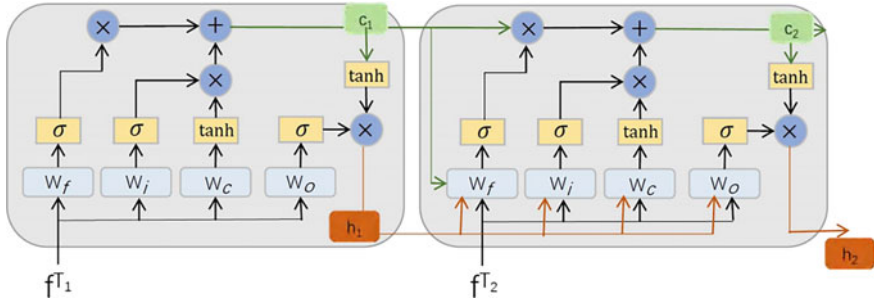


Fig. 13 The structure of LSTM

unit, and the input and output of the neural unit are controlled by the gates. These gates control the information flow of hidden neurons and retain the features extracted from the previous time steps.

As illustrated in Fig. 13, an LSTM-based recurrent layer maintains a series of memory cells c_t at time step t . The activation of the LSTM units can be calculated by:

$$h_t = o_t \times \tanh(c_t) \quad (12)$$

where $\tanh(\cdot)$ is the hyperbolic tangent function, and O_t is the output gate that controls the extent to which the memory content is exposed. The output gates are updated as:

$$o_t = \sigma(W_o^T \times f^{T_t} + U_o^T \times h_{t-1} + b_o) \quad (13)$$

where W_o and U_o are the input-output weight matrix and memory-output matrix, respectively, and b_o is the bias. The memory cell c_t is updated by partially discarding the present memory contents and adding new contents of the memory cells \tilde{c}_t

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t \quad (14)$$

where \otimes is the element-wise multiplication. The new memory contents are

$$\tilde{c}_t = \tanh(W_c^T \times f^{T_t} + U_c^T \times h_{t-1} + b_c) \quad (15)$$

Here, W_c and U_c are the input-memory weight matrix and hidden memory coefficient matrix, respectively; b_c is the bias; i_t is the input gate, which modulates the extent to which the new memory information is added to the memory cell; f_t is the forget gate, which controls the degree to which the contents of the existing memory cells are forgotten. The gates are computed as follows:

$$i_t = \sigma(W_i^T \times f^{T_t} + U_i^T \times h_{t-1} + V_i^T \times c_{t-1} + b_i) \quad (16)$$

$$f_t = \sigma(W_f^T \times f^{T_t} + U_f^T \times h_{t-1} + V_f^T \times c_{t-1} + b_f) \quad (17)$$

where $W_i, U_i,$ and V_i are the input-memory weight matrix of the input gate, hidden-memory coefficient matrix, and output weight matrix of the previous cell state, respectively; $W_f, U_f,$ and V_f are the input-memory weight matrix of the output gate, hidden-memory coefficient matrix, and output weight matrix of the previous cell state, respectively; b_i and b_f are the biases.

2.3.3 Mechanism of convLSTM

The CNN does not have additional complex operations of artificial neural networks for preprocessing and spatial distribution. Therefore, it uses a unique fine-grained feature extraction method to automatically process the spatial data. When dealing with time features, LSTM can effectively avoid the disappearance of valid information because of the long data interval span. There are certain limitations if parallel CNN and LSTM are used to extract spatial and temporal features. For example, in a parallel structure composed of a CNN and LSTM, the input and output of the two are relatively independent, and the extraction of the relationship between different features is ignored.

Conv-LSTM was born out of a precipitation prediction problem [54]. The problem is as follows: given a map of the precipitation distribution for the first few hours, predict the precipitation distribution for the next few hours. This was accomplished by replacing the input-to-state and state-to-state parts of the LSTM from feedforward calculations to convolution calculations. A cell diagram is shown in Fig. 14.

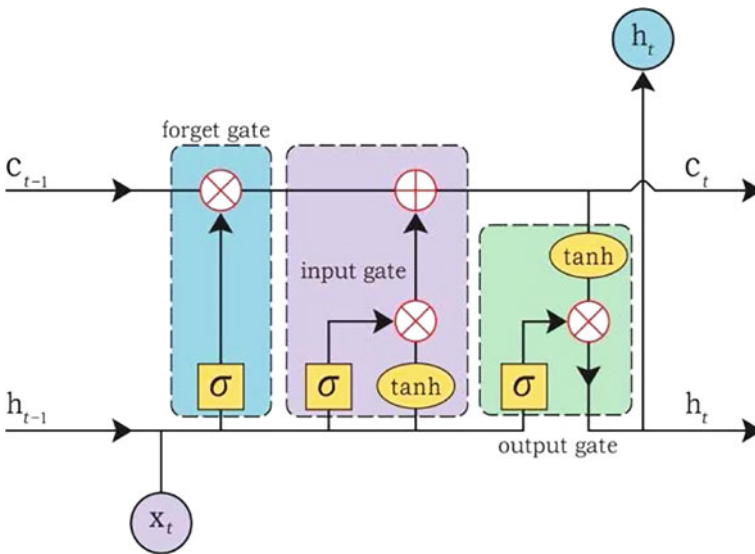


Fig. 14 The structure of Conv-LSTM

The principle of Conv-LSTM can be expressed by the following formula:

$$f_t = \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + W_{cf} \circ c_{t-1} + b_f) \quad (18)$$

$$i_t = \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + W_{ci} \circ c_{t-1} + b_i) \quad (19)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tanh(W_{xc} * x_t + W_{hc} * h_{t-1} + b_c) \quad (20)$$

$$o_t = \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + W_{co} \circ c_{t-1} + b_o) \quad (21)$$

$$h_t = o_t \circ \tanh(c_t) \quad (22)$$

where $*$ denotes the convolution and x, c, h, i, f, o are tensors. We can imagine Conv-LSTM as models that work on the eigenvectors of two-dimensional grids. It can predict the time-space features of the central grid based on the time-space features of the points around them.

2.4 Deep Generative Adversarial Network

CNNs and RNNs have been widely used in various fields, and have achieved good results. However, these methods need to rely on a large amount of labeled data. In actual research, we often encounter insufficient training sample data. This situation will lead to a decline in the recognition accuracy of our model. A generative adversarial network (GAN) can generate realistic sample data. If these generated sample data are used to train the model, the problem regarding the amount of training sample data can be solved. At present, GANs have been widely used in the fields of image and vision.

2.4.1 Architecture of Generative Adversarial Network

GAN [17] originated from the two-person zero-sum game. The two-person zero-sum game is a concept in game theory, which says that the sum of the interests of both parties in the game is always zero or remains unchanged. If one party gains, the other party must have a corresponding loss. GAN is composed of two parts: a generator and a discriminator. These two parts can be regarded as the two parties of the game. The optimization process of a GAN is equivalent to the two-person zero-sum game process.

The purpose of the GAN is to learn the distribution of real data, which can generate realistic data. The implementation of a GAN is shown in Fig. 15.

The generator G is used to capture the distribution of real data, and random noise is used as an input to generate the sample data. To capture the real data distribution, first, a random noise z that obeys the prior distribution $P_z(z)$ is given. Then the

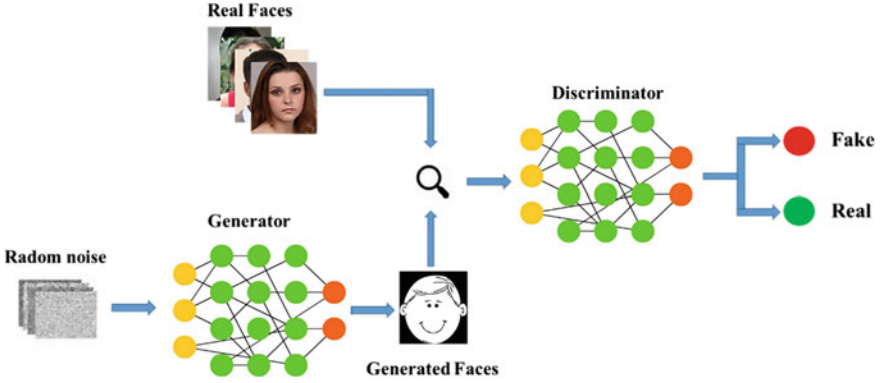


Fig. 15 The basic structure of the generative adversarial network

mapping space $G(z; \theta_g)$ is constructed using $P_z(z)$. The mapping space $G(z; \theta_g)$ is a generative model of the parameters θ_g . The random noise z is used as the input of the generator, and sample data is generated through the mapping space $G(z; \theta_g)$.

The discriminator D is used to determine whether the input sample is a real sample or a generated sample, which is equivalent to a two-classifier. Defining the mapping function $D(z; \theta_d)$, the input sample outputs a scalar between zero and one through the mapping function. This scalar represents the probability that the input sample is a real sample; the mapping function $D(z; \theta_d)$ is a discriminator for the parameters θ_d . It should be noted that the input of the discriminator consists of two parts. One part is the sample generated by the generator, and the other part is the real sample x that obeys the real data distribution.

2.4.2 Training of Generative Adversarial Network

In the GAN training process, generator G and discriminator D compete with each other, continuously alternating the iterative optimization. Finally, they gradually reach an equilibrium. The optimization function of GAN is as follows:

$$\max_G \max_D V(D, G) = E_{x \sim P_{data}(x)}[\log D(x)] + E_{z \sim P_z(z)}[\log(1 - D(G(z)))] \quad (23)$$

where $D(x)$ represents the probability that the real sample x is identified as a real sample after passing through the discriminator. $G(z)$ represents the sample data generated by random noise z through the generator. $D(G(x))$ represents the probability of the generated sample data being judged as a real sample after passing through the discriminator.

It can be seen that the optimization function of the GAN is equivalent to a min-max optimization problem. This optimization function has two steps. The first step is to optimize the discriminator D , and the second step is to optimize generator G .

The objective function can be regarded as alternately optimizing the following two objective functions:

$$\max_D V(D, G) = E_{x \sim P_{data}(x)}[\log D(x)] + E_{z \sim P_z(z)}[\log(1 - D(G(z)))] \quad (24)$$

$$\min_G V(D, G) = E_{z \sim P_z(z)}[\log(1 - D(G(z)))] \quad (25)$$

The label of the real sample is artificially defined as one, and the label of the generated sample is zero. The purpose of the discriminator is to distinguish between true and false samples. Thus, the discriminator hopes that $D(x)$ is as close to one as possible, and $D(G(x))$ is as close to zero as possible. Using sample data to optimize the discriminator based on these two conditions is equivalent to maximizing $V(D, G)$. The purpose of the generator is to generate sufficiently realistic sample data; hence, the generator hopes that $D(G(z))$ is as close to one as possible. For this purpose, using sample data to optimize the generator is equivalent to minimizing $V(D, G)$.

It should be noted that the parameters of G or D are always fixed, and the parameters of the other part are updated during training. Finally, $P(z)$ and $P_{data}(x)$ are infinitely close. The generator can generate samples in which the discriminator cannot distinguish authenticity.

2.4.3 Typical Generative Adversarial Networks

The conditional GAN [30] adds constraints to the standard GAN to guide the data generation process, thereby generating controllable samples. It solves the problem of GAN image generation being too free and difficult to control. The constraint condition can be the category label or semantics of the image. The implementation process of the conditional generation confrontation network is illustrated in Fig. 16.

The deep convolution generative adversarial network (DCGAN) [51] combines CNN and GAN to optimize the original GAN model from the network structure. DCGAN replaces the generator and discriminator in the original GAN with two CNNs to improve the quality of sample generation and the speed of model convergence.

2.4.4 Application of Generative Adversarial Network

A GAN can generate real-like samples without explicitly modeling any data distribution in the process of generating samples. Therefore, GANs have a wide range of applications in many fields, such as images and text.

One function of the GAN is to generate the data. A limitation of the development of deep learning is the lack of training data; only GAN-generated data can compensate for this shortcoming. For example, given the text description of a bird, such as some

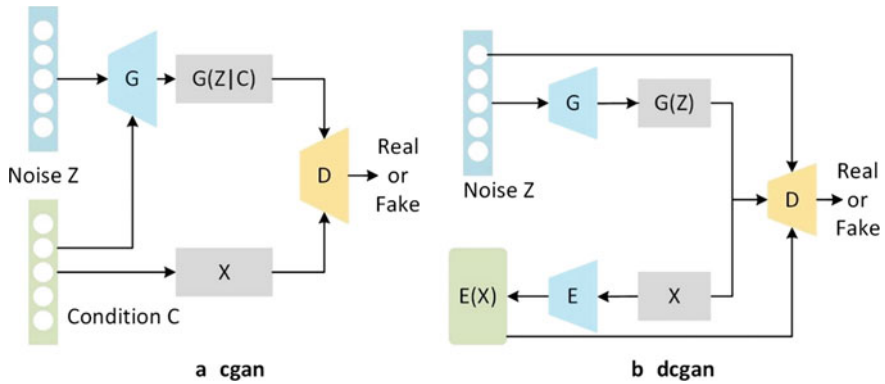


Fig. 16 Common Generative Adversarial Network Structure. **a** conditional generative adversarial nets. A network with conditional constraints (green tensors in the figure) **b** deep convolutional generative adversarial networks. A network composed of convolutional layers

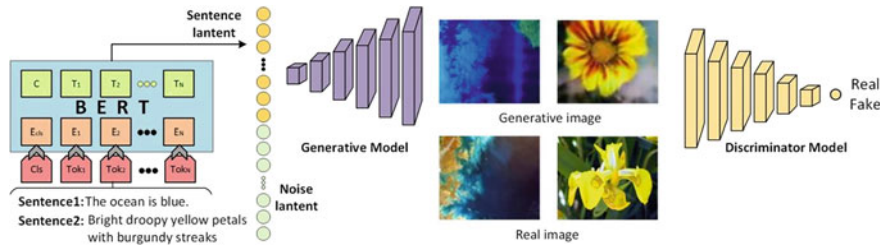


Fig. 17 Cross-modal image generation. The generative adversarial network can realize the task of generating from text to image

black and white on its head and wings, and a long orange beak, a trained GAN can generate images that match the description. Figure 17 shows the application of GAN in image generation

Another important application of GAN is image super-resolution, which refers to the process of recovering high-resolution (HR) images from low-resolution (LR) images. This is an important class of image processing techniques in computer vision and image processing. It enjoys a wide range of real-world applications, such as medical imaging, surveillance, and security amongst others. Other than improving image perceptual quality, it also helps to improve other computer vision tasks. However, generally, this problem is very challenging and inherently ill-posed since there are always multiple HR images corresponding to a single LR image. As shown in Fig. 18, relying on powerful image generation capabilities, GAN can decode and encode LR images into HR images.

The task of image translation can also be achieved through GAN. Image translation is the conversion of one (source domain) image to another (target domain) image. During the translation, the content of the source domain image will remain

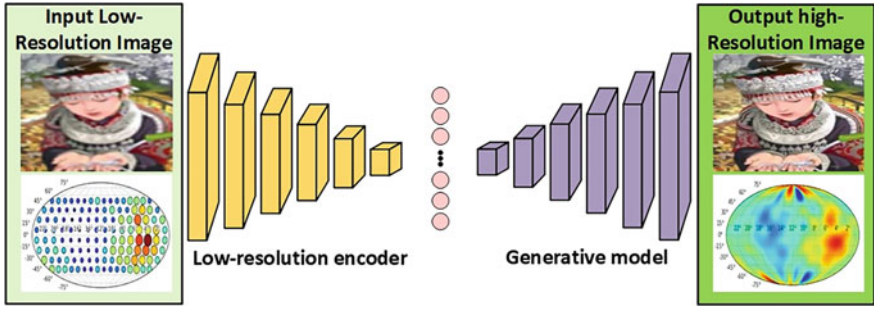


Fig. 18 Schematic diagram of super resolution. In the computer vision community, the super-resolution task makes blurry images clear. In oceanography, the purpose of the super-resolution task is to improve the temporal and spatial resolution of the numerical field

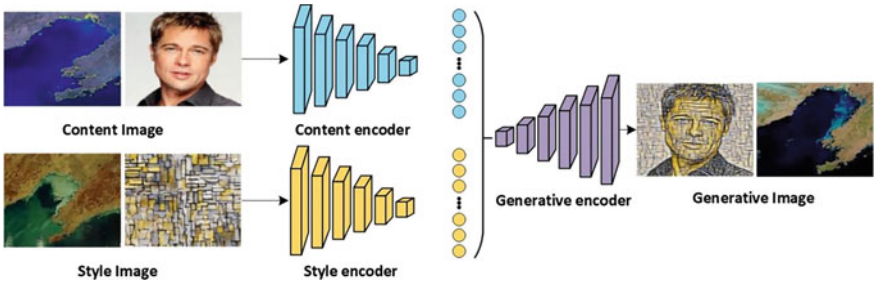


Fig. 19 Schematic diagram of image translation. This task merges two images to get a brand-new image, the generated image retains the content of one input and the style of the other input

unchanged. Nevertheless, the style or other attributes will be the same as the target domain, as shown in Fig. 19.

Image restoration is a technology that uses the learned image information to complement or modify the damaged image. Image restoration has various applications such as image completion and image deblurring. Owing to its good ability to fit the real distribution, GAN has shown good results in image restoration.

2.4.5 Problems of Generative Adversarial Network

GAN has become a popular research topic in recent years. Despite its recent genesis, it has developed rapidly and has made important contributions in many fields. However, owing to problems such as model collapse and gradient disappearance, its generation effect, training efficiency, and application range are still restricted.

(1) Low image-generation diversity

The diversity of image generation has always been an important issue in the field of GAN research. Traditional GAN algorithms can only fit simple datasets with small sizes, and the complexity of image generation is low. Therefore, the GAN

algorithm has been developed for image diversity. The existing GAN algorithm can generate indistinguishable high-quality images; however, many factors restrict the development of image diversity, which often conflict with other factors such as image size and model complexity.

(2) Insufficient model training efficiency

GAN has training instability problems, which are caused by model collapse and gradient disappearance. In addition, the complex model structure and redundant information causes the training cycle to be too long.

(3) The application field has not been extensively studied

GAN has been used in many fields in a relatively short period; Nonetheless, it is mostly limited to image processing. Many algorithms mention only their achievable functions without explaining their use-value. This development is slow in other fields such as NLP.

3 Perceptual Understanding Based on Neural Network

3.1 Recognition Based on Neural Network

Various neural network architectures support a wide variety of perceptual understanding applications. Currently, research on neural networks in natural language processing, visual data processing, speech signal processing, etc. is progressing rapidly. They have been widely used in industrial fields such as intelligent security, medical health, and industrial inspection. Figure 20 briefly depicts the applications of neural network-based deep learning research in some important fields. This section introduces neural network-based recognition, segmentation, and prediction applications.

3.1.1 Problem Description

Neural network-based recognition tasks involve both the extraction of features from the model input content and the establishment of mapping relationships between the extracted features and the identifiable attributes of the sample (category, location, etc.). The advent of CNNs has led to the rapid development of neural networks for visual recognition tasks. In this subsection, the classification, localization, and detection in recognition tasks are described.

Figure 21 depicts the flow of a neural network-based recognition task. First, the input image is subjected to a CNN to extract key features and represent them in a one-dimensional feature vector. Then, this feature vector is input to the classification module, localization module, or target detection module according to the different tasks; the corresponding output results are subsequently obtained. The classification module is used to determine the category of the target in the input image. The localization module is used to determine the location of the target in the picture.

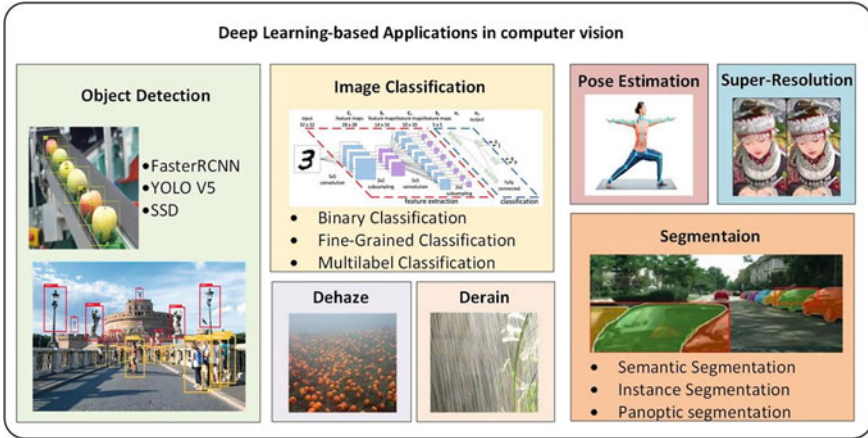


Fig. 20 Deep learning-based applications in computer vision. It briefly depicts applications of neural network-based deep learning research in some important fields



Fig. 21 Overview of a neural network-based recognition task. It contains the three basic computer vision tasks of classification localization and detection

The target detection module is a combination of classification and localization, that is, it determines the location of the target in the input image and its corresponding target category. The next section describes each of these three recognition tasks and their corresponding modules.

3.1.2 Classification

In computer vision, image classification is a crucial job. The objective of image classification is to discern what category the object in the image belongs to, such as whether it is a cat or a dog. As shown in Fig. 22, according to the difficulty of the classification task, it can be subdivided into dichotomous classification task, multi-classification task, or multi-label classification task, etc. The image classification task can be expressed by the following equation.

$$C = f_{fc}[f_{conv+pool}(x)] \tag{26}$$

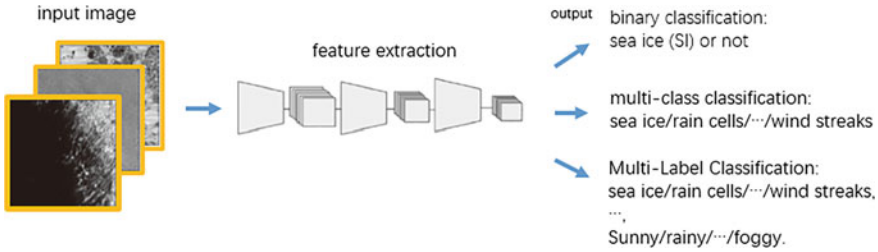


Fig. 22 Overview of a classification-based neural network. It includes binary classification, multi-class classification and multi-label classification

C denotes the category, $f_{conv+pool}$ denotes the convolutional and pooling layers, X denotes the input image, and f_{fc} denotes the fully connected layer.

Binary classification is the most basic type of image classification. It is used to identify whether the input image contains a certain category; the classification results are represented by zeros and ones. The binary classification task is used to identify whether the target is visible in the input. If the model result is zero, then it means that the input image does not contain the target object and vice versa.

Multi-category image classification is more widely used than binary classification. Its purpose is to classify the corresponding target class of an image that contains only one target class. Multi-category tasks are used to identify the specific category of the target in the input image, and the picture input to the network often contains only one category of targets. Multi-category image classification has now been integrated into all aspects of life and has been effectively used in a variety of sectors, such as facial recognition.

Multi-label classification is used to identify all the categories present in the input images. The pictures processed by this task often contain several different labels, and these labels are compatible with each other. The multi-label classification task can describe the information of pictures more graphically and has a more realistic meaning.

The success of deep learning classification tasks is inextricably linked to the development of supervised learning. The construction of large-scale datasets and the development of computational resources have made it possible to train the neural network parameters. The loss function, a metric used to measure how well a model predicts results, is an important component of the classification task and plays an important role in BP to update the network parameters. The purpose of the loss function is to update the parameters of the model to achieve better prediction results. Take the most commonly used cross-entropy loss function as an example, its loss function can be written as follows.

$$Loss = - \sum_{i=1}^n y_i \log(p(x_i)) \tag{27}$$

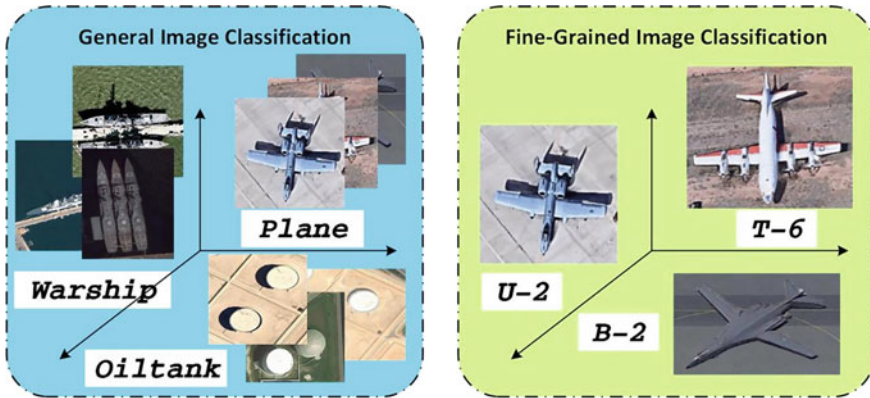


Fig. 23 Comparison of categories between traditional classification and fine-grained image classification. Fine-grained image classification can be used to distinguish different types of fighter jets

where $p(x_i)$ denotes the probability value that sample x is predicted to be the i -th category, y_i denotes the true label of the corresponding input sample x in the i -th category, which is one if it belongs to the i -th category, and zero otherwise.

For datasets such as ImageNet, which has more than 10 million images and 20,000 classes, the image classification level computer has surpassed that of humans; however, deep neural networks are not effective when recognizing subclasses under traditional categories, that is, discriminating between magpies and sparrows, etc., under the category of birds. Furthermore, the training of the model requires a large number of manually labeled tags, which is expensive. The cost of labeling increases exponentially with the number of targets and the difficulty of discernibility. To address these two challenges, fine-grained image classification and unsupervised image classification have emerged.

The distinction of fundamental categories and the performance of finer subcategories are the foundations of fine-grained image classification. Fine-grained image classification may be used to discriminate between sub-categories, such as various types of fighter airplanes, as illustrated in Fig. 23. Fine-grained image categorization may also be used to discern between different automobile types and battleship models, for example. This categorization has a wide range of practical uses.

Fine-grained images have a more similar appearance and features than coarse-grained images. In addition, there are the effects of poses, perspective, illumination, occlusion, and background interference, etc. in the acquisition. As a result, the data has a huge inter-class variability and a modest intra-class variability. This makes classification more difficult.

The above classification tasks are achieved via supervised learning. Each sample has its corresponding label, and the deep neural network is used to continuously learn the features corresponding to each label and achieve the classification. In this case, the size of the dataset and the quality of the labels often play a decisive role

in the performance of the model. High-quality datasets naturally bring difficulties in labeling and need a lot of human and financial resources. The aim of unsupervised image classification is to classify samples into several classes without using label information, thus greatly reducing the labor and time costs associated with data labeling.

3.1.3 Localization

The purpose of image localization is to determine the location of the target in the input image. An image is input to the model, and the model outputs the center coordinates, width, and height of the target location in the image. The result is a rectangular box with the location of the target in the input image, represented by the horizontal and vertical coordinates of the rectangular box and its width and height.

For example, a typical regional proposal network (RPN) incorporates a dichotomous classification problem, that is, whether the object in the location is an object or not. Rectangular boxes of different sizes and aspect ratios are first generated on a sliding window and labeled positively or negatively. The sample data for the RPN is organized as a binary classification labeling problem with multiple rectangular boxes within the input image and the presence or absence of objects within each rectangular box. The RPN maps each sample to a probability value and four localization values. The probability value reflects the probability of having an object in a rectangular box, and the four localization values are used to regress the center horizontal and vertical coordinates and the width and height of the target object location.

3.1.4 Detection

The image detection problem is equivalent to a combination of localization and classification problems. It needs to locate the location of the target and to deduce the class of the located target.

Current mainstream target detection algorithms can be divided into two-stage and one-stage detection. The former frames detection as a “coarse to fine” process, while the latter defines it as a “one-step completion”

The two-stage detection algorithm is relatively slower but more effective. Take the most typical two-stage target detection algorithm, faster RCNN, as an example. As shown in Fig. 24a, the first stage locates the target location in the input image through the region suggestion network. Then in the second stage, the localized targets are classified; finally, the rectangular box of the localized targets and their categories are obtained. There is a certain sequence between the two stages of this model, which makes a stronger connection between localization and classification. This approach allows for more robust results. The single-stage detection algorithm is fast but less accurate. Taking the most typical two-stage target detection algorithm YOLO v3 as an example, YOLO directly generates both coordinates and probabilities for each category at a time using regression, as shown in Fig. 24b. This improves the

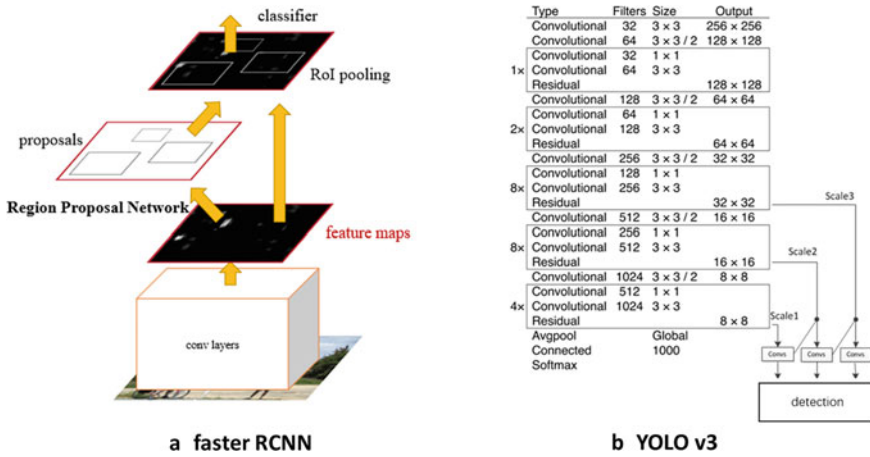


Fig. 24 Two representative detection framework architecture diagrams. **a** Faster RCNN, most typical two-stage target detection algorithm. **b** YOLO v3, an one-stage target detection algorithm

computational speed of the model but leads to poorer results because of the lack of correlation between localization and classification.

There are two main measures for the target detection task: IoU and confidence. The IoU was introduced in the previous section, and the other metric, confidence, is used to measure the confidence level of the detected targets. The higher the confidence level, the more certain the model is about the output.

Object detection has both classification and localization capabilities and has numerous applications. Examples include face detection, text detection, remote sensing target detection, pedestrian detection, and automatic detection of traffic signs and traffic signals. Target detection is widely used in military investigations, disaster rescue, and urban traffic management, and has gained wide attention in the fields of automatic driving, video surveillance, and criminal investigation, etc.

3.2 Segmentation Based on Neural Network

Neural network-based segmentation is most often found in vision tasks. The following section introduces image segmentation as an example.

Image segmentation is different from image classification and monitoring. The task of image classification is to identify the content of an image, whereas the task of image monitoring is to identify the content of an image and also monitor its location. Image segmentation is a pixel-level image classification task based on classifying each pixel of an image.

3.2.1 Problem Description

Image segmentation is a key operation in image processing. It refers to the use of several disjoint regions to represent a complete image, based on the references of grayscale, luminance, texture, and other characteristics of the image. It can simplify the representation of the image. Features show similarity or consistency in the same region, while they show clear differences in different regions.

Existing image segmentation is generally divided into semantic segmentation, instance segmentation, and panoramic segmentation, as shown in Fig. 25.

The semantic segmentation is to give each pixel a class label without distinguishing each instance of the same category. Instance segmentation is a combination of object detection and semantic segmentation. First, it detects the object in the image and then uses semantic segmentation on the detected objects. Moreover, it also distinguishes between different instances of the same kind. Panoramic segmentation is a combination of semantic segmentation and instance segmentation. It gives each pixel a class label while also distinguishing between different instances of the same kind.

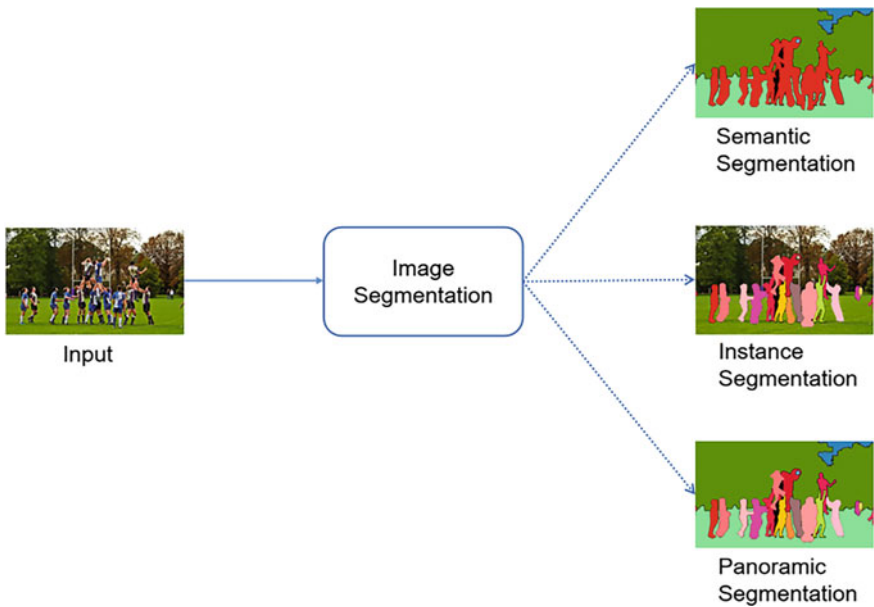


Fig. 25 Division of image segmentation tasks. According to the different segmentation tasks, image segmentation can be divided into semantic segmentation, instance segmentation and panoramic segmentation

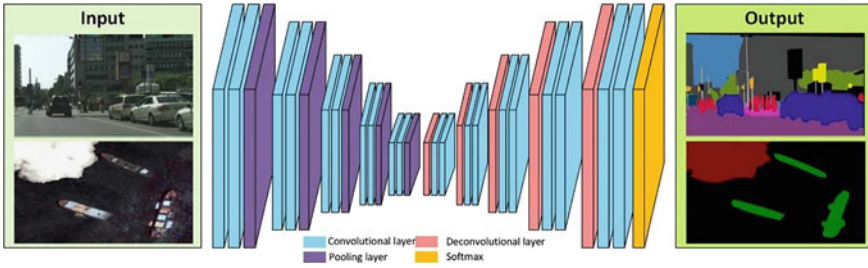


Fig. 26 Example of semantic segmentation based on neural networks. The image above represents an application in city street view. The image below represents an application in a marine environment

3.2.2 Semantic Segmentation

Semantics refers to the contents of an image. Semantic segmentation is performed on a pixel-by-pixel basis to label the class based on the semantics of the image, as shown in Fig. 26. Semantic segmentation uses feature extraction and classification to mark pixels of cars as blue, pixels of trees as green, pixels of buildings as gray, and so on.

There are three algorithm evaluation metrics for neural network-based semantic segmentation: accuracy, execution time, and memory consumption. Among these, execution time is the most important measure. In practical applications, datasets are generally very large, and computer hardware facilities are limited, and only a short execution time can make image segmentation more popular in daily applications. While memory consumption is also an important factor affecting semantic segmentation, memory is expandable in most scenarios. Accuracy is the most critical metric for semantic segmentation. Pixel accuracy and mean IoU are the common forms of accuracy. The number of successfully categorized pixels divided by the total number of pixels is the pixel accuracy. The mean IoU calculates the ratio of the intersection and the union of two sets, and in the field of semantic segmentation, the true and predicted values are the embodiment of the two sets.

The initial neural network-based semantic segmentation models are AlexNet, VGGNet, and ResNet. The emergence of fully convolutional networks later broke the previous segmentation method; thus, the accuracy in the PASCAL VOC dataset has substantially improved. FCN has greatly promoted the development of semantic segmentation algorithms. SegNet [1], RefineNet [32], PSPNet [55], and DeepLab [6] have been proposed one after another, and all of them have achieved good results.

3.2.3 Instance Segmentation

The network architecture of semantic segmentation aims to optimize the accuracy of segmentation results and improve segmentation efficiency; this should allow for applications in the field of image semantic real-time processing. However, seman-

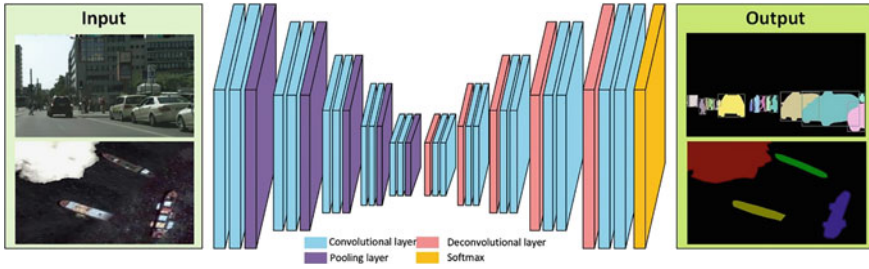


Fig. 27 Example of semantic segmentation based on neural networks. The mask and label of each object in the figure need to be segmented

tic segmentation can only judge categories and cannot distinguish individuals, and it is impossible to accurately understand semantic information or parse scenes in many complex real scenarios. The emergence of an instance segmentation algorithm effectively solves this problem.

The concept of instance segmentation was first proposed by Hariharan et al. [21], who aimed to detect objects in the input image and assign category labels to each pixel of the object. As shown in Fig. 27, unlike semantic segmentation, instance segmentation is able to distinguish between different instances with the same semantic category in the foreground.

Instance segmentation is essentially a combination of semantic segmentation and object detection. It not only has the characteristics of semantic segmentation to classify images at the pixel level but also has the characteristics of object detection to locate different instances of the same category in an image. As shown in Fig. 27, the instance segmentation technique based on deep neural networks usually consists of three parts: the image input, instance segmentation model, and the segmentation result output. First, a deep network model is designed according to the actual requirements, and the original image data is directly input to the network to extract image features. After the high-level abstract features are obtained, the instance segmentation model is used to process. The processing can first determine the location and category of object instances via object detection. It then performs segmentation in the selected region, or it can first implement the semantic segmentation task and then distinguish different instances. The final output is an instance segmented image with the same resolution as the input image.

In recent years, instance segmentation techniques have been rapidly developed. Mask R-CNN [22], which was developed based on the two-stage detector faster RCNN [42], is a direct and effective instance segmentation method. It has become the basic framework for some instance segmentation tasks because of its high accuracy and stability. YOLACT [3], an instance segmentation algorithm extended from a single-stage detector with high-speed detection, achieves real-time segmentation of video information and can obtain efficient processing ability with a small loss of accuracy.

3.2.4 Panoptic Segmentation

Panoramic segmentation combines the tasks of semantic segmentation and instance segmentation to generate a global unified segmented image. Instance segmentation only detects the objects in the image and segments the detected objects. Panoramic segmentation detects and segments all objects in the image including the background, achieving a panoramic understanding of the image. Figure 28 shows the panoramic segmentation. The information predicted by the panoramic segmentation is the most comprehensive, which includes not only the category classification of all pixel points using semantic segmentation but also the function of distinguishing between different instances in the instance segmentation task.

Compared to semantic segmentation, the difficulty of panoramic segmentation is to optimize the design of the fully connected network so that its network structure can distinguish between different categories of instances. The goal of panoramic segmentation is to assign a semantic label and an instance an ID to each pixel in the image, where the semantic label refers to the category of the object and the instance ID corresponds to different numbers of similar objects. Therefore, the overlapping phenomenon in instance segmentation cannot occur in panoramic segmentation.

The basic process of panoramic segmentation is shown in Fig. 29; it is mainly divided into feature extraction, semantic segmentation and instance segmentation processing, and sub-task fusion. The purpose of feature extraction is to obtain the feature representation of the input image and provide the necessary information for the two subsequent tasks. It relies on deep neural networks, and the main networks used include VGGNet, ResNet, MobileNet. The extracted features are shared by semantic

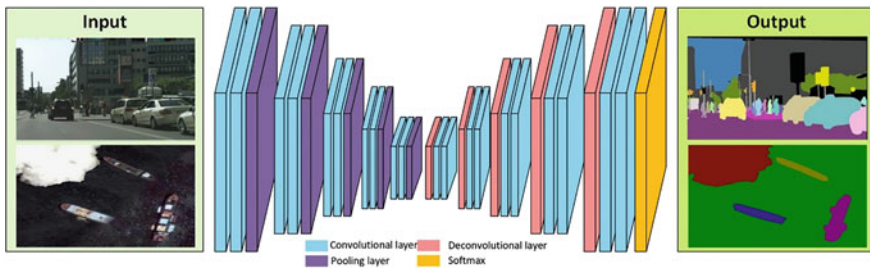


Fig. 28 Example of panoramic segmentation based on neural networks. Panoramic segmentation task is a combination of semantic segmentation task and instance segmentation task

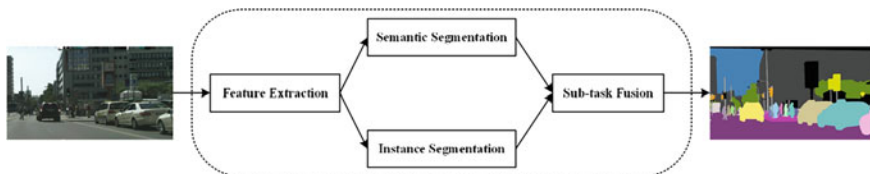


Fig. 29 The processing flow of panoramic segmentation

segmentation and instance segmentation. The semantic segmentation branch produces semantic segmentation predictions, while the instance segmentation branch produces instance segmentation predictions. The subtask fusion processes and fuses the prediction results of the above two branches in an appropriate way to produce the final panoramic prediction. Many works have been done to adopt the above basic process, such as JSIS-Net [9], AU-Net [31], Single network [10], OANet [34], etc.

Image segmentation can convert images into more meaningful and analyzable content expressions, which can effectively improve the processing efficiency of subsequent vision tasks. It is the basis of the computer vision scene being able to understand images and plays an important role in many scenes. In the field of medical image processing, by processing CT images of the organs of patients, it can accurately locate the boundaries of lesions; it can also automatically determine the location, shape, and size of the diseases, and assist doctors in lesion detection. In the field of remote sensing image processing, it can efficiently survey and plan the geographic spatial information such as topography and landform, water pattern direction, urban distribution, and farming planning. In the field of automatic driving, it can judge the surrounding environment of the road based on real-time road scenes, including lane line direction, traffic signs, and safety position of oncoming pedestrians or vehicles; it aims to provide correct guidance to vehicles and ensure driving safety. In the field of intelligent security, the object in the surveillance video is located and screened; it aims to play the role of security warning or object tracking. Furthermore, image segmentation can also be applied to augmented reality, text extraction, and industrial sorting, etc.

With the improvement of computer performance and the continuous optimization of image segmentation algorithm architecture, image segmentation technology based on deep neural networks has become an important task. While pushing the network in the direction of being lightweight, real-time, and highly accurate, more attention should be paid to technology implementation and scene promotion.

3.3 Prediction Based on Neural Network

This section introduces the application of neural networks to prediction problems. We will expand on regression, time-series prediction, one-dimensional signal prediction, and two-dimensional video prediction.

3.3.1 Regression

The link between the independent and dependent variables is predicted using regression [8]. The learning of the regression issue is similar to that of function fitting: select a function curve that fits the known data and accurately predicts the unknown data.

According to the number of independent variables, regression problems are split into unary and multiple regressions. It is separated into linear and nonlinear regression based on the relationship between the independent and dependent variables.

The square loss function is the most often used loss function for regression learning. The least-squares approach may be used to tackle the regression problem in this scenario.

Tasks in many fields can be formalized as regression problems. For example, regression can be used in the business field as a tool for market trend forecasting, product quality management, customer satisfaction surveys, and investment risk analysis.

3.3.2 Time Series Prediction

A time series is a sequence of numbers arranged in a specific order, and this order is usually determined by time [53]. It is an important means for people to understand the objective world and natural phenomena.

The development of time-series prediction is divided into two periods. The early-stage was before World War II, and financial and economic forecasting was the key. The second stage was from the mid-war to the 21st century. In this period, the application areas were more extensive; these include meteorology, aerospace, electronic computers, and mechanical vibration, etc. Time-series prediction has become a hot field pursued by experts in academic research.

The rapid development of artificial intelligence has a significant impact on time-series prediction methods. Currently, commonly used time-series prediction methods can be divided into traditional methods and methods based on deep learning.

Traditional time-series prediction methods are usually not ideal for non-wide stationary time series. Moreover, they are limited in forecasting by complex and highly nonlinear time series.

The emergence of neural networks has solved these problems. Neural networks have good learning capabilities. They can learn the underlying laws of the time series through multiple iterations based on the data itself. Compared to traditional methods, neural network methods are more accurate and can be applied to most time series. Among them, RNNs usually perform better when dealing with time-series problems.

3.3.3 One-dimensional Signal Prediction

Neural networks are also widely used in the prediction of one-dimensional signals.

In recent years, some scholars have discovered that high-frequency oscillation signals have a certain correlation with particular diseases, which may help to improve the accuracy of lesion location and promote the success rate of clinical operations. On the other hand, these findings and applications can help us understand the pathophysiological mechanism of human brain electrical activity and explore preventive treatments that predict disease.

3.3.4 Two-dimensional Video Prediction

Video prediction technology predicts subsequent video frames when several lengths of continuous video frames are provided [55]. It is an important topic in the field of computer vision and has significant application prospects.

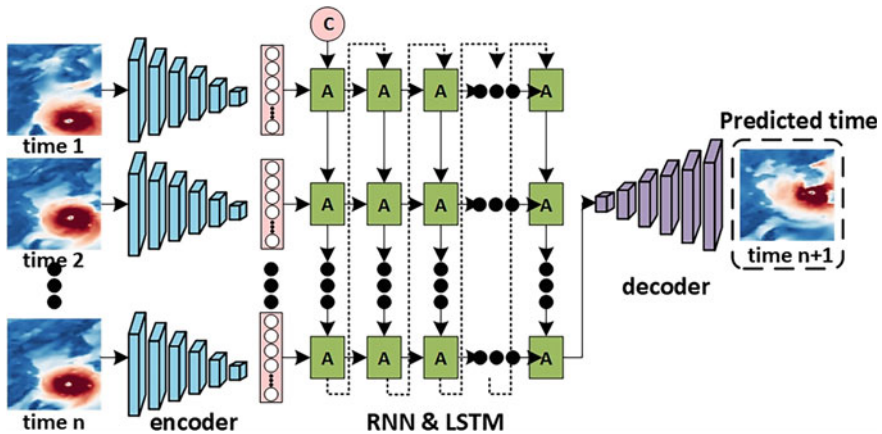


Fig. 30 Two-dimensional time series prediction. The typhoon position at time-1 to time-n are known, and the typhoon position at time n +1 is predicted

For example, in unmanned driving tasks, researchers can use the image information of historical frames to analyze the trajectory information of pedestrians and vehicles outside the car. In this way, the computer can predict the location of the objects outside the vehicle and make judgments in advance. Traffic accidents can also be avoided, and the safety of unmanned driving can also be improved.

In oceanography, the forecast of meteorological elements and ocean elements in a certain sea area is similar to the forecast of video sequences. The gridded two-dimensional sea area corresponds to a frame in the video, and the change of the sea area in a certain period corresponds to the change of the video frame on the timeline. As shown in Fig. 30, the figure depicts the neural network prediction of a typical typhoon path in the seas of eastern China. The neural network encodes the offshore wind current field in the eastern China sea at each known moment and predicts the process of the wind current field in the future through the RNN.

References

1. Badrinarayanan V, Kendall A, Cipolla R (2017) SegNet: a deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans Pattern Anal Mach Intell* 39(12):2481–2495

2. Berners-Lee T, Hendler J, Lassila O (2001) The semantic web. *Sci Am* 284(5):34–43
3. Bolya D, Zhou C, Xiao F, Lee YJ (2019) Yolact: real-time instance segmentation. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp 9157–9166
4. Buchanan B, Sutherland G (1968) Heuristic DENDRAL: A program for generating explanatory hypotheses in organic chemistry. Technical report, Stanford University California Department of Computer Science
5. Cauchy A (1847) Méthode générale pour la résolution des systemes d'équations simultanées. *Comp Rend Sci Paris* 25(1847):536–538
6. Chen LC, Zhu Y, Papandreou G, Schroff F, Adam H (2018) Encoder-decoder with atrous separable convolution for semantic image segmentation. In: *Proceedings of the European conference on computer vision (ECCV)*, pp 801–818
7. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. [arXiv:1406.1078](https://arxiv.org/abs/1406.1078)
8. Cox DR (1972) Regression models and life-tables. *J R Stat Soc: Ser B (Methodological)* 34(2):187–202
9. De Geus D, Meletis P, Dubbelman G (2018) Panoptic segmentation with a joint semantic and instance segmentation network. [arXiv:1809.02110](https://arxiv.org/abs/1809.02110)
10. de Geus D, Meletis P, Dubbelman G (2019) Single network panoptic segmentation for street scene understanding. In: *2019 IEEE intelligent vehicles symposium (IV)*. IEEE, pp 709–715
11. Duan Y, Chen X, Houthoofd R, Schulman J, Abbeel P (2016) Benchmarking deep reinforcement learning for continuous control. In: *International conference on machine learning*, PMLR, pp 1329–1338
12. Elman JL (1990) Finding structure in time. *Cogn Sci* 14(2):179–211
13. Fukushima K, Miyake S (1982) Neocognitron: a self-organizing neural network model for a mechanism of visual pattern recognition. In: *Competition and cooperation in neural nets*. Springer, pp 267–285
14. Gaschnig J (1982) Prospector: an expert system for mineral exploration. In: *Introductory readings in expert systems*. Gordon and Breach Science Publishers, New York
15. Gers FA, Schmidhuber J (2000) Recurrent nets that time and count. In: *Proceedings of the IEEE-INNS-ENNS international joint conference on neural networks*. IJCNN 2000. Neural computing: new challenges and perspectives for the new millennium, vol 3. IEEE, pp 189–194
16. Gers FA, Schmidhuber J, Cummins F (2000) Learning to forget: continual prediction with LSTM. *Neural Comput.* 12(10):2451–2471
17. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2020) Generative adversarial networks. *Commun ACM* 63(11):139–144
18. Graves A, Schmidhuber J (2005) Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw* 18(5–6):602–610
19. Graves A, Liwicki M, Fernández S, Bertolami R, Bunke H, Schmidhuber J (2008) A novel connectionist system for unconstrained handwriting recognition. *IEEE Trans Pattern Anal Mach Intell* 31(5):855–868
20. Graves A, Mohamed A, Hinton G (2013) Speech recognition with deep recurrent neural networks. In: *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, pp 6645–6649
21. Hariharan B, Arbeláez P, Girshick R, Malik J (2014) Simultaneous detection and segmentation. In: *European conference on computer vision*, Springer, pp 297–312
22. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask R-CNN. In: *Proceedings of the IEEE international conference on computer vision*, pp 2961–2969
23. Hebb DO (2005) *The organization of behavior: a neuropsychological theory*. Psychology Press
24. Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. *Neural Comput* 18(7):1527–1554
25. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
26. Hu X, Zhang J, Li J, Zhang B (2014) Sparsity-regularized HMAX for visual recognition. *PLoS One* 9(1):e81813

27. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708
28. Janocha K, Czarnecki WM (2017) On loss functions for deep neural networks in classification. [arXiv:1702.05659](https://arxiv.org/abs/1702.05659)
29. Jordan MI (1997) Serial order: a parallel distributed processing approach. In: Advances in psychology, vol 121. Elsevier, pp 471–495
30. Li X, Zhang Y, Zhang J, Chen Y, Li H, Marsic I, Burd RS (2017) Region-based activity recognition using conditional GAN. In: Proceedings of the 25th ACM international conference on Multimedia, pp 1059–1067
31. Li Y, Chen X, Zhu Z, Xie L, Huang G, Du D, Wang X (2019) Attention-guided unified network for panoptic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 7026–7035
32. Lin G, Milan A, Shen C, Reid I (2017) RefineNet: multi-path refinement networks for high-resolution semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1925–1934
33. Lipton ZC, Berkowitz J, Elkan C (2015) A critical review of recurrent neural networks for sequence learning. [arXiv:1506.00019](https://arxiv.org/abs/1506.00019)
34. Liu H, Peng C, Yu C, Wang J, Liu X, Yu G, Jiang W (2019) An end-to-end network for panoptic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6172–6181
35. Martin WA, Fateman RJ (1971) The MACSYMA system. In: Proceedings of the second ACM symposium on symbolic and algebraic manipulation, pp 59–75
36. McCarthy J (1959) Lisp: a programming system for symbolic manipulations. In: Preprints of papers presented at the 14th national meeting of the Association for Computing Machinery, pp 1–4
37. McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys* 5(4):115–133
38. Minsky M, Papert SA (2017) Perceptrons: an introduction to computational geometry. MIT press
39. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G (2015) Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533
40. Newell A, Shaw JC (1957) Programming the logic theory machine. In: Papers presented at the 26–28 Feb 1957, western joint computer conference: techniques for reliability, pp 230–240
41. Newell A, Simon H (1956) The logic theory machine—A complex information processing system. *IRE Trans Inf Theory* 2(3):61–79
42. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. *Adv Neural Inf Process Syst* 28:91–99
43. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. *Nature* 323(6088):533–536
44. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O (2017) Proximal policy optimization algorithms. [arXiv:1707.06347](https://arxiv.org/abs/1707.06347)
45. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. *IEEE Trans Signal Process* 45(11):2673–2681
46. Shortliffe EH (1974) MYCIN: a rule-based computer program for advising physicians regarding antimicrobial therapy selection. Technical report, Stanford University of California Department of Computer Science
47. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M (2016) Mastering the game of Go with deep neural networks and tree search. *Nature* 529(7587):484–489
48. Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A (2017) Mastering the game of go without human knowledge. *Nature* 550(7676):354–359

49. Turing AM (2009) Computing machinery and intelligence. In: Parsing the turing test. Springer, pp 23–65
50. Vinyals O, Babuschkin I, Czarnecki WM, Mathieu M, Dudzik A, Chung J, Choi DH, Powell R, Ewalds T, Georgiev P (2019) Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575(7782):350–354
51. Viola J, Chen Y, Wang J (2021) FaultFace: deep convolutional generative adversarial network (DCGAN) based ball-bearing failure detection method. *Inf Sci* 542:195–211
52. Wan L, Zeiler M, Zhang S, Le Cun Y, Fergus R (2013) Regularization of neural networks using dropout. In: International conference on machine learning, PMLR, pp 1058–1066
53. Weigend AS (2018) Time series prediction: forecasting the future and understanding the past. Routledge
54. Xingjian SHI, Chen Z, Wang H, Yeung DY, Wong WK, Woo Wc (2015) Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems, pp 802–810
55. Zhao H, Shi J, Qi X, Wang X, Jia J (2017) Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2881–2890
56. Zhu J, Hang SU, Zhang B (2020) Toward the third generation of artificial intelligence. *Scientia Sinica Informationis* 50(9):1281. <https://doi.org/10.1360/SSI-2020-0204>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits any noncommercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if you modified the licensed material. You do not have permission under this license to share adapted material derived from this chapter or parts of it.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Forecasting Tropical Instability Waves Based on Artificial Intelligence



Gang Zheng, Xiaofeng Li, Ronghua Zhang, and Bin Liu

1 Sea Surface Temperature and Tropical Instability Waves

With the development of earth observation satellites and various active and passive sensors, massive ocean data have been acquired. For instance, the cumulative satellite data archive volume at the National Oceanic and Atmospheric Administration's National Centers for Environmental Information reached ~7.5 petabytes in 2016. The projected volume by 2030 is ~50 petabytes [32]. Many oceanic gridded products (e.g., sea surface temperature (SST), sea surface winds, and sea surface height) have been generated from such deluges of satellite data. These products provide an unprecedented golden opportunity for in-depth research and demonstrate the urgent need to develop effective methods to explore time-series data. SST can be measured from space and has the longest history among satellite-derived oceanic products widely used to reveal the evolution of various important oceanic phenomena such as El Niño, western boundary current, and tropical instability wave (TIW) [18]. Thus, SST is a critical parameter in understating physical oceanography, biological

G. Zheng · B. Liu

State Key Laboratory of Satellite Ocean Environment Dynamics, Second Institute of Oceanography, Ministry of Natural Resources, Hangzhou 310012, China

X. Li (✉) · R. Zhang

CAS Key Laboratory of Ocean Circulation and Waves, Institute of Oceanology, Chinese Academy of Sciences, Qingdao 266071, China
e-mail: lixf@qdio.ac.cn

R. Zhang

Qingdao National Laboratory for Marine Science and Technology, Qingdao 266237, China

B. Liu

College of Marine Sciences, Shanghai Ocean University, Shanghai 201306, China

Key Laboratory of Marine Ecological Monitoring and Restoration Technologies, Ministry of Natural Resources, Shanghai 200137, China

© The Author(s) 2023

X. Li and F. Wang (eds.), *Artificial Intelligence Oceanography*,
https://doi.org/10.1007/978-981-19-6375-9_2

oceanography, and atmosphere-ocean interaction; it is also a key input parameter for climate and weather modeling. The models in traditional statistical analysis have relatively limited complexity. This could make the models not work well, when used to model the oceanic phenomena that are complicated by nature.

Recently, another new research and application front that utilizes available tremendous data using deep learning (DL) technology has emerged. With DL, substantially more complex models can be built to mine rules deeply hidden in SST data. DL is a subset of machine learning that teaches computers to learn and make decisions or predictions based on input data. The deep neural network (DNN) technique is one of the most popular and powerful DL techniques, achieving successes in computer vision and speech recognition [15, 17]. A DNN is a multilayer neural network (NN). In most network layers of a DNN, input values are weighted, combined, and then transformed by an activation function to incorporate nonlinearity into the network. The output values of a network layer are linked to the next layer as input. All weights of a DNN are iteratively optimized by combining error backpropagation and gradient-based optimization to make the DNN suitable for finding the underlying relationship among its inputs and outputs. Such a multilayer structure allows the DNN to learn data features with multiple abstraction levels, which is impossible to imagine by the human brain [15]. Convolutional layers, named for their mathematical form, are a core type of network layer widely used in DNN models. In a convolutional layer, the output value at a specific site is calculated by weighting and combining the nearby sites' input values. Each output site shares the same weights. Thus, a convolutional layer has fewer weights to be optimized than a traditional fully connected layer that uses independent weights to connect all input and output sites. As a result, using the convolutional layer is particularly efficient in processing multi-dimensional data. Therefore, compared with traditional statistical models, DNN-based DL models can be much more complex and thus, after trained by a large quantity of sample data, can more efficiently learn the inherent characteristics behind them. Recently, DL applications in the prediction of future images in videos have drawn extensive attention in the field of computer vision [24, 35]. Ocean SST forecasting is similar to image prediction in videos, where future SST maps are forecasted based on the previous maps using a DL model. Because of the abovementioned similarities, we believe DL technology will help us to model oceanic phenomena in a different and promising way that is driven by ever-increasing big ocean data, although DL applications in oceanography and other geosciences just begin in recent years [31]. Therefore, using the large accumulated amount and long time series of satellite SST data, we can build a pure data-driven SST forecasting model that capture the spatial-temporal variations of a complicated yet important oceanic phenomenon, TIW, which has effects on transport of heat, mass and momentum in the ocean, air-sea and biophysical interactions, climate change, etc. As an internally generated ocean variability with time scales of approximately 15-40 days, TIWs produce large perturbations to physical and biological fields in the ocean, including SST. Furthermore, TIW-produced SST perturbations induce almost instantaneous atmospheric surface wind responses, forming TIW-scale interactions between the atmosphere and ocean. Although TIWs are dominantly controlled by the background ocean state, TIW evolution and pre-

dictability are affected by air-sea coupling at TIW scales. TIW forecasting is a challenging task because the spatial-temporal variation of TIW is significant, with large shape distortions and deformations and seasonal and interannual variability caused by the El Niño-Southern oscillation. Both high-resolution grids of the space domain discretization and realistic parameterizations of the relevant physical processes are required, when we use numerical model TIWs. All these lead to substantial difficulties in realistic simulation of TIW-related oceanic and atmospheric responses and the coupled air-sea interactions. Dynamical equation-based numerical modeling for TIWs requires not only high spatial resolution but also realistic parameterizations of the relevant physical processes. As a result, substantial difficulties exist in realistically simulating TIW-related atmospheric responses and the coupled air-sea interactions. Therefore, the data-driven model was applied to the SST field in the eastern equatorial Pacific Ocean to show that the TIW propagation can be forecasted by the data-driven model.

Satellite-derived SSTs have long been assimilated into numerical models to improve their forecasts. Recently, the NN-based strategy was proposed to perform a similar role as data assimilation. For example, in [27], a NN model is used to find the bias correction term in a numerical SST forecasting model. Compared with a numerical model, a data-driven forecasting model is much simpler and computationally efficient. The forecast made by a data-driven model relies only on prior data of minimal physical parameters or even one parameter. As another example, an SST pattern time series can be expanded as the sum of products of time-dependent principal component scales and corresponding space-dependent eigenvectors following empirical orthogonal function (EOF) analysis. Thus, the forecast of the SSTs at grids can be approximately reduced to the forecasts of several SST leading principal components [40]. Recently, NN models were developed to directly forecast SSTs without EOF approximations, including both site-specific and -independent models. A site-specific model considers the site difference, so makes SST forecasts with different NN models at different sites [26]. However, as each site needs building a NN model, the computation cost is high in the NN-training phase of a site-specific model, and sufficient NN-training samples are also required at each site. When use a site-independent model to forecast SSTs, different sites share the same SST forecast model [2, 42, 44]. This makes site-independent models more efficient. However, when forecasting a future SST at one site, these recent models only utilize the prior SST series at the very close neighboring sites. The models may have limitations over a large area because the SST patterns controlled by large-scale phenomena could be related to each other within a vast ocean area. Thus, maybe a wider SST series centering at a forecast site should be utilized to forecast the future SST.

In the following section, we introduce a multi-scale scheme DNN with four stacked composite layers for SST forecasting in the eastern equatorial Pacific Ocean, which overcomes the shortcomings of previous data-driven SST forecasting models. The idea of a multi-scale scheme has achieved notable successes in the field of computer vision, e.g., DNN applications in semantic segmentation [21, 33], but has not been explored in the oceanography field. Considering the natural differences among different sites, we also build a space-dependent but time-independent bias correction

map and then combine it with the multi-scale DNN to develop the final data-driven SST forecasting model, named the DL model for brevity.

The developed DL model was applied to forecast the SST pattern variations associated with the TIWs in the eastern equatorial Pacific Ocean. TIWs are an important ocean dynamic phenomenon in both the equatorial Pacific and Atlantic Oceans. They were first captured in the current meter records and infrared satellite images in the 1970s [6, 18]. One prominent characteristic of Pacific TIWs is its cusp-shaped and westward-propagating waves at both flanks of the equatorial Pacific cold tongue where the north flank has stronger signal. Previous studies have estimated the wavelength, period, and phase speed of TIWs from various data sources, and their values are typically within the ranges of 600 to 2000 km, 15 to 40 days, and 17-86 km/day [3, 4, 12, 13, 19, 28, 29, 38, 39]. Previous studies also suggested that the generation of TIWs could be the result of barotropic and baroclinic instability processes of the meridional and vertical shear among the westward South Equatorial Current, the eastward Equatorial Undercurrent, and the North Equatorial Counter Current [4, 23, 30, 34]. As a result, TIWs are inactive/active during boreal spring/fall, because the current shear is weaker/stronger at that time. Moreover, TIWs are suppressed and even indiscernible during strong El Niño years when the Pacific cold tongue and the related equatorial current shear are too weak and vice versa during La Nina years [39]. Conversely, TIWs also have feedback to the El Niño-Southern Oscillation, affecting its asymmetry and irregularity [1, 10, 11]. The physical and biological processes of TIWs are complicated. As has been widely illustrated, TIWs have a profound effect on the distribution of SST, sea surface height anomaly, chlorophyll- α , rain, salinity, and winds in the eastern equatorial Pacific Ocean [3, 14, 28, 29, 38]. TIW induces horizontal convection and vertical mixing in the upper sea [12, 13, 20, 25]. The mixing reaches even the lower half of the thermocline, a fact that is still not well considered in most physical models [20]. TIWs affect the equatorial chlorophyll- α concentration by transporting nutrients to the upper ocean [7, 9, 43]. Conversely, modeling analyses indicate that chlorophyll- α may modulate solar radiation in the upper ocean and weaken TIWs [36, 37]. TIWs also interact with the atmosphere because of the sea surface wind modulation caused by the TIW-induced SST anomalies [21, 41, 45–47]. Moreover, a spatial correlation between SST and cloud patterns is observed during the TIW seasons. The clouds appearing in the warm troughs of the TIWs are usually generated by cool low-level winds crossing the SST fronts and, in turn, dampen the TIW-induced SST anomalies by reducing the incident solar radiation over the warm troughs [5]. More comprehensive physical models for TIW studies are still ongoing, and many of the above-mentioned aspects should be considered to make the models more realistic [12, 14, 20, 36, 37, 45–47], which is a difficult challenge. In contrast, the time series of data contain all these factors. Owing to the strong data-mining ability, a data-driven DL model can automatically learn comprehensive rules of SST spatial-temporal variations from the data, and does not depict various complex processes by using physical equations.

2 Data and Model of SST Forecasting

There are two parts in the model: a DNN and a constant map. The DNN is multi-scale, having a network structure of four stacked composite layers for different spatial resolutions. The DNN uses the SSTs from the preceding fourteen steps to estimate the SSTs at the following step. The interval between the two steps is five days. The DNN-made estimation is followed by the correction with the constant map for reducing bias. The details are given below.

2.1 *Satellite Remote Sensing SST Data*

The DL model was built and tested with the SST products of Remote Sensing Systems. The products were made from both microwave and infrared sensor measurements. Our studied area is a rectangular region spanning from 120°W to 180°W in longitude and from 10°S to 10°N in latitude. The products from 2006 to 2019 were collected in our study. These 9-km-grid products were averaged to the 18-km-grid SST data. The SST data were divided into two parts according to time. The first part (1st Jan 2006–31st Dec 2009) and the second part (1st Jan 2010–31st Mar 2019) were used to build and test the DL model, respectively. By considering that TIWs have about a fifteen-to-forty-days temporal scale, the time step of the DL model is set to five days. Based on the preceding thirteen and current-step SST maps, the DL model forecasts the SST map at the following future time step, the fifth step. Therefore, a sample in our study is an SST series consisting of sequent fifteen SST maps. Then, the SST series was shifted day by day to get the second, third, fourth, etc. The DL model forecasts the fifteenth-step SST map in each series based on the first-fourteen-steps SST maps. The forecasted SST map was then validated using the series's fifteenth-step SST map. Approximately one thousand four hundred series were generated in the first part of the SST data, and three thousand four hundred series samples were generated in the second part of the SST data. It should be noted that a significant El Niño event occurred during the period of 2014–2016, which is covered by the second part of the SST data.

2.2 *Architecture and Training of the DL Model*

As shown in Fig. 1, the DL model is composed of a trained multi-scale DNN and a time-independent bias-correction map. The DNN is a stack of four composite layers. And each composite layer has four cascaded convolutional layers. In this region, the value of SSTs range from 16°C to 34°C, and the range was rescaled to $[-1, 1]$. In order to feed to the corresponding composite layers at different stack levels, a 2×2 average pooling operation was used to downsample the SST maps. These composite

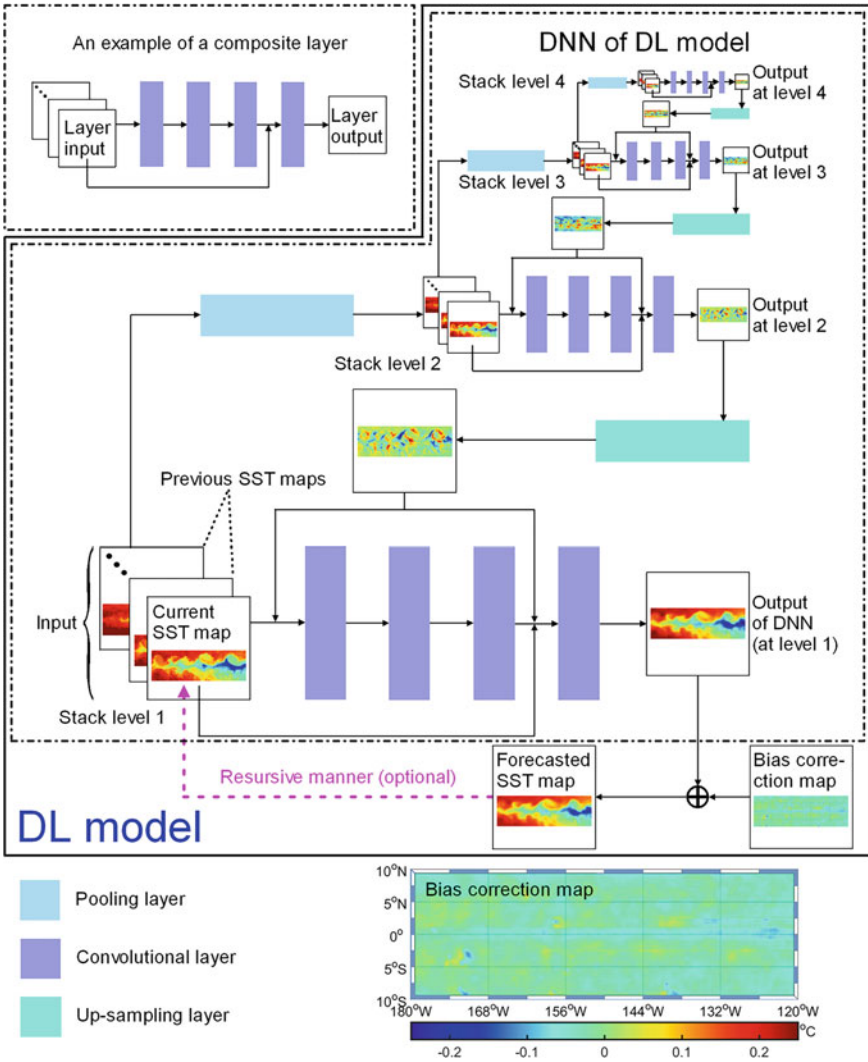


Fig. 1 The DL model receives SST maps at the previous and current time steps and then outputs the SST map at the future time step. The major part of the DL model is a DNN having four stacked composite layers. The bias correction map is added to the DNN output to obtain the forecasted SST map

layers process the SST maps at different spatial resolutions. The lower the stack level, the higher the resolution. Except the top level, each higher resolution composite layer at a lower stack level requires the output of the composite layer at the upper stack level. And the output need to be up-sampled. The input of the DNN consists of 14 SST maps at the current step and the previous 13 steps. Considering the input SST map at the current time step is more correlative to the future SST map, the DL model also directly linked the input SST map at the current step to the last convolutional layer along with the up-sampled output of the lower resolution composite layer at the upper stack level. The rectified linear unit function has better error gradient propagation [8], so it was used as the activation for the first three convolutional layers of each composite layer. The tanh activation was used for the last convolutional layer of each composite layer except for the bottom composite layer. The tanh activation rescales the output of each composite layer to $[-1, 1]$ that matches the input range of the higher resolution composite layer where the output is fed after the up-sampling. The activation of the last convolutional layer of the bottom composite layer is a linear function and is used to make the DNN output unbounded. The four convolutional layers of each composite layer include 8, 16, 32 and 1 channels. The kernel sizes of the four convolutional layers of the top composite layer are all 3×3 . Those of the other composite layers are 5×5 , 3×3 , 3×3 and 5×5 , respectively.

For a general network layer, one site in the output map is connected to multiple sites in the input map. Thus, the value at the output site is only dependent on the values at these input sites rather than the whole input map. These input sites form the receptive field of the output site. For instance, the input sites inside a receptive field of a convolutional layer are weighted and connected to the corresponding output site by the convolution kernel. The receptive field can be enlarged by using average pooling layers to down-sampling the inputs before feeding them to the subsequent layer. Then, the output can be treated with the same number of up-sampling layers to restore the resolution. SST variations in different locations may be correlated by oceanic phenomena with large scales. Considering this, we use the SST series of a wider area to forecast the SST at the area center. Therefore, the DNN is designed to be multi-scale to obtain the wider receptive field. After three down- and up-samplings among the four composite layers, the receptive field size of the whole DNN extended by about twelve times. For forecasting TIWs, this size is large enough.

The SST-map-series samples for building the DNN were divided into the training and validation datasets, according to the ratio of 3:1. The input area is set to be larger than the output (forecast) area in order to ensure that the input area covers the whole DNN receptive field. The following loss function is used to optimize the DNN:

$$Loss = \sum_{k=1}^K \sum_{(m,n) \in \text{Grids}_{\text{output}}} \left(SST_{\text{output}}^{(k)}(m, n) - SST_{\text{true}}^{(k)}(m, n) \right)^2 \quad (1)$$

where $SST_{\text{true}}^{(k)}(m, n)$ is the fifteenth-step satellite SST map. k denotes the k th sample, and K is the sample number of the training or validation dataset. (m, n) denote the grid (m, n) of the output area, and $\text{Grids}_{\text{output}}$ is the grid set. $SST_{\text{output}}^{(k)}(m, n)$ is the

DNN-forecasted SST. The Adam algorithm [16] was used to optimize the DNN parameters on the training dataset, and the maximum number of epochs was set to be 2500. The optimization was implemented using the CUDA technique on a NVidia Quadro M4000. The memory of the graphics card is eight GB. In order to avoid overfitting to the training dataset, the loss value on the validation dataset was also calculated during the optimization procedure. The smallest loss value (the validation dataset) was achieved at the one hundred and twenty-ninth epoch costing about one hundred and fourteen minutes. The parameter values corresponding to the smallest loss value were adopted.

Parameters in convolutional layers are the same for different sites. In addition, there is no optimizable parameter in both average pooling and up-sampling layers. Thus, the DNN is independent of the site. However, the environmental background of the study area is inhomogeneous. There is a spatial trend that the SST is overall higher in the west than in the east. This may cause evolution differences among the SST pattern in different areas. Therefore, an SST correction map is included in the DL model, which is added to the DNN-forecasted SST map to make the final forecast (Fig. 1). By using the samples during the training period, this SST correction map is generated by calculating the bias of the DNN at each grid after the optimization.

The operating efficiency of the developed DL model is very high. It only takes about 1 minute to forecast SSTs for all testing samples on an ordinary desktop computer.

3 SST Forecast of TIW Motion Using the DL Model During the Testing Period (2010/01–2019/03)

Figure 2a–c shows the satellite SST maps of the testing period, and Fig. 2d–f shows the SST forecast result by the DL model. The maps are matched closely in shape, where the most notable feature is the characteristic of TIWs that propagate westward. The characteristic is cusp-shaped and irregular deformations.

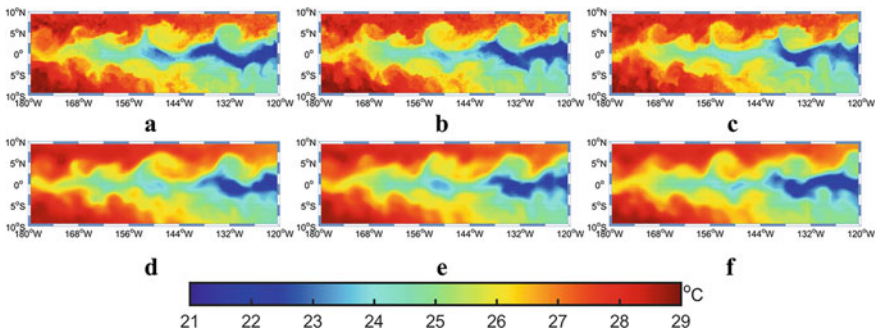


Fig. 2 Satellite SST maps a to c and DL-forecasted SST maps d to f

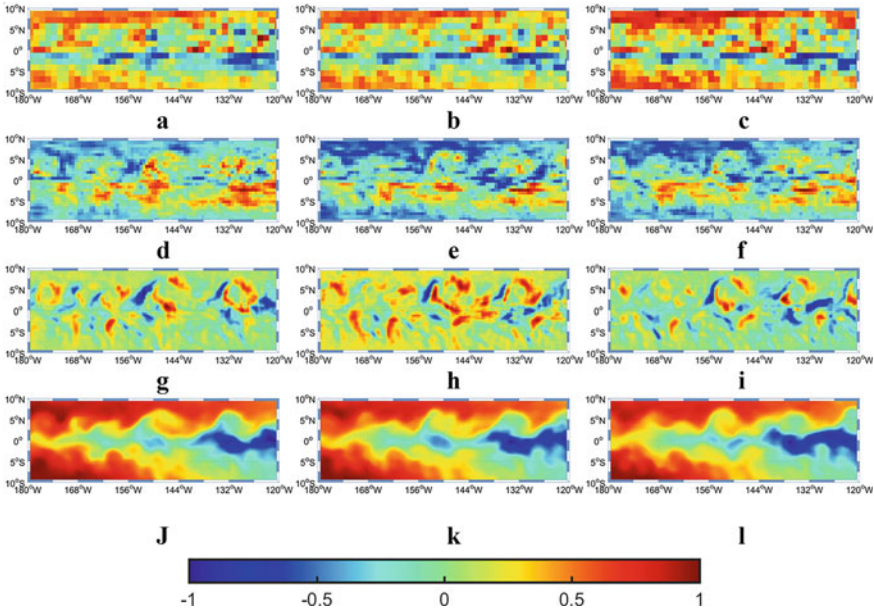


Fig. 3 The outputs at three consecutive time steps (the same to the steps in Fig. 2) of the fourth-(top)-stack-level composite layer of the DNN **a** to **c**, the third-stack-level composite layer, the third-stack-level composite layer **d** to **f**, the second-stack-level composite layer **g** to **i**, and the first-(bottom)-stack-level composite layer **j** to **l**

Figure 3 shows the output of the four composite layers in Fig. 1 at three continuous time steps and visualized from the first (bottom) to the fourth (top) stack level of the DNN. For the sake of clarity, the coarse-resolution results at higher levels are converted to the initial resolution using the nearest neighbor interpolation method. Then the results are rescaled to $[-1, 1]$. All outputs show a westward propagating signal similar to the satellite SST maps as shown in Fig. 2a-c. These maps are extracted from the DNN network during the training period(2006-2009) and show the temporal and spatial characteristics of TIW. Related parameters in the network are learned by DNN from sample data. The TIWs’ motion can be forecasted by these features.

The forecasted and satellite SST maps’ meridional averages (MAs) are calculated. The maximum detrended cross-correlation between the MAs at the current time step and the next step along the equator can estimate the westward propagation speed of the SST pattern.

During TIW Seasons, MAs calculated by SST can reflect the westward propagation signal of the SST pattern. The forecast area exists an approximately linear zonal trend of SST, which is warm in the western part and cold in the eastern part. Moreover, the trend is superimposed with the above signal. An instance of two zonal sequences of SST MAs at the longitudes of the grids of the forecast area and at two consecutive

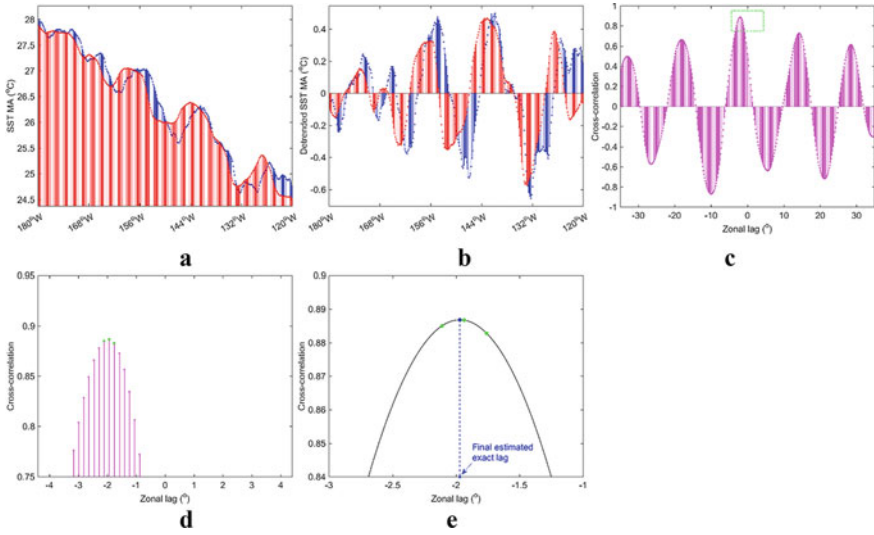


Fig. 4 Procedure of estimating SST pattern zonal speed: **a** Two zonal sequences of SST MAs at the longitudes of the grids are calculated from the SST maps at two times, where blue and red denote the first time and the second time, respectively. The first sequence is satellite MAs, and the second sequence is forecasted by the DL model. **b** Two sequences of SST MAs after their linear trends are removed. **c** Cross-correlations of the two sequences after the linear trends are removed. **d** An enlarged image of the green box in Fig. 4c, where the three green points are the maximum discrete cross-correlation and the two cross-correlations at the neighboring discrete zonal lags. **e** The three green points can be interpolated with a quadratic curve (black line), and the zonal lag corresponding to the peak of the curve is considered as the exact zonal lag with the maximum cross-correlation. The speed can then be estimated by dividing the exact zonal lag by the time interval

time steps is given (Fig. 4a). The red lines represent the MAs of the DL-forecasted SST map after five days (one time step), and the blue lines represent the MAs of the satellite SST map sequence. The westward propagation of the signal becomes more obvious after removing the linear zonal trend of the SST MAs (Fig. 4b). The two sequences of detrended SST MA series' cross-correlations can be calculated at the discrete zonal lags (Fig. 4c), and can find the discrete lag with the maximum cross-correlation and its two neighboring discrete lags (Fig. 4d). A quadratic curve can interpolate the cross-correlation of three discrete lags. The peak lag of the interpolated curve is considered to be the exact lag of the maximum cross-correlation between two non-trending SST MA sequences (Fig. 4e). In mathematical form, this is

$$lag_{\text{exact}} = \frac{1}{2} \cdot \frac{y_1(lag_2^2 - lag_3^2) + y_2(lag_3^2 - lag_1^2) + y_3(lag_1^2 - lag_2^2)}{y_1(lag_2 - lag_3) + y_2(lag_3 - lag_1) + y_3(lag_1 - lag_2)} \quad (2)$$

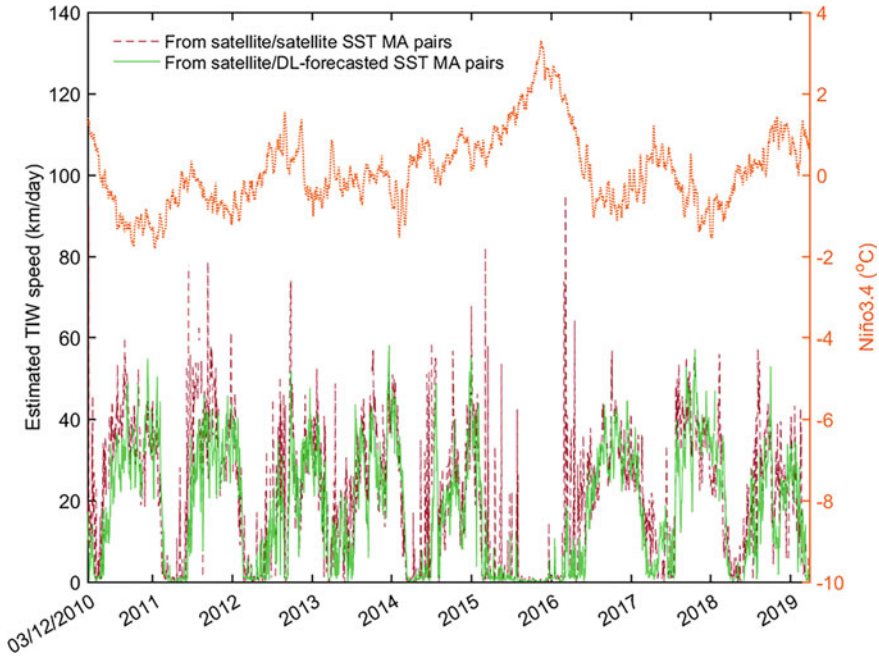


Fig. 5 Temporal variation of the SST pattern associated with TIW westward propagation during the testing period. We calculated the MAs of the satellite and forecasted SST maps and then estimated the speed of the SST pattern westward propagation based on the maximum detrended cross-correlation along the equator between the MAs of the satellite SST map at the current time step and those of the satellite or predicted SST map (brown dashed curve: the speed calculated by satellite/satellite pairs, green solid curve: the speed calculated by satellite/DL-predicted pairs, orange dotted curve: the daily Niño3.4 index)

where lag_1 , lag_2 , and lag_3 are the three discrete lags, and, y_1 , y_2 , and y_3 are the corresponding cross-correlations. Finally, the propagation speed can be obtained by dividing the exact lag by the time interval.

Figure 5 shows the estimated speeds mainly ranges from 0 to 100 km/day [3, 4, 12, 13, 19, 28, 29, 38, 39]. The green solid curve represents the SST pattern propagation velocity predicted by the DL model. The red dashed curve represents the velocity estimated by the satellite/satellite SST MA pairs. The two curves are in good agreement. Both curves show very consistent TIW seasonal fluctuations. In the TIW season, TIW controls the motion of the SST pattern. Thus, the DL-forecasted SST pattern propagation velocity can be regarded as the TIW speed. Nevertheless, the SST pattern is inert, and there is no apparent westward motion in the no- or weak-TIW seasons.

The DL model can also forecast recursively. In this recursive frame, the forecasted SST, the present satellite SST, and the previous 12 satellite SSTs were used to forecast the SST at the second recursive step, and then, the two forecasted SSTs, the current satellite SST and the previous 11 satellite SSTs were utilized to forecast the SST at

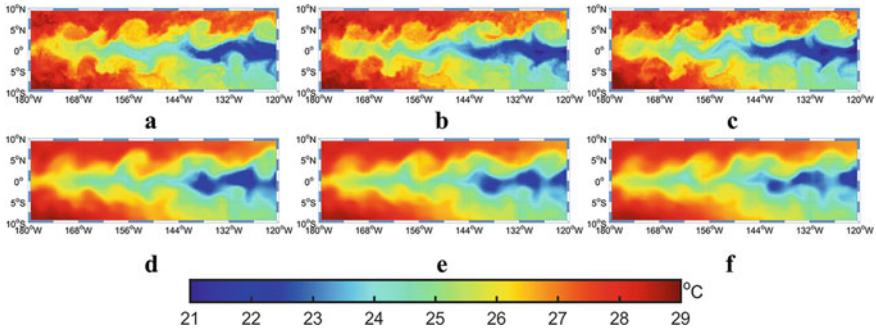


Fig. 6 Satellite-observed SST **a** to **c** and DL model-forecasted SST **d** to **f**

the third recursive step. Therefore, the DL model recursively forecasts the SST in the subsequent steps (the fourth, fifth, sixth, etc. recursive steps). Figure 6 shows an example of the recursively forecasted SST maps at the subsequent three time steps after the final time step in Fig. 2. As can be seen from the figure, the DL model can still work well and forecast the TIWs' westward motion in general.

4 Interannual Variation in TIW Westward Propagation

The daily Niño3.4 index data were also overlaid on Fig. 5, and denoted by orange dotted curve. The data was provided by the KNMI (the Royal Netherlands Meteorological Institute) Climate Explorer. Fig. 5 shows that the DL-forecasted TIW speed values and the Niño3.4 index values are 180 degrees out of phase. There is a major El Niño event from 2014 to 2016, and the TIW speeds were almost zero for the weakening of meridional SST gradients during this time. The measurements of mooring and Argo float from 2000 to 2010 also validate this fact, in which TIW kinetic energy and occurrence probability show negative correlation with the Niño3.4 index [11]. The correlation coefficient between the Niño3.4 index values and the speed values estimated from satellite/satellite SST MA pairs is -0.38 , with a P-value close to zero and a 95% confidence interval of $(-0.35, -0.41)$. The corresponding statistic results for the DL-forecasted speeds are -0.53 , with a P-value close to zero and $(-0.50, -0.55)$.

5 Zonally Westward Propagation of TIWs

Figure 7 gives the zonal TIW westward propagation speeds at 2-degree latitude bands, which were estimated from the satellite/satellite maps and the satellite/DL-forecasted SST maps, respectively. As can be seen from the figure, the estimated speed distribu-

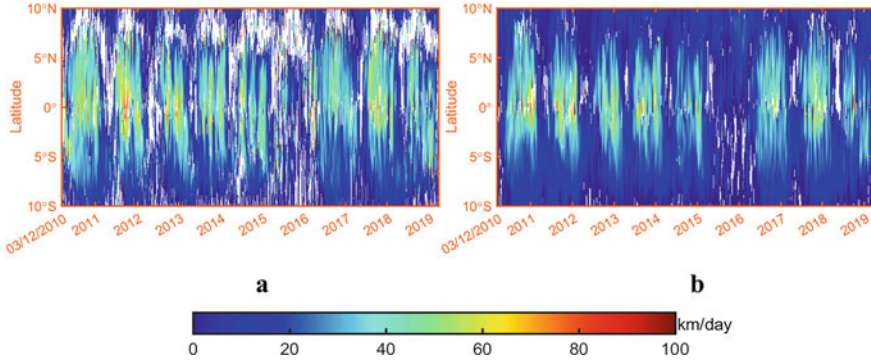


Fig. 7 Zonal TIW westward propagating speeds at 2-degree latitude bands. **a** Distribution of speeds estimated from satellite/satellite SST MA pairs and distributions of forecasted speeds estimated from **b** satellite/DL-forecasted SST MA pairs. The white blanks denote outliers beyond the range from 0 to 100 km/day

tions are consistent with each other and their temporal fluctuations are similar during TIW seasons. The fluctuations are also similar to the curves in Fig. 5. Furthermore, the equatorial bands have higher speeds than the higher-latitude bands. All these results are in agreement with the previous findings for the reason that TIWs at different latitudes are controlled by different dynamic mechanisms with their speeds determined by equatorial wave processes [22, 38].

6 Accuracy During the Testing Period (2010/01–2019/03)

The root mean square error (RMSE) and bias variation of the DL model over time were calculated during the testing period and are given in Fig. 8. From the figure, it can be seen that the RMSE and bias are generally stable. The RMSE fluctuates between 0.15 °C to 0.45 °C, while the bias fluctuates between -0.15 °C to 0.15 °C. Due to the rapid change of the SST pattern, the RMSE of the DL model is larger during the TIW seasons (Fig. 8a). There are approximately 3300 samples at each grid point. The RMSE and bias at each grid were calculated, and the RMSE and bias spatial distributions of the DL model are given in Fig. 9. The RMSE of the cold tongue area is higher than other areas. This is caused by the large spatial gradient and fast temporal variation of the SST in the cold tongue area. In the study area, the global RMSE of all grids and all samples is 0.29 °C and the bias is -0.01 °C.

For the recursive forecasting, the global RMSE and bias of the DL model from 5 days to 150 days after the current time step (i.e., recursive steps 1 to 30) are given in Fig. 10. It can be found that the DL model’s accuracy declines with the evolution of time. It should be noted that there will be no satellite SST in the model input after 14

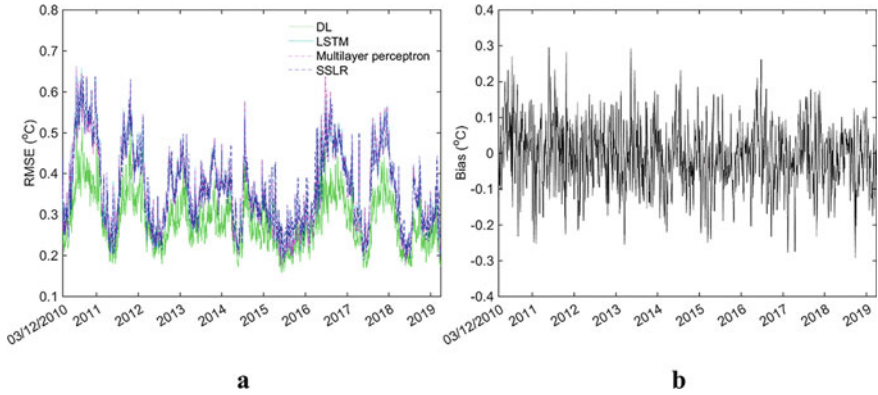


Fig. 8 RMSE **a** and bias **b** temporal trends. The RMSE and bias temporal trends were calculated sample by sample from the forecasting errors at all grids

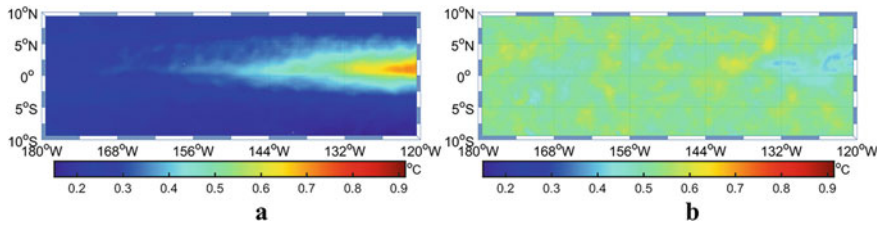


Fig. 9 RMSE **a** and bias **b** spatial distributions. The RMSE and bias spatial distributions were calculated grid by grid from the forecasting errors of all samples

recursive steps. Even so, the RMSE does not grow rapidly and is still smaller than 0.80 °C at the 15th recursive step. Meanwhile, the magnitude of the DL model’s bias is also smaller than 0.10 °C at the 30th recursive step.

7 Conclusions

In this chapter, a data-driven DL SST forecasting model using the DNN technique was built. The DL model accurately forecasted the spatial-temporal variation of the SST pattern with a RMSE of 0.29 °C and the TIW’s propagations that agree well with actual satellite observations.

The DL model is different from previous models. The DL model consists of a multi-scale DNN with four stacked composite layers and a time-independent but site-dependent bias correction map. In this design, the DL model takes the spatial dependence of a site-specific forecast over a large surrounding area and the bias correction of the DNN at different sites into consideration. The DL model was tested for nine years without overlapping with the training period. The results show that

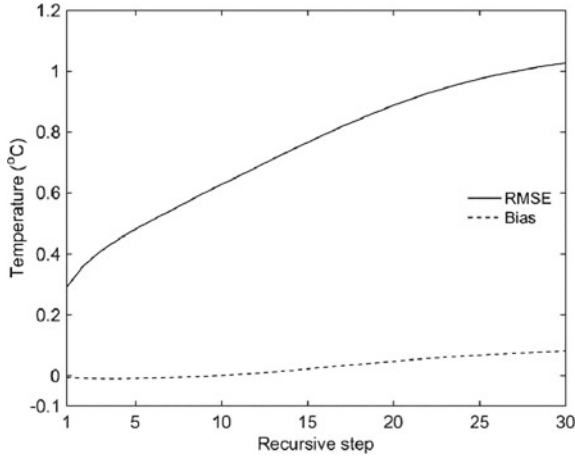


Fig. 10 The global RMSE and bias of DL model implemented recursively concerning the number of recursive steps. In the recursive model, the DL-forecasted SST at a future time step is fed back to the model input to forecast the SST at the next future time step. The recursive steps from 1 to 30 are correspond to 5 days to 150 days after the current time step. After 14 recursive steps, there is no satellite SST map at the model input, and all input SST maps are from the model’s forecast. The global RMSE and bias were calculated from the forecasting errors of all samples at all grids at each recursive step

the DL model effectively forecasts the SST variation associated with TIWs. The DL-forecasted TIW speed is in good agreement with that estimated from the satellite SST maps. Both of the speeds present the consistent seasonal cycle and interannual modulation, and the interannual modulation is negatively correlated with the Niño3.4 index. TIW speeds are higher in equator than other latitudes. The DL model can also forecast SSTs at future steps in a recursive manner, although the accuracy degrades with time for the loss of actual satellite SST input.

The developed model results show DNN’s great potential for marine forecasting utilizing gridded data. Compared with numerical forecasting models, DL forecast models are straightly driven by real measurements and elude the complex process, including model parameterizations and approximations, various physical equations, and a substantial computational burden. DL models are able to forecast accurately with the help of a few physical parameters’ prior information. In our case, only one SST parameter was used. Almost all of the DL model’s computational cost is spent on the iterative optimization of the weights. Emerging technologies on hardware, e.g., CUDA, can easily speed up this learning procedure. If the DNN has been trained and obtained the bias correction map, the DL model can make an efficient forecast with no iteration. Therefore, it can work very rapidly. In our case, it only takes about one minute to forecast the SST pattern of the testing period by an ordinary desktop computer. As far as DNN is a data-driven technology, whether training or

using, sufficient data is always the basic requirement. Fortunately, sufficient data and DNN's outstanding learning capability fully cater to the growing amount of marine satellite observations in the era of remote sensing big data.

References

1. An SI (2008) Interannual variations of the tropical ocean instability wave and ENSO. *J Clim* 21(15):3680–3686. <https://doi.org/10.1175/2008JCLI1701.1>
2. Aparna SG, D'Souza S, Arjun NB (2018) Prediction of daily sea surface temperature using artificial neural networks. *Int J Remote Sens* 39(12):4214–4231
3. Chelton DB, Wentz FJ, Gentemann CL, Szoeké RD, Schlax MG (2000) Satellite microwave SST observations of transequatorial tropical instability waves. *Geophys Res Lett* 27(9):1239–1242
4. Contreras Robert F (2011) Long-term observations of tropical instability waves. *J Phys Oceanogr* 32(9):2715–2722
5. Deser C, Wahl S, Bates JJ (1993) The influence of sea surface temperature gradients on stratiform cloudiness along the equatorial front in the Pacific Ocean. *J Clim* 6(6):1172–1180
6. Diing W, Hisard P, Katz E, Meincke J, Miller L, Moroshkin KV, Philander G, Ribnikov AA, Voigt K, Weisberg R (1975) Meanders and long waves in the equatorial Atlantic. *Nature* 257(5524):280–284
7. Evan W, Strutton PG, Chavez FP (2009) Impact of tropical instability waves on nutrient and chlorophyll distributions in the equatorial Pacific. *Deep-Sea Res Part I* 56(2):178–188
8. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. *JMLR Workshop and Conference Proceedings*, pp 315–323
9. Gorgues T, Menkes C, Aumont O, Vialard J, Dandonneau Y, Bopp L (2005) Biogeochemical impact of tropical instability waves in the equatorial Pacific. *Geophys Res Lett* 32
10. Holmes R, McGregor S, Santoso A, England M (2019) Contribution of tropical instability waves to ENSO irregularity. *Clim Dyn* 52(3–4):1837–1855. <https://doi.org/10.1007/s00382-018-4217-0>
11. Imada Y, Kimoto M (2012) Parameterization of tropical instability waves and examination of their impact on ENSO characteristics. *J Clim* 25(13):4568–4581
12. Inoue R, Lien RC, Moum JN (2012) Modulation of equatorial turbulence by a tropical instability wave. *J Geophys Res: Ocean*
13. Jochum M, Cronin MF, Kessler WS, Shea D (2007a) Observed horizontal temperature advection by tropical instability waves. *Geophys Res Lett* 34(9)
14. Jochum M, Deser C, Phillips A (2007) Tropical atmospheric variability forced by oceanic internal variability. *J Clim* 20(4):765–771
15. Jordan MI, Mitchell TM (2015) Machine learning: trends, perspectives, and prospects. *Science* 349(6245):255–260
16. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
17. Lecun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436
18. Legeckis R (1977) Long waves in the eastern equatorial Pacific Ocean: a view from a geostationary satellite. *Science* 197(4309):1179–1181
19. Legeckis R, Brown CW, Chang PS (2002) Geostationary satellites reveal motions of ocean surface fronts. *J Marine Syst* 37(1–3):3–15
20. Liu C, K'Hi A, Liu Z, Wang F, Stammer D (2016) Deep-reaching thermocline mixing in the equatorial Pacific cold tongue. *Nat Commun* 7:11576
21. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: 2015 IEEE Conference on computer vision and pattern recognition (CVPR)

22. Lyman JM, Johnson GC, Kessler WS (2007) Distinct 17- and 33-day tropical instability waves in subsurface observations. *J Physl Oceanogr* 37(4):855
23. Masina S, Philander S, Bush A (1999) An analysis of tropical instability waves in a numerical model of the Pacific Ocean: 2. Generation and energetics of the waves. *J Geophys Res: Ocean* 104(C12)
24. Mathieu M, Couprie C, LeCun Y (2016) Deep multi-scale video prediction beyond mean square error 1511.05440
25. Moum JN, Lien RC, Perlin A, Nash JD, Wiles PJ (2009) Sea surface cooling at the Equator by subsurface mixing in tropical instability waves. *Nat Geosci* 2(11):761–765
26. Patil K, Deo MC (2018) Basin-scale prediction of sea surface temperature with artificial neural networks. *J Atmos Ocean Technol* 35(7):1441–1455
27. Patil K, Deo MC, Ravichandran M (2016) Prediction of sea surface temperature by combining numerical and neural techniques. *J Atmos Ocean Technol* 33(8):1715–1726
28. Polito PS, Ryan JP, Liu WT, Chavez FP (2001) Oceanic and atmospheric anomalies of tropical instability waves. *Geophys Res Lett* 28(11):2233–2236
29. Qiao L, Weisberg RH (1995) Tropical instability wave kinematics: observations from the tropical instability wave experiment. *J Geophys Res Ocean* 100(C5):8677–8693
30. Qiao L, Weisberg RH (1998) Tropical instability wave energetics: observations from the tropical instability wave experiment. *J Phys Oceanogr* 28(2):345–360
31. Reichstein M, Camps-Valls G, Stevens B, Jung M, Denzler J, Carvalhais N, Prabhat (2019) Deep learning and process understanding for data-driven Earth system science. *Nature* 566(7743):195
32. Ritchey N (2017) NCEI's long term archive: infrastructure, processes, volume and trend. The 44th meeting of the working group on information systems & services (WGISS)
33. Ronneberger O, Fischer P, Brox T (2015) U-Net: convolutional networks for biomedical image segmentation. Springer International Publishing
34. Philander SGH (1976) Instabilities of zonal equatorial currents. *J Geophys Res* 81(21):3725–3735
35. Shi X, Chen Z, Wang H, Yeung DY, Wong WK, Woo WC (2015) Convolutional LSTM network: a machine learning approach for precipitation nowcasting. MIT Press
36. Tian F, Zhang R, Wang X (2018) A coupled ocean physics-biology modeling study on tropical instability wave-induced chlorophyll impacts in the Pacific. *J Geophys Res: Ocean*
37. Tian F, Zhang R, Wang X (2019) A positive feedback onto ENSO due to tropical instability wave (TIW)-induced chlorophyll effects in the Pacific. *Geophys Res Lett* 46(2):889–897
38. Tong L, Lagerloef G, Gierach MM, Kao H, Yueh S, Dohan K (2012) Aquarius reveals salinity structure of tropical instability waves. *Geophys Res Lett* 39
39. Willett CS, Leben RR, Lavin MF (2006) Eddies and tropical instability waves in the eastern tropical Pacific: a review. *Prog Oceanogr* 69(2/4):218–238
40. Wu A, Hsieh WW, Tang B (2006) Neural network forecasts of the tropical Pacific sea surface temperatures. *Neural Netw* 19(2):145–154
41. Xie Shang-Ping (2004) Satellite observations of cool ocean atmosphere interaction. *Bull Am Meteorol Soc* 85(2):195–208
42. Yang Y, Dong J, Sun X, Lima E, Mu Q, Wang X (2018) A CFCC-LSTM model for sea surface temperature prediction. *IEEE Geosci Remote Sens Lett* 15(2):207–211. <https://doi.org/10.1109/LGRS.2017.2780843>
43. Yoder JA, Ackleson SG, Barber RT, Flament P, Balch WM (1994) A line in the sea. *Nature* 371(6499):689–692
44. Zhang Q, Wang H, Dong J, Zhong G, Sun X (2017) Prediction of sea surface temperature using long short-term memory. *IEEE Geosci Remote Sens Lett* 14(10):1745–1749
45. Zhang R (2014) Effects of tropical instability wave (TIW)-induced surface wind feedback in the tropical Pacific Ocean. *Clim Dyn* 42(1–2):467–485
46. Zhang R (2016) A modulating effect of tropical instability wave (TIW)-induced surface wind feedback in a hybrid coupled model of the tropical Pacific. *J Geophys Res: Ocean* 121(10)
47. Zhang R, Busalacchi AJ (2008) Rectified effects of tropical instability wave (TIW)-induced atmospheric wind feedback in the tropical Pacific. *Geophys Res Lett* 35(5):94–96

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits any noncommercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if you modified the licensed material. You do not have permission under this license to share adapted material derived from this chapter or parts of it.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Sea Surface Height Anomaly Prediction Based on Artificial Intelligence



Yuan Zhou, Chang Lu, Keran Chen, and Xiaofeng Li

1 Significance of Sea Surface Height Anomaly Prediction

Sea surface height anomaly (SSHA) is one of the essential parameters for investigating ocean dynamics and climate change [4, 6, 32] and indicates mesoscale ocean dynamics features such as currents, tides, ocean fronts, and water masses. In addition, it is an important parameter for marine disaster emergency response [10, 30, 37, 39]. Historically, sea-level changes were computed using the tide gauge data [15], [40]. Compared with sparsely distributed tide gauge stations, recently, the advent of satellite altimetry has enabled constant sea-level measurements that include the entire sea area. The use of altimetry allows to acquire a particular sea-level change to the terrestrial reference frame, thus offering high-precision data for studying sea level [1, 8, 13, 25, 34].

2 Review of SSHA Predicting Methods

To predict sea-level changes, numerous approaches based on satellite altimetry data have been proposed, which can be categorized according to the type of model used as physical-based [7, 16, 27] and data-driven models [3, 31].

Physical-based models estimate sea-level changes by statistically combining related physics and dynamics equations. Reference [7] predicted the annual and semi-annual global average sea-level changes by employing a hydrological model.

Y. Zhou · C. Lu · K. Chen
School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China

X. Li (✉)
CAS Key Laboratory of Ocean Circulation and Waves, Institute of Oceanology, Chinese Academy of Sciences, Qingdao 266071, China
e-mail: lixf@qdio.ac.cn

Reference [16] compared the future worldwide and regional sea-level anomalies caused by the emissions of greenhouse gas in the 21st century in accordance with the Hadley Center climate simulations (HadCM2 and HadCM3). Reference [27] attempted to predict worldwide seasonal sea-level anomalies ahead of 7 months by developing the dynamic atmosphere-ocean coupled model.

Data-driven models build mapping correlations between sea-level anomaly records by using statistical methods and are comparatively more accurate in the prediction of sea-level changes. Reference [3] forecasted seasonal sea-level anomalies in the North Atlantic based on the auto-regressive integrated moving average. By using the least-squares (LS) method, [31] forecasted the global average and gridded sea level anomalies of the eastern equatorial Pacific by using a polynomial-harmonic model. To assess the sea-level changes on the mid-Atlantic coast of the United States, [12] developed a technique based on empirical mode decomposition (EMD). References [33] and [28] respectively studied sea level anomalies based on the changes in earth's temperature and ice sheet flow by using semi-empirical methods. Reference [14] proposed a hybrid model that combines the EMD, LS, and singular spectrum analysis to predict long-run sea-surface anomalies in the South China Sea (SCS). Reference [17] employed evolutionary support vector regression (SVR) and gene expression programming to predict sea-level anomalies in the Caspian Sea by using previous sea-level records. They also combined SVR with empirical orthogonal function (EOF) [18], wherein they adopted SVR for predicting sea-level anomalies in the tropical Pacific and simultaneously applied EOF for extracting the main components with the aim to lower data dimensions.

Deep learning (DL) is a data-driven approach that is well adapted to nonlinear relationships. Recently, DL has been used for forecasting time-series data [9, 19, 22, 36, 42]. The similarity between SSHA pattern prediction and time-series data prediction has prompted researchers to propose a variety of data-driven models for predicting SSHAs by using DL. Reference [5] adopted an RNN network for predicting and analyzing sea-level anomalies. RNN networks outperform simple regression models by extracting and fusing the characteristics of the time dimension [11]. Reference [23] proposed a DL model that integrates long short-term memory (LSTM) network with an attention mechanism [2] to reliably predict SSHAs. Reference [38] developed the merged-LSTM model and showed that it is superior to several advanced machine learning approaches in predicting sea-surface anomalies.

The aforementioned RNN/LSTM forecasting strategies focus on temporal change modeling, where constant state data updating is practiced within every LSTM unit over time. They perform SSHA estimation by utilizing the former SSHA series either at a single site or at its tightly adjoining sites, thus failing to consider the veiled information of the SSHA series at the remaining associated remote positions. However, the sea level of a region is influenced by both nearby and distant areas. In addition, spatial deformations and temporal dynamics have equal significance in the prediction of forthcoming SSHA fields.

In addition, the former approaches can only predict the SSHA at a single grid and not the SSHA over the entire region. In terms of each grid in the region, the models must be trained. Thus, to forecast the value of all grids in the area, each grid must be

trained numerous times to acquire diverse network parameters. In addition, the high model storage and retraining time are not acceptable.

In the following section, we introduce an SSHA prediction method named multi-layer fusion recurrent neural network (MLFrnn). MLFrnn can be used in the temporal and spatial domains and can accurately predict the SSHA map for the entire region.

3 Multi-Layer Fusion Recurrent Neural Network for SSHA Field Prediction

We introduce a classical spatiotemporal forecasting architecture in Sect. 3.1 and then present the network architecture of MLFrnn Sect. 3.2. Finally, the multilayer fusion cell as the fundamental building block of MLFrnn is discussed in Sect. 3.3.

3.1 A Classical Spatiotemporal Forecasting Architecture: ConvLSTM

Spatiotemporal zone modeling is necessary for the prediction of SSHA fields, an unachievable task using the RNN/LSTM forecasting strategies. For extracting spatiotemporal information from time-series data, [35] developed ConvLSTM, a variant of LSTM where three-dimensional (3D) tensors are used for representing all of the inputs, gates, cell states, and hidden states. ConvLSTM utilizes convolution operators to capture spatial characteristics by clearly encoding the spatial information into tensors. The critical equations of ConvLSTM are as follows:

$$\begin{aligned}
 f_t &= \sigma (W_{xf} * X_t + W_{hf} * H_{t-1} + b_f) \\
 i_t &= \sigma (W_{xi} * X_t + W_{hi} * H_{t-1} + b_i) \\
 g_t &= \tanh (W_{xg} * X_t + W_{hg} * H_{t-1} + b_g) \\
 C_t &= f_t \odot C_{t-1} + i_t \odot g_t \\
 o_t &= \sigma (W_{xo} * X_t + W_{ho} * H_{t-1} + b_o) \\
 H_t &= o_t \odot \tanh (C_t)
 \end{aligned} \tag{1}$$

In this architecture, $*$ denotes the convolutional operator, \odot denotes the Hadamard product, and σ refers to a sigmoid activation function with a value of $[0, 1]$ describing the amount of to-be-transmitted information about the state. The cell state conveyance is dependent on i_t, g_t, f_t and o_t which helps avoid fast gradient disappearance by capturing the state in the memory and helps alleviate the problem of long-run dependency.

By stacking ConvLSTM, an encoder-decoder network can be obtained, as shown in Fig. 1. The ConvLSTM network takes X_t as the input of the first layer and \tilde{X}_t as the

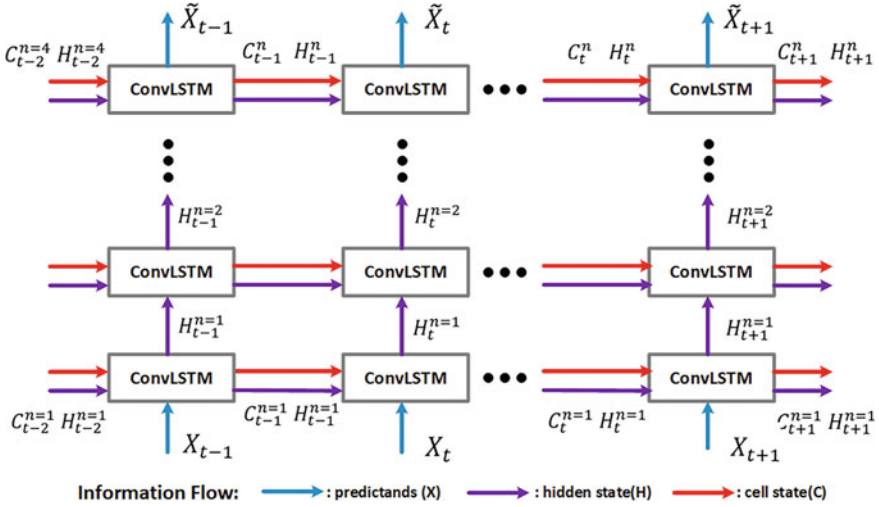


Fig. 1 ConvLSTM network

prediction result of the n -th layer. In addition, the ConvLSTM transmits hidden states in both vertical and horizontal directions. The update of the cell state is restricted within every layer of ConvLSTM and transmitted only in horizontal directions. The cell states of various layers are mutually irrelevant. Thus, every ConvLSTM layer will completely ignore cell states at lower layer. Moreover, the original layer (the lowest layer) does not reflect the memory contents of the deepest layer (the n -th layer) at the prior time step.

3.2 Architecture of MLFrnn

The states of different ConvLSTM layers are collectively irrelevant, and the interlayer connections are not explored sufficiently. To tackle this problem, the MLFrnn was developed in this study for the estimation of SSHA fields. As MLFrnn's fundamental building element, a novel type of multilayer fusion cells was designed that enable spatiotemporal trait acquisition at the nearby and remote positions for the SSHA fields. These features are delivered both horizontally and vertically.

Let's denote the SSHA field at time step t by X_t . The SSHA field can then be predicted based on the former t -days observations, as follows:

$$X'_{t+1} = \underset{X_{t+1}}{\operatorname{argmax}} p (X_{t+1} | X_1, X_2, \dots, X_t) \quad (2)$$

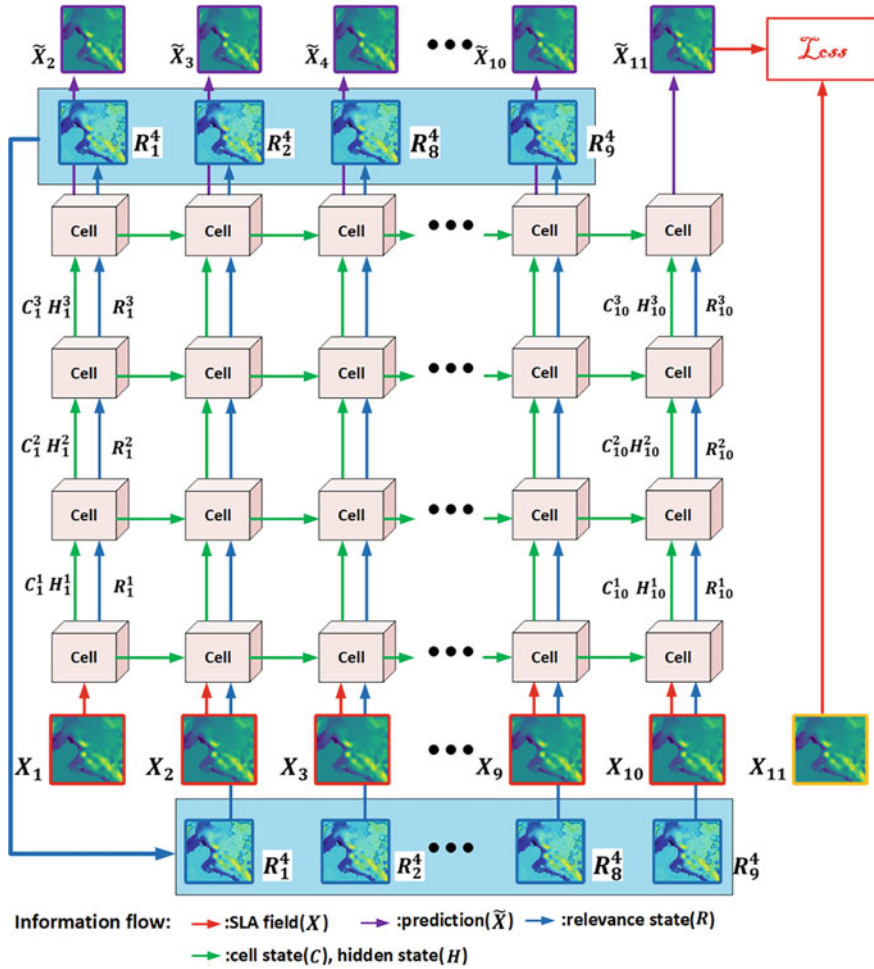


Fig. 2 MLFrnn model for forecasting SSHA fields. The red and purple arrows respectively indicate the directions of the information flows for inputs and forecasts, whereas the green arrows denote the flow direction of the cell/hidden state information that is transmitted both transversely and longitudinally. The blue arrows show the flow direction of relevance state information constituting the pristine-deepest layer connection

where $p(\cdot)$ represents a conditional probability and X'_{t+1} is the forecast SSHA field at time step $t + 1$.

Figure 2 presents the architecture of the MLFrnn model. It can extract cell state C_t^n and hidden state H_t^n (t denotes time and n represents the number of layers) for storing long-run and short-run memories. The cell state is expressed in vertical and horizontal ways rather than horizontally as in ConvLSTM. To study SSHA variations, the MLFrnn model fuses cell states and hidden states from diverse layers.

Additionally, in ConvLSTM, the pristine and deepest layers are irrelevant at the prior time step. To overcome this limitation, a supplementary relevance state R_t^n is added, which accomplishes the spatial trait storage for the SSHA fields and their straight transverse updating. In the subsequent time step, the pristine layer is fed with the relevance state of the deepest layer $R_t^{n=4}$, to allow a more abundant spatial feature.

The forecast of the SSHA field \tilde{X}_t is acquired by adopting a 1×1 convolution operation to the hidden state from the deepest layer $H_t^{n=4}$.

3.3 Multi-layer Fusion Cell

A recurrent forecasting module and a feature fusion module constitute a multilayer fusion cell, which completely fuses cell state, hidden state as well as relevance state to study the variations of SSHA fields (Fig. 3a). The structure of the recurrent forecasting module is presented, and then the feature fusion module is discussed.

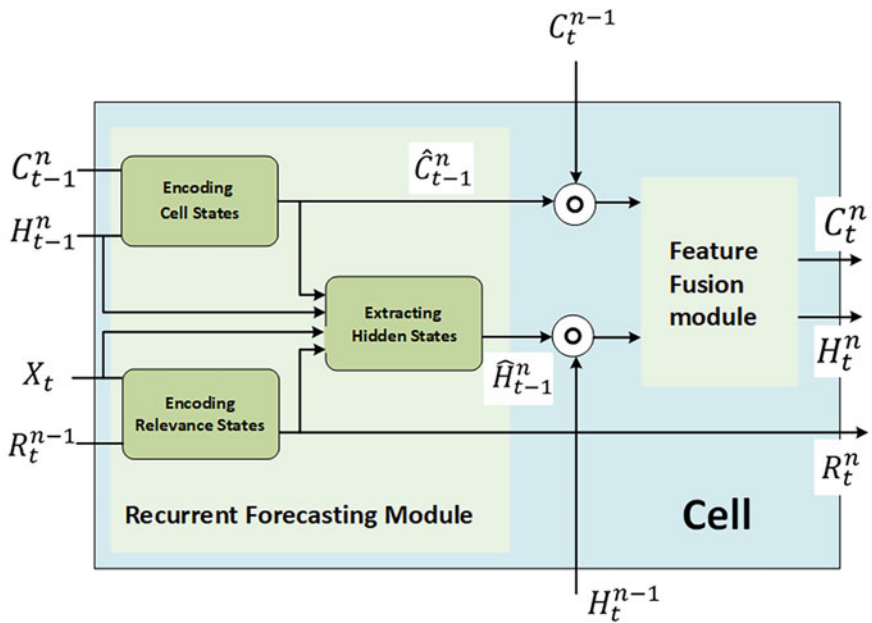
3.3.1 Recurrent Forecasting Module

The recurrent forecasting module can simultaneously encode the relevance state and the cell state. As shown in Fig. 3b, the structures in the red box encode the relevance state, and the structures in the blue box encode the cell state. Subsequently, the recurrent forecasting module extracts the hidden states transferred to the next time step cell via the output gate.

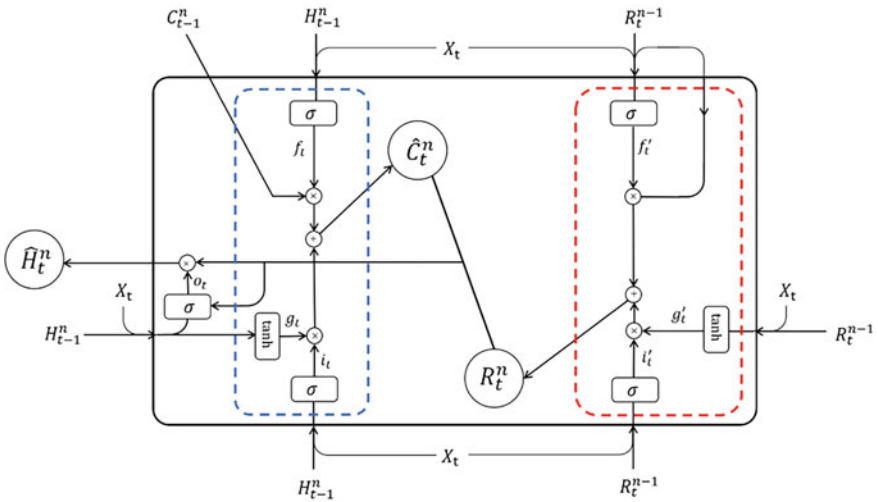
(1) Encoding relevance states: The time-series data X_t and relevance state R_t^{n-1} are considered as inputs. The input gate i'_t , forgetting gate f'_t , and input-modulation gate g'_t control the update of all the relevance states. The equations for encoding relevance states are:

$$\begin{aligned}
 f'_t &= \sigma \left(W'_{xf} * X_t + W_{rf} * R_t^{n-1} + b'_{f'} \right) \\
 i'_t &= \sigma \left(W'_{xi} * X_t + W_{ri} * R_t^{n-1} + b'_{i'} \right) \\
 g'_t &= \tan h \left(W'_{xg} * X_t + W_{rg} * R_t^{n-1} + b'_{g'} \right) \\
 R_t^n &= f'_t \odot R_t^{n-1} + i'_t \odot g'_t
 \end{aligned} \tag{3}$$

(2) Encoding cell states: The time-series data X_t and hidden state H_{t-1}^n are considered as inputs. The input gate i_t , forgetting gate f_t , and input-modulation gate g_t control the update of all the cell states. The equations for encoding cell states are:



a



b

Fig. 3 Multilayer Fusion Cell. **a** Structure of multilayer fusion cell: The multilayer fusion cell consists of a recurrent forecasting module and a feature fusion module. Concentric circles denote concatenation. **b** Recurrent forecasting module: The extraction of relevance states is shown in the red box. The extraction of cell states and hidden states is presented in the blue box

$$\begin{aligned}
f_t &= \sigma (W_{xf} * X_t + W_{hf} * H_{t-1}^n + b_f) \\
i_t &= \sigma (W_{xi} * X_t + W_{hi} * H_{t-1}^n + b_i) \\
g_t &= \tan h (W_{xg} * X_t + W_{hg} * H_{t-1}^n + b_g) \\
\widehat{C}_t^n &= f_t \odot C_{t-1}^n + i_t \odot g_t
\end{aligned} \tag{4}$$

(3) Extracting hidden states: The output gate of the recurrent forecasting module relies on \widehat{C}_t^n , H_{t-1}^n , and R_t^n . The output gate extracts the hidden state from the cell and relevance states as the output. The equations for extracting hidden states are:

$$\begin{aligned}
o_t &= \sigma (W_{xo} * X_t + W_{ho} * H_{t-1}^n + W_{co} * \widehat{C}_t^n + W_{ro} * R_t^n + b_o) \\
\widehat{H}_t^n &= o_t \odot \tan h (W_{1 \times 1} * [\widehat{C}_t^n, R_t^n])
\end{aligned} \tag{5}$$

where $*$ is the convolutional operator, \odot denotes the Hadamard product, and σ represents a sigmoidal activation function with a value of $[0, 1]$, describing the amount of each to-be-transmitted information about the state. The recurrent prediction module implements the spontaneous application of broad-spectrum receptive fields by using a series of convolutional operators so that the evolutions at the adjoining and distant sites of the SSHA domain can be portrayed [21].

3.3.2 Feature Fusion Module

Figure 4 shows the feature fusion module. In this study, \widehat{H}_{t-1}^n was concatenated with the hidden state from the former layer H_t^{n-1} as one of the inputs for feature fusion module. \widehat{C}_{t-1}^n was concatenated with cell state from the former layer C_t^{n-1} and taken as another input. The feature fusion module can be defined as

$$\begin{aligned}
f_t &= \sigma (W_{cf} * [C_t^{n-1}; \widehat{C}_t^n] + W_{hf} * [H_t^{n-1}; \widehat{H}_t^n] + b_f) \\
i_t &= \sigma (W_{ci} * [C_t^{n-1}; \widehat{C}_t^n] + W_{hi} * [H_t^{n-1}; \widehat{H}_t^n] + b_i) \\
g_t &= \tan h (W_{cg} * [C_t^{n-1}; \widehat{C}_t^n] + W_{hg} * [H_t^{n-1}; \widehat{H}_t^n] + b_g) \\
o_t &= \sigma (W_{co} * [C_t^{n-1}; \widehat{C}_t^n] + W_{ho} * [H_t^{n-1}; \widehat{H}_t^n] + b_o) \\
C_t^n &= f_t \odot Conv_{3 \times 3} [C_t^{n-1}; \widehat{C}_t^n] + i_t \odot g_t \\
H_t^n &= o_t \odot \tan h (C_t^n)
\end{aligned} \tag{6}$$

The cell and hidden states from different layers are integrated using the feature fusing module. While the states from the shallow layer comprise the nearby location information on a local scale, the states from the deep layer comprise the remote location information on a global scale. For the generation of subsequent SSHA fields, the local and worldwide information is integrated using the feature fusing module.

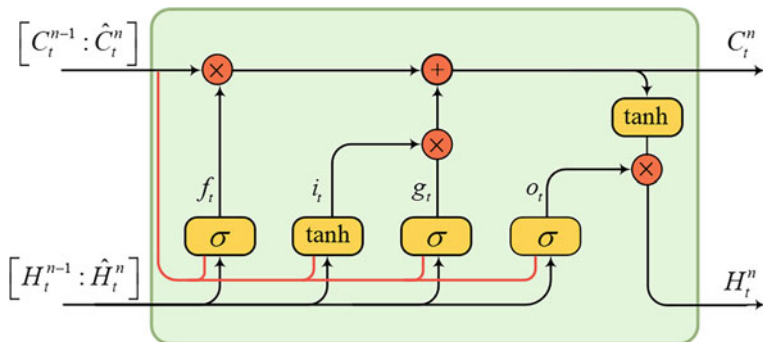


Fig. 4 Feature fusion module

4 Experimental Results and Discussion

The experimental results of the MLFrnn model are presented. First, the study area, dataset, and implementation details are presented. Next, the MLFrnn is compared with the currently available approaches. Moreover, the prediction results of the MLFrnn in different seasons are revealed. Finally, the impact of the different fusion modules and the number of layers on SSHA prediction is discussed.

4.1 Study Area and Dataset

As a semi-enclosed basin, the SCS connects the Pacific Ocean with the Indian Ocean. In addition, it features a complex seafloor topography accompanied by mesoscale eddies and frequent storm surges. It is considered as an area where oceanic and atmospheric modes exert a strong influence on the sea level [26, 29, 41, 43]. As a result, the fluctuating features of SSHA in the SCS are suitable for confirming the performance of the proposed network model. A subarea of the SCS was chosen as our study area, spanning 4.875°N-19.625°N and 109.875°E-119.625°E (red box in Fig. 5).

The altimeter data of satellites were sourced from different sensors, including Envisat, ERS-1/2, GFO, Jason-1/2/3, and T/P. The data generation was based on archiving, confirming, and interpreting satellite oceanographic (AVISO) data, while the data distributor was Copernicus Marine Environment Monitoring Service (CMEMS). The mean everyday data (0.125°N-25.125°N, 100.125°E-125.125°E) between January 1, 2001 and May 13, 2019 were utilized for experimentation, with a spatial resolution of $1/4^\circ$ latitude \times $1/4^\circ$ longitude. The training set comprised the SSHA fields recorded between January 1, 2001 and May 1, 2016, while the test set consisted of the SSHA fields recorded between May 2, 2016 and May 13, 2019. Among 6647 sequences of grouped data, 5570 sequences belonged to the training

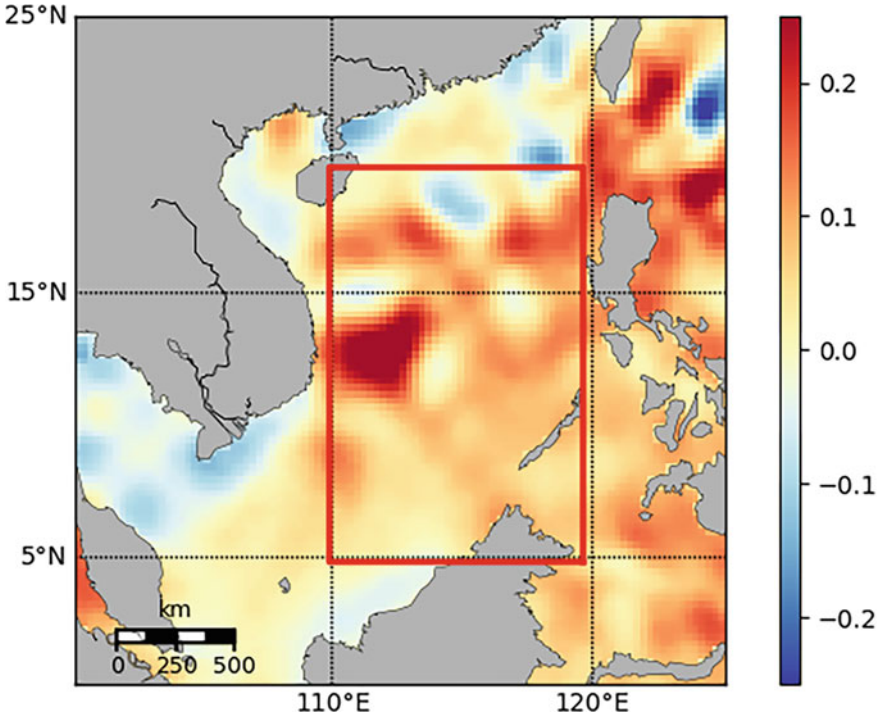


Fig. 5 Study area (boxed in red)

set, and 1077 sequences belonged to the test set. There were 31 tensors in every sequence, and the inter-tensor temporal interval was set as 1 day. In this study, the initial 10 tensors were regarded as the input, whereas the subsequent 21 tensors were regarded as the prediction reference. Prior to model feeding, the augmentation of data was accomplished using horizontal mirroring.

4.2 Implementation Detail

The spatial zone of the monitored SSHA field was considered a $H \times W$ grid involving L measurements. Accordingly, a 3D tensor X_t with dimensions $L \times W \times H$ was used to denote the daily SSHA data. A tensor sequence X_1, X_2, \dots, X_t was established by the monitoring results across t time steps from a temporal perspective. Prior to MLFrnn model feeding, the foregoing tensors were normalized within a $[0, 1]$ scope. The normalization procedure allowed better centralization of data so that the model training and convergence could be accelerated. In the following experiments, H and W were set as 100, L was set as 1, and t was set as 10.

At the model training stage, all the initial state parameters were assigned as zero for the hidden states $H_{t=0}^n$, cell states $C_{t=0}^n$, and relevance states $R_{t=0}^n$. Upon completion of 80,000 iterations, the training procedure was terminated, and every iteration had a mini-batch size of 8. The learning rate was 0.003 at baseline value, which was progressively decreased by a factor of 0.9 per 2500 iterations [24]. The MSE was adopted as the loss function, and the optimizer proposed by Adam [20] was used. A rapid decrease in the loss function was noted, ultimately converging to a small value. The repeated training outcomes differed only slightly.

For assessing the performance of the MLFrnn model, the following three metrics were adopted: the mean absolute error (MAE), the root mean square error (RMSE), and Pearson's correlation coefficient (r).

4.3 Experiment Results and Discussion

To predict the SSHA field for 21 d ahead and assess the performance of MLFrnn against the strong recent baseline ConvLSTM, the MLFrnn model was studied experimentally. The MLFrnn model was compared with the currently available DL approaches, namely the merged LSTM and attention-based LSTM from space and time dimensions (LSTM + STA).

In Table 1, the RMSE values of the MLFrnn and ConvLSTM models with a varying number of layers are presented for comparison. MLFrnn outperformed ConvLSTM in terms of predictive capacity, and with an increase in the time step, serious performance degradation was noted for ConvLSTM. Moreover, MLFrnn exhibited superior SSHA field predictability compared to ConvLSTM owing to the blending of spatiotemporal features (both local and global).

Table 1 RMSE values of SSHA field prediction 1–21 days ahead of four-layered MLFrnn (MLFrnn-4), single-layered ConvLSTM (ConvLSTM-1), and four-layered ConvLSTM (ConvLSTM-4)

Prediction lead (day)	1	2	3	4	5	6	7
ConvLSTM-1	0.00416	0.00678	0.00988	0.01350	0.01757	0.02191	0.02643
ConvLSTM-4	0.00353	0.00557	0.00802	0.01080	0.01389	0.01716	0.02054
MLFrnn-4	0.00324	0.00506	0.00714	0.00946	0.01200	0.01466	0.01739
Prediction lead (day)	8	9	10	11	12	13	14
ConvLSTM-1	0.03103	0.03564	0.04021	0.04474	0.04918	0.05353	0.05779
ConvLSTM-4	0.02396	0.02735	0.03067	0.03389	0.03698	0.03991	0.04266
MLFrnn-4	0.02014	0.02288	0.02557	0.02821	0.03078	0.03326	0.03565
Prediction lead (day)	15	16	17	18	19	20	21
ConvLSTM-1	0.06195	0.06602	0.07000	0.07389	0.07771	0.08145	0.08511
ConvLSTM-4	0.04523	0.04762	0.04983	0.05186	0.05372	0.05542	0.05696
MLFrnn-4	0.03795	0.04017	0.04230	0.04436	0.04634	0.04826	0.05012

Table 2 SSHA field predictability (RMSE) of four-layered MLFrnn (MLFrnn-4), LSTM + STA, and merged LSTM

Prediction lead (day)	1	2	3	4	5
Merged LSTM	0.0028	0.0059	0.0088	0.0120	0.0160
LSTM + STA	0.0038	\	\	\	\
MLFrnn-4	0.00324	0.00506	0.00714	0.00946	0.01200

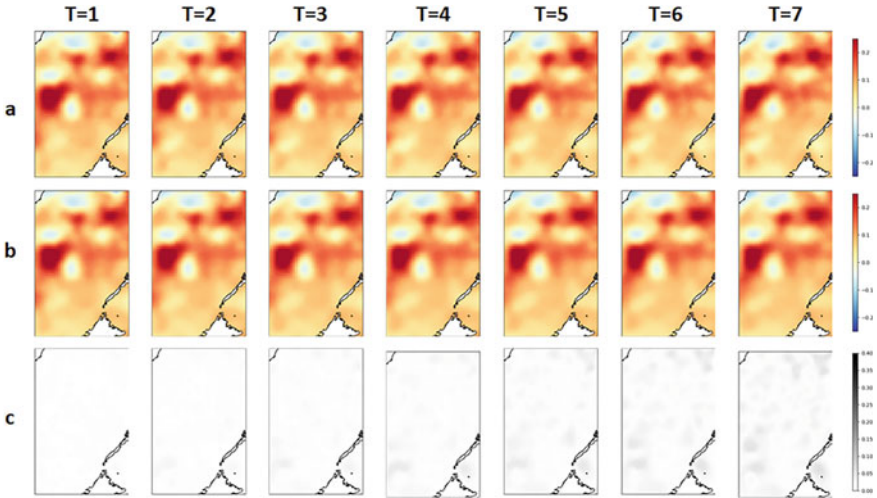


Fig. 6 Summertime MLFrnn qualitative findings for SSHA forecasts. To achieve SSHA-field estimation for 1–7 days ahead, a 10-day monitoring of the SSHA field was accomplished. **a** SSHA-field observations. **b** SSHA-field forecasts. **c** Deviations of SSHA-field observations from forecasts

The predictive behavior of the MLFrnn model was compared with that of LSTM + STA and merged LSTM (Table 2). The SSHA estimation data for the latter two models were available only for 1 and 5 d ahead, respectively. Therefore, MLFrnn was compared at identical prediction times. Interestingly, despite the design purpose of MLFrnn to estimate the future 21-day SSHAs, it was noted to be superior in the case of short-period prediction as well because the vector input shortcoming with the LSTM was surmounted by the MLFrnn model, which can accomplish concurrent SSHA field elucidation for spatiotemporal architectures.

For the performance characterization of the MLFrnn model, we studied the MLFrnn prediction outcomes across various seasons. The summertime and wintertime MLFrnn predictions are displayed in Figs. 6, 7, 8, 9, 10, and 11, where the SSHA observations, MLFrnn forecasts and, their deviations are presented in a top-bottom order. Quite evidently, there were small and permissible differences in the observa-

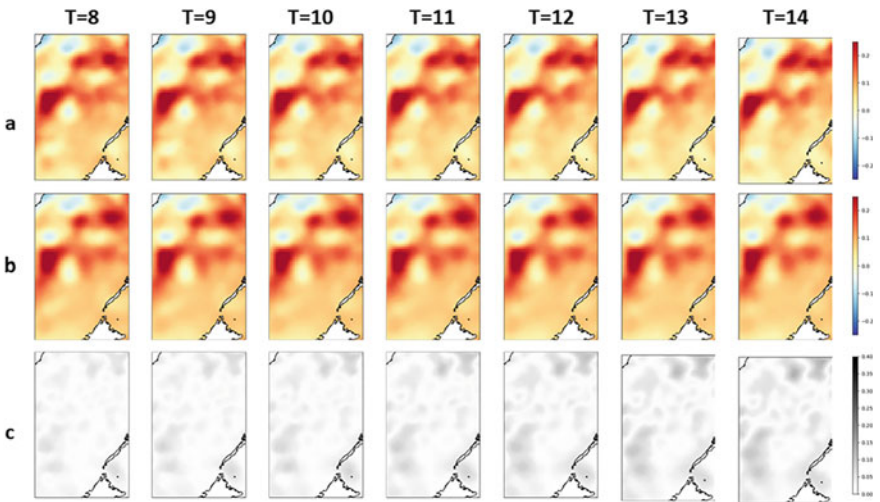


Fig. 7 Summertime MLFrnn qualitative findings for SSHA forecasts. To achieve SSHA-field estimation for 8–14 days ahead, a 10-day monitoring of the SSHA field was accomplished. **a** SSHA-field observations. **b** SSHA-field forecasts. **c** Deviations of SSHA-field observations from forecasts

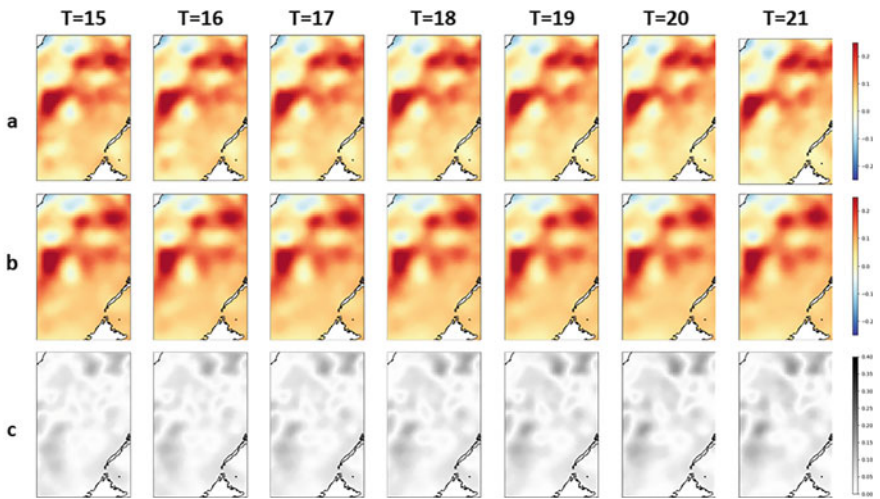


Fig. 8 Summertime MLFrnn qualitative findings for SSHA forecasts. To achieve SSHA-field estimation for 15–21 days ahead, a 10-day monitoring of the SSHA field was accomplished. **a** SSHA-field observations. **b** SSHA-field forecasts. **c** Deviations of SSHA-field observations from forecasts

tions made from the forecasts. As suggested by this finding, the SSHA predictability of MLFrnn is preferable across various seasons.

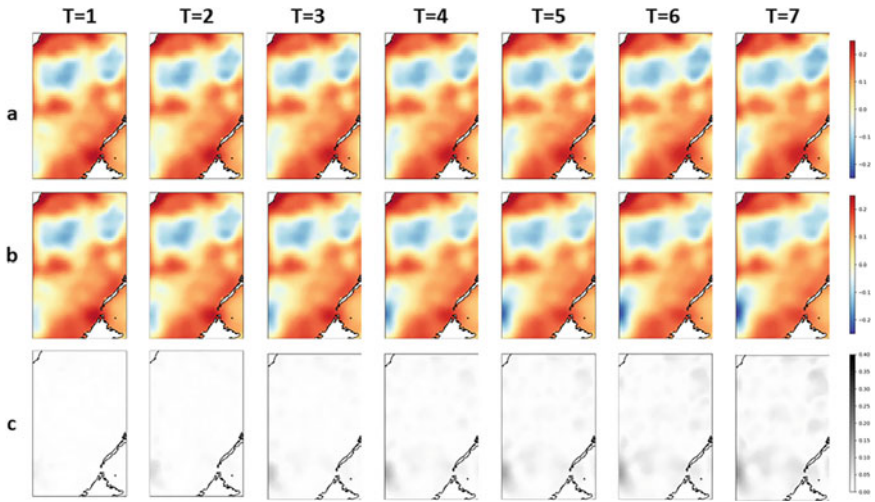


Fig. 9 Wintertime MLFrnn qualitative findings for SSHA forecasts. To achieve SSHA-field estimation for 1–7 days ahead, a 10-day monitoring of the SSHA field was accomplished. **a** SSHA-field observations. **b** SSHA-field forecasts. **c** Deviations of SSHA-field observations from forecasts

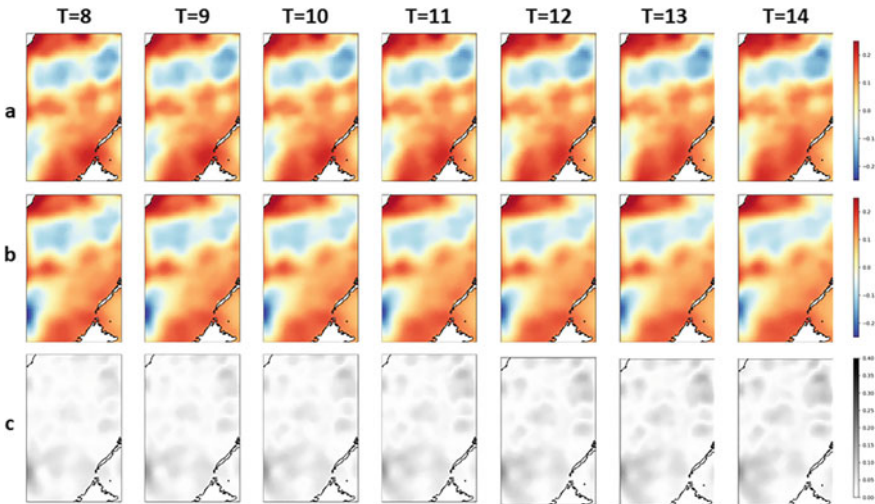


Fig. 10 Wintertime MLFrnn qualitative findings for SSHA forecasts. To achieve SSHA-field estimation for 8–14 days ahead, a 10-day monitoring of the SSHA field was accomplished. **a** SSHA-field observations. **b** SSHA-field forecasts. **c** Deviations of SSHA-field observations from forecasts

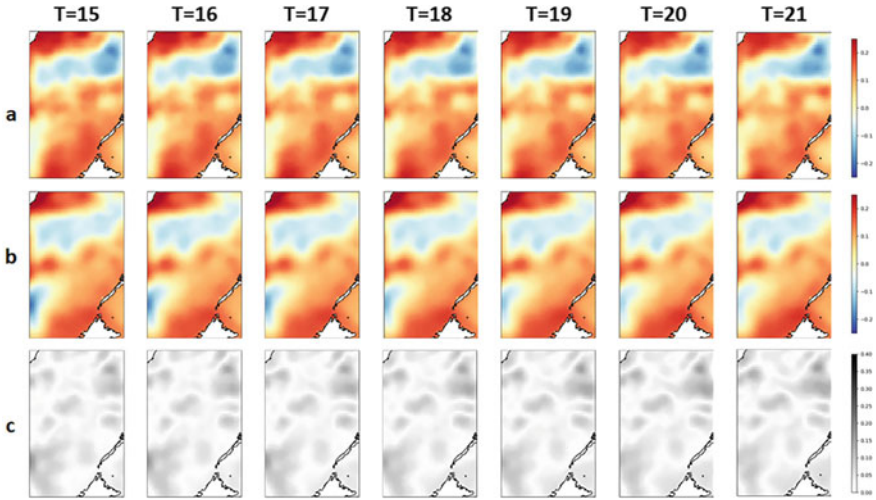


Fig. 11 Wintertime MLFrnn qualitative findings for SSHA forecasts. To achieve SSHA-field estimation for 15–21 d ahead, a 10-day monitoring of the SSHA field was accomplished. **a** SSHA-field observations. **b** SSHA-field forecasts. **c** Deviations of SSHA-field observations from forecasts

4.4 Ablation Study

To explore the contributions of the feature fusion module and the number of layers, an ablation study was conducted experimentally for the following models:

- (1) The two-layered MLFrnn based on the feature fusion module (MLF(F)-2).
- (2) The three-layered MLFrnn based on the feature fusion module (MLF(F)-3).
- (3) The four-layered MLFrnn based on the feature fusion module (MLF(F)-4).
- (4) The four-layered MLFrnn based on a 3×3 convolution as feature fusion module (MLF(Conv)-4).

The RMSE values are displayed in Fig. 12 for the future 21-day forecasts obtained using various models. MLF(F)-4 was compared with MLF(Conv)-4. The SSHA field evolutions were forecasted by MLFrnn by using the feature fusion module; thus, exhibiting higher accuracy. This was probably due to the preferable modeling of long-term SSHA dependencies by the MLFrnn owing to the feature fusion module.

Further comparison was made concerning the SSHA field predictive behaviors among the models having a different number of layers. MLFrnn-4 outperformed others in terms of predictability. The broader SSHAs from peripheral zones were due to the increased number of layers, which facilitated better accuracy of the forecasts. The MAE comparisons of the 1-21-day SSHA-field forecasts obtained using different models are presented in Fig. 13. The similarities in the trend of these data to the RMSEs in Fig. 12 were noted.

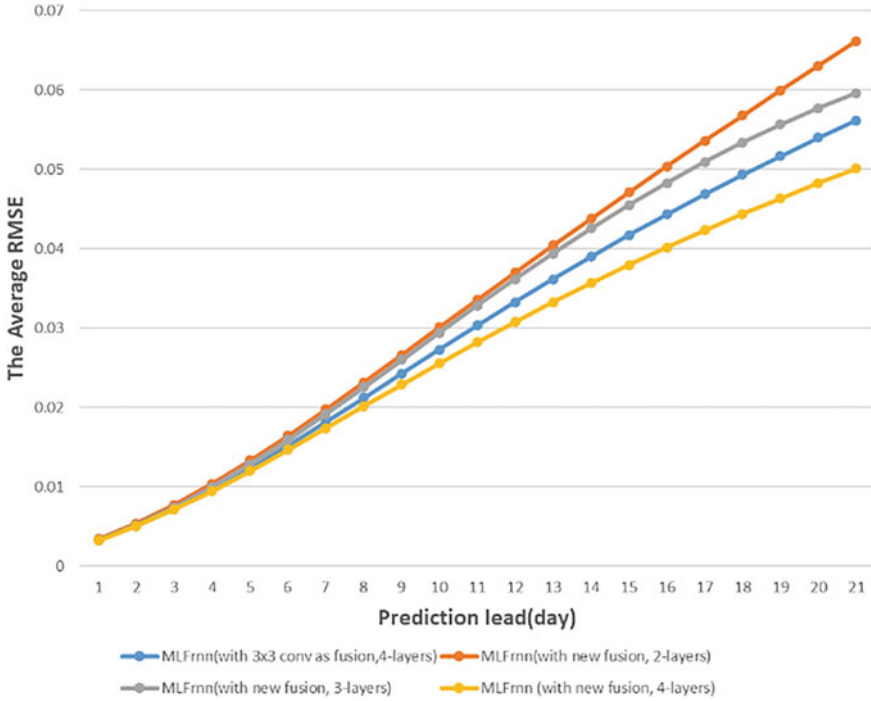


Fig. 12 Comparison of the RMSE values regarding the future 1–21-day SSHA-field forecasts obtained using different fusion modules and number of layers

The correlation coefficient reveals the degree of linearity among the investigated parameters. The Pearson’s correlation coefficient values (r) of SSHA forecasts and observations on the entire samples were determined and compared (Fig. 14). As is clear, compared to the remaining three models, MLFrnn-4, having a feature fusion module, exhibited greater r values all along. Moreover, a slower decrease in r was observed MLFrnn-4 as compared to the other three models. This confirms the positive linearity between the SSHA field (forecasted using the feature fusion module-based MLFrnn-4) and the true value.

5 Conclusion

In this chapter, we first introduced the significance of SSHA prediction and then presented a review of currently available SSHA prediction methods. Next, an SSHA forecasting model named MLFrnn was proposed. The main advantages of MLFrnn can be summarized as follows:

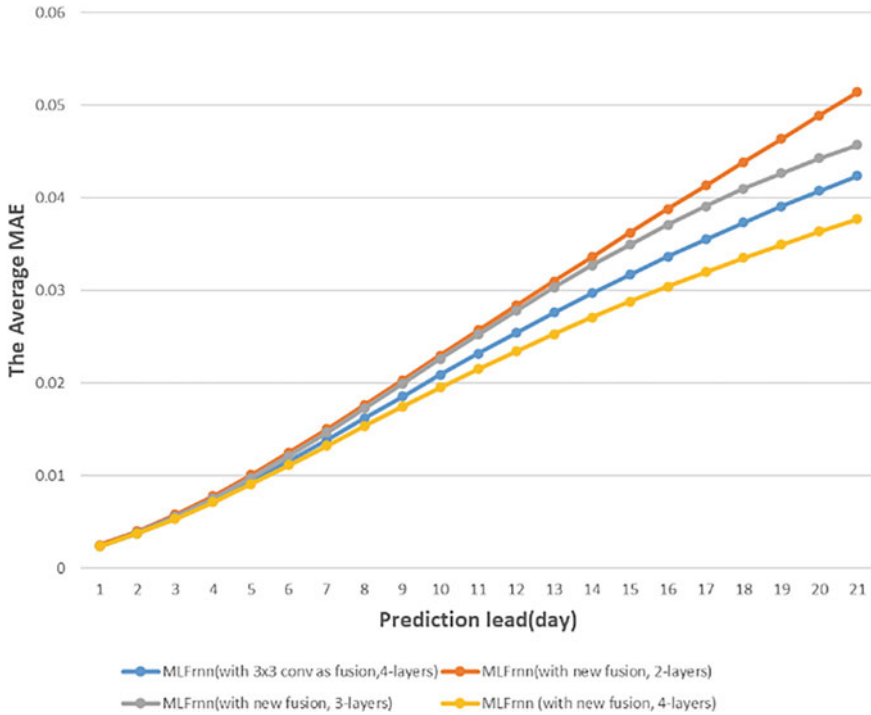


Fig. 13 Comparison of the MAE values regarding the future 1–21-day SSHA-field forecasts obtained using different fusion modules and the number of layers

- (1) The proposed MLFrnn model can acquire spatiotemporal traits from both nearby and distant sites. Existing RNN/LSTM-based studies emphasize temporal modeling while disregarding spatial information. In contrast, the proposed MLFrnn model prominently improves the SSHA predictability by modeling both the temporal variations and the spatial evolutions of the SSHA fields.
- (2) MLFrnn enables prediction for the entire SSHA map rather than single-site forecasts of the SSHA. Prior approaches could achieve SSHA estimation for only one grid and required repeated model training for performing predictions for the entire zone. In contrast, the MLFrnn model can perform predictions for the entire SSHA map accurately.
- (3) In this study, a type of multilayer fusion cells was developed for MLFrnn to fuse local and global spatiotemporal characteristics. In addition, reliable modeling of the SSHA field evolutions was achieved using the SSHAs from both nearby and remote locations.

Finally, we selected the SCS as study area and presented the experimental results of the MLFrnn model on the daily average satellite altimeter SSHA data for nearly

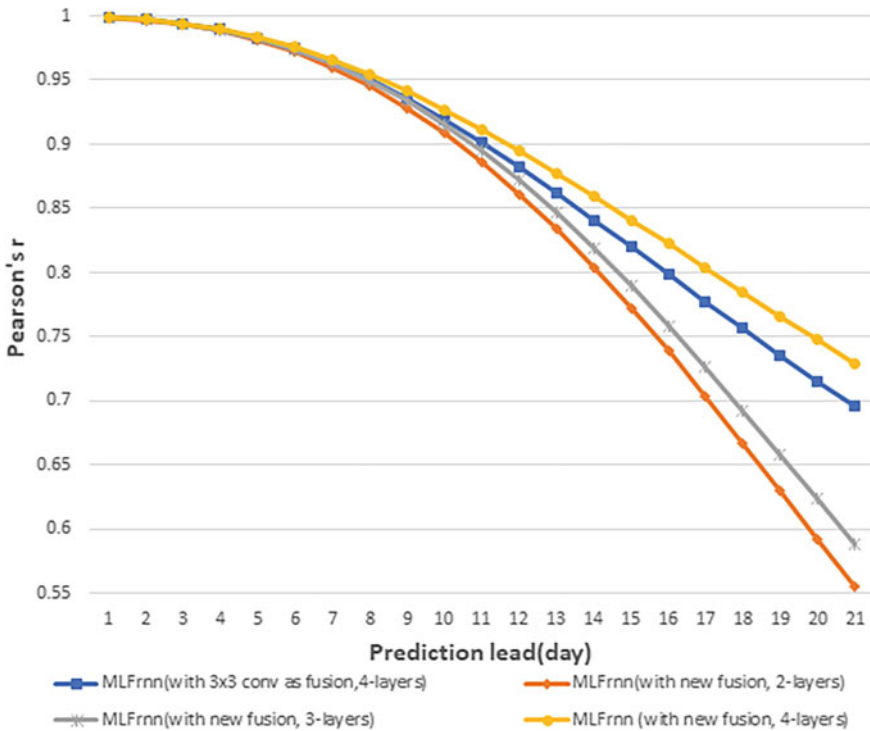


Fig. 14 Comparison of the Pearson's correlation coefficient values r regarding the future 1–21-day SSHA-field forecasts obtained using different fusion modules and the number of layers

19 years. The experimental results demonstrated that the MLFrnn model is effective and has better performance than the currently available DL networks in predicting the SSHA field.

References

1. Ablain M, Legeais JF, Prandi P, Marcos M, Fenoglio-Marc L, Dieng HB, Benveniste J, Cazenave A (2017) Satellite altimetry-based sea level at global and regional scales. *Surv Geophys* 38(1):7–31
2. Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by Jointly Learning to Align and Translate. arXiv preprint [arXiv:1409.0473](https://arxiv.org/abs/1409.0473)
3. Barbosa SM, Silva ME, Fernandes MJ (2006) Multivariate autoregressive modelling of sea level time series from TOPEX/Poseidon satellite altimetry. *Nonlinear Process Geophys* 13(2):177–184
4. Bonaduce A, Pinaridi N, Oddo P, Spada G, Larnicol G (2016) Sea-level variability in the Mediterranean Sea from altimetry and tide gauges. *Clim Dyn* 47(9):2851–2866
5. Braakmann-Folgmann A, Roscher R, Wenzel S, Uebbing B, Kusche J (2017) Sea level anomaly prediction using recurrent neural networks. arXiv preprint [arXiv:1710.07099](https://arxiv.org/abs/1710.07099)

6. Carton JA, Giese BS, Grodsky SA (2005) Sea level rise and the warming of the oceans in the Simple Ocean Data Assimilation (SODA) ocean reanalysis. *J Geophys Res: Oceans* 110(C9)
7. Chen JL, Wilson CR, Chambers DP, Nerem RS, Tapley BD (1998) Seasonal global water mass budget and mean sea level variations. *Geophys Res Lett* 25(19):3555–3558
8. Cheng Y, Xu Q, Li X (2018) Spatio-temporal variability of annual sea level cycle in the Baltic Sea. *Remote Sensing* 10(4):528
9. Chien JT, Ku YC (2015) Bayesian Recurrent Neural Network for language modeling. *IEEE Trans Neural Netw Learn Syst* 27(2):361–374
10. Church JA, White NJ, Aarup T, Wilson WS, Woodworth PL, Domingues CM, Hunter JR, Lambeck K (2008) Understanding global sea levels: past, present and future. *Sustain Sci* 3(1):9–22
11. Elman JL (1990) Finding structure in time. *Cogn Sci* 14(2):179–211
12. Ezer T, Atkinson LP, Corlett WB, Blanco JL (2013) Gulf Stream’s induced sea level rise and variability along the U.S. mid-Atlantic coast. *J Geophys Res: Oceans* 118(2):685–697
13. Fu LL, Chelton DB, Zlotnicki V (1988) Satellite altimetry: observing ocean variability from space. *Oceanography* 1(2):4–58
14. Fu Y, Zhou X, Sun W, Tang Q (2019) Hybrid model combining empirical mode decomposition, singular spectrum analysis, and least squares for satellite-derived sea-level anomaly prediction. *Int J Remote Sens* 40(20):7817–7829
15. Gornitz V, Lebedeff S (1987) Global sea-level changes during the past century
16. Gregory JM, Lowe JA (2000) Predictions of global and regional sea-level rise using AOGCMs with and without flux adjustment. *Geophys Res Lett* 27(19):3069–3072
17. Imani M, You RJ, Kuo CY (2014) Forecasting Caspian Sea level changes using satellite altimetry data (June 1992–December 2013) based on evolutionary support vector regression algorithms and gene expression programming. *Global Planet Change* 121:53–63
18. Imani M, Chen YC, You RJ, Lan WH, Kuo CY, Chang JC, Rateb A (2017) Spatiotemporal prediction of satellite altimetry sea level anomalies in the Tropical Pacific Ocean. *IEEE Geosci Remote Sens Lett* 14(7):1126–1130
19. Karevan Z, Suykens JAK (2020) Transductive LSTM for time-series prediction: an application to weather forecasting. *Neural Netw* 125:1–9
20. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
21. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp 1097–1105
22. Li X, Liu B, Zheng G, Ren Y, Zhang S, Liu Y, Gao L, Liu Y, Zhang B, Wang F (2020) Deep-learning-based information mining from ocean remote-sensing imagery. *Natl Sci Rev* 7(10):1584–1605
23. Liu J, Jin B, Wang L, Xu L (2020) Sea surface height prediction with deep learning based on attention mechanism. *IEEE Geosci Remote Sens Lett*
24. Loshchilov I, Hutter F (2017) Decoupled weight decay regularization. arXiv preprint [arXiv:1711.05101](https://arxiv.org/abs/1711.05101)
25. McGoogan JT (1975) Satellite altimetry applications. *IEEE Trans Microw Theory Tech* 23(12):970–978
26. Metzger EJ, Hurlburt HE (2001) The nondeterministic nature of Kuroshio penetration and eddy shedding in the South China Sea. *J Phys Oceanogr* 31(7):1712–1732
27. Miles ER, Spillman CM, Church JA, McIntosh PC (2014) Seasonal prediction of global sea level anomalies using an ocean-atmosphere dynamical model. *Clim Dyn* 43(7–8):2131–2145
28. Moore JC, Grinsted A, Zwinger T, Jevrejeva S (2013) Semiempirical and process-based global sea level projections. *Rev Geophys* 51(3):484–522
29. Nan F, He Z, Zhou H, Wang D (2011) Three long-lived anticyclonic eddies in the northern South China Sea. *J Geophys Res: Oceans* 116(C5)
30. Nicholls RJ, Cazenave A (2010) Sea-level rise and its impact on coastal zones. *Science* 328(5985):1517–1520

31. Niedzielski T, Kosek W (2009) Forecasting sea level anomalies from TOPEX/Poseidon and Jason-1 satellite altimetry. *J Geodesy* 83(5):469–476
32. Norris RD, Turner SK, Hull PM, Ridgwell A (2013) Marine ecosystem responses to cenozoic global change. *Science* 341(6145):492–498
33. Rahmstorf S (2007) A semi-empirical approach to projecting future sea-level rise. *Science* 315(5810):368–370
34. Rudenko S, Neumayer KH, Dettmering D, Esselborn S, Schöne T, Raimondo JC (2017) Improvements in precise orbits of altimetry satellites and their impact on mean sea level monitoring. *IEEE Trans Geosci Remote Sens* 55(6):3382–3395
35. SHI X, Chen Z, Wang H, Yeung DY, Wong WK, Woo Wc (2015) Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In: *Advances in Neural Information Processing Systems*, pp 802–810
36. Sivakumar S, Sivakumar S (2017) Marginally stable triangular recurrent neural network architecture for time series prediction. *IEEE Trans Cybern* 48(10):2836–2850
37. Slangen ABA, Katsman CA, Van de Wal RSW, Vermeersen LLA, Riva REM (2012) Towards regional projections of twenty-first century sea-level change based on IPCC SRES scenarios. *Clim Dyn* 38(5):1191–1209
38. Song T, Jiang J, Li W, Xu D (2020) A deep learning method with merged LSTM Neural Networks for SSHA Prediction. *IEEE J Sel Topics Appl Earth Obs Remote Sens* 13:2853–2860
39. Tandeo P, Chapron B, Ba S, Autret E, Fablet R (2013) Segmentation of mesoscale ocean surface dynamics using satellite SST and SSH observations. *IEEE Trans Geosci Remote Sens* 52(7):4227–4235
40. Unal YS, Ghil M (1995) Interannual and interdecadal oscillation patterns in sea level. *Clim Dyn* 11(5):255–278
41. Zhao Z, Liu B, Li X (2014) Internal solitary waves in the China seas observed using satellite remote-sensing techniques: a review and perspectives. *Int J Remote Sens* 35(11–12):3926–3946
42. Zheng G, Li X, Zhang RH, Liu B (2020) Purely satellite data-driven deep learning forecast of complicated tropical instability waves. *Sci Adv* 6(29):eaba1482
43. Zheng Q, Hu J, Zhu B, Feng Y, Jo YH, Sun Z, Zhu J, Lin H, Li J, Xu Y (2014) Standing wave modes observed in the South China Sea deep basin. *J Geophys Res: Oceans* 119(7):4185–4199

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits any noncommercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if you modified the licensed material. You do not have permission under this license to share adapted material derived from this chapter or parts of it.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Satellite Data-Driven Internal Solitary Wave Forecast Based on Machine Learning Techniques



Xudong Zhang, Quanan Zheng, and Xiaofeng Li

1 Introduction

Internal solitary wave (ISW) is a ubiquitous phenomenon in the world's oceans, particularly in continental and marginal waters [4, 10, 13, 28]. ISWs are generated at the mixed layer beneath the ocean surface and show less obvious manifestations on the ocean surface. ISWs can travel hundreds of kilometers while maintaining their waveform or amplitude, owing to nonlinear and dispersion effects. The resort force of ISWs is the reduced gravity, which promises the generation of large-amplitude ISWs. ISWs in the South China Sea (SCS) were observed with amplitude over 240 m. The length of wave crest (LWC) of ISWs can also extend to several hundreds of kilometers. ISWs are found to travel across the whole northern SCS, the Andaman Sea, and the Sulu-Celebes Sea within a few days.

Ocean habitats, off-shore engineering, ocean military, ocean mixing, and sediment resuspension can all be affected by the propagation and breaking of ISWs. While the wave crest of ISWs can be extended to hundreds of kilometers, the ISW scale across the wave crest only ranges from several hundreds of meters to several thousand. ISW propagation speed ranges between 2.0–3.0 m/s in the deep ocean and 1.0–2.0 m/s on the continental shelf. The across wave crest features of ISWs means the ISW will pass by a fixed location within a few tens of minutes. Considering the fast propagating and large amplitudes of ISWs, the ISWs are extremely dangerous to submarine or underwater vehicles. The propagation of ISWs will be accompanied by ISW-induced currents, and severe shear forces will endanger the safety of off-shore equipment, such as oil rigs. The propagation of large amplitude ISWs will

X. Zhang · Q. Zheng · X. Li (✉)

CAS Key Laboratory of Ocean Circulation and Waves, Institute of Oceanology, Chinese Academy of Sciences, Qingdao 266071, China

e-mail: lixf@qdio.ac.cn

Q. Zheng

Department of Atmospheric and Oceanic Science, University of Maryland, College Park, Maryland 20742, USA

© The Author(s) 2023

X. Li and F. Wang (eds.), *Artificial Intelligence Oceanography*,
https://doi.org/10.1007/978-981-19-6375-9_4

induce strong vertical mixing in the ocean and affect the distribution of suspension or sediments [8]. The importance of ISW makes the forecast of ISW propagation a meaningful but challenging task.

Global observation, random generation, fast propagation, and significant impact on the ocean have made ISWs a hot topic for decades. Various methods have been used to study ISWs, such as numerical models, in-situ observations, and satellite observations. Remote sensing techniques have developed rapidly and show significant advantages in ISW studies [10, 11]. To describe ISW propagations, various theories were developed, such as the Korteweg-de Vries equation (KdV) equation, the Benjamin-Ono (BO) equation, and the numerical models. The KdV equation for the propagation of ISWs is given by

$$\eta_t + c_0\eta_x + \alpha\eta\eta_x + \gamma\eta\eta_{xxx} = 0 \quad (1)$$

Here η is the amplitude of the solitary wave and c_0 is the linear phase speed. When the nonlinear term α is balanced with the dispersion term γ , one gets the solitary wave with an analytical form

$$\eta(x, t) = \eta_0 \operatorname{sech}^2[(x - ct)/L] \quad (2)$$

$$c = c_0 \left(1 + \frac{\eta_0}{2h}\right) \quad (3)$$

$$L = \sqrt{\frac{12\gamma}{\alpha\eta_0}} \quad (4)$$

where η_0 is the maximum amplitude, L is the characteristic length, h is the water depth, and g is the acceleration due to gravity. The KdV equation is commonly used to describe the propagation speed of ISWs, but the ISW amplitude needs to serve as preliminary information. Different theories have been developed for ISW propagations in different ocean areas, each with its advantages and disadvantages. Here we introduce a new data-driven model to forecast ISW propagation. The forecast model was trained using big data collected from multi-source remote sensing imageries. The model performance shows better results than the traditional equations and is more robust for errors included in the model inputs.

In the following chapter, we will first briefly overview the achievement of ISW studies using satellite observations. Then, machine learning techniques applied to the ISW studies will be introduced, and the establishment of the ISW forecast model will be presented. The model discussions and future works will be introduced in the last part of this chapter.

2 Satellite Observations of ISWs

Space-borne synthetic aperture radar (SAR) and optical satellite images are commonly used in ISW studies. SAR is an active microwave radar with all-weather, all-time, long-distance, and high-resolution detection advantages. The imaging principle of ISWs on SAR images is the Bragg backscattering mechanism [1]. Currents induced by ISW (usually refers to the first-mode depression ISW) modulate micro-scale waves on the sea surface, making convergent and divergent regions appear in the front and back of the wave, respectively. The sea surface roughness increases in the convergent region where the Bragg backscatter signal is enhanced and appears as a bright stripe on SAR images. The sea surface roughness decreases in the divergent region where the Bragg backscatter signal becomes weaker and appears dark on the SAR image. Figure 1a and b show that ISWs manifest as bright-dark stripes on SAR images. Although spatial resolutions of SAR images are relatively high, their time resolution is relatively low. The limited swath of SAR images also imposes restrictions on ISW studies. SAR images are generally used to study the mechanism of ISWs, such as inversion of amplitude [26], propagation speed [15], and energy analysis [17].

The imaging mechanism of ISW on optical images is the quasi-specular reflection. Optical remote sensing uses sunlight as the light source and receives ocean information from the sunlight reflected by the sea surface. The characteristics of ISW in optical images are more complicated than SAR images. Similarly, for depression ISW, the characteristics of different locations on the optical image are different. They may appear as bright-dark or dark-bright strips in sun-glint areas or non-sun-glint areas. Figure 1c and 1d show satellite observations of ISWs in the Sulu-Celebes Sea. Multiple ISW packets can be observed propagating in different directions with long wave crests.

Optical remote sensing images have advantages of high time resolution and wide swath. Take Moderate-Resolution Imaging Spectroradiometer (MODIS) image as an example, its swath can reach 2330 km, and the same ocean area can be observed twice in one day. In addition, the spatial resolution of optical satellites launched in recent years has also been greatly improved. For example, the highest spatial resolution of GF-1 remote sensing images of China's high-resolution series can reach 2 m, and the spatial resolution of GF-2 images has reached less than 1 m. However, the optical satellite images are heavily affected by the weather conditions, such as clouds and rain, which will limit its observation capability. Benefiting from the wide swath and high temporal resolution of optical satellite images, the temporal and spatial distribution characteristics of ISWs were studied in different ocean areas.

SAR and optical satellite images provide rich data sources for the research of ISWs. The generation mechanism, distribution, and propagation of the ISWs have been reported. Since the forecast of ISWs is significant for ocean environments, engineering, mixing, and military, the forecast of ISWs is meaningful. The ISW forecast is mainly conducted using empirical or numerical models [30]. In the SCS, ISWs have been predicted based on the west propagating barotropic tide in the Luzon Strait. ISW

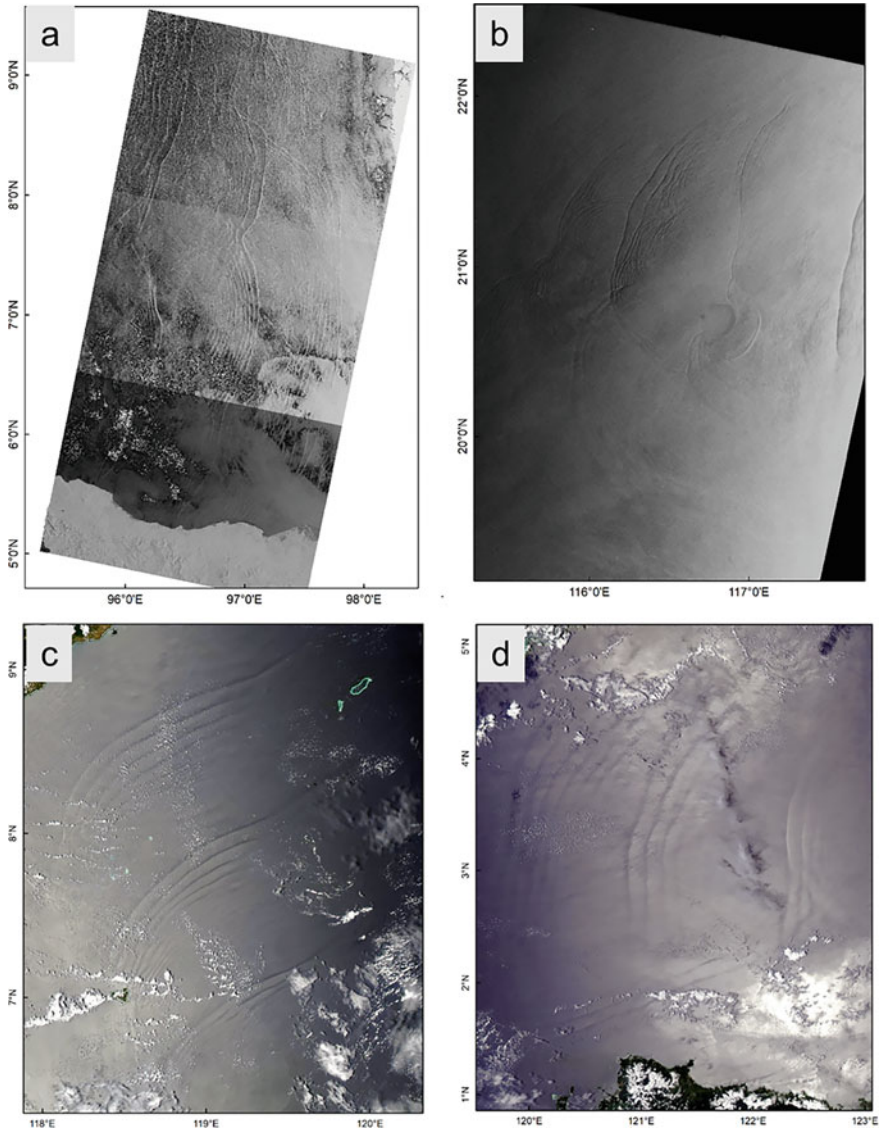


Fig. 1 Satellite detection of ISWs in the Andaman Sea **a**, South China Sea **b**, Sulu Sea **c**, and Celebes Sea **d**. **a** composite map of Sentinel-1 images acquired on 10 March 2019; **b** ENVISAT ASAR image acquired on 5 May 2004; **c** MODIS image acquired on 14 March 2020; **d** MODIS image acquired on 28 March 2020

positions can be predicted according to the distance to the Luzon Strait. Based on the relationship between the ISW generation and the tide information, [6] forecasted the possible ISW occurrence in the north region of the Andaman Sea. The ISWs are generated at multiple sources and have complex patterns as revealed in Fig. 1. The complex pattern of ISWs makes the forecast more difficult. The complexity of multiple sources and wave crest merging during the propagation make it hard to apply the empirical method. The numerical model needs large computation resources and is difficult to setup for ISWs propagating in the coastal areas. To overcome these difficulties, we proposed a machine learning model to forecast ISW propagation in different oceans with different ISW characteristics.

3 Machine-Learning-Based ISW Forecast Model

Machine learning techniques are fast evolving and have already demonstrated tremendous promise in oceanographic research [5, 23, 24]. Liu et al. [16] explored extracting coastal inundation mapping information from SAR imagery by applying deep learning techniques. Machine learning also allows for the creation of connections between multi-dimensional data. Pan et al. [19] employed textural information taken from optical satellite images and ocean environmental factors to determine the amplitude of ISWs using the back-propagation (BP) algorithm. Li et al. [12] used a U-net-based method to obtain ISW wave crest from satellite images and conducted a thorough evaluation of the use of machine learning approaches in satellite image information mining. Machine-learning approaches have previously been proven to offer benefits in maritime applications due to their high nonlinear mapping ability and multi-dimensional data processing.

The propagation of ISW is influenced by ISW features as well as the ocean influencing factors, such as topography and seasonal variations. Machine learning techniques are an excellent option to manage multi-dimensional impacting elements with no defined relationship, making them a strong choice for ISW propagation model construction.

3.1 Model Establishment

We use a fully connected neural (FCN) network to build relationships between the ISW propagations and its impacting factors [25, 29]. Figure 2 depicts a sketch map of the FCN network. We use the error back-propagation (BP) technique to train the FCN network. The desired output is calculated by the forward calculation procedure, and the back-propagation of errors between the model and the desired output is used to adjust the model. The weights of the neural network can be automatically adjusted based on the errors inversely fed into the model. The built dataset is categorized into the training dataset and validation dataset. The validation dataset watches the

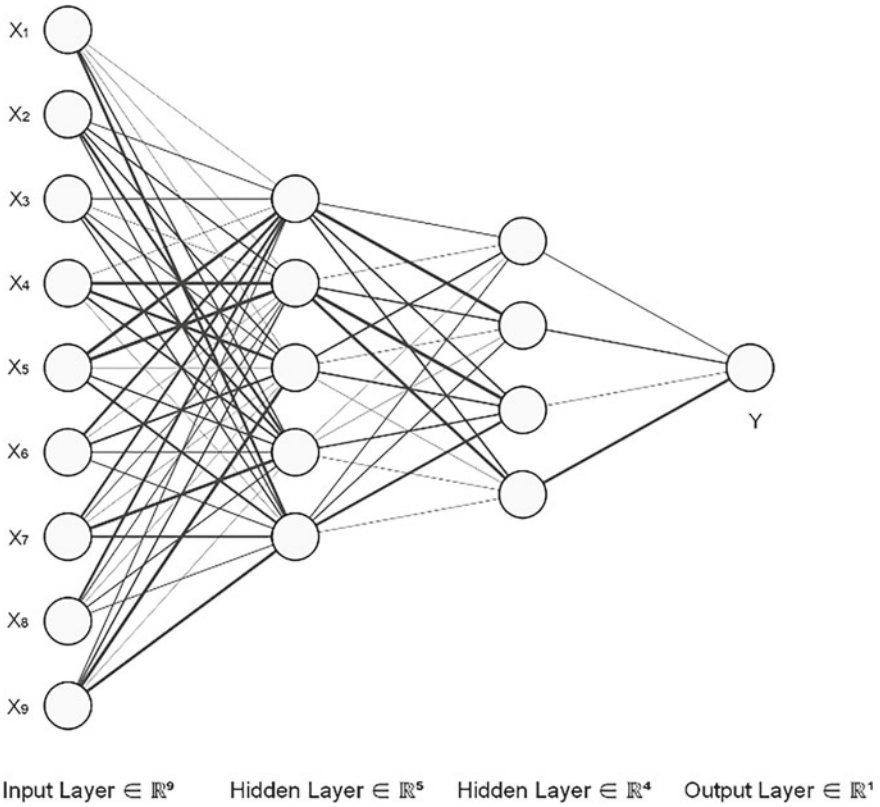


Fig. 2 Sketch map of an FCN network. The different thickness of lines connecting different neural indicate different weights

training procedure while the training dataset is employed for model training. The model will be trained for several epochs after being initialized randomly. The model will stop training and be further validated by a test dataset once it achieves the best validation results.

The forecast model’s input parameters are made up of ISW property-related inputs, i.e., the Peak-to-Peak (PP) distance and the LWC [31], as well as ocean environment-related inputs such as longitude, latitude, mixed layer depth, density difference, and water depth. Satellite images may be used to measure an ISW’s LWC and PP distance. The water depth can be interpolated from the ETOPO1 dataset. The World Ocean Atlas (WOA) 2018 dataset, where the temperature and salinity can be obtained, may be used to estimate ocean stratification. The buoyance frequency peak corresponds to the depth of the mixed layer.

There are two modules in the model: propagation speed and direction (PS and PD) module. Seven input parameters are included and the output layer is the propagation speed and direction. The initial propagation direction is an extra input parameter in the

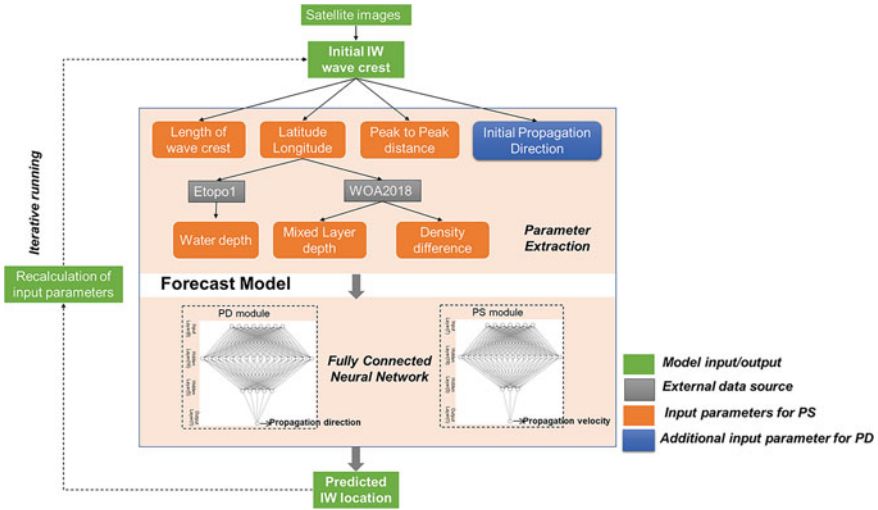


Fig. 3 Model structure for the ISW forecast model

PD module that is used to resolve occasions with cross-propagating ISWs. Figure 3 depicts the model framework for the ISW prediction model. Locations, ocean parameters, and ISW characteristics of an initial ISW location found from satellite images may all be gathered and used as model inputs. We may run the model for several time steps to get the expected ISW positions at each time step using the model’s predicted ISW propagation velocity and direction. The Levenberg-Marquardt algorithm-based training function ‘trainlm’ was chosen for its quick convergence rate [22]. The hyperbolic tangent sigmoid transfer (tansig) function is used to activate hidden layers, whereas the linear transfer (pureline) function is used to activate hidden and output layers. We used an early stopping strategy to combat the problem of over-fitting [20].

3.2 Model Training

Both optical and SAR images can be applied to extract the training data. The MODIS sensors are onboard the National Aeronautics and Space Administration (NASA) satellites Terra and Aqua. The MODIS image has a swath of 2330km and the highest spatial resolution of 250m. The Ocean and Land Color Instrument (OLCI) has five camera modules on board Sentinel-3. The OLCI has a swath of 1440km and the best spatial resolution of 300m. We use 123 MODIS images and 33 OLCI images in the Andaman Sea and 149 MODIS images and 8 VIIRS images in the Sulu-Celebes Sea to build the dataset.

On satellite images, ISWs appear as bright-dark bands. ISW locations can be saved as the GIS formatted file which is used to extract the spatial position. The LWC can

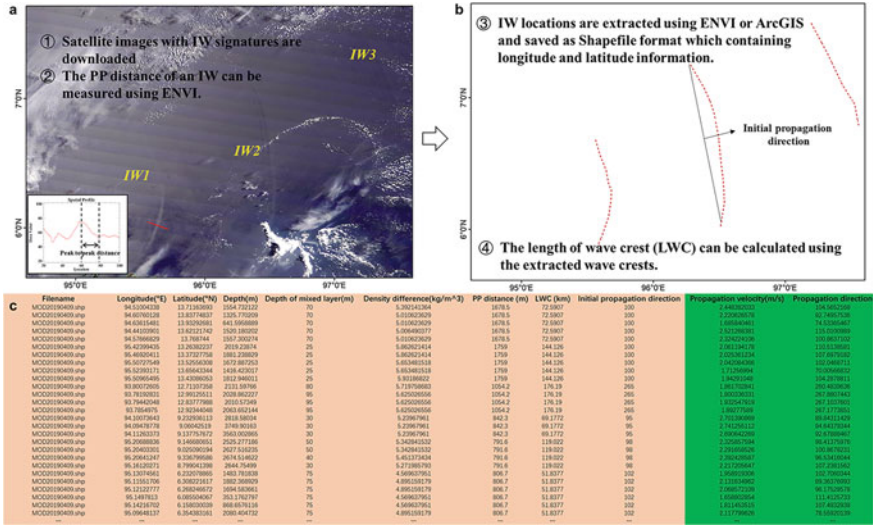


Fig. 4 Procedure to build the training dataset from satellite images. **a** A subset of the MODIS image showing three ISW packets, the PP distance can be measured from the satellite image. The insert map shows the extracted profile of the ISW and how the PP distance is measured; **b** extracted locations of ISW, the LWC was calculated from extracted wave crests, and how the initial propagation direction is obtained; **c** subset of the extracted dataset, the brown shaded area shows model input parameters and the green shaded area indicates model output parameters

be obtained using ISW labels and the PP distance equals the positive and negative peaks of the ISW profiles [31]. The propagation direction of the input ISW wave crest is introduced to solve cross-propagating ISW problems. A detailed procedure of how to build the training dataset is shown in Fig. 4.

We estimate the phase speed of ISWs based on the difference of ISW locations and image acquisition time. One utilizes the location and time difference of the ISWs to get the ISW propagation speed if an ISW was detected on two quasi-synchronous images. We assume the time difference between two ISWs in the same satellite image equals the period of the semi-diurnal tide [7, 9]. In the Andaman Sea, 1189 samples were extracted, while in the Sulu-Celebes Sea, 1546 samples were extracted. The training and independent test datasets were created from these samples, which were split by 80/20%.

Figure 5 shows the results of PS and PD modules for ISWs in the Andaman Sea. The Sulu-Celebes Sea forecast model yields similar results. The root mean square (RMSE) of the training (test) datasets for the PS module is 0.19 m/s (0.20 m/s), while the correlation coefficients (CC) are 0.90 (0.88). The results demonstrate that the loss of the forecast decreases over time, and the PS module’s mean square error (MSE) obtains its optimal validation performance at epoch 37. The PD module has an RMSE of 10°, and the CCs are over 0.99. The gradient was steadily reduced, and

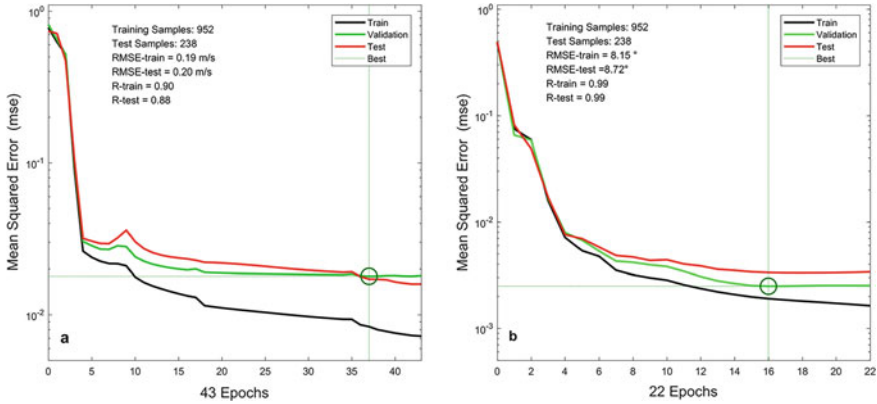


Fig. 5 Loss of the propagation velocity module **a** and the propagation direction module **b** for ISW forecast model

at epoch 16, the PD module obtained its optimal performance. The MSE does not decrease and validation tests rise at epoch 16, the model stops training.

ISWs in various ocean locations have varied features. In the Andaman Sea, we observe cross-propagating ISWs, while in the Sulu-Celebes Sea, we observe ISWs propagating in the opposite direction. We trained the model 30 times with and without initial primary propagation directions to see how they affected the predicted outcomes. Models having an initial major propagation direction as input performed better, with a reduced RMSE and more stable model performance, as illustrated in Fig. 6. The model without initial primary propagation directions had large deviations and lower correlation coefficients, indicating lower model generalizability. For circumstances with cross-propagating ISW patterns, it is required to add the propagation direction of the ISW wave crest in the model inputs.

3.3 Model Validation

The forecast model was validated for ISWs in the Andaman Sea and Sulu-Celebes Sea. The ISWs created by successive semi-diurnal tides in the Andaman Sea is depicted in Fig. 7. On the MODIS image, three ISW wave packets propagating eastward can be seen. IW1 and IW2 have LWCs of 146.41 and 242.26 km. IW1 and IW2 have PP distances of 1391.48 and 731.54 m. IW1 (IW2) parameters are used as model inputs, while IW2 (IW3) acts as model validation. In Fig. 7b and c, the time step is 6.21 h, and ISW positions after one time step are depicted with dashed lines. The model results (satellite observations) after one semi-diurnal tide are indicated by the black (red) lines in Fig. 7b and c. The model predicted results appear to be in good agreement with satellite data.

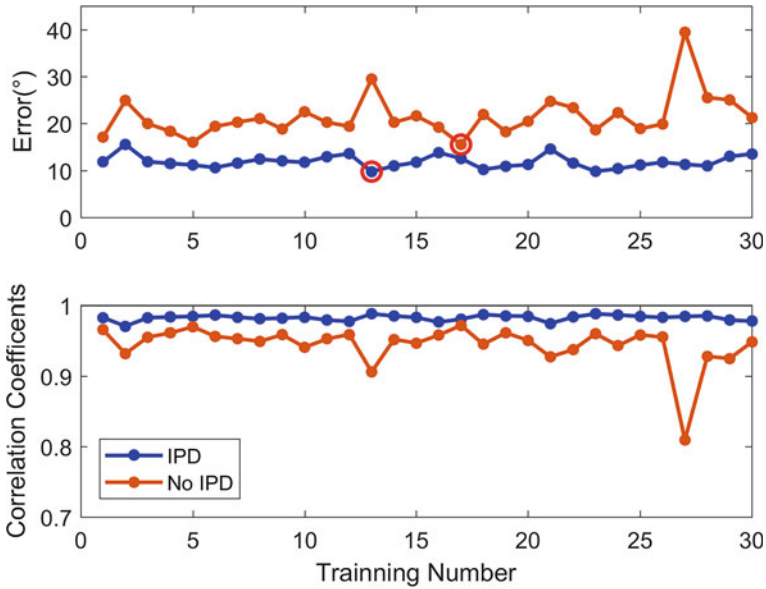


Fig. 6 Model tests with (blue lines) or without (orange lines) the propagation direction of ISW wave crests

We established three parameters, namely root-mean-square error (RMSE), Fréchet distance (FD), and CC, to qualitatively examine the effectiveness of the constructed ISW forecast model. The FD is a strict evaluation factor that considers the placement and order of points of the ISW wave crest. IW1 and IW2 have RMSEs of 6.10 and 2.50 km; the FDs is 18.28 and 9.06 km, and the CCs are 0.96 and 0.89, respectively. In the Andaman Sea, we examined 8 examples for ISWs, including distinct locations with different ISW features. Table 1 shows the statistical findings. The average CC value is 0.95, and the average FD is 11.46 km, showing that the model-predicted ISW positions and satellite observations have a good degree of agreement.

The results of the model validation for ISWs propagating in the Sulu-Celebes Sea are shown in Fig. 8. Three ISW wave packets have been detected moving northward (southward) in the Sulu (Celebes) Sea. For these three wave packets, the leading ISWs are called IW1, IW2, and IW3. The model input is the wave crest IW1 (IW2), and the model validation is IW2 (IW3). With solid black lines, the model predicted results are illustrated in Fig. 8. We can observe that the model outputs and satellite data are typically in agreement. Table 2 shows the statistical results of the nine validation instances that we gathered. The RMSE is 12.92 km, the FD is 18.73 km, and the CC is 0.98, which indicates the model performs well.

ISWs are consecutively generated by semi-diurnal cycles, so more than one tidal cycle can be observed by satellite images. We also tested the forecast model on these ISWs. The model runs iteratively to estimate ISW positions, the model predicted ISW locations is the input for the estimation of the next tidal cycles. The time

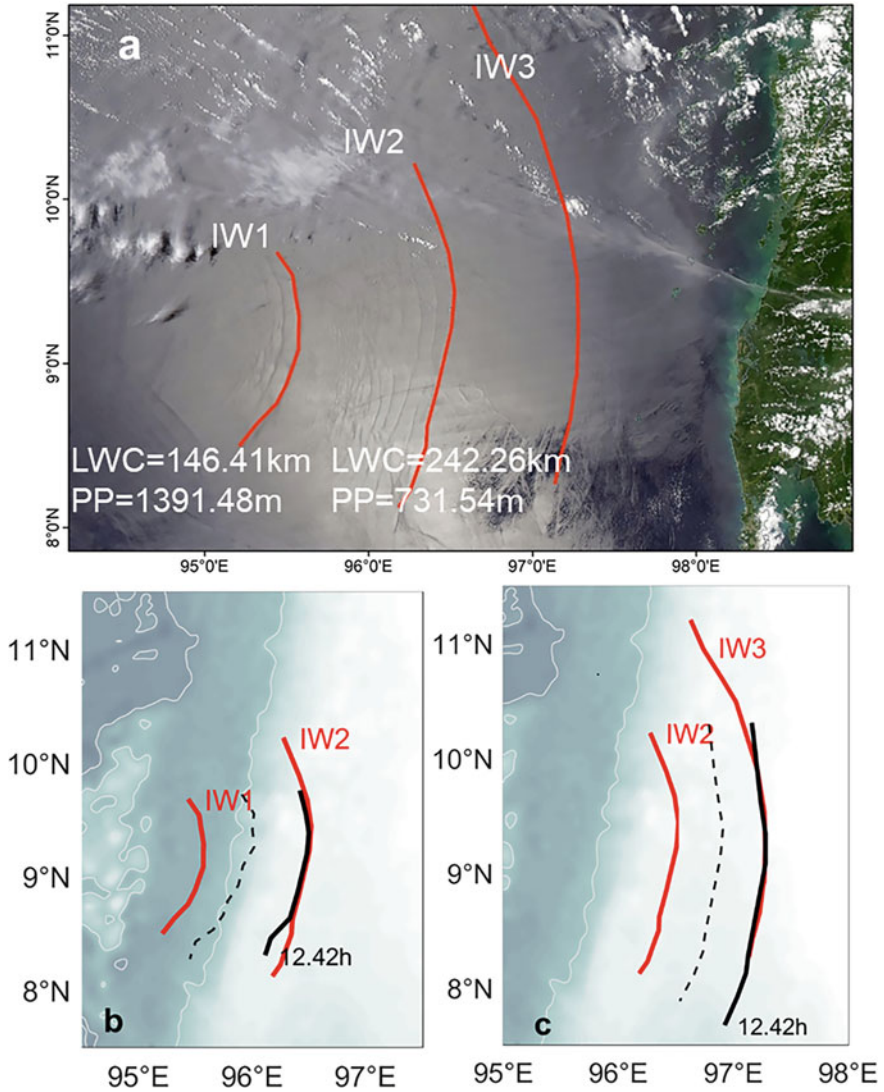


Fig. 7 MODIS images acquired on 9 May, 2017. The ISW locations extracted from satellite images are represented with *red lines*. The solid (*dashed*) *black lines* represent forecast positions after 12.42h (6.21h)

Table 1 Statistical results of forecast model in the Andaman Sea

	RMSE (km)	Fréchet distance (km)	Correlation Coefficient
Case 1	2.69	13.90	0.95
Case 2	2.99	10.66	0.97
Case 3	6.10	18.28	0.96
Case 4	2.50	9.06	0.89
Case 5	3.15	10.43	0.93
Case 6	2.37	6.42	0.93
Case 7	4.12	12.66	0.97
Case 8	1.79	10.28	0.99
Average	3.21	11.46	0.95

step is 6.21 h, three MODIS images were utilized to assess model performance. The predicted ISW positions are displayed as solid black lines. The forecast model estimated ISW positions were consistent with satellite observations, as shown in Fig. 9a and b. The deviations between satellite observations and model estimations are particularly pronounced in Fig. 9c. Because of the complex terrain in the north region of the Andaman Sea, significant variations in the PP distance of ISWs may impact model estimations.

The forecast model estimates were tested after two semi-diurnal tide cycles in the Sulu-Celebes Sea using a MODIS image collected on 29 October 2019 with distinct ISW signals. Figure 10a depicts the results. The Sulu-Celebes Sea ISW sites identified as IW1 are utilized as model input. In Fig. 10b, the ISW prediction is displayed after three semi-diurnal tidal cycles. After two or three semi-diurnal tidal cycles, the model estimations coincide well with satellite measurements.

4 Influence Factors on the ISW Forecast Model

While the model is validated and shows high accuracy as described above, some affecting factors will be discussed. When utilizing the estimated ISW positions of the first semi-diurnal tidal cycle as the model input for the following forecast, errors may be included. Based on the ISW locations, the locations and ocean environment characteristics could be changed, and the LWC could be computed. The ISW PP distance will remain constant in the forecast model's subsequent iterative runs. The predicted outcome deviations will accumulate, resulting in more severe discrepancies in the subsequent iterative prediction. Influences of the time step, input parameter errors, the influence of seasonal variations, and comparison with the KdV equation were discussed in this section.

Because ISWs are frequently generated by semidiurnal tidal cycles, the time step is set to 12.42h by default. When the time step is changed, we will see how

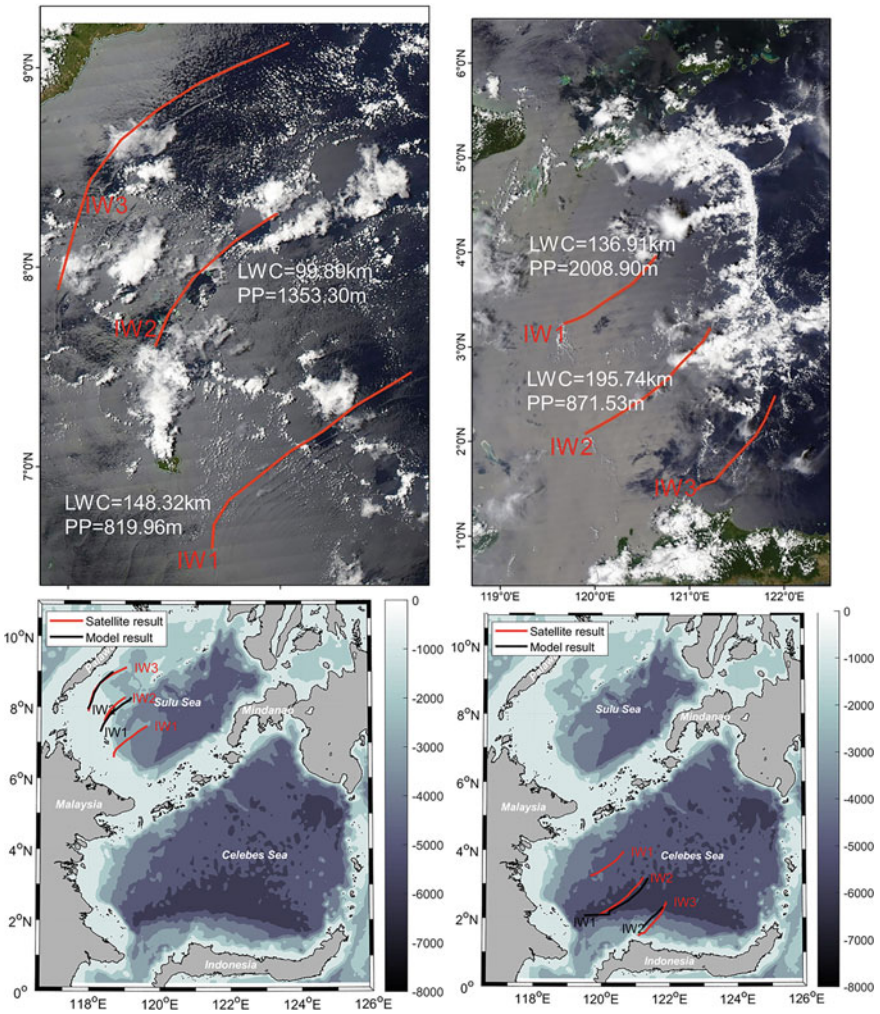


Fig. 8 Forecast model validation cases in the Sulu-Celebes Sea. MODIS image acquired on 18 May 2015 and 29 October 2019 (*upper left and right*) and corresponding forecast results (*lower left and right*)

it affects the model outcomes. Figure 11 depicts the effect of the model running with various time increments. We compared the model forecast results with satellite observations using time increments from 1/4 to one semi-diurnal tidal cycle. ISWs propagate from IW1 to IW2, the depth of water changes from about 2,000m to around 1,000m. The results are poorer using a time step equals 12.42h compared with smaller time steps. The model results were nearly the same when time steps were 3.11 and 4.14 h, and the disparity with the time step equals 6.21 h was similarly low. As a consequence, we infer that the ISWs cross over the steep isobaths when the

Table 2 Statistical results for validation cases in the Sulu-Celebes Sea

	Date	IWs	RMSE (km)	Fréchet Distance (km)	Correlation coefficient
Sulu Sea	18 May 2015	IW1	14.98	16.27	1.00
		IW2	4.72	9.02	0.99
	04 Aug. 2016	IW1	6.66	11.45	1.00
		IW2	5.56	6.15	1.00
Celebes Sea	03 Mar. 2014	IW1	17.06	30.42	0.96
		IW2	17.05	23.24	0.93
	29 Oct. 2019	IW1	16.10	27.56	1.00
		IW2	19.54	18.75	1.00
Average			12.92	18.73	0.98

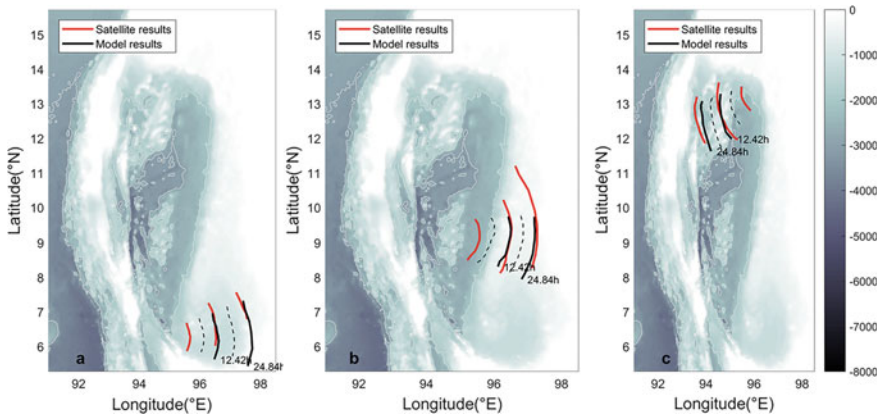


Fig. 9 Model results for ISW propagation after two semi-diurnal tidal cycles

terrain changes dramatically; a lower time step may enhance the prediction result. If the terrain changes gently, the time step may be set to be a large one.

There are eight input parameters for the model which were taken from satellite images or publicly accessible datasets. The forecast model’s outcomes will be influenced by the inaccuracy made in the input parameters. Except for the PP distance and initial propagation direction, all input parameters were modified when we ran the model repeatedly. In the following model predictions, initial ISW PP distance and propagation direction at the starting point will be used as corresponding inputs. When ISWs were not clearly spotted owing to unsatisfied imaging conditions, the PP distance may introduce errors. In four locations of the Andaman Sea, the effects of mistakes in PP distances and ISW propagation direction on the model estimations were studied. Figure 12 depicts the results.

An inaccuracy of $\pm 10^\circ$ was considered to analyze its impacts on the model predictions. The time step is 6.21 h, and the results are presented in Fig. 12a and b. The

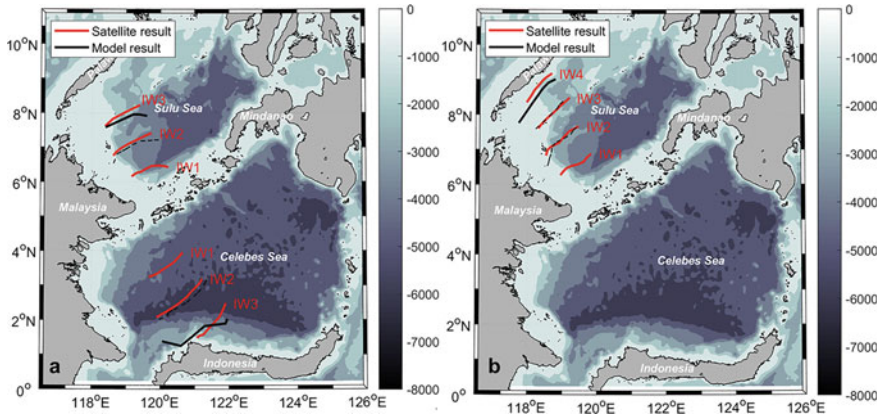


Fig. 10 Iterative forecast results for ISWs propagating after **a** two and **b** three semi-diurnal tidal cycles. ISWs are extracted from MODIS images acquired on **a** 29 October 2019 and **b** 25 February 2015. Black lines: model forecast results. Red lines: satellite observations of the ISW locations. Dashed lines: ISW locations every semi-diurnal tidal cycle

model predicted ISW positions that were near to each other and satellite data which promise the model tolerance on initial propagation direction errors.

Given that a one-pixel inaccuracy resulted in an error of ± 300 m to the PP distances, we compare the results with different PP distance inputs. In the Andaman Sea, two locations with significant and modest water depth fluctuations were examined. The results reveal that, despite the varying inaccuracies in the ISW PP distance, the predicted ISW positions were close to satellite measurements.

The results demonstrate that the proposed forecast model is extremely forgiving of inaccuracies in input parameters like the ISW PP distance and propagation direction of the input ISW wave crest. Minor inaccuracies in some input parameters had no effect on the model’s performance because the ISW propagation was defined by eight factors. Despite this, the model produced results that were close to satellite data. It’s important to remember that input parameter mistakes will accumulate over time when a model runs iteratively. However, the model’s predictions were remained valid after two or three tidal cycles, according to results presented in Fig. 10.

The stratification of the ocean fluctuates because of precipitation and other reasons, ISW propagation is impacted by seasonal variations. The dry season in the Andaman Sea starts from January to April and the rainy season start from May to November. We estimate ISW propagation in four Andaman Sea locations throughout the dry and wet seasons to see how seasonal differences affected the model’s results.

March (August) was selected to represent the dry (rainy) season. The density information was calculated from the WOA2018 dataset. In the two seasons, there were differences in the depth and density of the mixed layer. The predicted outcomes of the forecast model are displayed in Fig. 13. The model-predicted ISW positions were near to each other and also close to the satellite observations in Cases 13a,

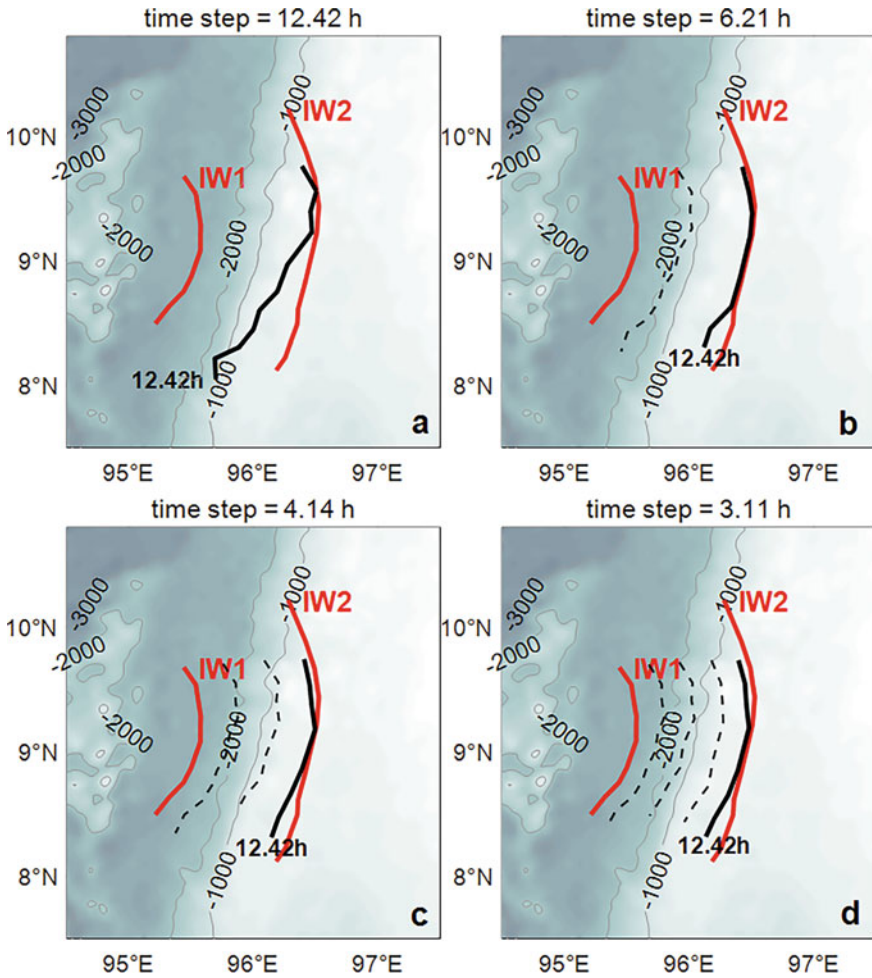


Fig. 11 Comparison of the model forecast results with different time steps

13b, and 13d. The ISW predictions in Fig. 13c exhibited more significant disparities between the outcomes. The findings suggest that ISW propagation shows small seasonal fluctuations in cases a, b, and d, whereas there are disparities in cases c.

Figure 13 depicts the buoyancy frequency of four locations in two seasons. The buoyancy frequency distribution in Fig. 13g and h showed more significant inconsistencies than in Figs. 13e and f. In the dry season, Fig. 13g and 13h had two peaks, while in the rainy season, there was only one peak. The most significant difference is found in Fig. 13g, indicating the most substantial fluctuations in ocean stratification. A larger buoyancy frequency peak indicates stronger ocean stratification. The ISW propagates faster with a larger density difference [27]. This explains why, in Fig. 13c, ISW spread quicker during the rainy season.

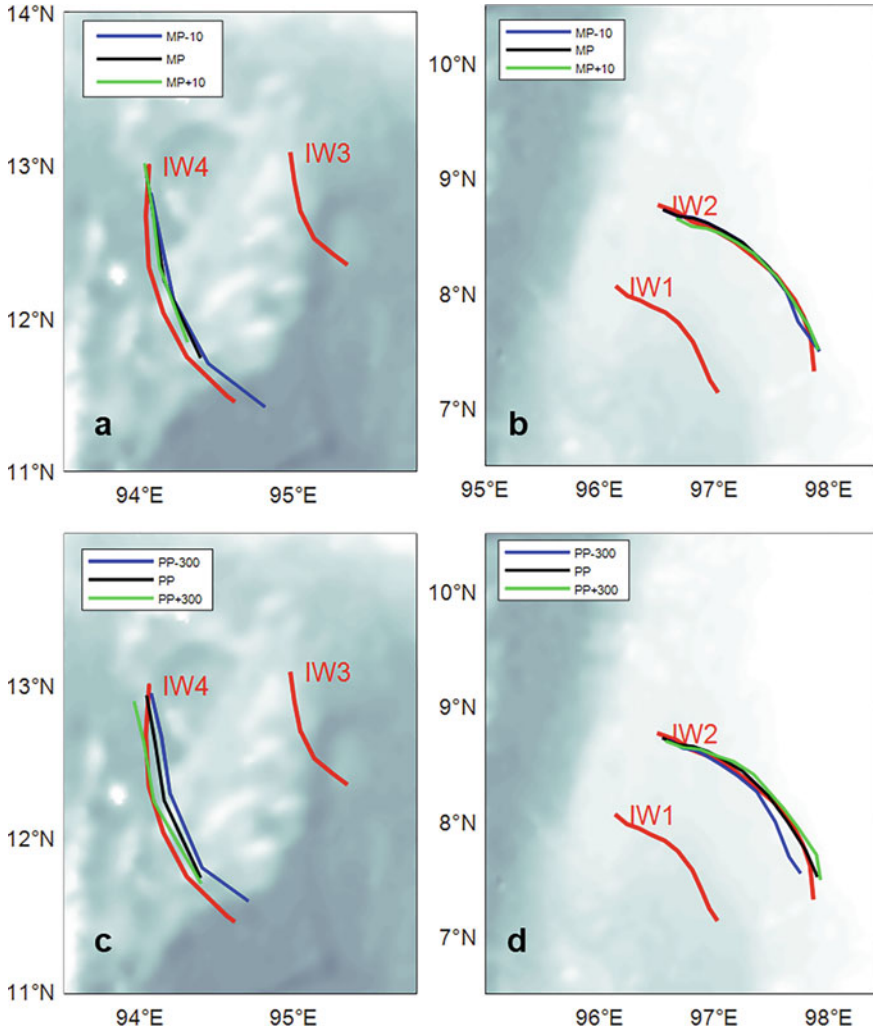


Fig. 12 Influence of input parameter errors for initial main propagation directions (a, b) and PP distances (c, d)

The nonlinear propagation velocity of ISWs are described by the KdV equation [3, 18]:

$$C_p = C_0 + \frac{\alpha}{3} A_0. \tag{5}$$

Where C_0 is the linear phase speed, α is the nonlinear coefficient, and A_0 is the ISW amplitude. The ISW propagation velocity is related to the ISW amplitude which is normally unknown [21]. Based on previous studies [2, 14], the amplitudes of ISWs in the Sulu-Celebes Sea vary from 30 to 90m. To compare the predicted results,

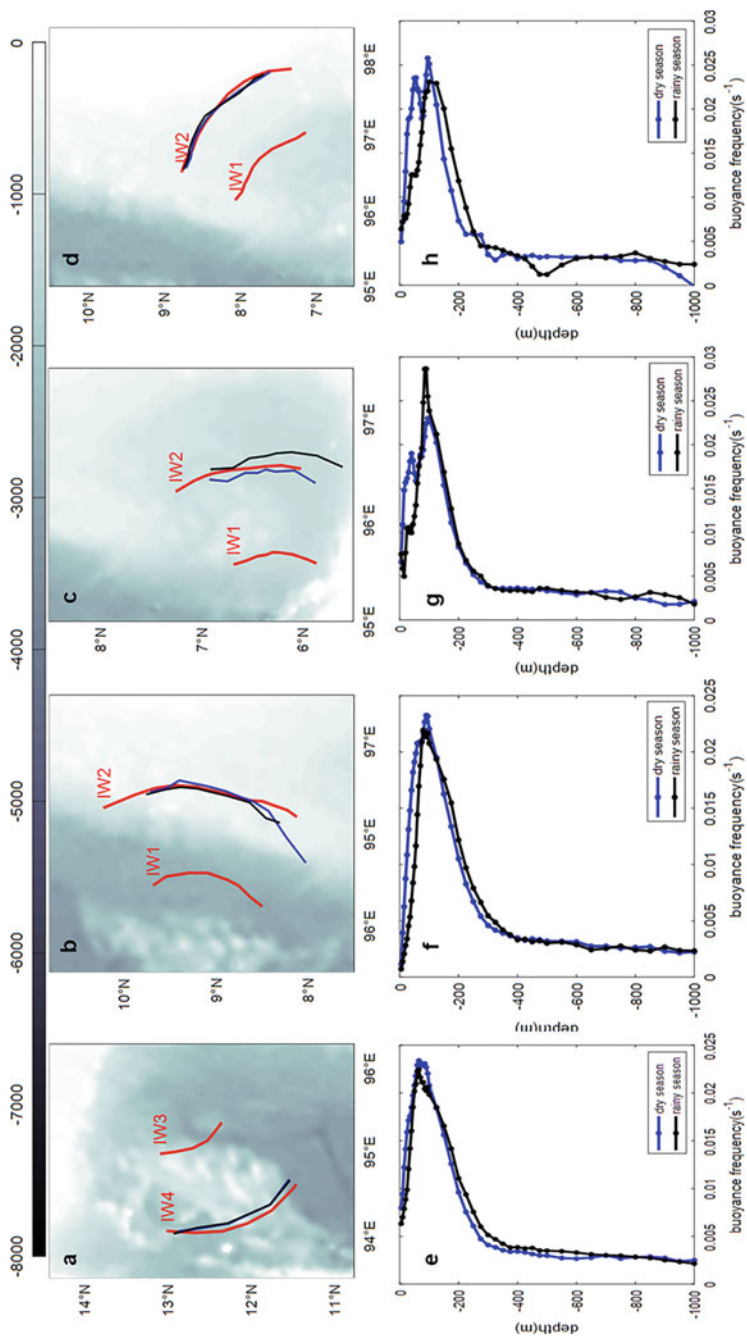


Fig. 13 ISW forecast in different seasons and locations in the Andaman Sea

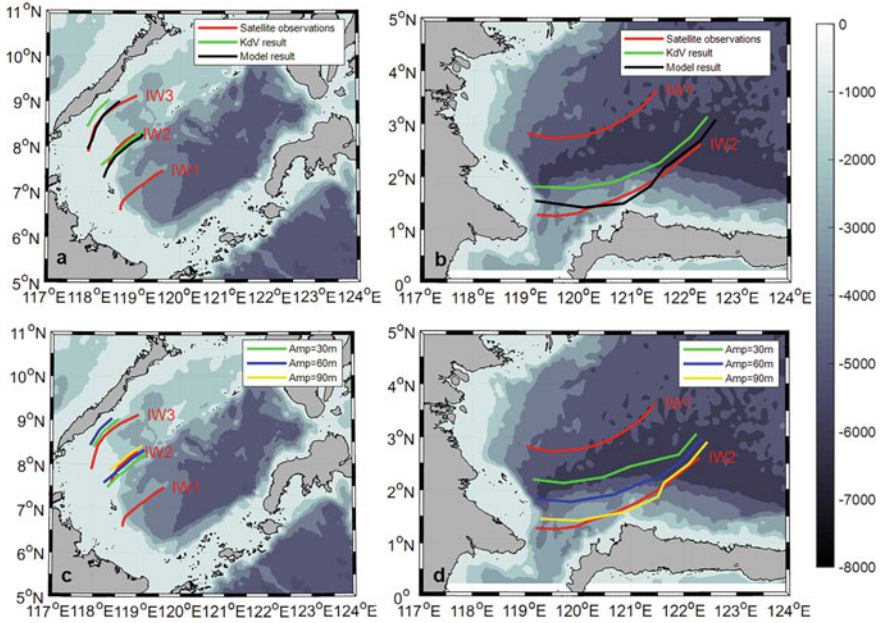


Fig. 14 Comparison of the forecast results of the proposed model and the KdV equation (*upper panels*) and sensitivity of the model results to the ISW amplitude (*lower panels*)

we set the ISW amplitude to 60 m. Figure 14 depicts the comparison results. The RMSE (FD) between the KdV-predicted ISW positions and satellite observations is 44.06 km (59.42 km). The developed model estimation has an RMSE (FD) of 14.67 km (28.35 km). The findings reveal that the proposed model’s predicted ISW positions are closer to satellite results. The KdV equation produces a larger ISW propagation velocity error as compared to satellite data.

ISW amplitudes range from tens to hundreds of meters. The anticipated ISW positions will deviate as a result of this uncertainty. Generally, we assume the linear propagation speed to 2.5 m/s, the mixed layer depth to 100 m, and the water depth is 3000 m. After one semi-diurnal tidal cycle, a 20 m uncertainty in the ISW amplitude results in an ISW position error of 32.42 km. Figure 14 shows the forecast results for three ISW amplitudes (30, 60, and 90 m) using the KdV equation. The results reveal that ISW locations predicted by the KdV equation are sensitive to ISW amplitudes. Because of the amplitude uncertainty, the predicted ISW positions will shift. Without any unknown characteristics, input parameters of the proposed model may be assessed from satellite observations or publically accessible datasets. When input parameters include errors, the proposed forecast model is more resilient.

5 Conclusions and Future Works

To forecast ISW propagations, the ISW forecast model was built using machine learning techniques in this chapter. The training dataset was built using samples extracted from satellite observations and the publicly accessible datasets ETOPO1 and WOA 2018. An FCN network with eight input parameters, which include ocean elements and ISW features, was proposed. The proposed forecast model can predict ISW positions after propagating several time steps when given an initial ISW position. The model's estimation is close to satellite data.

The impact of the model's time step on the predicted outcomes was investigated. When propagating ISWs pass over isobaths, a smaller time step yields better results. Measurements of the PP distance and propagation direction of the given ISW wave crest are easy to have errors. The PP distance and propagation direction of the given ISW wave crest is not modified in subsequent predictions when the forecast model runs repeatedly. The impact of input errors on model estimations was also investigated. The findings reveal that the proposed model is not sensitive to input parameter mistakes. An error of ± 300 m on the PP distance and 10° on the propagation direction did not affect the model estimation greatly. This result demonstrates that the proposed forecast model can still provide reliable results with errors included. The impact of seasonal variation on ISW propagation was investigated. The findings suggest that differences were discovered as a result of seasonal fluctuations in ocean stratification. Comparison with the KdV equation indicates that the forecast model produced superior forecast results and was more resilient.

In contrast to numerical models, the forecast model does not need prior knowledge, a rigid boundary, or beginning conditions. Only satellite observations and publicly available datasets are used. Hence, the model provides an alternate but easy way to forecast ISW propagation and can be readily modified to apply to other ocean regions. The initial position of an ISW wave crest is all that is required to run our forecast model. The machine learning algorithms used here have a lot of potential in oceanographic research for multi-dimensional data processing and forecasting.

References

1. Alpers W (1985) Theory of radar imaging of internal waves. *Nature* 314(6008):245–247
2. Apel JR, Holbrook JR, Liu AK, Tsai JJ (1985) The Sulu Sea internal soliton experiment. *J Phys Oceanogr* 15(12):1625–1651
3. Bai X, Liu Z, Li X, Hu J (2014) Generation sites of internal solitary waves in the southern Taiwan Strait revealed by MODIS true-colour image observations. *Int J Remote Sens* 35(11–12):4086–4098
4. Da Silva J, Magalhaes J (2016) Internal solitons in the Andaman Sea: a new look at an old problem. In: *Remote Sensing of the Ocean, Sea Ice, Coastal Waters, and Large Water Regions 2016*, International Society for Optics and Photonics, vol 9999, p 999907
5. Ham YG, Kim JH, Luo JJ (2019) Deep learning for multi-year ENSO forecasts. *Nature* 573(7775):568–572

6. Hyder P, Jeans D, Cauquil E, Nerzic R (2005) Observations and predictability of internal solitons in the northern Andaman Sea. *Appl Ocean Res* 27(1):1–11
7. Jackson C (2007) Internal wave detection using the moderate resolution imaging spectroradiometer (MODIS). *J Geophys Res: Oceans* 112(C11)
8. Jia Y, Tian Z, Shi X, Liu JP, Chen J, Liu X, Ye R, Ren Z, Tian J (2019) Deep-sea sediment resuspension by internal solitary waves in the northern South China Sea. *Sci Rep* 9(1):1–8
9. Li X, Clemente-Colón P, Friedman KS et al (2000) Estimating oceanic mixed-layer depth from internal wave evolution observed from Radarsat-1 SAR. *J Hopkins APL Tech Dig* 21(1):130–135
10. Li X, Zhao Z, Pichel WG (2008) Internal solitary waves in the northwestern South China Sea inferred from satellite images. *Geophys Res Lett* 35(13)
11. Li X, Jackson CR, Pichel WG (2013) Internal solitary wave refraction at Dongsha Atoll South China Sea. *Geophys Res Lett* 40(12):3128–3132
12. Li X, Liu B, Zheng G, Ren Y, Zhang S, Liu Y, Gao L, Liu Y, Zhang B, Wang F (2020) Deep-learning-based information mining from ocean remote-sensing imagery. *Natl Sci Rev* 7(10):1584–1605
13. Liu AK, Hsu M (2004) Internal wave study in the South China Sea using synthetic aperture radar (SAR). *Int J Remote Sens* 25(7–8):1261–1264
14. Liu AK, Holbrook JR, Apel JR (1985) Nonlinear internal wave evolution in the Sulu Sea. *J Phys Oceanogr* 15(12):1613–1624
15. Liu B, Yang H, Zhao Z, Li X (2014) Internal solitary wave propagation observed by tandem satellites. *Geophys Res Lett* 41(6):2077–2085
16. Liu B, Li X, Zheng G (2019) Coastal inundation mapping from bitemporal and dual-polarization SAR imagery based on deep convolutional neural networks. *J Geophys Res: Oceans* 124(12):9101–9113
17. Magalhaes J, da Silva J (2012) SAR observations of internal solitary waves generated at the Estremadura Promontory off the west Iberian coast. *Deep Sea Res Part I* 69:12–24
18. Osborne A, Burch T (1980) Internal solitons in the Andaman Sea. *Science* 208(4443):451–460
19. Pan X, Wang J, Zhang X, Mei Y, Shi L, Zhong G (2018) A deep-learning model for the amplitude inversion of internal waves based on optical remote-sensing images. *Int J Remote Sens* 39(3):607–618
20. Prechelt L (1998) Automatic early stopping using cross validation: quantifying the criteria. *Neural Netw* 11(4):761–767
21. Wang P, Wang X, Chong J, Lu Y (2016) Optimal parameter estimation method of internal solitary waves in SAR images and the Cramér-Rao bound. *IEEE Trans Geosci Remote Sens* 54(6):3143–3150
22. Wilamowski BM, Yu H (2010) Improved computation for Levenberg-Marquardt training. *IEEE Trans Neural Networks* 21(6):930–937
23. Xu G, Cheng C, Yang W, Xie W, Kong L, Hang R, Ma F, Dong C, Yang J (2019) Oceanic eddy identification using an AI scheme. *Remote Sensing* 11(11):1349
24. Zhang L, Zhang L, Du B (2016) Deep learning for remote sensing data: a technical tutorial on the state of the art. *IEEE Geosci Remote Sensing Mag* 4(2):22–40
25. Zhang X, Li X (2020) Combination of satellite observations and machine learning method for internal wave forecast in the Sulu and Celebes seas. *IEEE Trans Geosci Remote Sens* 59(4):2822–2832
26. Zhang X, Wang J, Sun L, Meng J (2016) Study on the amplitude inversion of internal waves at Wenchang area of the South China Sea. *Acta Oceanol Sin* 35(7):14–19
27. Zhang X, Jie Z, Fan C, Wang J, Meng J (2018) Observations of internal waves with high sampling data of radar altimetry and MODIS images. *Int J Remote Sens* 39(21):7405–7416
28. Zhang X, Li X, Zhang T (2020) Characteristics and generations of internal wave in the Sulu Sea inferred from optical satellite images. *J Oceanol Limnol* 38(5):1435–1444
29. Zhang X, Li X, Zheng Q (2021) A machine-learning model for forecasting internal wave propagation in the Andaman Sea. *IEEE J Sel Topics Appl Earth Obs Remote Sensing* 14:3095–3106

30. Zhang Z, Fringer O, Ramp S (2011) Three-dimensional, nonhydrostatic numerical simulation of nonlinear internal wave generation and propagation in the South China Sea. *J Geophys Res Oceans* 116(C5)
31. Zheng Q, Yuan Y, Klemas V, Yan XH (2001) Theoretical expression for an ocean internal soliton synthetic aperture radar image and determination of the soliton characteristic half width. *J Geophys Res: Oceans* 106(C12):31415–31423

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits any noncommercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if you modified the licensed material. You do not have permission under this license to share adapted material derived from this chapter or parts of it.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



AI-Based Subsurface Thermohaline Structure Retrieval from Remote Sensing Observations



Hua Su, Wenfang Lu, An Wang, and Tianyi Zhang

1 Introduction

The ocean acts as a heat sink and is vital to the Earth's climate system. It regulates and balances the global climate environment through the exchange of energy and substances in the atmosphere and the water cycle. As a huge heat storage, the ocean collects most of the heat from global warming and is sensitive to global climate change. The global ocean hold over 90% of the Earth's increasing heat as a response to the Earth's Energy Imbalance (EEI), leading to substantial ocean warming in recent decades [24, 36]. Subsurface thermohaline are basic and essential dynamic environmental variables for understanding the global ocean's involvement in recent global warming caused by the greenhouse gas emissions. Moreover, many significant dynamic processes and phenomena are located beneath the ocean's surface, and there are many multiscale and complicated 3D dynamic processes in the ocean's interior. To completely comprehend these processes, it is necessary to accurately estimate the thermohaline structure in the global ocean's interior [43].

The ocean has warmed dramatically as a result of heat absorption and sequestration during recent global warming. Meanwhile, the heat content of the ocean has risen rapidly in recent decades [3, 14]. The global upper ocean warmed significantly from 1993 to 2008 [6]. The rate of heat uptake in the intermediate ocean below 300 m has increased much more in recent years [2]. It shows that the warming of the ocean above 300 m slows down, while the warming of the ocean below 300 m speeds up. The ocean system accelerates heat uptake, leading to significant and unprecedented

H. Su (✉) · A. Wang · T. Zhang

Key Laboratory of Spatial Data Mining and Information Sharing of Ministry of Education,
The Academy of Digital China, Fuzhou University, Fuzhou 350108, China
e-mail: suhua@fzu.edu.cn

W. Lu

School of Marine Sciences, Sun Yat-Sen University, and Southern Marine Science and
Engineering Guangdong Laboratory (Zhuhai), Zhuhai 519000, China
e-mail: luwf6@sysu.edu.cn

© The Author(s) 2023

X. Li and F. Wang (eds.), *Artificial Intelligence Oceanography*,
https://doi.org/10.1007/978-981-19-6375-9_5

heat content increasing and worldwide ocean warming, particularly in the subsurface and deeper ocean. This has caused the global ocean heat content hitting a record high in recent years [14, 15]. In addition, the ocean salinity as another key dynamic variable is also crucial for investigations on ocean variability and warming. The salinity mechanism has been proposed to expound how the upper ocean's warming heat transferred to the subsurface and deeper ocean [12], which highlights the importance of salinity distribution in the heat redistribution and the process of ocean warming. Furthermore, the global hydrological cycle is modulated by ocean salinity [4]. The thermohaline expansions, which contribute significantly to sea-level rise, are also linked to ocean temperature and salinity [9]. Therefore, to improve the understanding of the dynamic process and climate variability in subsurface and deeper ocean, deriving and predicting subsurface thermohaline structure is critical [31].

Due to the sparse and uneven sampling of float observations and the lack of time-series data in the ocean, there are still large uncertainties in the estimation of the ocean heat content and the analysis of the ocean warming process [13, 42]. In the era of ship-based measurement, large areas of the global ocean are without or lack of in-situ observation data, especially in the Southern Ocean. The data obtained by the traditional ship-based method not only has limited coverage, but also can't achieve uniform spatiotemporal measurement, hindering the multi-scale studies on the ocean processes. Since 2004, the Argo observation network has achieved the synchronous observation for the upper 2000 m of the global ocean in space and time [39, 51]. However, the number of Argo floats is currently insufficient and far from enough for the global ocean observation, which cannot provide high-resolution internal observation and cannot meet the requirements of global ocean processes and climate change study. Given that satellite remote sensing can obtain large-scale sea surface range and high-resolution sea surface observation data, satellite remote sensing has become an essential technique for ocean observation. Although sea surface satellites can provide large-scale, high-resolution sea surface observation data, they cannot directly observe the ocean subsurface temperature structure [1]. Since many subsurface phenomena have surface manifestations that can be interpreted with the help of satellite measurements, it is able to derive the key dynamic parameters (especially the thermohaline structure) within the ocean from sea surface satellite observations by certain mechanism models. Deep ocean remote sensing (DORS) has the ability to retrieve ocean interior dynamic parameters and enables us to characterize ocean interior processes and features and their implications for the climate change [25].

Previous studies have demonstrated that the DORS technique has a great potential to detect and predict the dynamic parameters of ocean interior indirectly based on satellite measurements combined with float observations [41, 43]. DORS methods mainly include numerical modeling and data assimilation [25], dynamic theoretical approach [30, 48, 50], and empirical statistical and machine learning approach [23, 41]. The accuracy of numerical and dynamic modeling for subsurface ocean simulation and estimation at large scale is not guaranteed due to the complexity and uncertainty of these methods. Reference [47] empirically estimated mesoscale 3D oceanic thermal structures by employing a two-layer model with a set of parameters. Reference [35] determined the vertical structure and transport on a transect across

the North Atlantic Current by integrating historical hydrography with acoustic travel time. Reference [34] estimated the 4D structure of the Southern Ocean from satellite altimetry by a gravest empirical mode projection. However, in the big ocean data and artificial intelligence era, data-driven models, particularly cutting-edge artificial intelligence or machine learning models, perform well and can reach high accuracy in DORS techniques and applications. So far, the empirical statistical and AI models have been well developed and applied, including the linear regression model [19, 23], empirical orthogonal function-based approach [32, 37], geographically weighted regression model [43], and advanced machine learning models, such as artificial neural networks [1, 45], self-organizing map [10], support vector machine [28, 41], random forests (RFs) [43], clustering neural networks [31], and XGBoost [44]. Although traditional machine learning methods have made significant contributions to DORS techniques, they are unable to consider and learn the spatiotemporal characteristics of ocean observation data. In the big earth data era, deep learning has been widely utilized for process understanding for data-driven Earth system science [38]. Deep learning techniques offer great potential in DORS studies to help overcome limitations and improve performance [46]. For example, Long Short-Term Memory (LSTM) can well capture data time-series features and achieves time-series learning [8], and Convolutional Neural Networks (CNN) take into account data spatial characteristics to easily realize spatial learning [5]. Deep learning technique has unleashed great potential in data-driven oceanography and remote sensing research.

This chapter proposes several novel approaches based on ensemble learning and deep learning to accurately retrieve and depict subsurface thermohaline structure from multisource satellite observations combined with Argo in situ data, and highlight the AI applications in the deep ocean remote sensing and climate change studies. We aim to construct AI-based inversion models with strong robustness and generalization ability to well detect and describe the subsurface thermohaline structure of the global ocean. Our new methods can provide powerful AI-based techniques for examining subsurface and deeper ocean thermohaline change and variability which has played a significant role in recent global warming from remote sensing perspective on a global scale.

2 Study Area and Data

The ocean plays a significant role in modulating the global climate system, especially during recent global warming and ocean warming [51]. It serves as a significant heat sink for the Earth's climate system [12], and also acts as an important sink for the increasing CO₂ caused by anthropogenic activities and emissions. The study area focused here is the global ocean which includes the Pacific Ocean, Atlantic Ocean, Indian Ocean, and Southern Ocean (180° W~180° E and 78.375° S~77.625° N).

The satellite-based sea surface measurements adopted in this study include sea surface height (SSH), sea surface temperature (SST), sea surface salinity (SSS), and sea surface wind (SSW). Here, the SSH is obtained from AVISO satellite altimetry. The

SST is acquired from Optimum Interpolation Sea-Surface Temperature (OISST) data. The SSS is obtained from the Soil Moisture and Ocean Salinity (SMOS). The SSW is acquired from Cross-Calibrated Multi-Platform (CCMP). The longitude (LON) and latitude (LAT) georeference information are also employed as supplementary input parameters. All sea surface variables above have the same $0.25^\circ \times 0.25^\circ$ spatial resolution. The subsurface temperature (ST) and salinity (SS) data are from Argo gridded products with $1^\circ \times 1^\circ$ spatial resolution. This study adopted Argo gridded data for subsurface ocean upper 1,000 m with 16 depth levels as labeling data. We initially applied the nearest neighbor interpolation approach to unify the satellite-based sea surface variables to $1^\circ \times 1^\circ$ spatial resolution.

All the aforementioned satellite-based sea surface variables and Argo gridded data should be subtracted their climatology (baseline: 2005–2016) to obtain their anomaly fields in order to avoid the climatology seasonal variation signal [41]. In this study, We primarily focus on the nonseasonal anomaly signals, which are more difficult to detect but more significant for climate change. We applied a maximum-minimum normalization approach to normalize the training dataset to the range of [0, 1]. The testing dataset was likewise subjected to the corresponding normalization, which can effectively prevent data leakage during the modeling.

3 Retrieving Subsurface Thermohaline Based on Ensemble Learning

Here, the specific procedure of subsurface thermohaline retrieval based on machine learning approaches contains three technical steps. Firstly, the training dataset for the model was constructed. We selected the satellite-based sea surface parameters (SSH, SST, SSS, SSW) as input variables for AI-based models, and the subsurface temperature anomaly (STA) and salinity anomaly (SSA) from Argo gridded data were adopted as data labels for training and testing. Moreover, all the input surface and subsurface datasets were uniformly normalized and randomly separated into a training dataset (60%) and a testing dataset (40%), which were utilized to train and test the AI-based models, respectively. Secondly, the model was trained using the training dataset. The model's hyper-parameters were tuned by using Bayesian optimization approach, and then a proper machine learning model was well set up using the optimal input parameters. Finally, the prediction was performed based on the trained model. We predicted the STA and SSA by the optimized model, and then evaluated the model performance and accuracy by determination coefficient (R^2) and root-mean-square error (RMSE).

3.1 *EXtreme Gradient Boosting (XGBoost)*

Gradient Boosting Decision Tree (GBDT) as a boosting algorithm is an iterative decision trees algorithm and is composed of multiple decision trees [16]. EXtreme

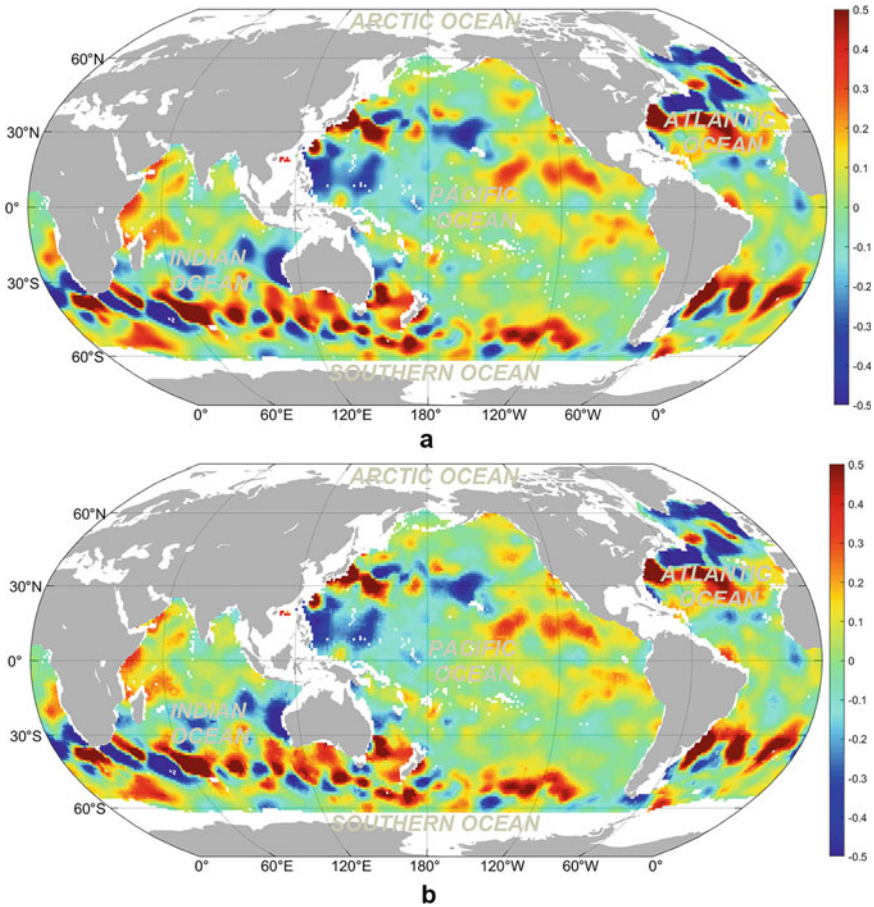


Fig. 1 Spatial distribution of the **a** Argo STA and the **b** XGBoost-estimated STA in December 2015 at 600 m depth

Gradient Boosting (XGBoost) is an upgraded GBDT ensemble learning algorithm [11], as well as an optimized distributed gradient boosting library. XGBoost implements an ensemble machine learning algorithm based on decision tree that adopts a gradient boosting framework, and also provides a parallel tree boosting that solve many data science problems in an efficient, flexible and accurate way. To achieve the optimal model performance, the parameter tuning is essential during the modeling. XGBoost contains several hyper-parameters which are related to the complexity and regularization of the model [49], and they must be optimized in order to refine the model and improve the performance. Here, we used the well-performed Bayesian optimization approach to tune the XGBoost hyper-parameters.

Figures 1–2 show the spatial distribution of subsurface temperature and salinity anomalies (STA and SSA) of the global ocean from the XGBoost-based result and Argo gridded data in December 2015 at 600 m depth. It is clear that both the XGBoost-

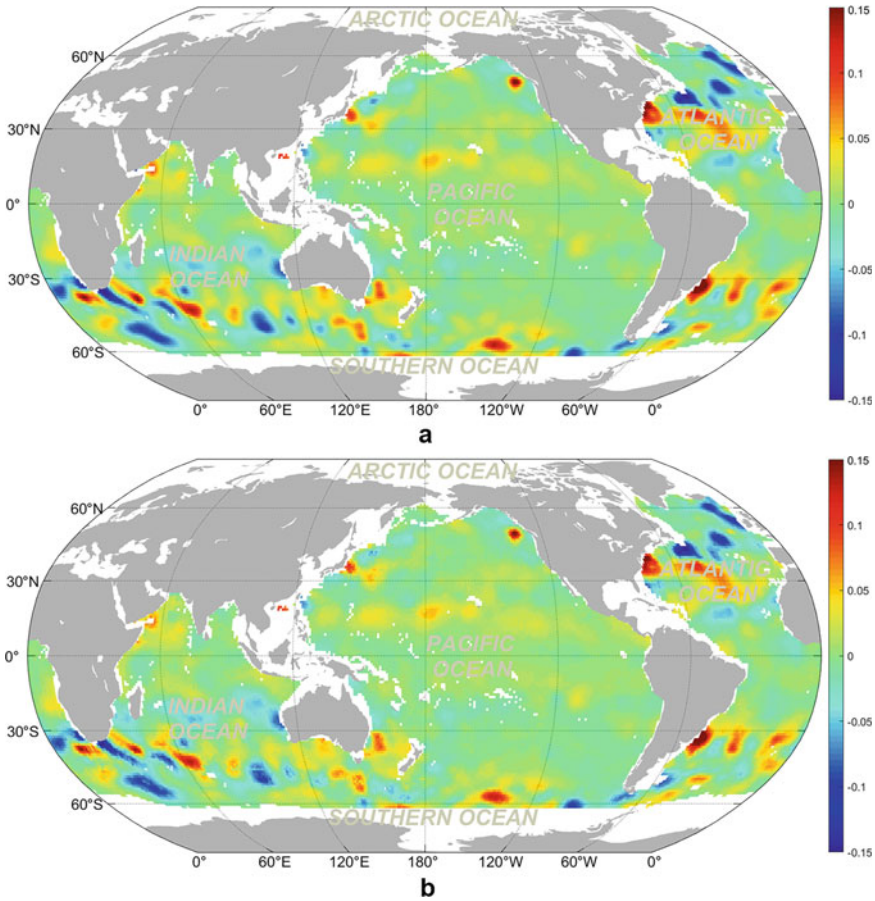


Fig. 2 Spatial distribution of the **a** Argo SSA and the **b** XGBoost-estimated SSA in December 2015 at 600 m depth

estimated STA and SSA were significantly consistent with the Argo gridded STA and SSA at 600 m depth. The R^2 of STA/SSA between Argo gridded data and XGBoost-estimated result is 0.989/0.981, and the RMSE is 0.026 °C/0.004 PSU.

3.2 Random Forests (RFs)

Random Forests (RFs) are a popular and well-used ensemble learning method for data classification and regression. Reference [7] proposed the general strategy of RFs, which fit numerous decision trees on various data subsets by randomly resampling the training data. RFs adopt averaging to improve the prediction accuracy and control overfitting, and correct for the decision tree’s tendency of overfitting. RFs have been

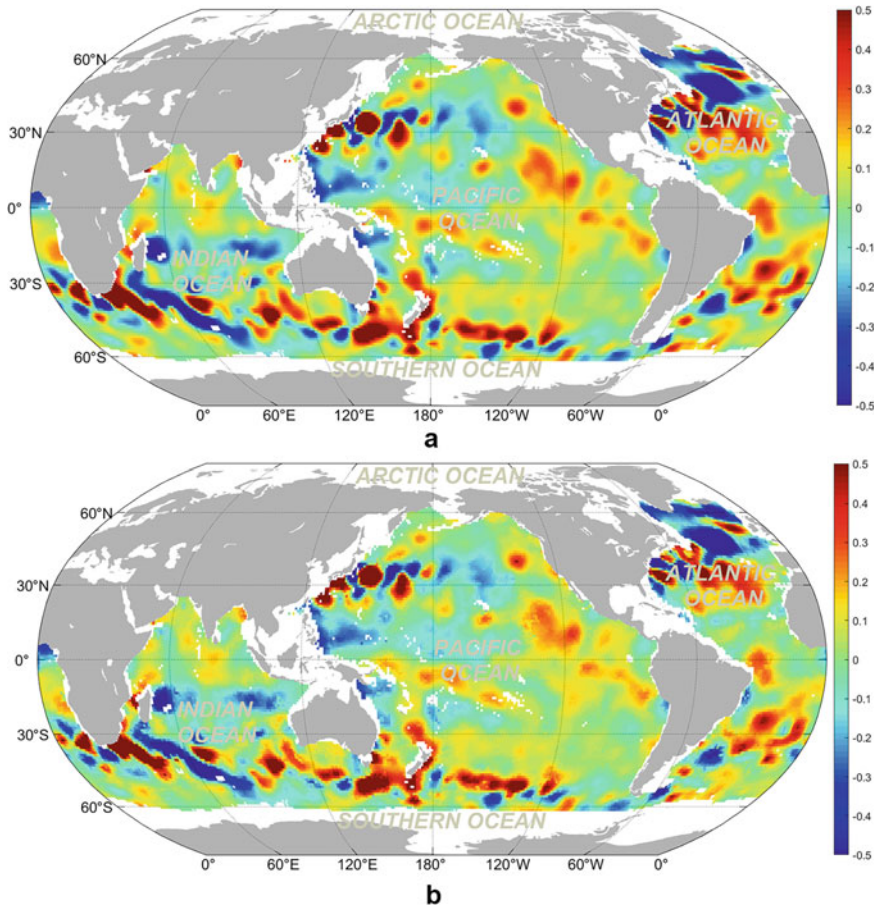


Fig. 3 Spatial distribution of the **a** Argo STA and the **b** RFs-estimated STA in June 2015 at 600 m depth

effectively applied in varying remote sensing fields [21, 53] and generally perform very well. Several advantages make RFs well-suited to remote sensing studies [20, 52].

The basic strategy of RFs is to grow a number of decision trees on random subsets of the training data [40], and determine the decision rules, and choose the best split for each node splitting [29]. This strategy performs well compared to many other classifiers and makes it robust against overfitting [7]. RFs only require two input parameters for training, the number of trees in the forest (n_{tree}) and the number of variables/features in the random subset at each node (m_{try}), and both parameters are generally insensitive to their values [29].

Figures 3–4 show the spatial distribution of subsurface thermohaline anomalies of the global ocean from RFs-based result and Argo gridded data in June 2015 at

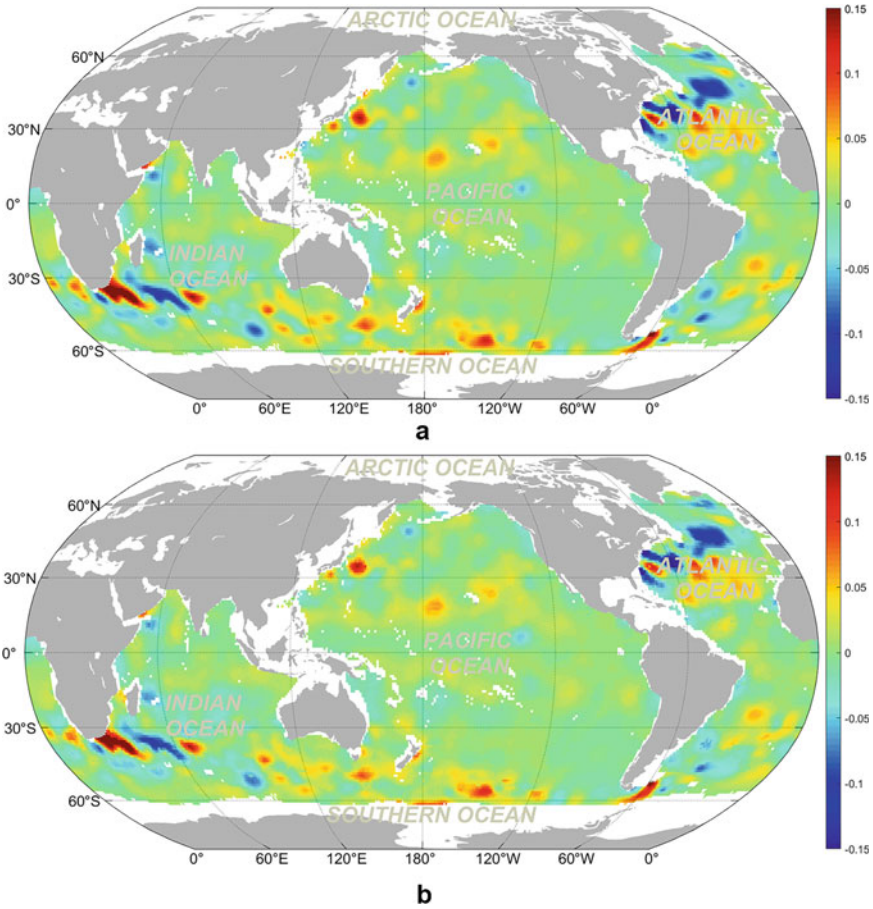


Fig. 4 Spatial distribution of the **a** Argo SSA and the **b** RFs-estimated SSA in June 2015 at 600m depth

600 m depth. It is clear that the spatial distribution and pattern between RFs-estimated results and Argo gridded data are quite similar. The R^2 of STA/SSA between Argo data and XGBoost-estimated result is 0.971/0.972, and the RMSE is 0.042 °C/0.005 PSU.

4 Predicting Subsurface Thermohaline Based on Deep Learning

The predicting process for subsurface thermohaline based on deep learning includes three steps. Firstly, the training dataset combined satellite-based sea surface parameters (SSH, SST, SSS, SSW) with Argo subsurface data as training label were prepared. Secondly, we carried out a hyperparameter tuning based on a grid-search strategy to achieve an optimal deep learning model by training. Here, we set up the time-series deep learning models by adopting the time-series data as the training dataset and the rest as the testing dataset, so as to realize time-series subsurface thermohaline prediction. Finally, the performance measures of RMSE and R^2 were adopted to evaluate the model performance and accuracy.

4.1 Bi-Long Short-Term Memory (Bi-LSTM)

The LSTM is a sort of recurrent neural network [22], which is well-suited to time-series modeling and has been widely applied in natural language processing and speech recognition. The primary principle behind LSTM is to leverage the target variable's historical information. Unlike traditional feedforward neural networks, the training errors in an LSTM propagate over a time sequence, capturing the time-dependent relationship of the training data's historical information [18]. Bi-Long Short-Term Memory (Bi-LSTM) is an upgraded LSTM algorithm. The Bi-LSTM consists of two unidirectional LSTM that processes the input sequence forward and backward meanwhile, and captures the information ignored by the unidirectional LSTM.

To ensure the Bi-LSTM model can achieve good performance and high accuracy, it is necessary to select and tune the proper hyperparameters as the input of Bi-LSTM model. Here, we randomly picked 20% of the training dataset for Bi-LSTM hyperparameter tuning, so as to achieve the optimal model input. The Bayesian optimization approach was utilized in this study to obtain the best number of layers and neurons for Bi-LSTM network. By model testing, we finally selected a neural network with three layers and neuron counts of 32, 64, and 64 for respective layer. Moreover, the batch normalization was conducted after the hidden layer of each network. According to the previous practice, the optimal performance could be effectively attained with mini-batch sizes ranging from 2 to 32 [33]. Thus, the best batch size was set to 32 for the model. In addition, the optimal epoch of the STA network was set to 257, while the best one of the SSA network was set to 81. Moreover, We adopted the RMSE, R^2 , and Spearman's rank correlation coefficient (ρ) to obtain the optimal Bi-LSTM timestep. The results demonstrate that the Bi-LSTM model performs optimally when the network timestep is set to 10. Thus, the timestep here was set as 10.

Table 1 The different datasets feeded to Bi-LSTM model

Target month	Training dataset	Predicting dataset	Testing dataset
2015.12	2010.12–2015.11	2015.03–2015.12	2015.12

We employed the data from December 2010 to November 2015 as the training dataset and the data in December 2015 as the testing dataset. The testing dataset adopted the target month dataset for performance evaluation (Table 1). In general, Bi-LSTM was characterized by a whole temporal sequence in both training and prediction, but for the accuracy validation, we only focused on the target month.

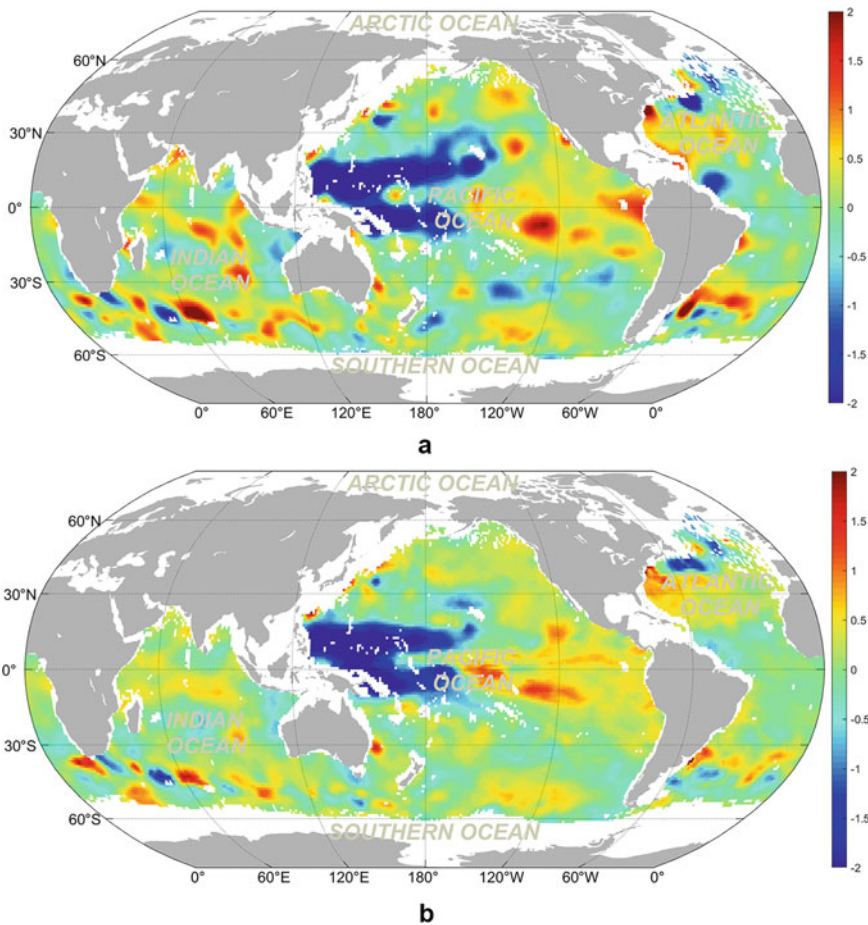


Fig. 5 Spatial distribution of the **a** Argo STA and the **b** LSTM-predicted STA in December 2015 at 200m depth

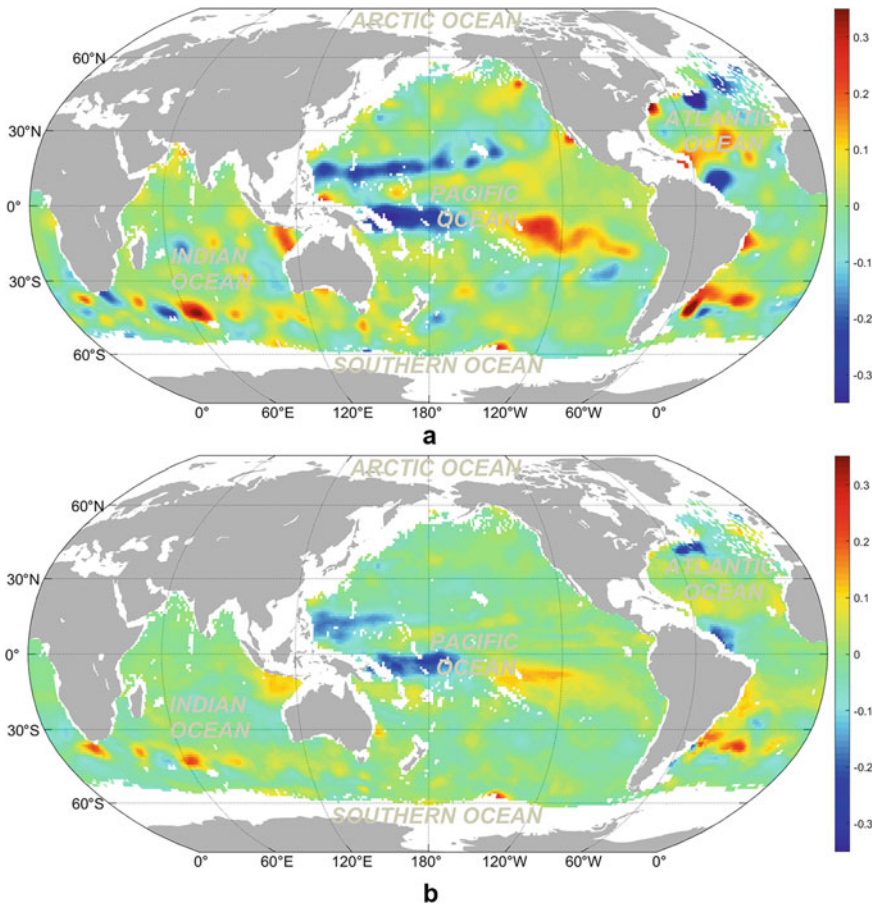


Fig. 6 Spatial distribution of the **a** Argo SSA and the **b** LSTM-predicted SSA in December 2015 at 200 m depth

When constructed the input dataset for Bi-LSTM, we restructured the data grid by grid with time sequence according to the rule of $X_{i=1}^{j=1}, X_{i=1}^{j=2} \dots X_{i=1}^{j=60}, X_{i=2}^{j=1} \dots X_{i=2}^{j=60}, \dots, X_{i=24922}^{j=1} \dots X_{i=24922}^{j=60}$ (i represents the grid point, j represents the month).

Figures 5–6 show the spatial distribution of subsurface temperature and salinity anomalies of the global ocean from the LSTM-predicted result and Argo gridded data in December 2015 at 200 m depth. It is clear that the LSTM-predicted result can accurately retrieve and capture most anomaly signals in the subsurface ocean. The R^2 of STA/SSA between Argo gridded data and LSTM-predicted result is 0.728/0.476, and the RMSE is 0.378 °C/0.055 PSU.

Figure 7 is the meridional profile (at longitude 190°) for Argo gridded and LSTM-predicted STA for vertical comparison and validation. The results presented that the two vertical profiles are highly consistent in the vertical distribution pattern, and over

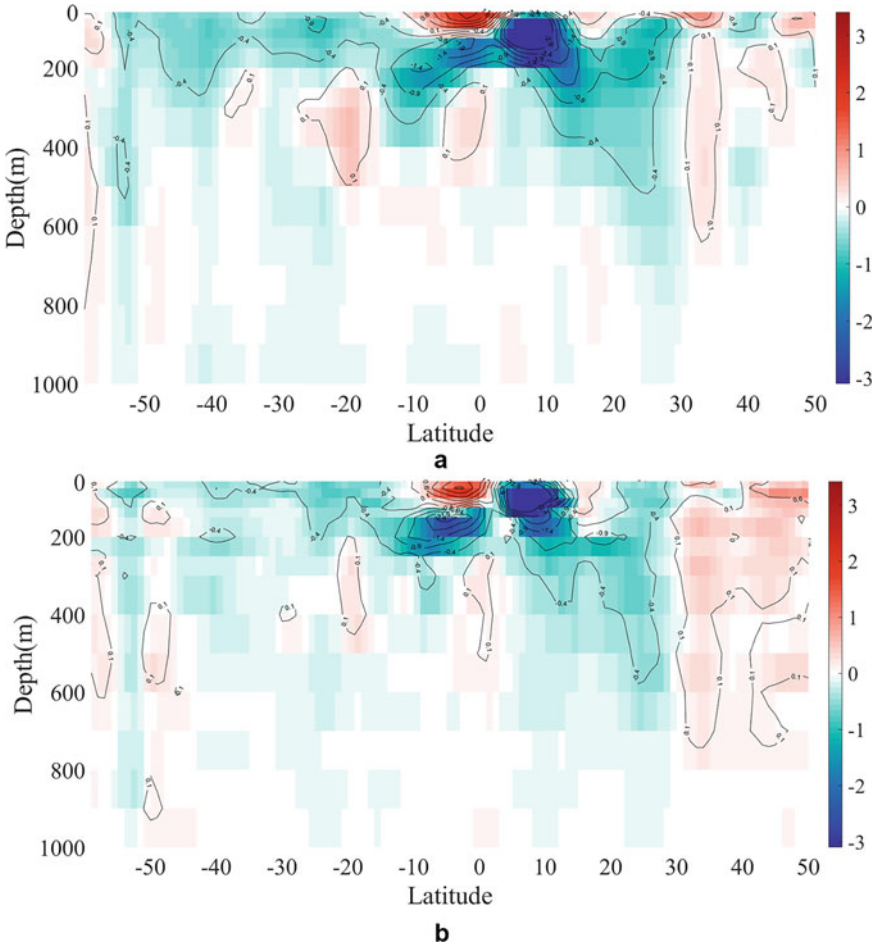


Fig. 7 The meridional vertical profile of the STA in December 2015 at the longitude of 190° for **a** Argo gridded data, and **b** LSTM-predicted result

99.75% of the profile points were within $\pm 1^\circ\text{C}$ prediction error, while over 99.44% of the profile points were within $\pm 0.5^\circ\text{C}$ error. Figure 8 is the same meridional profile for Argo gridded and LSTM-predicted SSA for vertical comparison and validation. The results indicated that the two vertical profiles match well in the vertical distribution pattern, and over 99.55% of the profile points were within ± 0.2 PSU prediction error, while over 99.36% of the profile points were within ± 0.1 PSU error. The results demonstrated that the model prediction performance for STA and SSA are excellent with high accuracy.

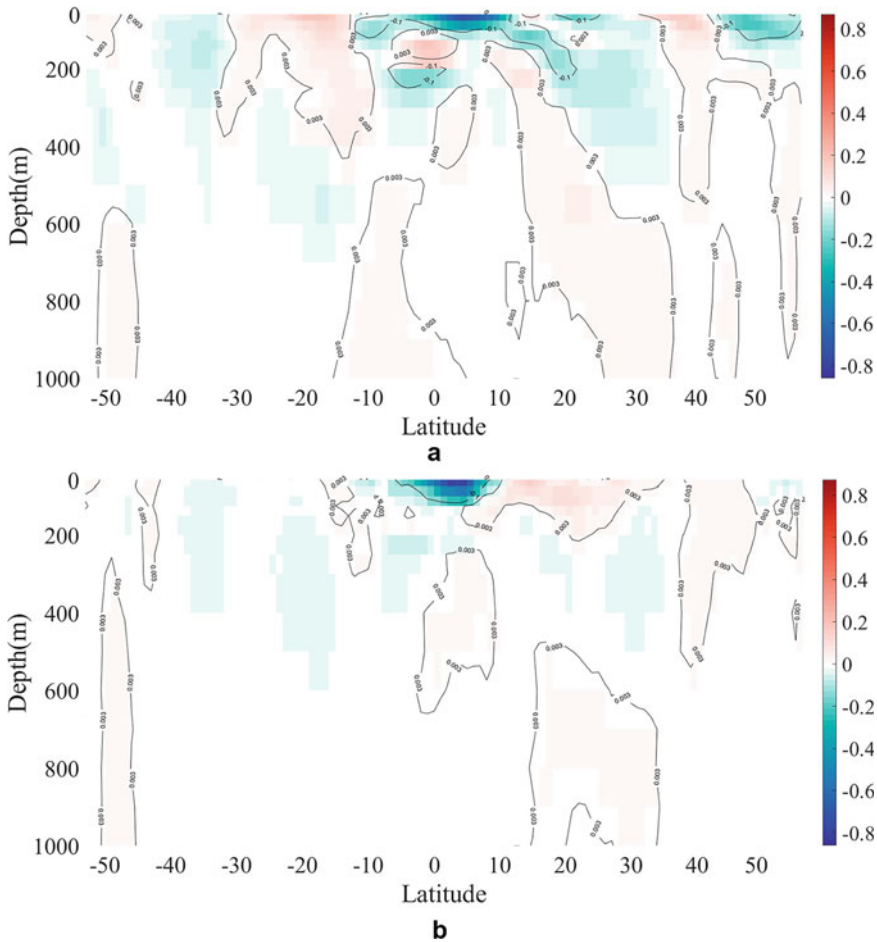


Fig. 8 The meridional vertical profile of the SSA in December 2015 at the longitude of 190° for **a** Argo gridded data, and **b** LSTM-predicted result

4.2 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a well-known deep learning algorithm. [17] proposed a neural network structure, including convolution and pooling layers, which can be regarded as the first implementation of the CNN model. On this basis, [27] proposed the LeNet-5 network, which used the error backpropagation algorithm in the network structure and was considered a prototype of CNN. Until 2012, the deep network structure and dropout method were applied in the ImageNet image recognition contest [26], and significantly reduced the error rate, which opened a new era in the image recognition field. So far, the CNN technique has already been

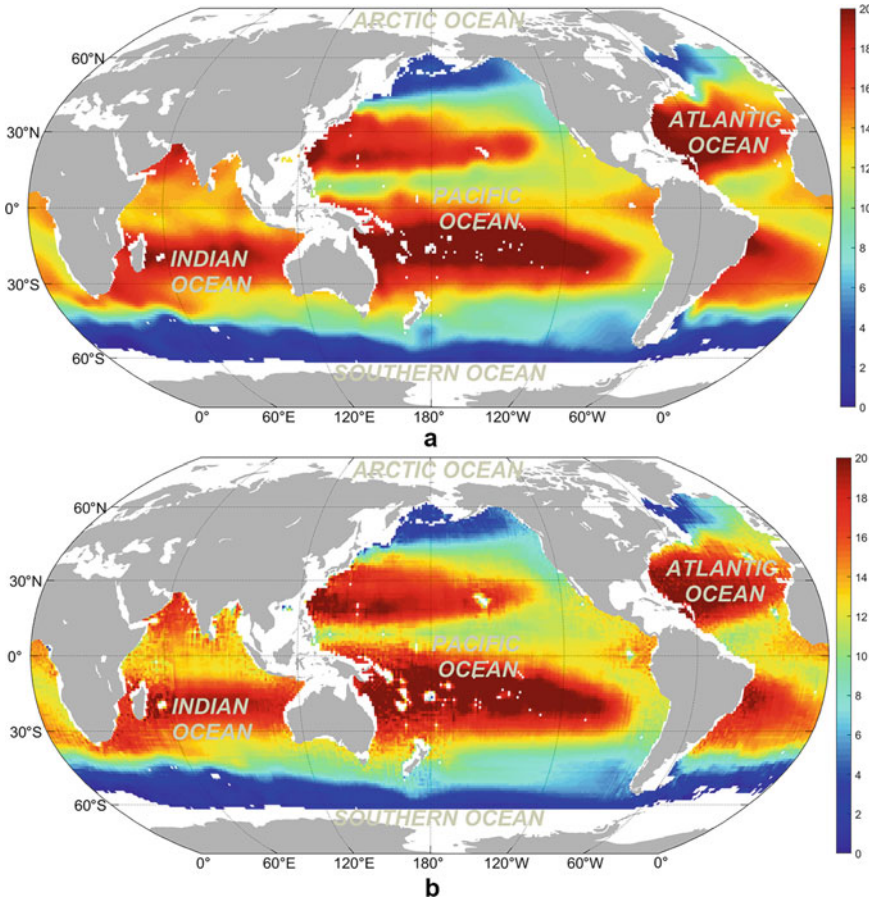


Fig. 9 Spatial distribution of the **a** Argo ST and the **b** CNN-predicted ST in December 2015 at 200 m depth

widely utilized in a variety of applications, including climate change and marine environmental remote sensing applications [5]. Here, the CNN algorithm combined with satellite observations was employed to predict ocean subsurface parameters.

We utilized the CNN approach to retrieve ocean subsurface temperature (ST) and salinity (SS) based on satellite remote sensing data directly. Figures 9–10 show the spatial distribution of subsurface thermohaline of the global ocean from the CNN-predicted and Argo gridded data in December 2015 at 200 m depth. The R^2 of STA/SSA between Argo gridded data and CNN-predicted result is 0.972/0.822, and the RMSE is 0.924 °C/0.293 PSU.

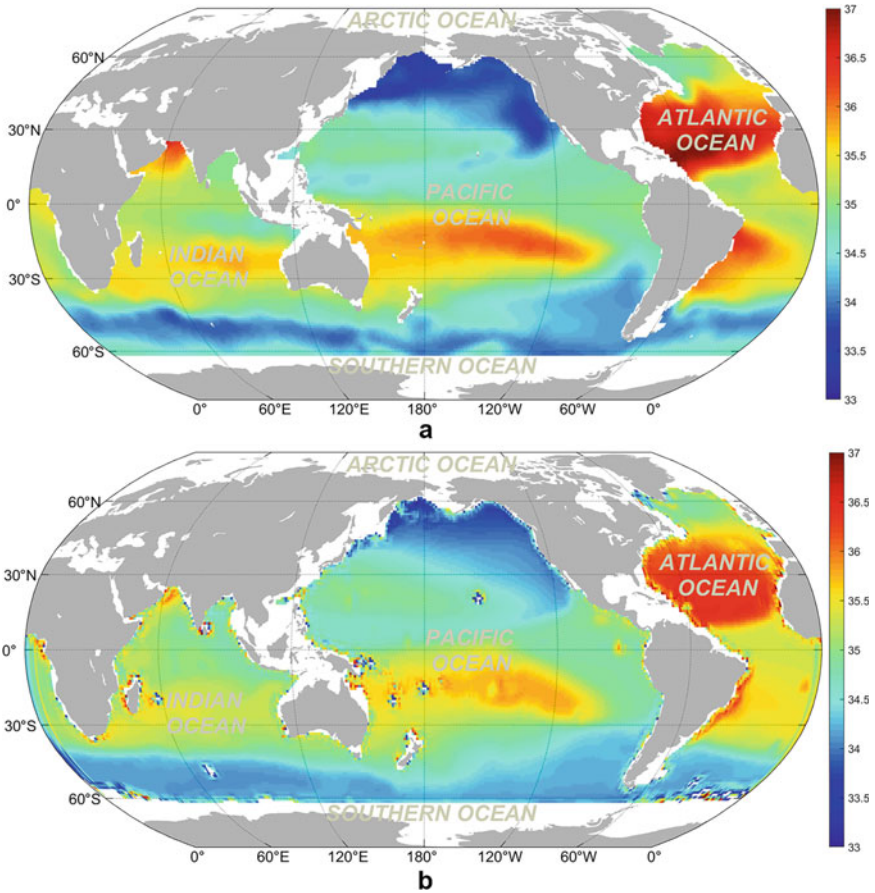


Fig. 10 Spatial distribution of the **a** Argo SS and the **b** CNN-predicted SS in December 2015 at 200m depth

5 Conclusions

This chapter proposes several AI-based techniques (ensemble learning and deep learning) for retrieving and predicting subsurface thermohaline in the global ocean. The proposed models are proved to estimate the subsurface temperature and salinity structures accurately in the global ocean through multisource satellite remote sensing observations (SSH, SST, SSS, and SSW) combined with Argo float data. The performance and accuracy of the models are well evaluated by Argo in situ data. The results demonstrate that the AI-based model has strong robustness and generalization ability, and can be well applied to the prediction and reconstruction of subsurface dynamic environmental parameters.

We employ XGBoost and RFs ensemble learning algorithms to derive the subsurface temperature and salinity of the global ocean, and the R^2 /RMSE of XGBoost retrieved STA and SSA are 0.989/0.026 °C and 0.981/0.004 PSU, and the R^2 /RMSE of RFs retrieved STA and SSA are 0.971/0.042 °C and 0.972/0.005 PSU. Moreover, Bi-LSTM and CNN deep learning algorithms are adopted to time-series predicting of subsurface thermohaline, the R^2 /RMSE of Bi-LSTM predicted STA and SSA are 0.728/0.378 °C and 0.476/0.055 PSU, the R^2 / RMSE of CNN predicted ST and SS are 0.972/0.924 °C and 0.822/0.293 PSU (CNN to predict the ST and SS directly). Overall, ensemble learning algorithms which are suited for small data modeling can be used to well retrieve mono-temporal subsurface thermohaline structure, while deep learning algorithms which are fit for big data modeling can be well adopted to predict time-series subsurface thermohaline structure.

In the future, we can employ longer time-series of remote sensing data for modeling and utilize more advanced deep learning algorithms to improve the model applicability and robustness. We should further promote the application of AI and deep learning techniques in the deep ocean remote sensing and data reconstruction for revisiting global ocean warming and climate change. The powerful AI technology shows great potential for detecting and predicting the subsurface environmental parameters based on multisource satellite measurements, and can provide a useful technique for promoting the studies of deep ocean remote sensing as well as ocean warming and climate change during recent decades.

References

1. Ali M, Swain D, Weller R (2004) Estimation of ocean subsurface thermal structure from surface parameters: a neural network approach. *Geophys Res Lett* 31(20)
2. Allison L, Roberts C, Palmer M, Hermanson L, Killick R, Rayner N, Smith D, Andrews M (2019) Towards quantifying uncertainty in ocean heat content changes using synthetic profiles. *Environ Res Lett* 14(8):084037
3. Balmaseda MA, Trenberth KE, Källén E (2013) Distinctive climate signals in reanalysis of global ocean heat content. *Geophys Res Lett* 40(9):1754–1759
4. Bao S, Zhang R, Wang H, Yan H, Yu Y, Chen J (2019) Salinity profile estimation in the Pacific Ocean from satellite surface salinity observations. *J Atmos Oceanic Tech* 36(1):53–68
5. Barth A, Alvera-Azcárate A, Licer M, Beckers JM (2020) DINCAE 1.0: a convolutional neural network with error estimates to reconstruct sea surface temperature satellite observations. *Geosci Model Develop* 13(3):1609–1622
6. Boyer T, Domingues CM, Good SA, Johnson GC, Lyman JM, Ishii M, Gouretski V, Willis JK, Antonov J, Wijffels S et al (2016) Sensitivity of global upper-ocean heat content estimates to mapping methods, XBT bias corrections, and baseline climatologies. *J Clim* 29(13):4817–4842
7. Breiman L (2001) Random forests. *Mach Learn* 45(1):5–32
8. Buongiorno Nardelli B (2020) A deep learning network to retrieve ocean hydrographic profiles from combined satellite and in situ measurements. *Remote Sensing* 12(19):3151
9. Cazenave A, Meyssignac B, Ablain M, Balmaseda M, Bamber J, Barletta V, Beckley B, Benveniste J, Berthier E, Blazquez A et al (2018) Global sea-level budget 1993-present. *Earth System Science Data* 10(3):1551–1590

10. Chen C, Yang K, Ma Y, Wang Y (2018) Reconstructing the subsurface temperature field by using sea surface data through self-organizing map method. *IEEE Geosci Remote Sens Lett* 15(12):1812–1816
11. Chen T, Guestrin C (2016) XGBoost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp 785–794
12. Chen X, Tung KK (2014) Varying planetary heat sink led to global-warming slowdown and acceleration. *Science* 345(6199):897–903
13. Cheng L, Zhu J (2014) Uncertainties of the ocean heat content estimation induced by insufficient vertical resolution of historical ocean subsurface observations. *J Atmos Oceanic Tech* 31(6):1383–1396
14. Cheng L, Abraham J, Zhu J, Trenberth KE, Fasullo J, Boyer T, Locarnini R, Zhang B, Yu F, Wan L et al (2020) Record-setting ocean warmth continued in 2019. *Adv Atmos Sci* 37(2):137–142
15. Cheng L, Abraham J, Trenberth KE, Fasullo J, Boyer T, Locarnini R, Zhang B, Yu F, Wan L, Chen X et al (2021) Upper ocean temperatures hit record high in 2020. *Adv Atmos Sci* 38(4):523–530
16. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. In: *The Annals of Statistics*, pp 1189–1232
17. Fukushima K, Miyake S (1982) Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In: *Competition and cooperation in neural nets*. Springer, pp 267–285
18. Goodfellow I, Bengio Y, Courville A (2016) *Deep learning*. MIT Press
19. Guinehut S, Dhomps AL, Larnicol G, Le Traon PY (2012) High resolution 3-D temperature and salinity fields derived from in situ and satellite observations. *Ocean Sci* 8(5):845–857
20. Guo L, Chehata N, Mallet C, Boukir S (2011) Relevance of airborne lidar and multispectral image data for urban scene classification using Random Forests. *ISPRS J Photogramm Remote Sens* 66(1):56–66
21. Ham J, Chen Y, Crawford MM, Ghosh J (2005) Investigation of the Random Forest framework for classification of hyperspectral data. *IEEE Trans Geosci Remote Sens* 43(3):492–501
22. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
23. Jeong Y, Hwang J, Park J, Jang CJ, Jo YH (2019) Reconstructed 3-D ocean temperature derived from remotely sensed sea surface measurements for mixed layer depth analysis. *Remote Sensing* 11(24):3018
24. Johnson GC, Lyman JM (2020) Warming trends increasingly dominate global ocean. *Nat Clim Chang* 10(8):757–761
25. Klemas V, Yan XH (2014) Subsurface and deeper ocean remote sensing from satellites: An overview and new results. *Prog Oceanogr* 122:1–9
26. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst* 25:1097–1105
27. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86(11):2278–2324
28. Li W, Su H, Wang X, Yan X (2017) Estimation of global subsurface temperature anomaly based on multisource satellite observations. *J Remote Sens* 21:881–891
29. Liaw A, Wiener M et al (2002) Classification and regression by Random Forest. *R news* 2(3):18–22
30. Liu L, Xue H, Sasaki H (2019) Reconstructing the ocean interior from high-resolution sea surface information. *J Phys Oceanogr* 49(12):3245–3262
31. Lu W, Su H, Yang X, Yan XH (2019) Subsurface temperature estimation from remote sensing data using a clustering-neural network method. *Remote Sens Environ* 229:213–222
32. Maes C, Behringer D, Reynolds RW, Ji M (2000) Retrospective analysis of the salinity variability in the western tropical Pacific Ocean using an indirect minimization approach. *J Atmos Oceanic Tech* 17(4):512–524
33. Masters D, Luschi C (2018) Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612*

34. Meijers A, Bindoff N, Rintoul S (2011) Estimating the four-dimensional structure of the Southern Ocean using satellite altimetry. *J Atmos Oceanic Tech* 28(4):548–568
35. Meinen CS, Watts DR (2000) Vertical structure and transport on a transect across the North Atlantic Current near 42°N: Time series and mean. *J Geophys Res: Oceans* 105(C9):21869–21891
36. Meyssignac B, Boyer T, Zhao Z, Hakuba MZ, Landerer FW, Stammer D, Köhl A, Kato S, L'ecuyer T, Ablain M, et al (2019) Measuring global ocean heat content to estimate the Earth Energy Imbalance. *Front Marine Sci* 6:432
37. Nardelli BB, Santoleri R (2005) Methods for the reconstruction of vertical profiles from surface data: Multivariate analyses, residual GEM, and variable temporal signals in the North Pacific Ocean. *J Atmos Oceanic Tech* 22(11):1762–1781
38. Reichstein M, Camps-Valls G, Stevens B, Jung M, Denzler J, Carvalhais N et al (2019) Deep learning and process understanding for data-driven Earth system science. *Nature* 566(7743):195–204
39. Roemmich D, Gilson J (2009) The 2004–2008 mean and annual cycle of temperature, salinity, and steric height in the global ocean from the Argo program. *Prog Oceanogr* 82(2):81–100
40. Stumpf A, Kerle N (2011) Object-oriented mapping of landslides using Random Forests. *Remote Sens Environ* 115(10):2564–2577
41. Su H, Wu X, Yan XH, Kidwell A (2015) Estimation of subsurface temperature anomaly in the Indian Ocean during recent global surface warming hiatus from satellite measurements: A support vector machine approach. *Remote Sens Environ* 160:63–71
42. Su H, Wu X, Lu W, Zhang W, Yan XH (2017) Inconsistent subsurface and deeper ocean warming signals during recent global warming and hiatus. *J Geophys Res: Oceans* 122(10):8182–8195
43. Su H, Li W, Yan XH (2018) Retrieving temperature anomaly in the global subsurface and deeper ocean from satellite observations. *J Geophys Res: Oceans* 123(1):399–410
44. Su H, Yang X, Lu W, Yan XH (2019) Estimating subsurface thermohaline structure of the global ocean using surface remote sensing observations. *Remote Sensing* 11(13):1598
45. Su H, Zhang H, Geng X, Qin T, Lu W, Yan XH (2020) OPEN: A new estimation of global ocean heat content for upper 2000 meters from remote sensing data. *Remote Sensing* 12(14):2294
46. Su H, Zhang T, Lin M, Lu W, Yan XH (2021) Predicting subsurface thermohaline structure from remote sensing data based on long short-term memory neural networks. *Remote Sens Environ* 260:112465
47. Takano A, Yamazaki H, Nagai T, Honda O (2009) A method to estimate three-dimensional thermal structure from satellite altimetry data. *J Atmos Oceanic Tech* 26(12):2655–2664
48. Wang J, Flierl GR, LaCasce JH, McClean JL, Mahadevan A (2013) Reconstructing the ocean's interior from surface data. *J Phys Oceanogr* 43(8):1611–1626
49. Xia Y, Liu C, Li Y, Liu N (2017) A boosted decision tree approach using bayesian hyperparameter optimization for credit scoring. *Expert Syst Appl* 78:225–241
50. Yan H, Wang H, Zhang R, Chen J, Bao S, Wang G (2020) A dynamical-statistical approach to retrieve the ocean interior structure from surface data: SQG-mEOF-R. *J Geophys Res: Oceans* 125(2):e2019JC015840
51. Yan XH, Boyer T, Trenberth K, Karl TR, Xie SP, Nieves V, Tung KK, Roemmich D (2016) The global warming hiatus: Slowdown or redistribution? *Earth's Future* 4(11):472–482
52. Yu X, Hyypä J, Vastaranta M, Holopainen M, Viitala R (2011) Predicting individual tree attributes from airborne laser point clouds based on the random forests technique. *ISPRS J Photogramm Remote Sens* 66(1):28–37
53. Zhang Y, Zhang H, Lin H (2014) Improving the impervious surface estimation with combined use of optical and SAR remote sensing images. *Remote Sens Environ* 141:155–167

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits any noncommercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if you modified the licensed material. You do not have permission under this license to share adapted material derived from this chapter or parts of it.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Ocean Heat Content Retrieval from Remote Sensing Data Based on Machine Learning



Wenfang Lu and Hua Su

1 Introduction

In recent decades, the imbalance in the top-of-atmosphere radiation, termed the Earth's energy imbalance (EEI) [49], has been continuously promoted changes in the global climate system, leading to continued global warming. The EEI must be accurately quantified in order to investigate and comprehend the past, present, and future state of climate change [38], which is defined by the net heat gaining in the Earth's climate system calculating the difference between the energy entering into and reflected by the Earth [50]. Due to its small magnitude compared with solar radiation, the EEI is difficult to quantify accurately [38]. Yet, more than 93% of the EEI of the Earth system is sequenced in the ocean as ocean heat content (OHC) changes [5, 7]. Naturally, this is due to the large heat capacity and gigantic volume of seawater, which accounts for ~71% of the world's surface area and ~97% of total water volume. Therefore, the OHC variability is slower and can better capture low-frequency climate variability. These make OHC the most suitable variable to detect and track EEI changes than sea surface temperature (SST) [26, 50].

OHC is driven by both human activity and natural variability. The anthropogenic forcing has been reflected in the OHC, leading to speeding OHC warming rate [35], and therefore the former serves as an essential indicator of ocean variability. In turn, OHC also feedbacks to the climate change [10, 25]. On the multi-decadal timescale natural variability, the global OHC is of high relevance to the Earth's heat balance [3]. OHC is also closely associated with the El Niño Southern Oscillation (ENSO),

W. Lu (✉)

School of Marine Sciences, Sun Yat-Sen University, and Southern Marine Science and Engineering Guangdong Laboratory (Zhuhai), Zhuhai 519000, China
e-mail: luwf6@sysu.edu.cn

H. Su

Key Laboratory of Spatial Data Mining and Information Sharing of Ministry of Education, The Academy of Digital China, Fuzhou University, Fuzhou 350108, China

© The Author(s) 2023

X. Li and F. Wang (eds.), *Artificial Intelligence Oceanography*,
https://doi.org/10.1007/978-981-19-6375-9_6

125

dominating the interannual variability. In recent years, hemispheric asymmetry of OHC changes has emerged, which can be likely explained by internal dynamics instead of different surface forcing [39]. The world ocean of 2021 was the hottest ever recorded by human beings despite the La Niña conditions [17]. In summary, the accurate quantification of the OHC is crucial to understanding EEI [10, 50].

Remote sensing can provide wide and near-real-time coverage as well as a vast collection of spatial and temporal information. However, in most circumstances, only water surface can be seen from the remote sensing which cannot penetrate to the ocean's interior. Recently, a series of subsurface and deeper ocean remote sensing (DORS) methods were developed to unlock the enormous potential of remote sensing data in sensing the ocean interior [31]. Particularly, Artificial Intelligent (AI) methods have provided cutting-edge tools and infrastructures [41].

Different remote sensing data were applied to derive subsurface thermal information via various methods [18, 31, 32, 52]. These early initiatives demonstrated the concept of DORS to tackle the issue of data sparsity. Recent research has shown that surface remote sensing data may be effectively used to retrieve STA via machine-learning or AI approaches. For instance, [27] and [23] proved that subsurface structures were dominated by the first baroclinic mode, and thus can be estimated from SSH. By merging remote sensing data with a Self-Organization Map (SOM) approach, [51] further confirmed the theory's credibility in the Northern Atlantic Ocean. Reference [36] used a clustered shallow neural network (NN) to obtain subsurface temperature, demonstrating the promise of NNs as a category of generic techniques with powerful regression capabilities. Other relevant contributions have been made [2, 8, 20, 21, 23], to mention a few.

Among the several methodologies, neural networks (NN), as the foundation of contemporary deep learning breakthroughs, have demonstrated the capability in the regression problems of the ocean subsurface estimation [2, 27–29, 48]. Yet, the application of NN models to temporally extrapolate remote sensing data was limited. This is partly because of the difficulty in time series estimation, and the fact that STA was indirectly influenced by the surface signals in the deep ocean. The exact physical controls are fundamentally nonlinear. In this regard, OHC is more tightly coupled with the surface forcing [40], which may lead to a more physically consistent DORS application. So far, only a few studies have used surface data to retrieve OHC [27, 54]. In the ground-breaking work of [27], an NN was trained to derive discrete site-wise Indian Ocean OHC. Given that different ocean basins have different OHC dynamics and thus linkages to the surface, the first goal of this study is to determine whether this approach can be extended to the entire global ocean. We will answer this question by developing an NN model driven by big data to estimate OHC, that is accurate for the global ocean and for temporally extending OHC data to the pre-Argo era of 1993 onward to 2004.

The method will also be used to generate an OHC product using this NN approach, hindcasting the OHC before the Argo era. Since the early 2000s, when Argo floats have been continuously deployed, the ability to accurately quantify OHC has unprecedentedly increased [44]. To present, a network of over 4000 Argo floats has been detecting robust climate signals in the global ocean's large-scale dynamic

features. However, prior to the Argo era, there was no reliable full-coverage ocean interior data. As a result, there are discussions and debates regarding various climate issues. Consider the trend of heat redistribution during the “hiatus” period between 1998 and the late 2010s, when global warming appeared to be slowing [53]. Various climate signals have been detected, each backed by different data products [45], which puts a strain on the quality of ocean interior data in order to give comprehensive and effective support for climate research throughout this time period [53]. For example, there are two broad opinions on driving processes: one is the Atlantic meridional overturning circulation-controlled mechanism [11], and the other is Indo-Pacific-originated mechanisms [33].

In this chapter, we describe the NN model yielding an NN-based global OHC product named Ocean Projection and Extension neural Network (OPEN) [47]. The technical details will be described with a focus on the NN approach. This chapter is structured as follows. After presenting data in Sect. 2, the NN method is detailed in Sect. 3. We also present the design of experiments to optimize the network. In Sect. 4, we first test the sensitivity of the network parameters and structure. The OHC is then reconstructed, extended to the pre-Argo era, from 1993 to 2020. In addition, OPEN and other renowned near-global OHC products are evaluated in terms of linear trends and variability modes. Finally, in Sect. 5, we summarize the results and provide prospects for future studies.

2 Data

A summary of all data sets utilized in this chapter is shown in Table 1, including an Argo-based three-dimensional temperature product to derive OHC, multi-source satellite remote sensing data, and OHC products from different sources.

The sea surface height (SSH) is from the Absolute Dynamic Topography products of Archiving, Validation, and Interpretation of Satellite Oceanographic (AVISO). The SST is from the Optimum Interpolation Sea Surface Temperature (OISST). The sea surface wind (SSW) is from the Cross Calibrated Multi-platform (CCMP). These three products have a common spatial resolution of one quarter. The sea surface salinity (SSS) is adopted from the Soil Moisture Ocean Salinity (SMOS) product. The SMOS product has a spatial resolution of one degree. We linearly interpolated all the products to a one-degree grid except for SSS.

The OHC ‘ground truth’ was derived from Roemmich and Gilson [44] gridded Argo product, which consists of 27 standard levels of 0–2000 m. The variables include pressure, temperature, and salinity. Dynamic heights were also provided from the T/S profiles. It has a monthly time interval from 2005 to the present, and a spatial resolution is $1^\circ \times 1^\circ$. By definition, the OHC can be calculated by conducting depth integral of temperature T from the surface to a particular level z .

$$\text{OHC} = \rho C_p \int_0^z T dz \quad (1)$$

Table 1 Data sets applied in this chapter

Data abbreviation	Full name	Time span	Source	Purpose	Reference
Argo OHC	Ocean Heat Content (OHC) generated from gridded Argo 3D temperature	2005–2018	http://apdrc.soest.hawaii.edu/projects/Argo/data/gridded/On_standard_levels/index-1.html	For training and testing the NN model	Roemmich and Gilson [44]
AVISO SSH	Archiving, Validation, and Interpretation of Satellite Oceanographic	1993–present	https://resources.marine.copernicus.eu/?option=com_csw&view=details&product_id=SEALEVEL_GLO_PHY_L4_REP_OBSERVATIONS_008_047		/
OISST	Optimum Interpolation Sea Surface Temperature	1983–present	https://www.nci.noaa.gov/products/optimum-interpolation-ssst	For training the NN model	Reynolds et al. [43]
CCMP	Cross Calibrated Multi-platform	1983–present	https://rda.ucar.edu/datasets/ds745.1		Atlas et al. [4]
SMOS	Soil Moisture and Ocean Salinity	2010–present	https://www.catds.fr/Products/Available-products-from-CEC-OS/CEC-Lops-SSS-SMOS-SMAP-OI-L4		
IAP	Institute of Atmospheric Physics	1940–present	http://159.226.119.60/cheng		Cheng et al. [13]
NCEI	National Centers for Environmental Information	1955–present	https://www.nodc.noaa.gov/OC5/3M_HEAT_CONTENT		Levitus et al. [35]
EN4	Version 4 of the Met Office Hadley Centre “EN” series of data sets	1900–present	https://www.metoffice.gov.uk/hadobs/en4/download-en4-2-1.html	For comparison with OPEN OHC dataset	Good et al. [22]
GLORYS2V4	Global Ocean Reanalysis and Simulation	1993–2019	https://resources.marine.copernicus.eu/?option=com_csw&view=details&product_id=GLOBAL_REANALYSIS_PHY_001_026		/
ARMOR3D	/	1993–present	https://resources.marine.copernicus.eu/?option=com_csw&view=details&product_id=MULTIOBS_GLO_PHY_REP_015_002		Guinehut et al. [23]

In the integration, ρ is the seawater density and C_p is the heat capacity. Constant values of $1025 \text{ kg} \cdot \text{m}^{-3}$ and $3850 \text{ J} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}$ were applied. OHC300, OHC700, OHC1500, and OHC2000 refers to the OHC of top 300, 700, 1500, and 2000 m, respectively, where integration is done. The reference of OHC, with regard to the climatological mean of 2005–2015, was then computed and removed to provide OHC anomalies. Hereafter, we report OHC anomaly unless otherwise indicated.

Other near-global OHC products will be compared with the OPEN product. These data sets are: National Centers for Environmental Information (NCEI) data by [35], Institute of Atmospheric Physics (IAP) data by [13], EN4 from the Met Office of United Kingdom by [22], empirical DORS-based ARMOR3D data by [23], and numerical reanalysis GLORYS2V4. Among these data products, NCEI, IAP, and EN4 are all optimal interpolated (mapped) one-degree products from a common collection of discrete station and profiling data. The source of the in-situ data includes Argo profilers, conductivity-temperature-depth (CTD), and expendable bathythermograph (XBT). The ARMOR3D and GLORY2V4 both have a 0.25-degree resolution. Because we only use basin OHC summations, the different resolution is not an issue.

3 Method

3.1 Neural Network

The NN with a total of o layers (\mathbf{h} as hidden layers) applied in this chapter can be generally formularized as:

$$\text{Neurons in input layer : } \mathbf{h}_1 = f_1(\mathbf{x}; \theta_1) = \sigma_1 \left(\mathbf{b}_1 + \sum_i^{\text{features}} \mathbf{w}_1 \mathbf{x}_i \right) \quad (2)$$

$$\text{Neurons in hidden layer(s) : } \mathbf{h}_2 = f_2(\mathbf{h}_1; \theta_2) = \sigma_2 \left(\mathbf{b}_2 + \sum_j^{\text{neurons}} \mathbf{w}_2 \mathbf{h}_{1,j} \right) \quad (3)$$

...

$$\text{Neuron in output layer : } \hat{y} = f_o(\mathbf{h}_{o-1}; \theta_o) \quad (4)$$

For a regression problem, one may express mathematically an NN as an approximation function $\hat{y} = f(\mathbf{x}; \theta)$ from the inputs \mathbf{x} to the OHC \hat{y} with parameters θ . θ include weights \mathbf{w} , biases \mathbf{b} , and activation functions σ for each neuron in the hidden layer.

Generally, in a network, one input layer, one or more hidden layers, and one output layer are essential. Layers inter-connects each other in a manner of stacks. The input layer collects input features and hence has the same number of neurons. Each neuron

in each layer computes the weighted average from its previous layer's outputs. The neuron then computes the nonlinear outputs with its activation function. And the next layer receives the output results from its previous layer. The number of neurons is often described as the *width*, with the number of hidden layers as the *depth*, i.e., a *deep* NN has more hidden layers.

Such an NN is essentially an optimization problem to find the parameters θ leading to the minimized cost function J , which is the mean squared error. This can be formularized as:

$$\arg_{\theta} \min J(\theta) = \sum_{(x,y) \in Tr} \frac{(y - \hat{y})^2}{N} \quad (5)$$

In the equation, Tr refers to the training set with to N samples in a training set. We applied a Bayesian regularization for the NN, following our previous study of [36]. Our experience suggests that by smoothing the cost function J , the Bayesian regularization approach can efficiently avoid overfitting [19]. This trait is advantageous for temporal projection because smoothness is more likely to work effectively when fresh data are provided. An ensemble technique was applied. Six subsets of training periods were defined, that starts from 2005, 2006, 2007, 2008, 2009, and 2010, and ends in 2013, 2014, 2015, 2016, 2017, and 2018, respectively. Except for the training period, all remaining data were utilized as the testing set. The uncertainty may be evaluated using three times standard deviations, which are distributed across six ensemble members. For each depth, a distinct NN was trained. Once the remote sensing data are provided, the OHC field can be derived. The ensemble average will be reported as our hindcast of OHC in the following chapters.

3.2 Design of Experiments

The NN relies heavily on the proper combination of sea surface variables. In AI field, these variables are referred to as input *features*. One may certainly train NN for any unrelated input-output data, yet this often results in overfitted NN. It is envisaged that an NN model can successfully extrapolate to unknown data provided there is a clear input-output relationship that the NN can learn. Furthermore, in the practice of optimization, choosing the greatest feature combination might be paradoxical at times [46]. As a result, features are frequently chosen haphazardly in a process known as feature engineering. The availability of historical data also influences feature selection. In the current study, to find the best combination of the features, we designed 16 experiments as shown in Table 2. For these experiments, we chose the OHC300 in January 2011 as the target to be hindcasted with the tuned NN and a data subset from the 12 months of 2010 as the training data. Note that the conclusions here are insensitive to the data subsetting. Case A and Case R are included in each case. Case R uses remote sensing SSH, in addition to the surface SST and SSS, while Case A employs those from the (surface) Argo data of the

uppermost level. Both experiment series shared the same SSW data set. This is to test if Argo ‘surface’ data can be transited to remote sensing OHC estimation. In addition, we designed several experiments to evaluate the role of temporal and spatial information, involving day of year (DOY), longitude (LON), and latitude (LAT). At last, the normalized root-mean-square error (NRMSE) and determination coefficient (R^2) were used to measure the network performance. NRMSE is the ratio of root-mean-square error to corresponding standard deviation.

4 Results and Analysis

4.1 Optimization of Feature Combinations

Table 2 shows the R^2 and NRMSE values. Overall, the SSH anomaly is the leading factor affecting OHC, followed by the SST anomaly. This can be seen from the retrieval accuracy for Case 1, which has already very high retrieval accuracy considering these two features (Table 2). Cases 1A and Case 1R show that the accuracy is fairly good, with the retrieved OHC explaining 70% of the variance. The retrieval accuracy with satellite data is slightly higher. This could be because Argo’s SSH is actually the dynamic height integrated from temperature and salinity [44] while the contribution from volume changes was missing. Consistent with our previous work [36], it is clear that including spatiotemporal information, i.e., LON, LAT, and DOY, enhances the training for both data sets substantially when comparing Case 1 and Case 2. We suspect that including DOY improves the NN because it allowed it to learn the seasonal cycle, which is the most dominating signal in OHC. By comparing Cases 2 and Case 4, or Cases 1 and Case 3, we can find that SSW only improves the accuracy by $\sim 1\%$. When comparing Cases 1 and Case 5, SSS increases the retrieval in Case R, but has a suppressing effect of 6% of Argo data. This is because Argo SSS differs to a large extent from the remote sensing SSS. When utilizing the measured SSS to train and the remotely sensed SSS to predict, this discrepancy resulted in considerably lower accuracy. On the other hand, Case R is generally better than Cases A. This is not surprising, given that SSH primarily represents the inner dynamics of the first baroclinic mode [9, 40], as well as the mismatch between dynamic height and satellite-based absolute dynamic topography. Not surprisingly, directly using remote sensing data for the training is a better way. Case 8R had the highest accuracy successfully captured 80% variabilities; nevertheless, because SSS is only available for recent years, the feature combination in Case 4R is chosen as the optimized network features.

Table 2 Design of experiments and corresponding results for testing OHC300^a

Experiment ^b	Input Features	R^2 (Case A/Case R)	NRMSE (Case A/Case R)
Case 1A, Case 1R	<u>SSH</u> <u>SST</u>	0.69/0.71	0.39/0.38
Case 2A, Case 2R	<u>SSH</u> <u>SST</u> DOY LON LAT	0.79/0.80	0.36/0.34
Case 3A, Case 3R	<u>SSH</u> <u>SST</u> SSW	0.70/0.72	0.38/0.38
Case 4A, Case 4R	<u>SSH</u> <u>SST</u> SSW DOY LON LAT	0.79/0.80	0.36/0.34
Case 5A, Case 5R	<u>SSH</u> <u>SST</u> <u>SSS</u>	0.64/0.71	0.40/0.39
Case 6A, Case 6R	<u>SSH</u> <u>SST</u> <u>SSS</u> DOY LON LAT	0.67/0.79	0.49/0.35
Case 7A, Case 7R	<u>SSH</u> <u>SST</u> <u>SSS</u> SSW	0.64/0.72	0.41/0.38
Case 8A, Case 8R	<u>SSH</u> <u>SST</u> <u>SSS</u> SSW DOY LON LAT	0.71/0.81	0.44/0.33

^a These values were achieved after training with 2010 data (12 months), while the testing was performed with January 2011 data. Noting that all these features are anomalies

^b Case R indicates that SSH and SST are from remote sensing data. Case A indicates those are from the surface record of Argo. The underlines indicate variables are different in Case A and Case R. Notice that these differences are merely in the training; in the testing, all are from remote sensing products

4.2 Deep, or Shallow—That Is the Question

One comment perspective is a deep NN has a stronger capability to regress the complex hidden relationship between input and output features. Theoretically, the universal approximation theorem demonstrated that a one-hidden-layer NN with sufficient neurons can approximate any continuous function [24]. To confirm this concept in retrieving the OHC, the optimal hyperparameters will be discovered using a grid-search method. We designed several experiments in which the NN was deepened from two hidden layers to six. The performance of networks with different hyperparameters is examined using a subset of data (Fig. 1).

As Fig. 1 demonstrates, as the neuron number increases, the retrieving accuracy of two-/three-layer networks first improves, then declines. Generally, three-layer networks have steeper declines, i.e., being more prone to overfitting. This means that keeping a basic shallow network structure is better for the current issue. We also observe that increasing the complexity of NN reduces linear trends. The following part will deal with the global and basin-wide warming trends of OHC. These results agree with our previous application of NN for subsurface temperature estimation [36]. In summary, adding more hidden layers to a network can improve its capacity to fit a complicated input-to-output mapping function. It might, however, raise the probability of overfitting and make the training more difficult.

The choice of activation functions is also influencing (Fig. 1). For two-layer networks, the combination of ReLU and sigmoid functions is not as good as the sigmoid

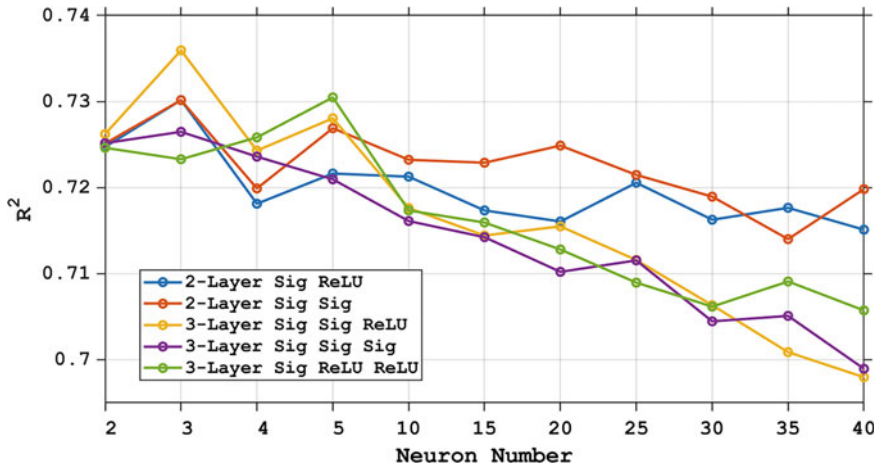


Fig. 1 Determination coefficients as a function of the number of neurons for different NN structures. Different activation function combinations and hidden-layer numbers are represented by different line colors. The training data is OHC300 of 2005–2013, while testing data is those of 2017. The configuration of Case 4 was adopted. Sig means the activation function of tangential sigmoid

function alone. The ReLU activation, on the other hand, outperforms all three-layer networks. Furthermore, the ReLU activation function is predicted to be more efficient than the sigmoid function in terms of computation, but this advantage is negligible for shallow networks. This result shows that for a shallow NN, the nonlinear sigmoid function is a better choice, emphasizing the above-mentioned universal approximation theorem [24]. The optimum NN design was determined by these studies to be a three-layer NN with three neurons and a combined sigmoid with ReLU activation. This architecture will be used to report findings by default in the following text.

4.3 Data Reconstruction

We used the ensemble approach to train the model using data from 2005 to 2018. We further hindcasted the data from 1993, the earliest year with global altimetry coverage, to the year 2020. OPEN OHC data are compared to six datasets, i.e., NCEI, EN4, IAP, ARMOR3D, and GLORYS2V4, with an emphasis on interannual variabilities and decadal trends. These data sets are summarized in Table 1.

Figure 2 presents OHC300, OHC700, and OHC2000 from OPEN and Argo for January 2011 (as Table 2). The NRMSE values were 0.36, 0.34, and 0.37 for the three depth integrals, while the retrieval R^2 is 0.80, 0.82, and 0.80, respectively. Across different depths, the accuracy changes are small, suggesting the robustness of NN networks. The spatial distribution hindcasted by OPEN closely agrees with the Argo OHC. The spatial distribution of the OHC is dominated by the ENSO

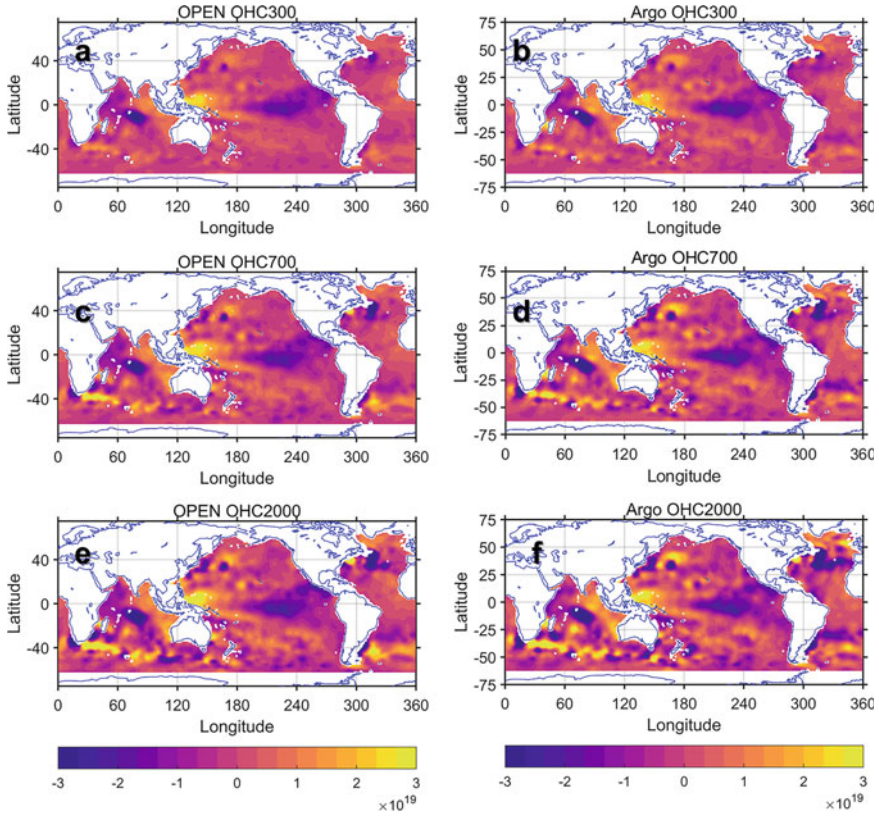


Fig. 2 The ocean heat content (OHC) in Joule for OPEN-hindcast and Argo data in January 2011 for (top) 0–300 m; (middle) 0–700 m; and (bottom) 0–2000 m. The OHC reference is from 2005 to 2015

fluctuation. In the tropical Indo-Pacific waters, the OHC has high values for all depths; in the eastern tropical Pacific, the OHC is lower. This pattern suggests a La Niña state, consistent with a multivariate Niño index of -1.83 . In the southern hemisphere, the meandering of the Agulhas retroflexion is discernible, showing alternating warming and cooling patterns [6]. For different depths, the OHC patterns are consistent; however, the magnitude is different, which gradually increases with depth. The most major difference is between OHC300 and OHC2000. Significant OHC changes can be found in the Pacific and Indian seas for OHC300, while changes can be found in all basins for OHC2000.

For hindcasting 1993–2004, by using IAP data as the true value, Fig. 3a shows the pattern correlation of the hindcasted OPEN OHC, while the temporal correlation and errors are displayed in Fig. 3b–d. The total R^2 was higher than 0.98, with an NRMSE of 12%. The R^2 and NRMSE interannual fluctuations are quite minimal, indicating a consistent performance. For long-term temporal extrapolation, this is preferable. In

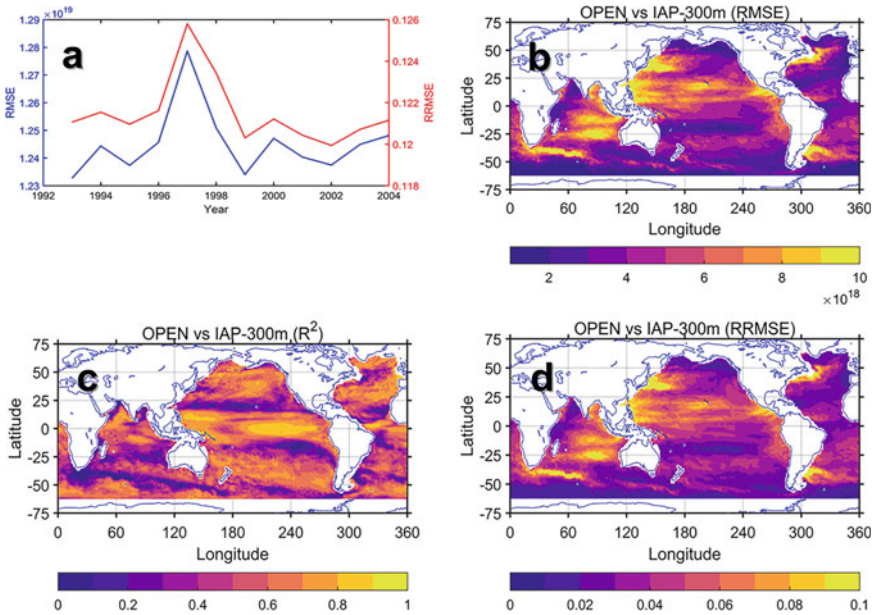


Fig. 3 Comparing OPEN OHC300 with the IAP data set. **a** Pattern RMSE and NRMSE showing as time series. Spatial maps of temporal **b** RMSE, **c** determination correlation, and **d** NRMSE

1997, there is an extremely high error, which is most likely due to the strong ENSO signature of this year. ENSO might disturb the ocean surface, causing the network to deviate from its learned association.

At the extension of the western boundary current system, OPEN’s error is relatively higher, as well as in the two zonal bands cross the subtropical Pacific Ocean in the northern and southern hemispheres at $\sim 25^\circ$, and in the Agulhas retroflexion region. All these systems have nonlinear circulation and complex dynamics. In other regions, OPEN has a high correlation and low RMSE in terms of site-wise OHC time series but also presents heterogeneous structures. Overall, the hindcast of OHC300 in the global ocean presents $\sim 10\%$ error with respect to the spatiotemporal standard deviation of OHC300.

Table 3 summarizes the statistic matrix between OPEN OHC and other products. The R^2 are all greater than 0.988 (OHC300: 0.993; OHC700: 0.988; OHC1500: 0.988; OHC2000: 0.989) when compared to the Argo OHC over the training period (2005–2018), whereas the NRMSE values are all less than 11% (OHC300: 0.09, OHC700: 0.109, OHC1500: 0.111, and OHC2000: 0.106). The best agreement between OPEN and EN4 products can be found, while differs from IAP (Table 3). In summary, OPEN OHC can be reliably reconstructed to the pre-Argo period since the overall accuracy is high.

We further compare the global OHC300 from all the products shown in Fig. 4, and the corresponding linear trends for two distinct time periods of 1993–2010 and

Table 3 Accuracy of OPEN compared with other data sets

	Matrix	Argo	IAP	EN4	ARMOR3D	GLORYS2V4
OHC300	R^2	0.993	0.984	0.990	0.991	0.988
	RMSE ($\times 10^{19}$ J)	0.841	1.237	0.964	0.907	1.062
	NRMSE (%)	8.6	12.0	8.2	8.5	10.1
OHC700	R^2	0.988	0.971	0.986	0.987	0.982
	RMSE ($\times 10^{19}$ J)	1.642	2.542	1.767	1.713	2.009
	NRMSE (%)	10.9	15.8	9.2	9.6	11.9
OHC1500	R^2	0.988	0.958	0.985	0.984	0.975
	RMSE ($\times 10^{19}$ J)	2.256	4.111	2.472	2.563	3.142
	NRMSE (%)	11.1	19.1	9.2	10.8	13.5
OHC2000	R^2	0.989	0.953	0.985	0.982	0.972
	RMSE ($\times 10^{19}$ J)	2.333	4.755	2.643	2.957	3.645
	NRMSE (%)	10.6	20.3	8.9	11.2	14.2

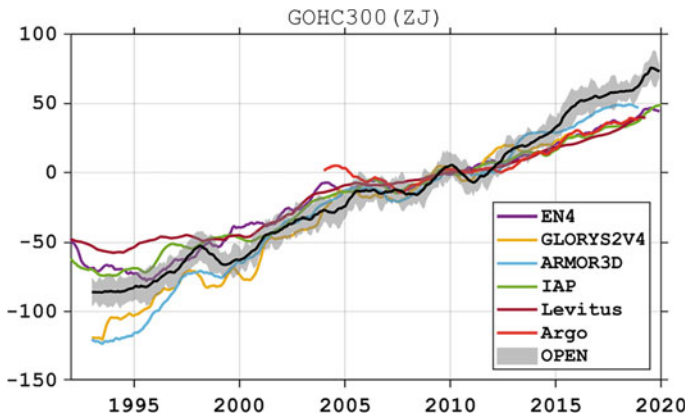


Fig. 4 12-month moving averaged global OHC300 (unit: ZJ, i.e., $\times 10^{21}$ J) referred to the 2005–2014 period. The *thick black line with gray envelopes* is the ensemble average and three standard deviations for six OPEN ensemble members

Table 4 Warming rates^a of the global ocean OHC at different depths in $\times 10^{22}$ J/decade

Depth (m)	EN4	GLORYS2V4	ARMOR3D	IAP	NCEI	OPEN
	OHC trends					
0–300	4.86/3.93 ^b	6.95/6.12	7.45/5.68	4.62/4.18	3.78/3.83	3.71/3.95
0–700	7.86/7.11	12.41/9.88	13.15/8.71	6.93/6.68	5.81/5.70	7.92/8.16
0–1500	10.46/10.31	15.55/13.55	18.31/12.92	9.19/9.07	– ^c	9.78/10.63
0–2000	10.98/11.20	16.05/14.61	18.83/13.41	9.79/9.77	–	10.10/11.21

^a These trends were computed from the 12-month moving mean of the global integration of each product

^b The two values are for 1993–2010 and 1998–2015 periods

^c NCEI (Levitus) data only available for upper 700 m

1998–2015. In the second period, the surface warming hiatus occurred. From the trends, it is reflected the global ocean’s ongoing warming. For instance, in 2018 and 2019, record high were reached in the OHC [15, 16]. Interannual variabilities such as the ENSO fingerprinted the OHC, showing an abrupt high in 1997 and 1998. This signal is less visible for deeper OHC (0–700, 0–1500, and 0–2000 m), but more so for the upper OHC300. Because the OHC300 is more sensitive to the surface thermal forcings, the ENSO signature is more prominent. In Table 4, it is noticeable that OPEN has a higher OHC warming trend than IAP, while that of EN4 is very close to the latter. Since the two data sets were both from mapping techniques from a similar database of in-situ observation, this similarity is not surprising, especially for OHC300 and less so for the deeper OHC. In these depths, two statistic-based products (GLORYS2V4 and ARMOR3D) present even larger inconsistency and stronger trends. Summarizing across all the products in Fig. 4, our NN-based OPEN product agrees well with other products, falling within the range of all data sets. Similarly for OPEN and ARMOR3D, the OHC300 presents a high bias after the year 2015, which is likely due to the same source of remote sensing data as the major inputs of estimation. Further improvements can be achieved by using more sophisticated AI approaches, which are ongoing efforts to predict OHC by the use of time sequence learning and spatial autoencoding-decoding structures. Given the large uncertainties among various estimations and the core role of OHC in understanding ocean warming and heat transfers in the Earth system, the importance of accurate quantifying the OHC is further emphasized.

For different ocean basins, the OHC variabilities and trends are shown in Fig. 5 and Table 5. The representative pattern of linear trends of IAP and OPEN is shown in Fig. 6. For all the major basins of oceans, there are consistent warming trends exist, which reflects the overall ocean warming by the anthropogenic forcing. Consistent with the findings of [33], during the two time periods, steadily highest OHC increase can be found in the Indo-Pacific basins and the warm pool area. This highlighted the Indo-Pacific role driving the recent global warming hiatus. In contrast, the Southern Ocean illustrated the lowest warming rates; this estimation was accompanied by large uncertainties, which can be attributed to the low Argo coverage therein. From 1993 to 2010, IAP and OPEN both present a basin-wide dipole warming and cooling pattern

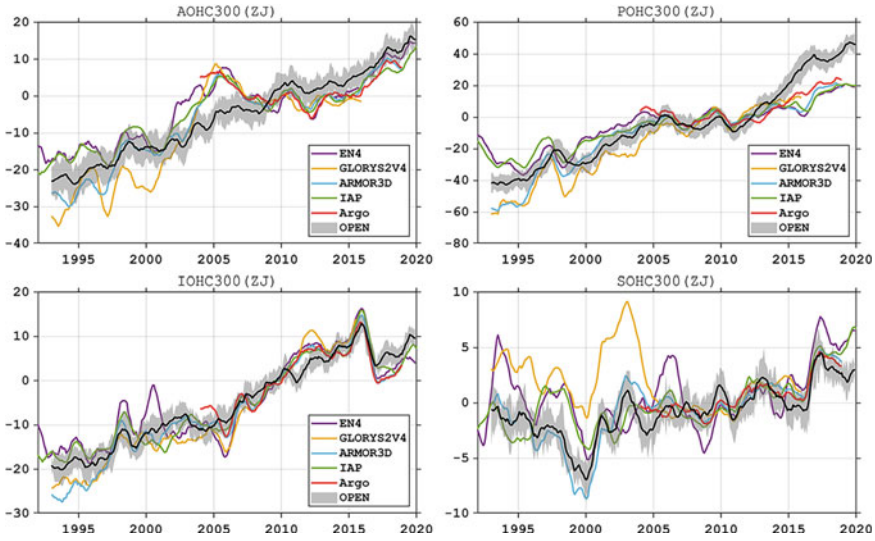


Fig. 5 12-month moving averaged OHC300 (unit: ZJ, i.e., $\times 10^{21}$ J) for four major ocean basins. Because Argo data has a shorter temporal coverage so the reference here is 2005–2014. The *thick black line with gray envelopes* is the ensemble average and three standard deviations for six OPEN ensemble members

Table 5 The OHC linear trends^a for different ocean basins (unit: $\times 10^{22}$ J/decade)

Basin	EN4	GLORYS2V4	ARMOR3D	IAP	OPEN
	OHC trends				
Atlantic ocean	1.27/0.62 ^b	2.24/1.25	1.93/0.88	1.27/0.66	1.47/1.28
Pacific ocean	2.29/1.33	3.52/3.12	3.49/1.93	1.88/1.57	2.57/2.45
Indian ocean	0.80/1.36	1.26/1.75	1.43/1.50	0.95/1.48	1.17/1.36
Southern ocean	-0.15/0.12	-0.25/-0.15	0.16/0.36	0.01/0.20	0.14/0.30

^a These trends were computed from the 12-month moving mean of the regional integration of each product

^b The two values are for 1993–2010 and 1998–2015 periods

in the Pacific Ocean, with positive trends in the western part and negative trends in the east (Fig. 6). The structure mimics the Pacific Decadal Oscillation negative pattern, which was supported by the transition from positive to negative phase reported in literature [37]. For the later period, both the IAP and OPEN show bulk warming Indian Ocean, but less homogeneous for other basins. These consistencies prove the capability of OPEN data to reflect OHC trends in both Argo and pre-Argo eras.

We now focus on the inconsistency. Compared with the majority of data sets, OPEN differs to the largest degree for the Pacific. Because the ENSO’s signature is exaggerated in the Pacific and lower in the other oceans. For OPEN, a significant jump after 2015 can be unexpectedly seen for the Pacific OHC, which is not found

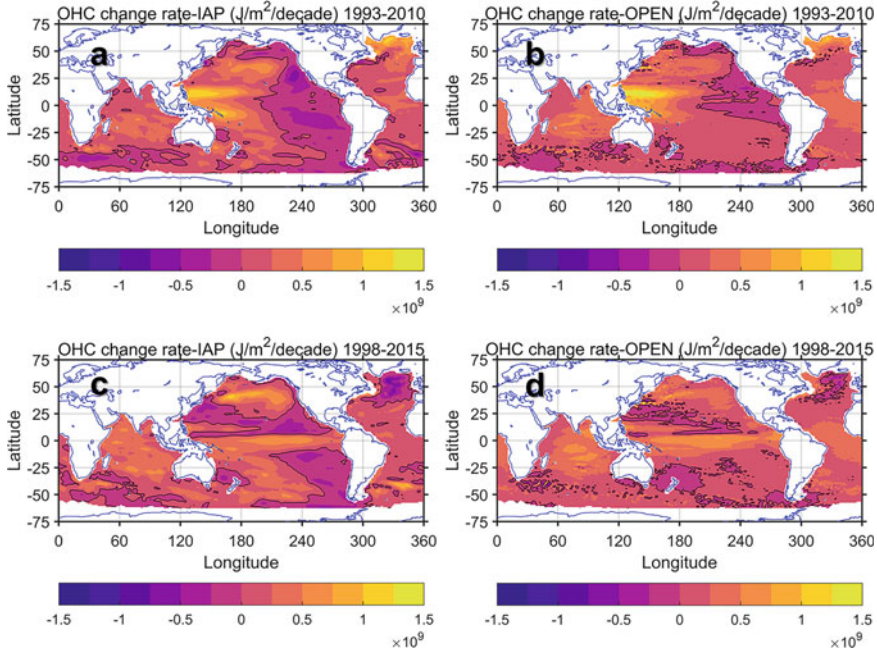


Fig. 6 Linear trends for (left) IAP and (right) OPEN OHC300. The upper row shows those of 1993–2010 and the lower row is for 1998 to 2015. Zeros are depicted with black lines

in the global OHC (Fig. 4). Further speculation shows that the different references contributed to approximately half of this jump, i.e., OPEN has a lower reference compared to ARMOR3D and GLORYS2V4. On the other hand, OPEN OHC300 presents a minimal envelope of uncertainty in the Pacific basin (Fig. 5), which implies that the jump is not due to random errors but contains mostly systematic biases. To reduce the error, one option is to utilize a more sophisticated deepened NN, such as the deep convolutional NN that will be discussed in the following chapters of this book, to extract the complicated link between surface variables and OHC. Alternatively, one can also adopt the strategy to use the clustering technique to subset the global ocean into distinct thermal provinces, each can be represented by a simple but different surface-subsurface relationship and thus better estimated by NN. This strategy has been shown viable in our previous effort [36]. These will be tested in future research.

We notice that IAP has a reference that is about 10% higher than that of Argo and OPEN. To further examine this mismatch, we show the non-anomaly OHC300 from these three products, which is one particular snapshot as an example (Fig. 7). The IAP product is significantly larger, i.e., warm bias, than OPEN and Argo, which is especially noticeable in the Pacific Ocean’s subtropical gyres, despite the fact that the three products show relatively similar patterns. This is very likely due to the errors of the XBT correction scheme of IAP. Compared with more accurate CTD measurements, the XBT measurements have a well-documented warm bias, despite

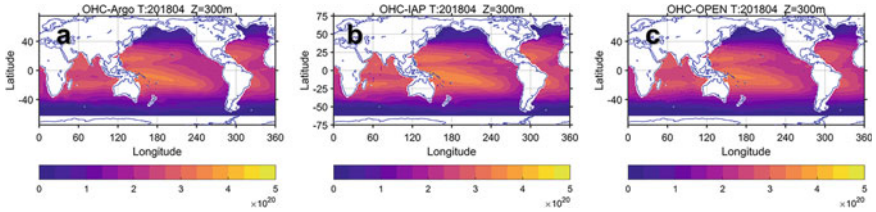


Fig. 7 Non-anomaly OHC300 (non-anomaly, unit: J) for **a** Argo, **b** IAP, and **c** OPEN showing a particular snapshot at April average of 2018

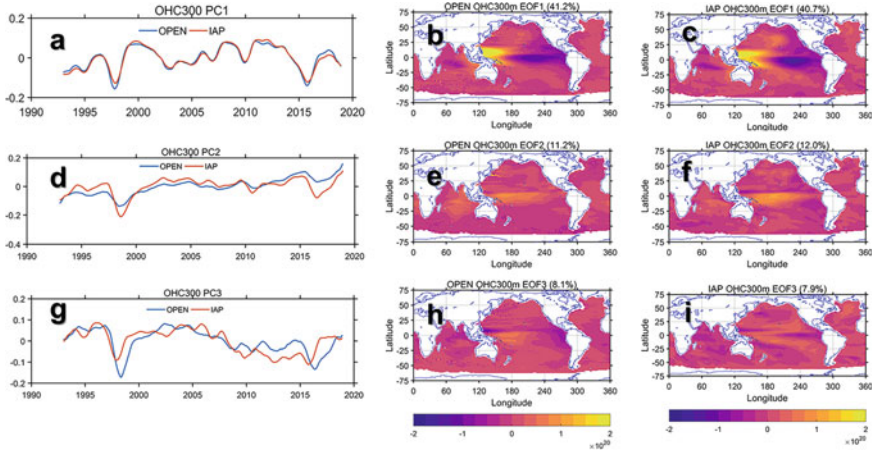


Fig. 8 The PCs (*left*), EOFs of OPEN (*middle*), and EOF of IAP (*right*) from EOF analysis of OHC300, with the corresponding percentage of explanation shown in the title. The explained percentage of each mode is listed above the corresponding EOFs. Before EOF analysis, linear trends and high-frequency signals (higher than 12 months) were removed

the advanced time-varying correction recently developed [12, 34], which accounts for $\sim 0.5^\circ\text{C}$ warm bias. Yet, noting that this error is systematic, there are limited signatures in the decadal trends, but this highlights the value of having additional independent OHC datasets.

To further demonstrate the OHC300's spatiotemporal variability, we apply the empirical orthogonal function (EOF) analysis to OPEN and IAP (Fig. 8). The EOF analysis was conducted after linear trends and seasonal variation (by a 12-month lowpass filter) were removed. Those spatial EOFs and temporal PCs show high agreements between OPEN and IAP data sets (Fig. 8). For instance, the first mode accounts for 41.2% variability in IAP and 40.7% in OPEN, which is very close to each other. Only some small differences can be found in the corresponding PC1 (Fig. 8a). The EOF both have a tropical Pacific dipole higher in the warm pool and lower in the western part. In the extratropic, the difference is still small. Other modes present some visibly larger differences, but considering the smaller percentage ($< 12\%$) of

these modes, the contribution to the OHC difference is small. The same conclusions will be drawn by analyzing OPEN and other products, further demonstrating the validity of OPEN.

5 Summary and Conclusions

In this chapter, we describe the AI technique of DORS and its application for studying climate change. An NN approach was developed to estimate OHC from remote sensing data sets, yielding a new ocean heat content estimation, which was termed Ocean Projection and Extension neural Network (OPEN) product [47]. By using the $1^\circ \times 1^\circ$ gridded Argo OHC data as the true values, and taking advantage of remote sensing products of SSH, SST, and SSW with near-global coverage and higher spatiotemporal resolution, we trained four NNs, each for estimating the OHC from the surface to 300, 700, 1500, and 2000 m depth. The NNs were trained with the 2005–2018 data in a way that enables the temporal extrapolation of OHC. By testing a variety of architecture of NN and feature combinations, the NN was optimized. Generally, a simple shallow NN was favorable for temporal extrapolation. The final choice for NN architecture had three hidden layers, each with three neurons. In this way, the four-depth OPEN OHC product was extended to the 1993 period covering the pre-Argo era, with a very high accuracy of $R^2 > 0.95$ and NRMSE $< 20\%$. We also estimated the uncertainty of OHC by using an ensemble technique, which demonstrated that OPEN also had low uncertainties from the NN technique. Comparisons of OPEN against other widely applied OHC data sets showed the good performance of OPEN in terms of trends and variabilities.

Various contributions have emphasized the need for more trustworthy OHC products for the sake of understanding the Earth's climate, e.g., [42, 54]. As we mentioned before, all estimations are subjected to different sources of uncertainties. In-situ mapping-based products (IAP, EN4, and NCEI) have inconsistent observation records and uncertainties in mapping schemes. Numerical models (GLORYS2V4) may incorporate imperfect representations of physics. Despite the favorable performance of the NN-based OPEN product, it has limitations. It was trained from gridded Argo data. Although the gridded Argo product is often treated as observation, it is subjected to its own mapping and instrumental errors. For instance, [30] has found a larger errors of such product in western boundary current systems, where nonlinear dynamics are characterized. The unevenly distributed Argo profiles also contributed to the spatial errors.

For the oceanography community, one haunting skepticism to AI technique is that: what can these techniques do to solve real-world oceanography problems? To date, many applications of AI oceanography are still very preliminary, '*in their infancy*' [55], far away from product-level outcomes. This chapter shows a promising application of AI techniques in climatic and ocean sciences, in addition to the currently available DORS studies. Presumably, the application of OPEN will also serve as a base for future AI studies. Several future directions of AI application concern-

ing OHC and its climatic effects are: (1) extending the global OHC product to a longer time span, favorably covering several quick-warming and particularly surface warming ‘hiatus’ periods to understand the phenomenology [53]; (2) generating a downscale OHC product with higher spatial/temporal resolution; (3) developing AI method that digging into multiple datasets and digesting physics laws; and (4) projecting future OHC. These are all playgrounds where the AI Oceanography approach can unleash its potential.

References

1. Akbari E, Alavipanah S, Jeihouni M, Hajeb M, Haase D, Alavipanah S (2017) A review of ocean/sea subsurface water temperature studies from remote sensing and non-remote sensing methods. *Water* 9(12):936. <https://doi.org/10.3390/w9120936>
2. Ali MM, Swain D, Weller RA (2004) Estimation of ocean subsurface thermal structure from surface parameters: A neural network approach. *Geophys Res Lett* 31(20). <https://doi.org/10.1029/2004gl021192>, <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2004GL021192>
3. Antonov JI (2005) Thermosteric sea level rise, 1955–2003. *Geophys Res Lett* 32(12). <https://doi.org/10.1029/2005gl023112>
4. Atlas R, Hoffman RN, Ardizzone J, Leidner SM, Jusem JC, Smith DK, Gombos D (2011) A cross-calibrated, multiplatform ocean surface wind velocity product for meteorological and oceanographic applications. *Bull Am Meteor Soc* 92(2):157–174. <https://doi.org/10.1175/2010BAMS2946.1>
5. Balmaseda MA, Trenberth KE, Källén E (2013) Distinctive climate signals in reanalysis of global ocean heat content. *Geophys Res Lett* 40(9):1754–1759. <https://doi.org/10.1002/grl.50382>
6. Beal LM, De Ruijter WP, Biastoch A, Zahn R, Group SWIW (2011) On the role of the Agulhas system in ocean circulation and climate. *Nature* 472(7344):429–36. <https://doi.org/10.1038/nature09983> www.ncbi.nlm.nih.gov/pubmed/21525925
7. Bindoff NL, Willebrand J, Artale V, Cazenave A, Gregory JM, Gulev S, Hanawa K, Le Quere C, Levitus S, Nohji Y, et al. (2007) Observations: oceanic climate change and sea level
8. Charantonis AA, Badran F, Thiria S (2015) Retrieving the evolution of vertical profiles of Chlorophyll-a from satellite observations using Hidden Markov Models and Self-Organizing Topological Maps. *Remote Sens Environ* 163:229–239. <https://doi.org/10.1016/j.rse.2015.03.019>
9. Chelton DB, deSzoek RA, Schlax MG, El Naggar K, Siwertz N (1998) Geographical variability of the first baroclinic rossby radius of deformation. *J Phys Oceanogr* 28(3):433–460. [https://doi.org/10.1175/1520-0485\(1998\)028<0433:GVOTFB>2.0.CO;2](https://doi.org/10.1175/1520-0485(1998)028<0433:GVOTFB>2.0.CO;2)
10. Chen J, Shum C, Wilson C, Chambers D, Tapley B (2000) Seasonal sea level change from TOPEX/Poseidon observation and thermal contribution. *J Geodesy* 73(12):638–647
11. Chen XY, Tung KK (2014) Varying planetary heat sink led to global-warming slowdown and acceleration. *Science* 345(6199):897–903. <https://doi.org/10.1126/science.1254937>. <Go to ISI>://WOS:000340524700037
12. Cheng L, Abraham J, Goni G, Boyer T, Wijffels S, Cowley R, Gouretski V, Reseghetti F, Kizu S, Dong S, Bringas F, Goes M, Houpert L, Sprintall J, Zhu J (2016) XBT science: Assessment of instrumental biases and errors. *Bull Am Meteor Soc* 97(6):924–933. <https://doi.org/10.1175/bams-d-15-00031.1>
13. Cheng L, Trenberth KE, Fasullo J, Boyer T, Abraham J, Zhu J (2017) Improved estimates of ocean heat content from 1960 to 2015. *Sci Adv* 3

14. Cheng L, Abraham J, Zhu J, Trenberth KE, Fasullo J, Boyer T, Locarnini R, Zhang B, Yu F, Wan L (2020a) Record-setting ocean warmth continued in 2019. *Adv Atmos Sci* 37:137–142
15. Cheng LJ, Abraham J, Hausfather Z, Trenberth KE (2019) How fast are the oceans warming? *Science* 363(6423):128–129. <https://doi.org/10.1126/science.aav7619>. <Go to ISI>://WOS:000455320600030
16. Cheng LJ, Abraham J, Zhu J, Trenberth KE, Fasullo J, Boyer T, Locarnini R, Zhang B, Yu FJ, Wan LY, Chen XR, Song XZ, Liu YL, Mann ME (2020) Record-setting ocean warmth continued in 2019. *Adv Atmos Sci* 37(2):137–142. <https://doi.org/10.1007/s00376-020-9283-7> <Go to ISI>://WOS:000519079600002
17. Cheng L, Abraham J, Trenberth KE, Fasullo J, Boyer T, Mann ME, Zhu J, Wang F, Locarnini R, Li Y, Zhang B, Tan Z, Yu F, Wan L, Chen X, Song X, Liu Y, Reseghetti F, Simoncelli S, Gouretski V, Chen G, Mishonov A, Reagan J (2022) Another record: ocean warming continues through 2021 despite La Niña conditions. *Adv Atmos Sci*. <https://doi.org/10.1007/s00376-022-1461-3>
18. Chu PC, Fan C, Liu WT (2000) Determination of vertical thermal structure from sea surface temperature. *J Atmos Oceanic Tech* 17(7):971–979
19. Foresee FD, Hagan MT (1997) Gauss-Newton approximation to Bayesian learning. In: Proceedings of international conference on neural networks (ICNN'97), IEEE, vol 3, pp 1930–1935
20. Friedrich T, Oschlies A (2009) Neural network-based estimates of North Atlantic surface pco2 from satellite data: A methodological study. *J Geophys Res: Oceans* 114(C3). <https://doi.org/10.1029/2007jc004646>. <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2007JCO04646>
21. Garcia-Goriz E, Garcia-Sanchez J (2007) Prediction of sea surface temperatures in the western Mediterranean Sea by neural networks using satellite observations. *Geophys Res Lett* 34(11). <https://doi.org/10.1029/2007gl029888>. <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2007GL029888>
22. Good SA, Martin MJ, Rayner NA (2013) EN4: Quality controlled ocean temperature and salinity profiles and monthly objective analyses with uncertainty estimates. *J Geophys Res: Oceans* 118(12):6704–6716. <https://doi.org/10.1002/2013JC009067>
23. Guinehut S, Dhomps AL, Larnicol G, Le Traon PY (2012) High resolution 3-d temperature and salinity fields derived from in situ and satellite observations. *Ocean Sci* 8(5):845–857. <https://doi.org/10.5194/os-8-845-2012> www.ocean-sci.net/8/845/2012/
24. Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. *Neural Netw* 2(5):359–366
25. Hoskins BJ, McIntyre M, Robertson AW (1985) On the use and significance of isentropic potential vorticity maps. *Q J R Meteorol Soc* 111(470):877–946
26. IPCC (2014) Climate Change 2014: Impacts, Adaptation, and Vulnerability. Part A: Global and Sectoral Aspects. Contribution of Working Group II to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA
27. Jagadeesh PSV, Kumar MS, Ali MM (2015) Estimation of heat content and mean temperature of different ocean layers. *IEEE J Sel Topics Appl Earth Obs Remote Sensing* 8(3):1251–1255. <https://doi.org/10.1109/JSTARS.2015.2403877>
28. Jain S, Ali MM (2006) Estimation of sound speed profiles using artificial neural networks. *IEEE Geosci Remote Sens Lett* 3(4):467–470. <https://doi.org/10.1109/LGRS.2006.876221>
29. Jain S, Ali MM, Sen PN (2007) Estimation of sonic layer depth from surface parameters. *Geophys Res Lett* 34(17). <https://doi.org/10.1029/2007gl030577>. <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2007GL030577>
30. Jeong Y, Hwang J, Park J, Jang CJ, Jo YH (2019) Reconstructed 3-D ocean temperature derived from remotely sensed sea surface measurements for mixed layer depth analysis. *Remote Sensing* 11(24):3018
31. Klemas V, Yan XH (2014) Subsurface and deeper ocean remote sensing from satellites: an overview and new results. *Prog Oceanogr* 122:1–9. <https://doi.org/10.1016/j.pocean.2013.11.010>. www.sciencedirect.com/science/article/pii/S0079661113002310

32. Klemas V, Yan XH (2014) Subsurface and deeper ocean remote sensing from satellites: An overview and new results. *Prog Oceanogr* 122:1–9. <https://doi.org/10.1016/j.pocean.2013.11.010>. <Go to ISI>://WOS:000334006100001
33. Lee SK, Park W, Baringer MO, Gordon AL, Huber B, Liu Y (2015) Pacific origin of the abrupt increase in Indian Ocean heat content during the warming hiatus. *Nature Geosci* 8(6):445–449. <https://doi.org/10.1038/ngeo2438>. <http://dx.doi.org/10.1038/ngeo2438www.nature.com/ngeo/journal/v8/n6/pdf/ngeo2438.pdf>
34. Levitus S, Antonov JI, Boyer TP, Locarnini RA, Garcia HE, Mishonov AV (2009) Global ocean heat content 1955–2008 in light of recently revealed instrumentation problems. *Geophys Res Lett* 36(7). <https://doi.org/10.1029/2008gl037155>
35. Levitus S, Antonov JI, Boyer TP, Baranova OK, Garcia HE, Locarnini RA, Mishonov AV, Reagan JR, Seidov D, Yarosh ES, Zweng MM (2012) World ocean heat content and thermosteric sea level change (0–2000 m), 1955–2010. *Geophys Res Lett* 39(10):n/a–n/a. <https://doi.org/10.1029/2012gl051106>
36. Lu W, Su H, Yang X, Yan XH (2019) Subsurface temperature estimation from remote sensing data using a clustering-neural network method. *Remote Sens Environ* 229:213–222. <https://doi.org/10.1016/j.rse.2019.04.009> www.sciencedirect.com/science/article/pii/S0034425719301464
37. Mantua NJ, Hare SR, Zhang Y, Wallace JM, Francis RC (1997) A pacific interdecadal climate oscillation with impacts on salmon production. *Bull Am Meteor Soc* 78(6):1069–1080
38. Meyssignac B, Boyer T, Zhao Z, Hakuba MZ, Landerer FW, Stammer D, Köhl A, Kato S, L'Ecuyer T, Ablain M, Abraham JP, Blazquez A, Cazenave A, Church JA, Cowley R, Cheng L, Domingues CM, Giglio D, Gouretski V, Ishii M, Johnson GC, Killick RE, Legler D, Llovel W, Lyman J, Palmer MD, Piotrowicz S, Purkey SG, Roemmich D, Roca R, Savita A, Kv Schuckmann, Speich S, Stephens G, Wang G, Wijffels SE, Zilberman N (2019) Measuring global ocean heat content to estimate the Earth Energy Imbalance. *Front Marine Sci* 6. <https://doi.org/10.3389/fmars.2019.00432>
39. Rathore S, Bindoff NL, Phillips HE, Feng M (2020) Recent hemispheric asymmetry in global ocean warming induced by climate change and internal variability. *Nat Commun* 11(1):2008. <https://doi.org/10.1038/s41467-020-15754-3> www.ncbi.nlm.nih.gov/pubmed/32332758
40. Rebert JP, Donguy JR, Eldin G, Wyrski K (1985) Relations between sea level, thermocline depth, heat content, and dynamic height in the tropical pacific ocean. *J Geophys Res* 90(C6). <https://doi.org/10.1029/JC090iC06p11719>
41. Reichstein M, Camps-Valls G, Stevens B, Jung M, Denzler J, Carvalhais N, Prabhat, (2019) Deep learning and process understanding for data-driven earth system science. *Nature* 566(7743):195–204. <https://doi.org/10.1038/s41586-019-0912-1>. www.ncbi.nlm.nih.gov/pubmed/30760912
42. Resplandy L, Keeling RF, Eddebar Y, Brooks M, Wang R, Bopp L, Long MC, Dunne JP, Koeve W, Oschlies A (2019) Quantification of ocean heat uptake from changes in atmospheric O₂ and CO₂ composition. *Sci Rep* 9(1):20244. <https://doi.org/10.1038/s41598-019-56490-z>
43. Reynolds RW, Smith TM, Liu C, Chelton DB, Casey KS, Schlax MG (2007) Daily high-resolution-blended analyses for sea surface temperature. *J Clim* 20(22):5473–5496. <https://doi.org/10.1175/2007jcli1824.1>
44. Roemmich D, Gilson J (2009) The 2004–2008 mean and annual cycle of temperature, salinity, and steric height in the global ocean from the Argo program. *Prog Oceanogr* 82(2):81–100. <https://doi.org/10.1016/j.pocean.2009.03.004>
45. Su H, Wu XB, Lu WF, Zhang WW, Yan XH (2017) Inconsistent subsurface and deeper ocean warming signals during recent global warming and hiatus. *J Geophys Res-Oceans* 122(10):8182–8195. <https://doi.org/10.1002/2016jc012481>. <Go to ISI>://WOS:000415893300022
46. Su H, Yang X, Lu W, Yan XH (2019) Estimating subsurface thermohaline structure of the global ocean using surface remote sensing observations. *Remote Sensing* 11(13). <https://doi.org/10.3390/rs11131598>

47. Su H, Zhang H, Geng X, Qin T, Lu W, Yan XH (2020) OPEN: A new estimation of global ocean heat content for upper 2000 meters from remote sensing data. *Remote Sensing* 12(14):2294. <https://www.mdpi.com/2072-4292/12/14/2294>
48. Swain D, Ali MM, Weller RA (2006) Estimation of mixed-layer depth from surface parameters. *J Mar Res* 64(5):745–758. <https://doi.org/10.1357/002224006779367285> www.ingentaconnect.com/content/jmr/jmr/2006/00000064/00000005/art00005doi.org/10.1357/002224006779367285
49. Trenberth KE, Fasullo JT, Balmaseda MA (2014) Earth's energy imbalance. *J Clim* 27(9):3129–3144
50. von Schuckmann K, Palmer MD, Trenberth KE, Cazenave A, Chambers D, Champollion N, Hansen J, Josey SA, Loeb N, Mathieu PP, Meyssignac B, Wild M (2016) An imperative to monitor Earth's Energy Imbalance. *Nat Clim Chang* 6(2):138–144. <https://doi.org/10.1038/nclimate2876>
51. Wu X, Yan XH, Jo YH, Liu WT (2012) Estimation of subsurface temperature anomaly in the North Atlantic using a Self-Organizing Map neural network. *J Atmos Oceanic Tech* 29(11):1675–1688. <https://doi.org/10.1175/jtech-d-12-00013.1>
52. Yan XH, Schubel JR, Pritchard DW (1990) Ocean upper mixed layer depth determination by the use of satellite data. *Remote Sens Environ* 32(1):55–74. [https://doi.org/10.1016/0034-4257\(90\)90098-7](https://doi.org/10.1016/0034-4257(90)90098-7). <Go to ISI>://WOS:A1990DZ26200005
53. Yan XH, Boyer T, Trenberth K, Karl TR, Xie SP, Nieves V, Tung KK, Roemmich D (2016) The global warming hiatus: slowdown or redistribution? *Earth's Future* 4:472–482. <https://doi.org/10.1002/2016ef000417>
54. Zanna L, Khatiwala S, Gregory JM, Ison J, Heimbach P (2019) Global reconstruction of historical ocean heat storage and transport. *Proc Natl Acad Sci U S A* 116(4):1126–1131. <https://doi.org/10.1073/pnas.1808838115> www.ncbi.nlm.nih.gov/pubmed/30617081
55. Zheng G, Li X, Zhang RH, Liu B (2020) Purely satellite data-driven deep learning forecast of complicated tropical instability waves. *Sci Adv* 6

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits any noncommercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if you modified the licensed material. You do not have permission under this license to share adapted material derived from this chapter or parts of it.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Detecting Tropical Cyclogenesis Using Broad Learning System from Satellite Passive Microwave Observations



Sheng Wang and Xiaofeng Yang

1 Introduction

Tropical cyclone (TC), as one of the most violent phenomena of air-sea interaction, often brings disastrous storm surges and flooding and causes significant damage to human life, agriculture, forestry, fisheries, and infrastructure. Therefore, the knowledge of TC track, intensity, structure, and evolution is required to guide severe weather forecasting and risk assessment. Generally, the formation of TCs needs the support of dynamic environmental conditions and thermodynamically favorable environmental conditions [19]. Because only a small percentage of convective disturbances are developing into TCs, it is still challenging to predict the TC formation accurately.

Since the Dvorak Technique (DT) was proposed and developed [8, 21, 28], it has been widely used in TCs intensity estimation [14, 22, 25, 29] and TC formation prediction [6, 20, 32, 33]. However, DT is based on the infrared technique, whose observations may be obscured by significant convection or cirrus clouds. In contrast, microwave radiation images can capture the strong convective areas and cloud organization. Therefore, it is potential to predict the formation of TCs with microwave remote sensing data.

With the advancements in high-performance computing, machine learning methods based on big datasets are widely used in tropical cyclogenesis detection of TC

S. Wang · X. Yang

State Key Laboratory of Remote Sensing Science, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100101, China

S. Wang

State Key Laboratory on Internet of Things for Smart City and Department of Civil and Environmental Engineering, University of Macau, Macau 999078, China

X. Yang (✉)

Key Laboratory of Earth Observation of Hainan Province, Sanya 572029, China
e-mail: yangxf@radi.ac.cn

© The Author(s) 2023

X. Li and F. Wang (eds.), *Artificial Intelligence Oceanography*,
https://doi.org/10.1007/978-981-19-6375-9_7

formation. Based on the decision tree method, a series of classification rules are constructed to predict future tropical cyclone (TC) genesis events, and the overall prediction accuracy is 81.72% [35]. Using the dataset established with WindSat wind products, [23] established a classification model for tropical cyclogenesis detection. The validation shows that the model produced a positive detection rate of approximately 95.3% and a false alarm rate of 28.5%. This study confirmed the potential of microwave remote sensing observation in detecting typhoon formation [23]. Recently, based on the internal structure information of tropical cyclones obtained by satellite remote sensing, [27] effectively improved the prediction accuracy of the rapid enhancement process of tropical cyclones and reduced the false alarm rate using the machine learning method. Moreover, [13] compared different machine learning algorithms' TC formation detection performance. Their results prove that the machine learning method performs better than the traditional linear discriminant analysis.

However, with the continuous accumulation of remote sensing data, traditional machine learning methods cannot deal with massive data perfectly. Fortunately, powerful deep learning has demonstrated its more significant superiority over traditional physical or statistical-based algorithms for image information extraction [18]. In ocean remote sensing applications, the deep learning methods are used in hurricane intensity estimation [7, 24], sea ice concentration prediction [5, 9, 11], sea surface temperature estimation [1, 30] and other fields [10, 26, 37]. A deep learning approach has been proposed to identify tropical cyclones (TCs) and their precursors based on twenty-year simulated outgoing longwave radiation (OLR) calculated with a cloud-resolving global atmospheric simulation [19]. In the Northwest Pacific in the period from July to November, the probability of detection (POD) of the model is 79.9–89.1%, and the false alarm ratio (FAR) is 32.8–53.4%. In addition, this study reveals that the detection performance is correlated with the amount of training data and TC lifetimes.

Although deep learning is increasingly widely used in ocean remote sensing [18], the disadvantages of deep learning are also evident. It requires high computing power and a long training time. Moreover, most deep learning models do not have incremental learning capacity, which means the model needs to be retrained if updating the dataset. In ocean remote sensing, the satellite-based data increases every day, so the size of datasets is expected to expand further to improve the generalization ability and identification accuracy of models. Therefore, the defect of no incremental learning is not friendly to storage resources or model update times. Fortunately, the capacity of incremental learning of the Broad Learning System (BLS) [3] makes it have the potential to be applied in the field of ocean remote sensing. Meanwhile, the BLS is a time-cost-friendly learning strategy due to its flatted network. These advantages can compensate for the disadvantage of its accuracy compared with deep learning, so it has been widely used as soon as it is proposed. Recently, it has successfully been applied in seismic attenuation modeling [16], model updating [17], hyperspectral imagery classification [34], and crack detection [36].

In this chapter, we proposed a tropical cyclogenesis detection algorithm based on Special Sensor Microwave Imager (SSM/I) brightness temperature data. The proposed model based on BLS has three unique features: low hardware requirements,

fast computation speed, and incremental learning ability. In Sect. 2, the dataset used in this study is presented. In Sect. 3, the details of BLS are introduced. The experimental results are shown in Sect. 4, and the conclusion is given in Sect. 5.

2 Data Description

The dataset used in this chapter is extracted from the brightness temperature (TB) observations acquired by SSM/I. This series of instruments is carried onboard Defense Meteorological Satellite Program (DMSP) near-polar orbiting satellites. The SSM/I is a conically scanning sensor that measures the natural microwave emission from the Earth in the spectral band from 19 GHz to 85 GHz with different polarization (See Table 1). The parameters derived from these radiometer observations include surface wind speed, atmospheric water vapor, cloud liquid water, and rain rate [31]. Comparing the feature of TB images in different channels/polarizations, the 37 GHz H-polarization (37H) channel is selected due to its clear description of the features of disturbances and tropical cyclones.

To collect the sample images covered TCs or non-developed disturbances (non-TC), the TC best tracks and tropical cloud cluster (TCC) tracks during 2005–2009 are used as auxiliary data. This information can be obtained from the International Best Track Archive for Climate Stewardship dataset (IBTrACS) [15] and Global Tropical Cloud Cluster dataset [12], respectively. The time resolution of these two datasets is three hours. Note that not all the best track records in the TC evolution period are used, but those during the TC formation period are selected. Specifically, the time when the TC maximum wind speed reaches 25 knots for the first time is defined as the starting time. Then, the 72 h after this time is defined as the TC formation period [23]. For the TCC tracks, only the records that have not developed into TCs are selected. The preprocessing steps for extracting the TC and non-TC images are described as follow:

- (1) For each TC/non-TC track record, determine the matching SSM/I TB data within the absolute time difference of 1.5 h.
- (2) Take the track record as the image center position, and extract the sub-images with the size of $8^\circ \times 8^\circ$ from the SSM/I TB observations.
- (3) The sub-images with more than 60% non-empty pixels are retained as qualified samples (see Fig. 1); otherwise, the invalid data will be excluded (see Fig. 2).

Table 1 Channel characteristics of SSM/I

Band/GHz	Polarization	Spatial resolution/km × km
9.35	V/H	69 × 43
23.235	V	50 × 40
37.0	V/H	37 × 28
85.5	V/H	15 × 13

* H-horizontal polarization, V-vertical polarization

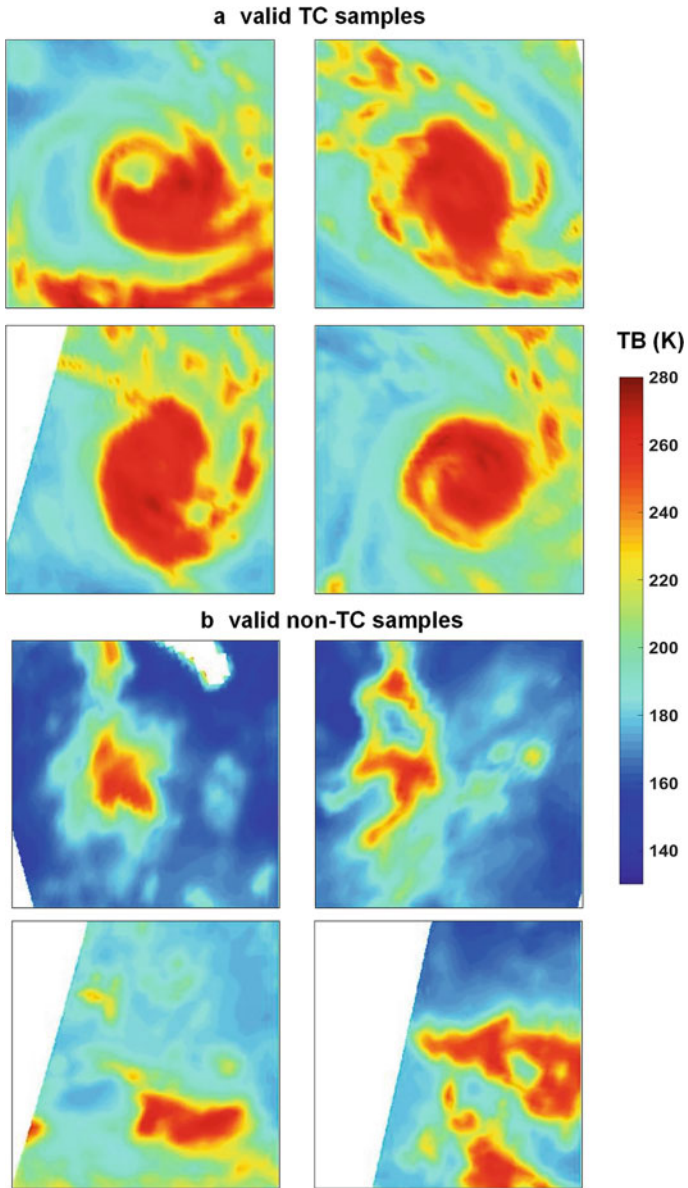


Fig. 1 Qualified samples: **a** valid TC samples and **b** valid non-TC samples

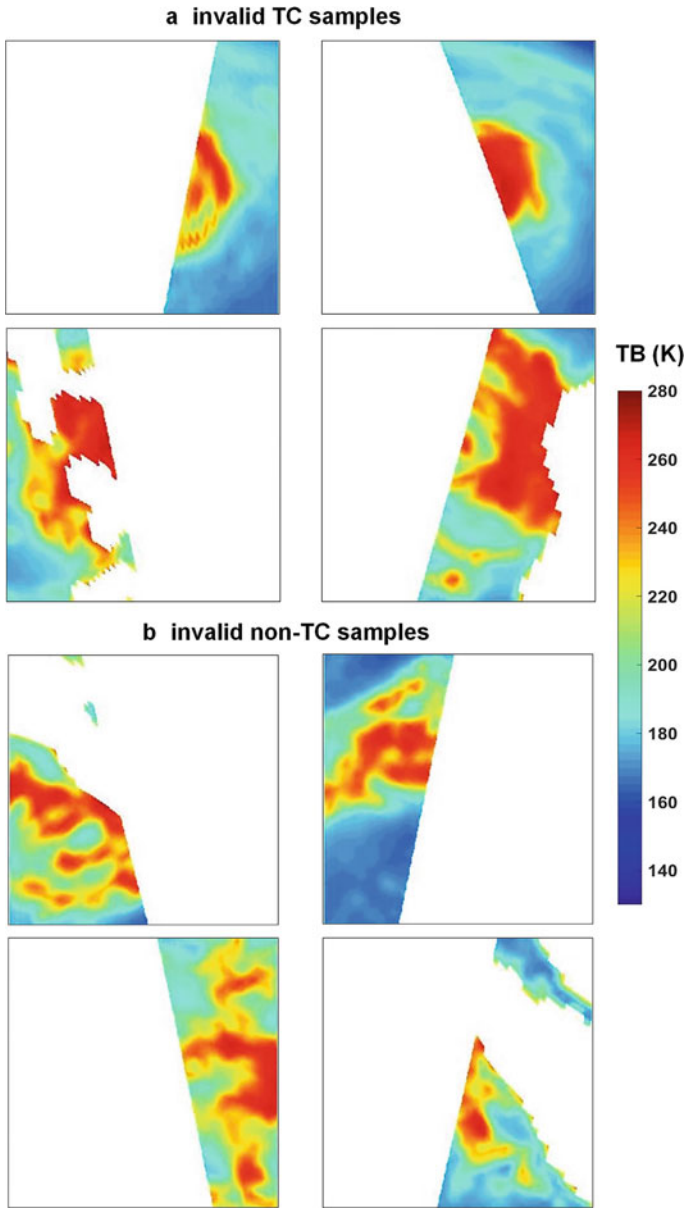


Fig. 2 Unqualified samples: **a** invalid TC samples and **b** invalid non-TC samples

Following the above steps, 880 TC samples and 6268 non-TC samples were obtained from the SSM/I observations in 2005–2009. Due to the significant number difference between the two samples, only 2506 non-TC samples in 2005–2006 and 880 TC samples in 2005–2009 were selected to form the final dataset. Each sample is in the size of 224×224 pixels with RGB channels. Finally, These two datasets are randomly divided into the training set and the testing set in the ratio of 4:1, respectively.

3 Broad Learning System for Tropical Cyclogenesis Detection

Once the dataset is established, the tropical cyclogenesis detection can be executed with the broad learning system. In contrast to deep learning methods, the BLS provides a time-cost-friendly learning strategy due to its flatted network. The main structure of BLS consists of the input layer, node layer, and output layer. Specifically, the node layer includes the feature nodes and enhancement nodes. Generally, the input data is mapped to feature nodes with random weights. Then, the feature nodes are further mapped to enhancements with new random weights. Finally, the final weights of BLS can be trained by estimating the output data with these feature nodes and enhancement nodes. Figure 3 shows the architecture of this study, the definition of variables are: \mathbf{X} is the input data, \mathbf{F} is the feature node, and \mathbf{E} is the enhancement node. \mathbf{Y} is the respective classification labels of the input data \mathbf{X} . The details of BLS are presented as follows.

3.1 Broad Learning Model

Assume that the input data is \mathbf{X} , so the feature vector \mathbf{F} mapped with random weight can be described as

$$F_i = \phi(XW_{e_i} + b_{e_i}), i = 1, \dots, n \quad (1)$$

where F_i is the i -th feature node, W_e and b_e are the random weights and biases with the proper dimensions, respectively. Denote $F^n \equiv [F_1, \dots, F_n]$, which is the concatenation of all the first n groups of mapped features. Then, the enhancement nodes can be given by:

$$E_m = \xi(F^n W_{h_m} + b_{h_m}) \quad (2)$$

where E_m is the m -th enhancement node, W_h and b_h are the random weights and biases with the proper dimensions, respectively. Similarly, the concatenation of all the first m groups of enhancement nodes are denoted as $E^m \equiv [E_1, \dots, E_m]$. Therefore, the broad model can be represented as the equation of the form

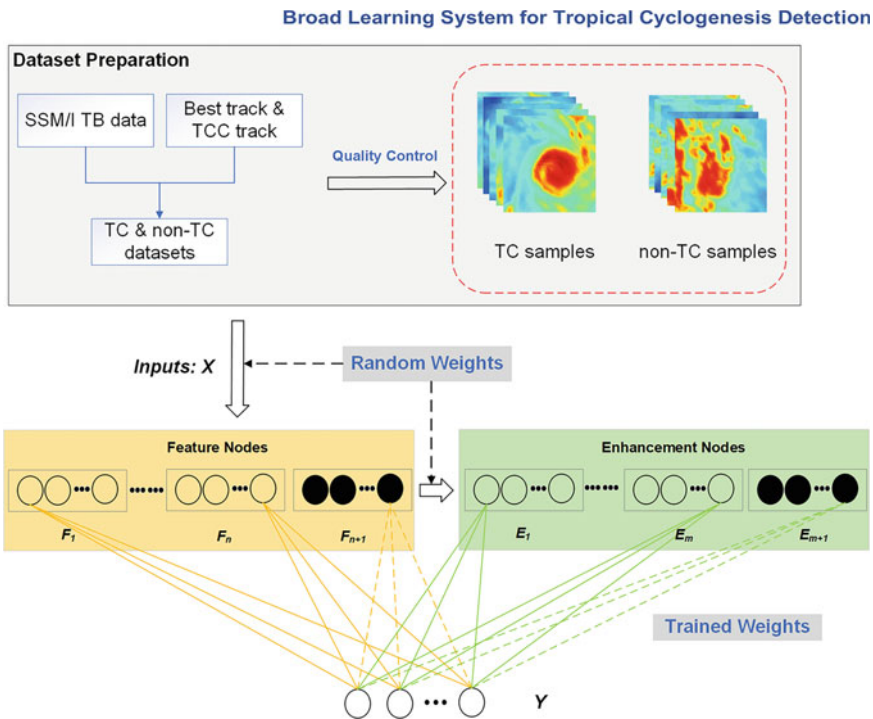


Fig. 3 The architecture of BLS

$$\begin{aligned}
 Y &= [F_1, \dots, F_n | \xi(F^n W_{h_1} + b_{h_1}), \dots, \xi(F^n W_{h_m} + b_{h_m})] W^m \\
 &= [F_1, \dots, F_n | E_1, \dots, E_m] W^m \\
 &= [F^n | E^m] W^m
 \end{aligned}
 \tag{3}$$

where $W^m = [F^n | E^m]^+ Y$ are the connecting weights for the broad structure to be computed and $[F^n | E^m]^+$ is the pseudo-inverse of $[F^n | E^m]$. In a flatted network, pseudo-inverse can be considered a very convenient approach to solving the output-layer weights of a neural network. However, a straightforward solution is too expensive, especially when the training samples and input patterns suffer from high volume, high velocity, and/or high variety [4]. Under this situation of the expensive cost for directly computing the pseudo-inverse, the solution can be approximated by ridge regression:

$$A^+ = [F^n | E^m]^+ = (\lambda I + [F^n | E^m][F^n | E^m]^T)^{-1} [F^n | E^m]^T
 \tag{4}$$

where λ is the regularization parameter. Finally, the model weights are given by

$$W = A^+ Y
 \tag{5}$$

During the BLS computation, it should be noted that the number of enhancement nodes is a hyperparameter (N_3), and the number of feature nodes is the combination of two hyperparameters: the number of feature windows (N_1) and the number of nodes in each feature window (N_2). Here, the Bayesian optimization method is used to find the optimal model hyperparameters, and it can be easily executed with the Hyperopt package [2].

3.2 Incremental Learning of BLS

When a deep learning model works not well, the number of convolutional kernels or the number of convolutional layers will increase. This will lead to expensive computation and long time costing. However, BLS usually uses incremental learning to address the low model accuracy caused by insufficient mapping nodes. Generally, the incremental learning part can improve the model performance. There are two ways to expand the broad structure: (1) increment of enhancement nodes and feature nodes, and (2) adding input data.

3.2.1 Increment of the Feature Nodes and Enhancement Nodes

Assume that the initial BLS has n feature nodes and m enhancement nodes that is $A = [F^n \mid E^m]$. In the adding process, the $(n+1)$ -th feature node is given by:

$$F_{n+1} = \phi(XW_{e_{n+1}} + b_{e_{n+1}}) \quad (6)$$

So that the corresponding enhancement node to this feature node is given by:

$$E_{ex_m} = [\xi(F_{n+1}W_{ex_1} + b_{ex_1}), \dots, \xi(F_{n+1}W_{ex_m} + b_{ex_m})] \quad (7)$$

Then, there are additional p enhancement nodes added to the BLS structure, and the $(m+1)$ -th enhancement node is given by

$$E_{m+1} = \xi(F^n W_{h_{m+1}} + b_{h_{m+1}}) \quad (8)$$

Therefore, the final node layer matrix is combined as

$$A' = [A \mid F_{n+1} \mid E_{ex_m} \mid E_{m+1}] \quad (9)$$

Then, the pseudo-inverse of A' is computed with

$$(A')^+ = \begin{bmatrix} A^+ - DB^T \\ B^T \end{bmatrix} \quad (10)$$

where $D = (A)^+ [F_{n+1} \mid E_{ex_m} \mid E_{m+1}]$,

$$B^T = \begin{cases} C^+ & C \neq 0 \\ (1 + D^T D)^{-1} B^T A^+ & C = 0 \end{cases} \quad (11)$$

and $C = [F_{n+1} \mid E_{ex_m} \mid E_{m+1}] - AD$. Finally, the new weights are

$$W' = \begin{bmatrix} W - DB^T Y \\ B^T Y \end{bmatrix} \quad (12)$$

As seen in Eq. 12, the updated weights consist of the initial and the new parts. There is no need to re-calculate the pseudo-inverse for the whole nodes but only compute the added nodes.

3.2.2 Increment of the Input Data

The increment of feature nodes and enhancement nodes mentioned above are for the fixed dataset. However, in most learning models, the input dataset is the core factor influencing the prediction accuracy. As for deep learning, once some new data is added to the training dataset, the existing model needs to be retrained. This is time-consuming and reduces the timeliness of model updating and application. Fortunately, there is no need to retrain the whole model for BLS after adding input data. The BLS will train only the added ones.

Denote X_a as the new inputs, the respective increment of mapped feature nodes and enhancement nodes are:

$$F_x^n = [\phi(X_a W_{e_1} + b_{e_1}), \dots, \phi(X_a W_{e_n} + b_{e_n})] \quad (13)$$

$$E_m^x = [\xi(F_x^n W_{h_1} + b_{h_1}), \dots, \xi(F_x^n W_{h_m} + b_{h_m})] \quad (14)$$

where the W_{e_i} , W_{h_i} and b_{e_i} , b_{h_i} are randomly generated during the initial BLS. Hence, the updating matrix is

$$A' = \begin{bmatrix} A \\ A_x^T \end{bmatrix} \quad (15)$$

where $A_x^T = [F_x^n \mid E_m^x]$. The associated updating pseudo-inverse could be deduced as follows:

$$W_n^x = W + (Y_a - A_x^T W) B \quad (16)$$

where Y_a are the respective labels of additional X_a . Similar to the two increment processes mentioned above, only the pseudo-inverse associated with new inputs is calculated. It greatly improves the update speed of the BLS model.

4 Results

Before training the model, one should determine the hyperparameters first. For the basic BLS, the hyperparameters include the number of feature windows (N_1), the number of nodes in each feature window (N_2), and the number of enhancement nodes (N_3). To compare the training time and model accuracy, the classical ResNet50 model is also applied to the same dataset. For this ResNet50 model, the initial learning rate is 10^{-3} , the multiplicative factor of learning rate decay is 0.5, the batch size is 16, and the number of epochs is 20. Before training the ResNet50 network, all input images were resized to 224×224 . To fairly compare the training time of these two networks, the training tasks were operated on a computer with an Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz and 64GB RAM.

4.1 Basic BLS Results

Using the Hyperopt, the hyperparameters are optimized as: $N_1 = 5$, $N_2 = 24$, $N_3 = 2332$. The training and testing accuracies and training time of these two methods are shown in Table 2. The testing accuracy of BLS is 86.83%, which is slightly lower than the 91.88% of ResNet50. On the other hand, though the ResNet50 is operated with an accelerating GPU, the training time of 2090.45 s is still 20 times than 60.52 s of BLS. Therefore, BLS has obvious advantages in computational efficiency, but it is not as accurate as the deep learning network because it is insensitive to the image features. Furthermore, we compared the hit rate (HR) and false alarm rate (FAR) of these two models. Table 3 lists the HR and FAR for the training and testing processes. The testing HR and FAR are 81.14 and 11.18%, respectively. Compared to the 79.9–89.1% of HR and 32.8–53.4% of FAR of existing deep learning research [19], our results are competitive. But it should be noted that the size of the dataset used in [19] is 50000 TCs and 500000 non-TCs, which is significantly larger than our dataset.

Table 2 The results for BLS and ResNet50

Model	Training accuracy (%)	Testing accuracy (%)	Training time
BLS	99.96	86.83	60.52 (CPU)
ResNet50	98.30	91.88	2090.45 (GPU)

Table 3 HR and FAR of BLS and results of [19]

Prediction accuracy (%)	BLS (%)	Matsuoka et al. [19] (%)
HR	81.14	79.9–89.1
FAR	11.18	32.8–53.4

Table 4 Prediction accuracy with different hyperparameters

Feature nodes (N ₁ , N ₂)	Enhancement nodes	Training time (s)	Testing accuracy (%)	HR (%)	FAR(%)
5, 24	2332	58.68	86.83	81.14	11.18
5, 24	1000	57.59	86.24	72.57	8.98
5, 24	1500	59.78	85.35	75.43	11.18
5, 24	2000	58.28	84.62	74.29	11.78
5, 24	2500	58.82	85.50	74.88	10.78
5, 24	3000	59.68	85.50	74.86	10.78
6, 24	3000	68.01	84.47	77.71	13.17
7, 24	3000	73.34	86.24	78.28	10.98
8, 24	3000	77.02	85.50	78.86	12.18

In contrast to the massive number of hyperparameters, the BLS only has primary hyperparameters to influence the prediction accuracy. To know the prediction performance of BLS influenced by different combinations of these three hyperparameters, we trained the same dataset several times, and the results are listed in Table 4. The combination of the number of feature nodes and enhancement nodes is essential for the accuracy of the model. It shows that the more nodes, the longer the training time, but the changing trend of model testing accuracy, HR, and FAR is inconsistent. However, the optimized hyperparameter combination has the highest HR because the optimization process takes the HR as the selection standard. Therefore, It is indispensable to determine these parameters using the optimization algorithm.

4.2 Incremental Learning Results

The capacity of incremental learning is the prominent superior feature of BLS to most traditional deep learning models. For most deep networks, the structures are fixed once the training process is finished. In contrast, the BLS can be updated by adding new nodes or updating the dataset. There is no need to retrain the whole network, which significantly saves the costing time of model updating. The dataset size is 3386, and we set the initial size as 1386 with the adding input patterns of 400. First, we trained the initial model with the initial samples. Then, the incremental learning method was operated to add corresponding input patterns each time until

Table 5 Prediction accuracy and CPU time of incremental learning

Number of inputs	Training accuracy (%)	Testing accuracy (%)	Training time (s)
707	100.00	26.33	31.53
708–1107	27.46	61.83	28.93
1108–1507	53.35	74.11	28.79
1508–1907	63.14	74.11	28.43
1908–2307	69.53	74.11	28.56
2308–2707	74.03	74.11	29.09

all training samples were input. During these incremental learning steps, the value of hyperparameters is still as $N_1 = 5$, $N_2 = 24$, $N_3 = 2332$. The results are shown in Table 5 and note that the testing dataset is unchanged during the incremental learning. The results in the table show that with the size of inputs increasing, the training accuracy grows. For the initial process, the small size of the dataset leads to unreasonable accuracies. Because the test dataset for each incremental learning is the same, the testing accuracy tends to be stable.

4.3 Case Study: Hurricane Wilma (2005)

Once the model training/testing processes are finished, the model hyperparameters (N_1 , N_2 , and N_3) and the weights of nodes (feature nodes and enhancement nodes) are fixed. Based on the trained BLS, the prediction task of specific TC cases can be executed. Here, we select Hurricane Wilma (2005) as the study case to validate the effectiveness of the proposed model. Wilma was an extremely intense hurricane over the northwestern Caribbean Sea. It had the all-time lowest central pressure for an Atlantic basin hurricane. According to the statistics, twenty-three deaths have been directly attributed to Wilma, and the total economic losses reached 16 billion to 20 billion dollars. As the best tracks in Table 6 shown, Wilma developed into a tropical storm (TS) from tropical depression (TD) at 06:00 UTC 17 October, and then strengthened into a hurricane (HU) at 12:00 UTC 18 October. We planned to collect the samples from 18:00 UTC 15 October to 15:00 UTC 18 October during the data preparation. However, there were only four qualified samples were retained (see Fig. 4). The corresponding best track records of Fig. 4a–d are 00:00 UTC 17 October, 12:00 UTC 17 October, 00:00 UTC 18 October, and 12:00 UTC 18 October, respectively.

Table 7 lists the prediction results of these four samples. It shows that Fig. 4b is incorrectly classified into a non-TC label, and the remaining three samples are correctly identified. Figure 4a is the first sample captured by SSM/I during Wilma's formation time, and its correct classification means the proposed model can detect tropical cyclogenesis as early as possible. Figure 4c, d are the observations during

Table 6 The best track information during the formation time of Wilma (2005)

Time (UTC)	Status	Longitude	Latitude	Wind speed
10-15 18:00	TD	78.50° W	17.60° N	25
10-15 21:00	TD	78.66° W	17.61° N	25
10-16 00:00	TD	78.80° W	17.60° N	25
10-16 03:00	TD	78.91° W	17.55° N	27
10-16 06:00	TD	79.00° W	17.50° N	30
10-16 09:00	TD	79.10° W	17.49° N	30
10-16 12:00	TD	79.20° W	17.50° N	30
10-16 15:00	TD	79.30° W	17.51° N	30
10-16 18:00	TD	79.40° W	17.50° N	30
10-16 21:00	TD	79.52° W	17.49° N	30
10-17 00:00	TD	79.60° W	17.40° N	30
10-17 03:00	TD	79.61° W	17.18° N	32
10-17 06:00	TS	79.60° W	16.90° N	35
10-17 09:00	TS	79.64° W	16.59° N	37
10-17 12:00	TS	79.70° W	16.30° N	40
10-17 15:00	TS	79.75° W	16.12° N	42
10-17 18:00	TS	79.80° W	16.00° N	45
10-17 21:00	TS	79.86° W	15.89° N	50
10-18 00:00	TS	79.90° W	15.80° N	55
10-18 03:00	TS	79.88° W	15.70° N	57
10-18 06:00	TS	79.90° W	15.70° N	60
10-18 09:00	TS	80.04° W	15.91° N	62
10-18 12:00	HU	80.30° W	16.20° N	65
10-18 15:00	HU	80.68° W	16.44° N	70

*TD-Tropical Depression, TS-Tropical Storm, HU-Hurricane

Wilma’s mature period, and its TC structure is relatively stable and complete. So it is predictable to obtain the correct results. However, the negative result of Fig. 4b proves that the quality of samples brings uncertainty and error to the model prediction. Specifically, compared with the other three samples, Fig. 4b loses nearly half of the TC system information, which reduces the number of effective pixels and damages spiral or TB distribution characteristics of TCs. This could be the main reason for the poor prediction.

However, not all samples with missing information will be incorrectly identified, but the core TC system information loss will lead to misclassification. To verify this conclusion, we select the tropical storm Hilda (2009) as the test case. Figure 5 shows Hilda’s four qualified TC samples, and Table 8 lists the corresponding prediction results. It shows that all the samples are correctly classified though parts of information lost in Fig. 5. In particular, the size of the missing part in Fig. 5a, b is similar to

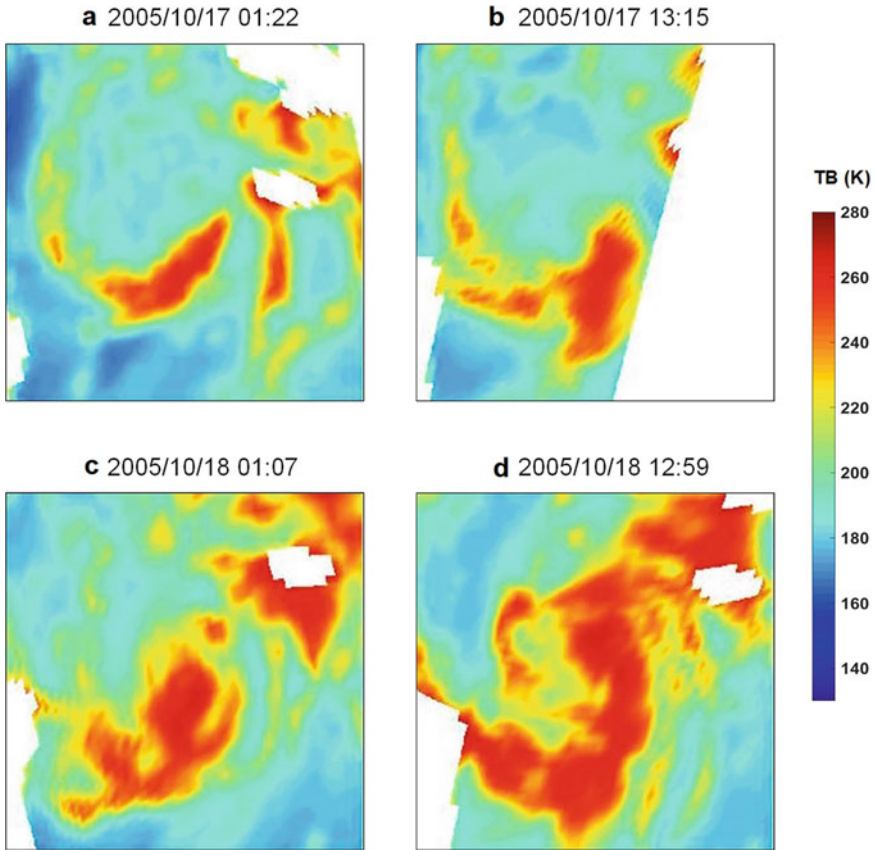


Fig. 4 The bright temperature images during the formation of Wilma (2005)

Table 7 Prediction results for the four samples of Wilma (2005)

Time of samples (UTC)	Label	Prediction label	Result	Operating time (s)
0–17 01:22	1	1	True	<0.01
10–17 13:15	1	0	False	<0.01
10–18 01:07	1	1	True	<0.01
10–18 12:59	1	1	True	<0.01

* Label 1-TC, Label 0-non-TCC

that in Fig. 4b, but there is less TC system information in the missing part. The TC structure and TB distribution pattern are not contaminated significantly. Therefore, these two samples can still be correctly identified. All in all, the proposed model can well detect tropical cyclogenesis with high-quality data.

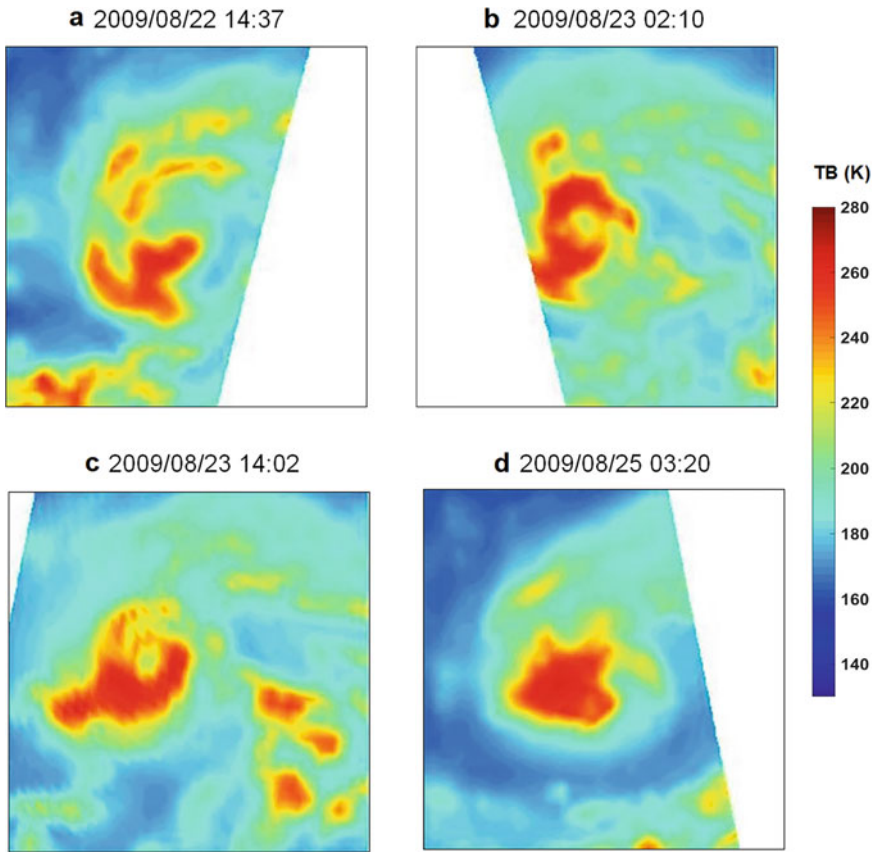


Fig. 5 The bright temperature images during the formation of Hilda (2009)

Table 8 Prediction results for the four samples of Hilda (2009)

Time of samples (UTC)	Label	Prediction label	Result	Operating time (s)
08-22 14:37	1	1	True	<0.01
08-22 02:10	1	1	True	<0.01
08-23 14:02	1	1	True	<0.01
08-25 03:20	1	1	True	<0.01

* Label 1-TC, Label 0-non-TC

5 Conclusion

In this study, a tropical cyclogenesis detection model is proposed using BLS. In contrast to the deep network methods, the new model is a lightweight flatted network, leading to lower computation and shorter training time. Meanwhile, the capacity of incremental learning of BLS is consistent with the continuously updated and accumulated remote sensing data. Adding new input data does not need to retrain the whole updated dataset. Based on the dataset consisting of 3386 TB images, the testing accuracy, HT, and FAR of BLS are 86.83%, 81.14%, and 11.18% respectively. This study confirms the applicability of BLS in the binary classification problem in ocean remote sensing. It also proves the possibility of detection of TC formation from satellite microwave TB data.

Although the BLS has shown great power in the classification problem, two defects need to be addressed: (1) the BLS is insensitive to the features of images, which will lead to poor accuracy when the image features are complicated. Inspired by the powerful ability of convolution neural network (CNN) to capture and learn image features, we will add a feature extracting mode before our BLS to improve its image processing ability. (2) The size of the dataset is too small to support the learning requirements perfectly. There are three ways to expand the dataset: One is to add the TB data from other channels (e.g., 19H/V), the other is to obtain samples in a longer period, and the last is to utilize the TB observations from other microwave radiometers (e.g., Microwave Imager onboard FengYun series satellites).

References

1. Aparna SG, D'Souza S, Arjun NB (2018) Prediction of daily sea surface temperature using artificial neural networks. *Int J Remote Sens* 39(11–12):4214–4231
2. Bergstra J, Yamins D, Cox D (2013) Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In: *International Conference on Machine Learning*, PMLR, pp 115–123
3. Chen CLP, Liu Z (2018) Broad learning system: An effective and efficient incremental learning system without the need for deep architecture. *IEEE Trans Neural Netw Learn Syst* 29(1):10–24. <https://doi.org/10.1109/TNNLS.2017.2716952>
4. Chen CLP, Zhang CY (2014) Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Inf Sci* 275:314–347. <https://doi.org/10.1016/j.ins.2014.01.015>
5. Chi J, Kim Hc (2017) Prediction of arctic sea ice concentration using a fully data driven deep neural network. *Remote Sens* 9(12). <https://doi.org/10.3390/rs9121305>
6. Cossuth JH, Knabb RD, Brown DP, Hart RE (2013) Tropical cyclone formation guidance using Pregenesis Dvorak Climatology. part i: Operational forecasting and predictive potential. *Weather Forecast* 28(1):100–118. <https://doi.org/10.1175/WAF-D-12-00073.1>
7. Dawood M, Asif A, Minhas FuAA (2020) Deep-PHURIE: deep learning based hurricane intensity estimation from infrared satellite imagery. *Neural Comput Appl* 32(13):9009–9017. <https://doi.org/10.1007/s00521-019-04410-7>

8. Dvorak VF (1975) Tropical cyclone intensity analysis and forecasting from satellite imagery. *Mon Weather Rev* 103(5):420–430
9. Gao Y, Gao F, Dong J, Wang S (2019) Transferred deep learning for sea ice change detection from synthetic aperture radar images. *IEEE Geosci Remote Sens Lett* 16(10):1655–1659. <https://doi.org/10.1109/LGRS.2019.2906279>
10. Ham YG, Kim JH, Luo JJ (2019) Deep learning for multi-year ENSO forecasts. *Nature* 573(7775):568+. <https://doi.org/10.1038/s41586-019-1559-7>
11. Han Y, Gao Y, Zhang Y, Wang J, Yang S (2019) Hyperspectral sea ice image classification based on the spectral-spatial-joint feature with deep learning. *Remote Sens* 11(18). <https://doi.org/10.3390/rs11182170>
12. Hennon CC, Helms CN, Knapp KR, Bowen AR (2011) An objective algorithm for detecting and tracking tropical cloud clusters: Implications for tropical cyclogenesis prediction. *J Atmos Oceanic Tech* 28(8):1007–1018. <https://doi.org/10.1175/2010JTECHA1522.1>
13. Kim M, Park MS, Im J, Park S, Lee MI (2019) Machine learning approaches for detecting tropical cyclone formation using satellite data. *Remote Sens* 11(10). <https://doi.org/10.3390/rs11101195>
14. Knaff JA, Brown DP, Courtney J, Gallina GM, Beven JL II (2010) An evaluation of Dvorak technique-based tropical cyclone intensity estimates. *Weather Forecast* 25(5):1362–1379. <https://doi.org/10.1175/2010WAF2222375.1>
15. Knapp KR, Kruk MC, Levinson DH, Diamond HJ, Neumann CJ (2010) The international best track archive for climate stewardship (IBTrACS) unifying tropical cyclone data. *Bul Am Meteorol Soc* 91(3):363+. <https://doi.org/10.1175/2009BAMS2755.1>
16. Kuok SC, Yuen KV (2020) Broad learning for nonparametric spatial modeling with application to seismic attenuation. *Comput-Aided Civil Infrastruct Eng* 35(3):203–218
17. Kuok SC, Yuen KV (2020) Multi-resolution broad learning for model updating using incomplete modal data. *Struct Control Health Monit* 27(8):e2571
18. Li X, Liu B, Zheng G, Ren Y, Zhang S, Liu Y, Gao L, Liu Y, Zhang B, Wang F (2020) Deep-learning-based information mining from ocean remote-sensing imagery. *Natl Sci Rev* 7(10):1584–1605. <https://doi.org/10.1093/nsr/nwaa047>
19. Matsuoka D, Nakano M, Sugiyama D, Uchida S (2018) Deep learning approach for detecting tropical cyclones and their precursors in the simulation by a cloud-resolving global nonhydrostatic atmospheric model. *Progress Earth Planet Sci* 5. <https://doi.org/10.1186/s40645-018-0245-y>
20. Nakano M, Kubota H, Miyakawa T, Nasuno T, Satoh M (2017) Genesis of super cyclone pam (2015): Modulation of low-frequency large-scale circulations and the madden-julian oscillation by sea surface temperature anomalies. *Mon Weather Rev* 145(8):3143–3159. <https://doi.org/10.1175/MWR-D-16-0208.1>
21. Olander TL, Velden CS (2007) The advanced dvorak technique: Continued development of an objective scheme to estimate tropical cyclone intensity using geostationary infrared satellite imagery. *Weather Forecast* 22(2):287–298. <https://doi.org/10.1175/WAF975.1>
22. Olander TL, Velden CS (2019) The advanced dvorak technique (ADT) for estimating tropical cyclone intensity: Update and new capabilities. *Weather Forecast* 34(4):905–922. <https://doi.org/10.1175/WAF-D-19-0007.1>
23. Park MS, Kim M, Lee MI, Im J, Park S (2016) Detection of tropical cyclone genesis via quantitative satellite ocean surface wind pattern and intensity analyses using decision trees. *Remote Sens Environ* 183:205–214. <https://doi.org/10.1016/j.rse.2016.06.006>
24. Pradhan R, Aygun RS, Maskey M, Ramachandran R, Cecil DJ (2018) Tropical cyclone intensity estimation using a deep convolutional neural network. *IEEE Trans Image Process* 27(2):692–702. <https://doi.org/10.1109/TIP.2017.2766358>
25. Rozoff CM, Velden CS, Kaplan J, Kossin JP, Wimmers AJ (2015) Improvements in the probabilistic prediction of tropical cyclone rapid intensification with passive microwave observations. *Weather Forecast* 30(4):1016–1038. <https://doi.org/10.1175/WAF-D-14-00109.1>
26. Scher S, Messori G (2019) Weather and climate forecasting with neural networks: using general circulation models (GCMs) with different complexity as a study ground. *Geosci Model Dev* 12(7):2797–2809. <https://doi.org/10.5194/gmd-12-2797-2019>

27. Su H, Wu L, Jiang JH, Pai R, Liu A, Zhai AJ, Tavallali P, DeMaria M (2020) Applying satellite observations of tropical cyclone internal structures to rapid intensification forecast with machine learning. *Geophys Res Lett* 47(17). <https://doi.org/10.1029/2020GL089102>
28. Velden C, Olander T, Zehr R (1998) Development of an objective scheme to estimate tropical cyclone intensity from digital geostationary satellite infrared imagery. *Weather Forecast* 13(1):172–186. [https://doi.org/10.1175/1520-0434\(1998\)013<0172:DOAOST>2.0.CO;2](https://doi.org/10.1175/1520-0434(1998)013<0172:DOAOST>2.0.CO;2)
29. Velden C, Harper B, Wells F, Beven JL II, Ehr R, Olander T, Mayfield M, Guard CC, Lander M, Edson R, Avila L, Burton A, Turk M, Caroff A, Christian A, Caroff P, McCrone P, (2006) The Dvorak tropical cyclone intensity estimation technique. *Bull Am Meteorol Soc* 87(9):1195–1210. <https://doi.org/10.1175/BAMS-87-9-1195>
30. Wei L, Guan L, Qu L, Guo D (2020) Prediction of sea surface temperature in the China Seas based on long short-term memory neural networks. *Remote Sens* 12(17). <https://doi.org/10.3390/rs12172697>
31. Wentz F (1997) A well-calibrated ocean algorithm for special sensor Microwave/Imager. *J Geophys Res-Oceans* 102(C4):8703–8718. <https://doi.org/10.1029/96JC01751>
32. Xiang B, Lin SJ, Zhao M, Zhang S, Vecchi G, Li T, Jiang X, Harris L, Chen JH (2015) Beyond weather time-scale prediction for Hurricane Sandy and super typhoon Haiyan in a global climate model. *Mon Weather Rev* 143(2):524–535. <https://doi.org/10.1175/MWR-D-14-00227.1>
33. Yamaguchi M, Koide N (2017) Tropical cyclone genesis guidance using the early stage dvorak analysis and global ensembles. *Weather Forecast* 32(6):2133–2141. <https://doi.org/10.1175/WAF-D-17-0056.1>
34. Yi K, Wang X, Cheng Y, Chen C (2018) Hyperspectral imagery classification based on semi-supervised broad learning system. *Remote Sens* 10(5):685
35. Zhang W, Fu B, Peng MS, Li T (2015) Discriminating developing versus nondeveloping tropical disturbances in the Western North Pacific through Decision Tree Analysis. *Weather Forecast* 30(2):446–454. <https://doi.org/10.1175/WAF-D-14-00023.1>
36. Zhang Y, Yuen KV (2021) Crack detection using fusion features-based broad learning system and image processing. *Comput-Aided Civil Infrastruct Eng.* <https://doi.org/10.1111/mice.12753>
37. Zheng G, Li X, Zhang RH, Liu B (2020) Purely satellite data-driven deep learning forecast of complicated tropical instability waves. *Sci Adv* 6(29). <https://doi.org/10.1126/sciadv.aba1482>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits any noncommercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if you modified the licensed material. You do not have permission under this license to share adapted material derived from this chapter or parts of it.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Tropical Cyclone Monitoring Based on Geostationary Satellite Imagery



Chong Wang, Qing Xu, Xiaofeng Li, Gang Zheng, and Bin Liu

1 Introduction

Tropical cyclones (TCs) are extreme weather processes developed over the tropical ocean. They are called typhoons in the northwestern Pacific and hurricanes in the eastern Pacific and Atlantic. The Northwest Pacific Ocean is the basin with the largest number of TCs in the world [5, 11], where TCs can be observed throughout the year [48]. TCs can cause huge losses to marine production and transportation, such as sweeping away the fishing nets, destroying the fish rans and cages for breeding, leading to the death of a large number of fish and shellfish, destroying the breakwaters and offshore oil platforms, overturning ships and aircraft, etc. After landing, they usually cause hazardous disasters such as storm surges, mountain torrents, urban waterlogging, landslides and debris flows owing to the destructive wind and low

C. Wang · X. Li (✉)

CAS Key Laboratory of Ocean Circulation and Waves, Institute of Oceanology,
Chinese Academy of Sciences, Qingdao 266071, China
e-mail: lixf@qdio.ac.cn

Q. Xu

College of Marine Technology, Faculty of Information Science and Technology,
Ocean University of China, Qingdao 266100, China

G. Zheng

State Key Laboratory of Satellite Ocean Environment Dynamics, Second Institute of
Oceanography, Ministry of Natural Resources, Hangzhou 310012, China

B. Liu

College of Marine Sciences, Shanghai Ocean University, Shanghai 201306, China

State Key Laboratory of Satellite Ocean Environment Dynamics, Second Institute of
Oceanography, Ministry of Natural Resources, Hangzhou, Zhejiang 310012, China

Key Laboratory of Marine Ecological Monitoring and Restoration Technologies,
Ministry of Natural Resources, Shanghai 200137, China

pressure, which poses a serious threat to coastal areas and brings great damages to human life and property.

The annual number of TCs generated in the Northwest Pacific Ocean accounts for 36% of the total number of TCs in the world [55]. In this basin, TCs occur most frequently from July to September. They can easily develop into super TCs under the action of high temperature and high humidity. Under the influence of marine and atmospheric environment, TCs usually move from the southeast to the northwest. Many countries have listed the TC as one of the major natural disasters affecting national public security, and clearly proposed to strengthen the research on key technologies of TC monitoring and early warning [20, 54].

Locating the TC center and estimating its intensity have been widely considered as an important part of TC monitoring by meteorological forecasting agencies. Under the background of global warming, the intensity of global TCs shows a significant increasing trend [20]. Accurate TC center position and intensity information can better initialize the numerical models and make them more accurate to predict the intensity and movement direction of TCs, especially the rapidly strengthening TCs [20, 50], and thus help people to take precautions in advance and reduce losses.

Traditional TC monitoring platforms include coastal stations, buoys, oil drilling platforms, etc.. But the recorded TC data is very few due to the sparse distribution of these platforms. With the development of airborne remote sensing technology, real-time observation of TCs in a large space and time range is realized by using airborne radars or radiometers, but is still limited by extreme weather conditions. Since 1970, a major breakthrough has been made in satellite remote sensing. Many countries have launched geostationary meteorological satellites in succession, which can realize real-time and continuous observation of the earth. Spaceborne sensors generally have high spatial resolution and have become an important tool for TC monitoring. Although there are more and more observation methods, the monitoring of TCs still relies on the experience of experts to a certain extent. Particularly, there is a lack of objective and effective monitoring methods for weak TCs at the stage of formation or extinction.

A variety of TC center location methods based on infrared images, scatterometer data or synthetic aperture radar images have been developed. These methods can be divided into four categories. The first category is subjective method, which locates the TC center based on the forecaster's experience judgment on the Central Dense TC Overcast in the satellite images [7–10, 37]. The second is the threshold method, which is used to segment and identify the TC's eye area from satellite images and determine the morphological center of the eye area [2, 14, 28, 30, 32, 47]. The third is the spiral curve method. As is well known, the structure of the TC cloud system is not symmetrical, but spiral. The concept of spiral can be expressed by vector distance and the spiral center is the TC center [23, 51]. The fourth category is wind vector or cloud motion wind (CMW) method, which uses the wind field retrieved by scatterometers or time series of infrared satellite images to establish the relationship between the wind vector or CWM and the movement variation of a TC so as to locate the TC center [17, 21, 31, 33, 34, 40, 52, 57].

For TC intensity estimation using satellite images, two traditional methods are Dvorak technology [7, 10, 12, 19, 56] and empirical regression method [3, 16, 24, 26, 36, 38, 42–44]. The Dvorak technology was first proposed by Dvorak in 1975 [7] and has been operated for more than 40 years. The method assumes that the rotation and shape of a TC eye area are related to the strengthening and weakening of a TC, and TCs with similar intensities have similar morphological characteristics. In the empirical regression method such as the deviation-angle variance technique (DAVT) [3], the TC intensity is determined by establishing the relationship between the TC intensity and different characteristics. Both methods rely on high levels of artificial features converted from satellite images of TCs. However, it is difficult to obtain general characteristics of TCs and establish empirical regression models for TCs at different development stages and in different regions.

In the past few years, meteorologists and oceanographers have introduced the convolutional neural network (CNN) into TC monitoring. CNN is a kind of feedforward deep neural network, which is trained through back propagation algorithm, and its inspiration comes from the natural visual cognitive mechanism of organisms [22]. A complete CNN model consists of convolutional layers, pooling layers and full connection layers. The convolution layer is used to extract features from the image, the pooling layer filters and reduces the number of features, and the full connection layer learns the relationship between these features and the model output. The CNN model not only avoids the complex image preprocessing, but also does not depend on the priori knowledge of TCs, and thus can meet the requirements of automatic and objective TC center location and intensity estimation.

The CNN has achieved great success in image classification and target recognition [15, 22, 24, 26, 38, 42, 43, 46] and also shows great application potential in TC intensity estimation. Pradhan et al. [36] collected 8138 TC images over the North Pacific Ocean and the Atlantic Ocean from 1998 to 2012, and labelled the images into eight categories based on the Saffir-Simpson hurricane scale and the HURDAT2 Best Track dataset. They used the CNN model to classify the TCs and then the maximum wind speed (MWS) was calculated according to the probability of each TC category. The root mean square error (RMSE) of the derived MWS is 10.19 kt. However, there are green shorelines in the satellite images used in their study, which may affect the accuracy of TC intensity estimation [16]. Combinido et al. [6] used a CNN regression model to estimate TC intensity, but the RMSE of the MWS (13.23 kt) was larger than that of the CNN classification model proposed by [36]. Also with the CNN regression model, Chen et al. calculated the intensity of TCs in the global ocean based on more TC images [36, 44] and the RMSE was reduced to 10.58 kt. Recently, Tian et al. combined the CNN classification and regression models to estimate TC intensity [3], and they obtain a smaller RMSE of 8.91 kt.

The above results show that the CNN classification model or regression model performs well in TC intensity estimation. Compared with the Dvorak technology, the CNN model does not rely on the subjective judgment of the forecaster, which ensures the objectivity of the method. Compared with the empirical regression method, the neural network reduces the requirement of the knowledge level of the person who uses this method. In addition, although the TC center position is necessary in the CNN

model, the information is only used to cut the input images. Therefore, compared with the empirical regression method, the CNN model does not require a very high TC positioning accuracy.

In conclusion, the accuracy of TC center location and intensity estimation technology has been greatly improved in recent years, but there are still some limitations. For operational TC monitoring, objective, we need a fast and robust objective method, and the CNN just meets this requirement. In this chapter, a set of CNN models were designed to determine the TC center position and intensity from Himawari-8 geostationary satellite images in the Northwest Pacific Ocean. By discussing the influence of the sensor channel, the number of satellite images and the configuration of the neural network on the model performance, an optimal CNN model was developed for automatic TC monitoring. The data and structure of the CNN model are described in Sect. 2. Sect. 3 and 4 present TC center location and intensity estimation results, respectively. Section 5 is the summary.

2 Data and Methodology

2.1 Data

The Japan Meteorological Agency (JMA) launched the Himawari-8 (H-8) geostationary meteorological satellite in October 2014. The Advanced Himawari Imager (AHI) onboard H-8 provides observations of different regions at different modes: Full Disk (global scope), Japan Area (scope of two Japanese regions), Specific Area (scope of two regions), and Landmark Area (scope of two regions). The Full Disk and Japan area are fixed, while the other two specific areas and landmark areas can be adjusted flexuously. The scanning range is shown in Fig. 1 (60°S-60°N, 80°E-160°W). The on orbit working life of H-8 is 8 years [4]. After calibration and calibration test, the data were provided since July 2015 <http://www.eorc.jaxa.jp/ptree>.

AHI's 16 channels cover the whole Northwest Pacific: three visible, three near-infrared, and ten thermal-infrared channels. Full-disk observations are taken every 10 minutes. The brightness temperature data from five infrared (IR, Channels 7, 8, 13, 14, and 15, see Table 1 for details from <http://www.eorc.jaxa.jp/ptree>.) channels was obtained.

Based on AHI data, we collected 6,690 satellite images of 97 TCs over the Northwest Pacific with a time interval of 3 hours during the whole life cycle of the TCs from 2015 to 2018, and the image resolution is 5 km. As shown in Fig. 2, the brightness temperature data of 5 channels (7, 8, 13, 14, 15) with higher transmittance near the large window (center wavelength are 3.9, 6.2, 10.4, 11.2, 12.4 μm , respectively) were selected to locate TC center and estimate TC intensity. Channel 7 is mainly used for observing cloud and natural disasters in the lower layer, channel 8 for observing water vapor content in the upper and middle layer, channel 13 for observing

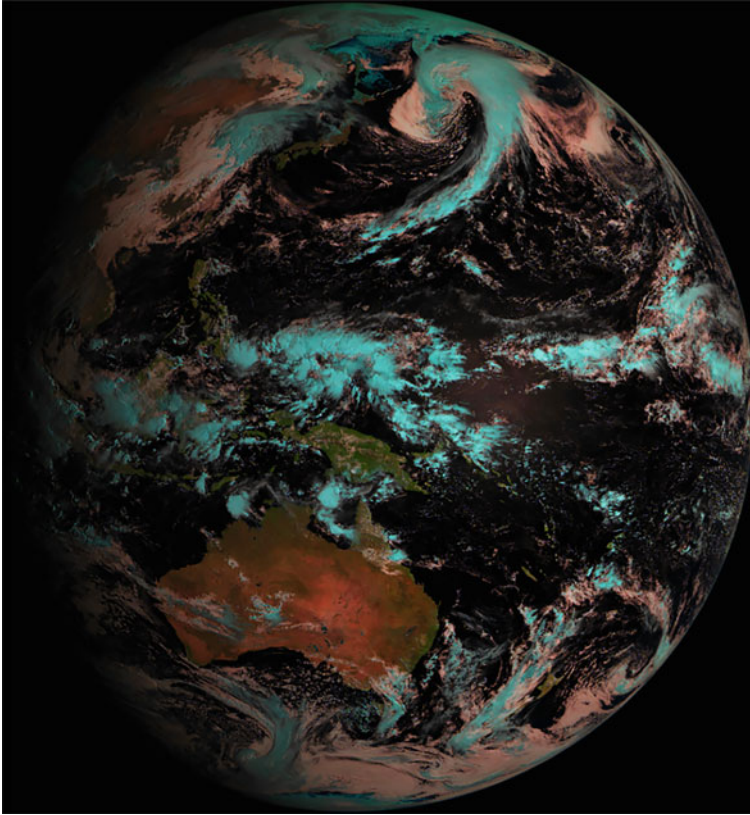


Fig. 1 Himawari-8 satellite scanning range

cloud images and cloud top conditions, and channels 14 and 15 are mainly used for observing cloud images and sea surface temperature [4].

We use the TC Best Track dataset provided by the tropical cyclone information center of China Meteorological Administration (CMA) <http://tcdata.TC.org.cn> as ground truth data. The data set was compiled by Shanghai Typhoon Institute. The TC Best Track dataset over the Northwest Pacific from 1949 to 2019 includes the TC number and name, time, longitude and latitude of the TC center, the MSW (2 minute average wind speed), the minimum air pressure, etc.. The time interval is 6 hours before 2017 and has been encrypted to 3 hours for landing TCs since 2017. Starting from 2018, the Best Track data has provided 3-h TC information 24 hours before the landing activities [1, 45]. Consistent with H-8 satellite observations, we downloaded the Best Track data of 97 TCs over the Northwest Pacific during 2015-2018.

Table 1 Himawari-8 channel setup [4]

Channel	Wavelength (μm)	Observation object
1	0.46	Vegetation, aerosol observation and color image synthesis
2	0.51	Vegetation, aerosol observation and color image synthesis
3	0.64	Sublayer cloud and color image synthesis
4	0.86	Vegetation and aerosol observation
5	1.6	Cloud phases
6	2.3	The effective radius of cloud droplets
7	3.9	Sublayer cloud and natural disasters
8	6.2	Water vapor in the upper and middle layers
9	7.0	Water vapor in the middle layers
10	7.3	Water vapor in the middle layers
11	8.6	Cloud phases and SO ₂
12	9.6	O ₃
13	10.4	Cloud image and cloud top image
14	11.2	Cloud image and sea surface temperature
15	12.3	Cloud image and sea surface temperature
16	13.3	Cloud height

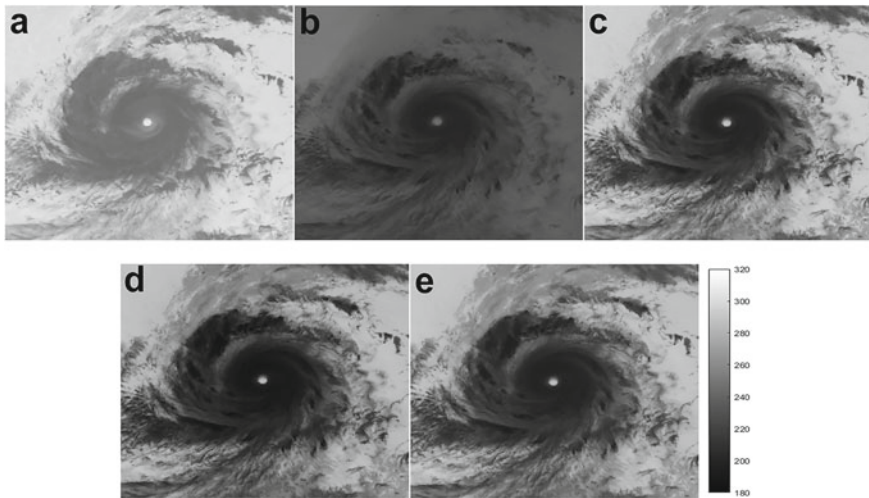


Fig. 2 Bright temperature(unit: K) image of different channels of “Soudelor” obtained by AHI on Himawari-8 satellite at 18:00 (UTC) on August 15, 2015 **a** channel 7, **b** channel 8, **c** channel 13, **d** channel 14, **e** channel 15. The image space range is 1255 km×1255 km

2.2 Data Pre—Processing

We selected H-8 satellite images synchronized with the 6-h Best Track dataset to design the CNN-based TC center location (CNN-L) model. There are 3298 images in total, among which 1971 are used for model training, 657 for validation and 670 for testing. For each training or validation image, we re-extracted a sub-image covering an area of $1500 \text{ km} \times 1500 \text{ km}$ (pixel size 301×301), and then cut the pixel size from 301×301 to 151×151 three times by moving it up, down, left or right randomly. Finally, we labeled the processed image with the number and direction of moving pixels. For example, if a TC center is shifted left by 5 pixels, and up by 10 pixels, the image is labeled $(-5, 10)$. The test images were only cropped. Finally, we obtained 5913 training images, 1971 validation images, and 670 test images with a reduced size of 151×151 .

More training images may also help to improve the CNN-based TC intensity estimate (CNN-I) model performance. Hence, before the estimation of TC intensity, we interpolated the MSW provided by the Best Track dataset every 3 hours to increase the data samples. The total of 6690 3-h images with a size of 251×251 was labeled correctly according to eight categories using the Saffir–Simpson hurricane wind scale (H1 to H5) along with intensity categorization for the tropical storm (TS) and tropical depression (TD) as TC intensity categories (Table 2). The total 6690 images are divided into the training, validation and testing images with 4014, 1338 and 1338 images, respectively. Studies show that the rotation of images in a CNN model can reduce the sensitivity of orientation and does not affect the classification accuracy. Therefore, the normalized training and validation images were artificially rotated 90° , 180° and 270° clockwise. In this way, the number of images is increased by three times, and finally, we obtained 16,056 training images and 5352 validation images for the construction of the CNN based TC intensity estimation model.

Table 2 Saffir–Simpson hurricane wind scale and related classifications (The Saffir–Simpson Team, 2012). MSW is ten-minute averaged maximum sustained wind speed

Symbol	Category	MSW (kt)
NC	No Category	≤ 20
TD	Tropical depression	20–33
TS	Tropical storm	34–63
H1	Category 1	64–82
H2	Category 2	83–95
H3	Category 3	96–112
H4	Category 4	113–136
H5	Category 5	≥ 137

2.3 Methodology

The convolutional neural network (CNN) model consists of convolutional layers (C), pooling layer (P) and fully connected layer (FC). The convolutional layer is used to extract image features. In order to reduce the number of features and speed up the calculation, the pool layer uses the filtering methods such as maximum, average and minimum values to process these features. The nonlinear relationship between the filtered features and the output results is learnt by the full connection layer. To prevent overfitting, a dropout layer is usually added before the full connection layer [26], which invalidates part of the connection between the upper and lower layers. CNN is usually designed as a feed forward network that can be trained with a backpropagation algorithm. When the error propagates back in the model, the network is optimized by updating the weights and deviations to minimize the value of the loss function [22].

The CNN model architecture for TC monitoring is shown in Fig. 3. It consists of one input layer, four convolutional layers, four pooling layers, one dropout layer, two fully connected layers, and one output layer. Table 3 lists the parameter setting of the CNN model as shown in Fig. 3. The parameters include the convolutional kernel shape, output shape and number of parameters, as well as the convolution kernel shape used in each convolution layer, the filter shape used in each pooling layer and the step size. Taking the CNN model shown in Fig. 3 as an example, the satellite

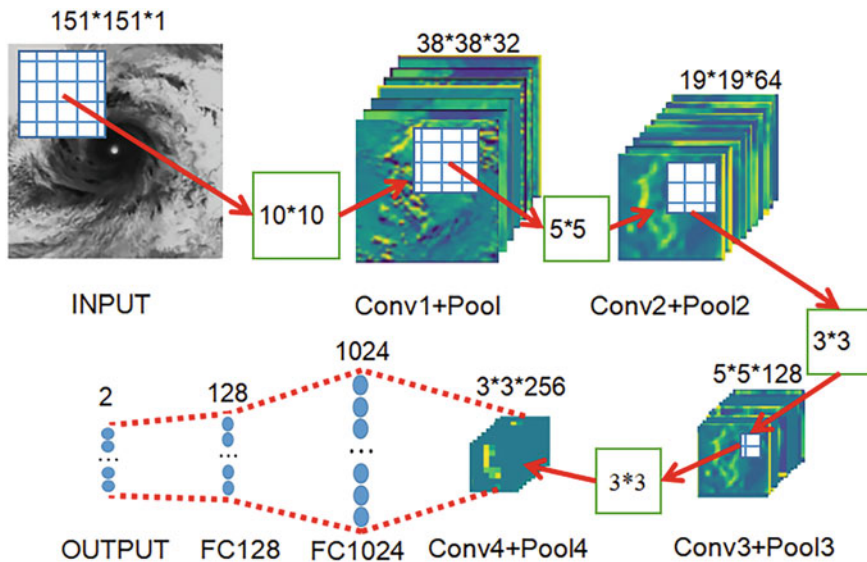


Fig. 3 Framework of the CNN based TC center location model

Table 3 The MLE of CNN based TC center location model

Model number	Wavelength (μm)	MLE (km)
CNN-L 1	10.4	40.3
CNN-L 2	11.2	40.5
CNN-L 3	2.3	40.1

image with a size of 151×151 was input into the CNN model. The first convolution operation is performed on the satellite image. The size of each convolution kernel was 10×10 , and the step size was 2. In this process, the model needs to learn 3232 parameters; In the first pooling layer, 32 feature images with the size of 3×3 and step size of 2 are filtered, and 32 feature images with the size of 38×38 are obtained by using the maximum pooling method. After four convolutional layers and four pooling layers, 256 abstract feature images of the size of 3×3 are finally generated. These feature images are expanded into 1-dimensional data of the size of 2304, input into the dropout layer (the dropout rate is 0.5), and then connected with the first fully connected layer. The model needs to learn 2360320 parameters. Finally, the model output is obtained in the output layer. In the whole training process, the model needs to learn a total of 2915298 parameters.

For TC center location and intensity estimation from multi-channel satellite images, the input channel, the image resolution and model parameters would affect the training efficiency and accuracy of the CNN model. By Setting up a group of sensitivity experiments to investigate the influence of different factors on the model performance, we aim to develop an optimal CNN model for TC monitoring.

3 TC Center Location

The CNN model for TC center location, i.e., the CNN-L model, as shown in Figure 3.3 is established in this section. We use the mean location error (MLE) to evaluate the performance of the model:

$$MLE = \frac{\sum_{i=1}^n \sqrt{(x_i - x'_i)^2 + (y_i - y'_i)^2}}{n} \quad (1)$$

where n is the number of test images; (x, y) and (x', y') are TC center positions from the Best Track data and CNN-L model outputs, respectively.

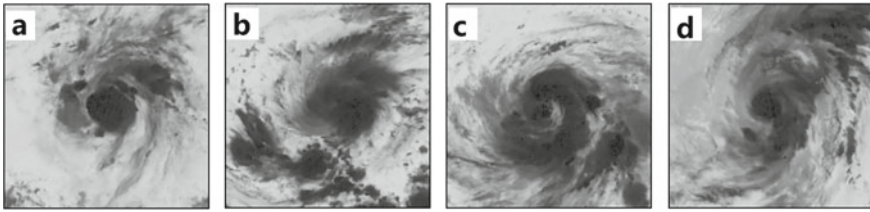


Fig. 4 Himawari-8 Channel 7 images of TCs: **a** Halola (9:00 UTC on 13 July 2015), **b** Soudelor (18:00 UTC on 30 July 2015), **c** Atsani (12:00 UTC on 16 August 2015), **d** Meranti (12:00 UTC on 14 September 2016)

As shown in Fig. 4, there are sometimes some noises in the image of channel 7 or 8 due to the sensor calibration problem, which would seriously affect the TC location accuracy. Therefore, we only use the images of channels 13, 14 and 15 as the input of the CNN-L model.

We take TC “Maria” occurring during 4 July to 11 July, 2018 as an example. Figure 5 shows the TC center location results using the network configuration listed in Table 3. The mean location errors between different CNN-L model outputs and observations are similar, suggesting that the image channel has little effect on the accuracy of the TC center location model.

Using Channel 15 data as the input of the CNN-L 3 model, a variety of network configurations listed in Table 4 were further tested. CNN-L 3, CNN-L 4, and CNN-L 5 models all consist of three or four convolutional layers, three or four pooling layers, and two FC layers. However, each model has a different number of kernels in the convolutional layers. The stride and zero-padding, the shape of the FC layer, and the dropout in each model are listed in Table 4. One can see that CNN-L 3 produces the lowest MLE, indicating that too many or few convolutional layers do not help to improve the CNN-L model.

The results of CNN-L 3 model for different categories of TCs are shown in Figs. 5, 6 and Table 5. Strong TCs generally demonstrate a more distinct and stable structure and more obvious TC eye area than weak TCs. As a result, the mean location error of the CNN-L 3 model also decreases rapidly with the increase of TC intensity. The average MLEs of H1-H5 and H4-H5 TCs are 30 km and less than 25 km, respectively. As shown in Table 5, the accuracy of our CNN based TC center location model is comparable to that of some techniques that also locate TCs from IR images with spatial resolutions of 2.5–12.5 km (Table 6).

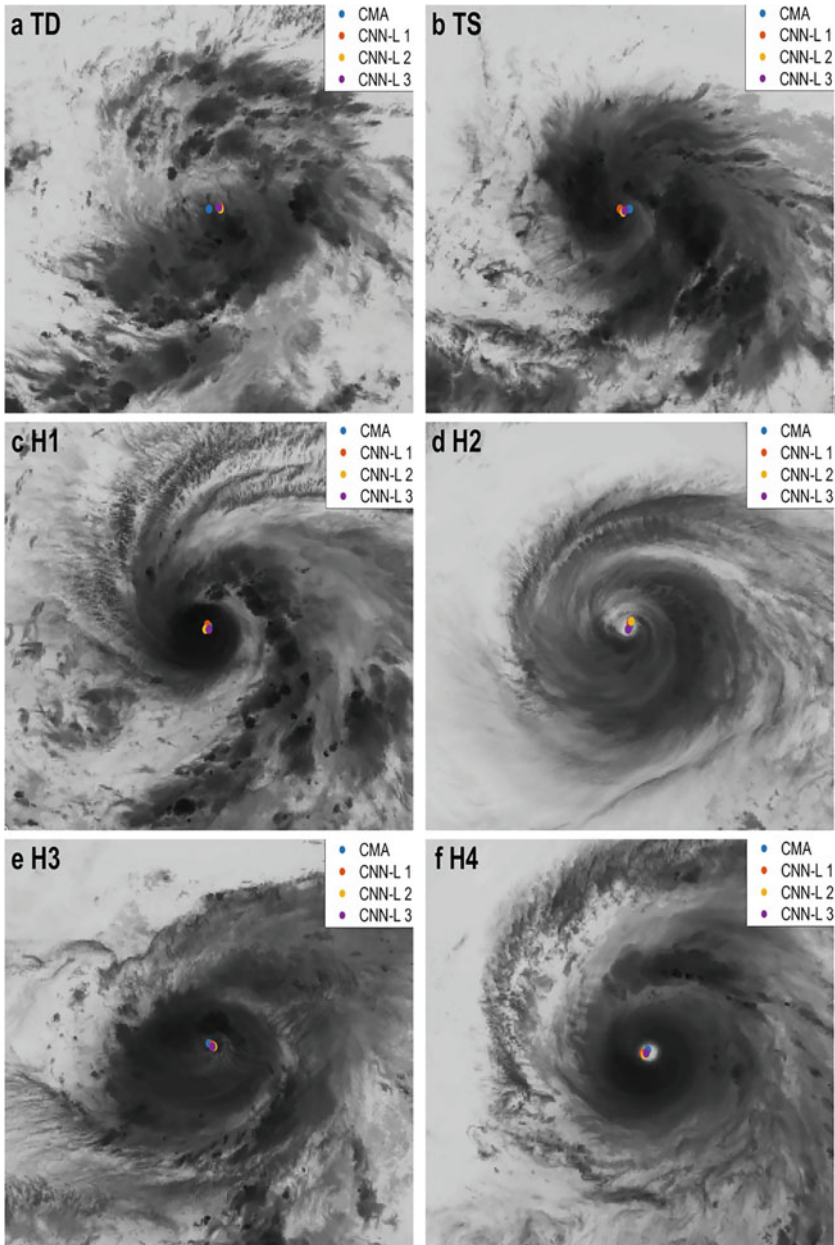


Fig. 5 CNN based TC center location model results for TC “Maria” occurring from 4 July to 11 July, 2018 at different stages: **a** Tropical depression, **b** Tropical storm, **c** Category 1, **d** Category 2, **e** Category 3, **f** Category 4. The red, yellow and purple dots represent the TC center positions determined by the CNN-L model with brightness temperature of Channels 13, 14 and 15 as the input, respectively. The blue dot shows the TC center position from the Best Track dataset provided by CMA

Table 4 Results of the CNN based TC center location models with different parameters(For example, C1 means the first convolutional layer and P1 means the first pooling layer, C1(32@3 × 3) denotes 32 kernels in the first convolutional layer with a size of 3 × 3)

Model number	Parameters	MLE (km)
CNN-L 3	C1(32@10 × 10), P1(3 × 3)	40.1
	C2(64@5 × 5), P2(3 × 3)	
	C3(128@3 × 3), P3(3 × 3)	
	C4(256@3 × 3), P4(2 × 2)	
	Dropout = 0.5, FC1024, FC128	
CNN-L 4	C1(32@10 × 10), P1(3 × 3)	42.3
	C2(64@5 × 5), P2(3 × 3)	
	C3(128@3 × 3), P3(3 × 3)	
	Dropout = 0.5, FC1024, FC128	
CNN-L 5	C1(32@10 × 10), P1(3 × 3)	41.8
	C2(64@8 × 8), P2(3 × 3)	
	C3(128@4 × 4), P3(3 × 3)	
	C4(256@4 × 4), P4(2 × 2)	
	Dropout = 0.5, FC1024, FC128	

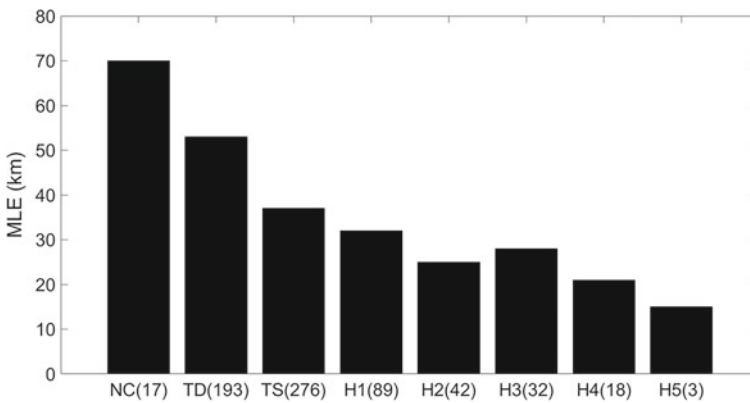


Fig. 6 Mean location error (km) of the CNN-L 3 model for different categories of TCs. Numbers in brackets represent the number of samples in the test group for this category

Table 5 Mean location error (km) of the CNN-L 3 model

TC category	NC	TD	TS	H1	H2	H3	H4	H5
Data number	17	203	298	99	43	33	18	3
MLE (km)	90	53	37	32	25	28	21	15

Table 6 Performance of the CNN model in this study and other methods for TC center location

Literature	MLE (km)
Pal P et al. [32]	11–79
Jin S et al. [23]	42.1
Our model	40.1

4 TC Intensity Estimation

Similar to Sec.3, the influence of the selection of input data and model parameters on the performance of the CNN based TC intensity estimation model, i.e., the CNN-I model shown in Fig. 7, is investigated in this section. The possible solution of the side effects of imbalanced dataset is also discussed.

We evaluate the performance of the CNN-I model from three aspects:

(1) Accuracy

The number of exact-hits, which refers to the correct classification of a TC with the highest confidence, is the accuracy metric.

(2) Root mean square error (RMSE) and mean average error (MAE) of TC intensity

For categories TD through H4, we define the estimated intensity or MSW of a TC as the weighted average of the two highest categories for their probabilities.

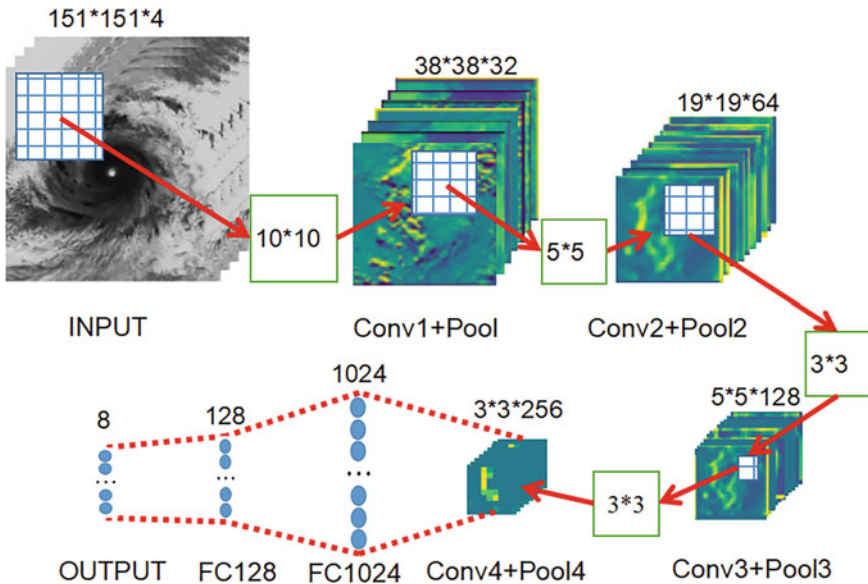


Fig. 7 Framework of the CNN based TC intensity estimation model

Table 7 Results of the CNN based TC intensity estimation model with images of different channels as the input

Model number	Wavelength (μm)	Accuracy (%)	RMSE (kt)
CNN-I 1	3.9	78.1	12.14
CNN-I 2	6.2	78.2	12.41
CNN-I 3	10.4	80.3	11.40
CNN-I 4	11.2	80.3	11.54
CNN-I 5	12.3	80.5	11.28
CNN-I 6	3.9,12.3	81.0	11.25
CNN-I 7	6.2,12.3	81.3	11.03
CNN-I 8	10.4,12.3	81.5	11.05
CNN-I 9	11.2,12.3	80.7	11.71
CNN-I 10	3.9,10.4,12.3	82.7	10.76
CNN-I 11	6.2,10.4,12.3	82.5	10.89
CNN-I 12	3.9,6.2,10.4,12.3	82.9	10.64

Otherwise, we use the mean speed of the category that has the highest confidence. The MSW (W) is evaluated as [35]:

$$W = U_1 \times P_1 + U_2 \times P_2 \quad (2)$$

where P_1 and P_2 are the probabilities which output by the model of the TC categories with the highest and second-highest confidence, respectively; U_1 and U_2 are the mean wind speed of the corresponding category.

(3) Confusion matrix and classification report

As shown in Table 9, The confusion matrix depicts a model's overall classification performance. The number along the diagonal line in a confusion matrix represents the number of correctly identified images for any category. A CNN model's precision (P), recall (R , or confidence of detection), and f1-score ($F1$) are described in the classification report. The ratios of real positive class values to total positive classifications and the number of positive class values in the test data, respectively, are P and R . $F1 = 2P \times R / (P + R)$ is the harmonic mean of recall and precision.

Table 7 shows the results of the CNN based TC intensity estimation model based on 1338 test images with the model parameters shown in Fig. 7.

As shown in Table 7. for CNN-I models with single-channel input, Channels 13-15 can achieve a higher accuracy. Channels 14 and 15 correspond to the thermal-IR bands, which are often used to observe the cloud and calculate surface temperature (SST). From a different perspective, the data support the conclusion of [41] that TC intensity is connected with the temperature deficit of cloud top against sea surface. We first combined Channel 15 with the other channels in CNN-I 6 through CNN-I 9 models to examine the use of multi-channel combination in TC intensity estimate. Comparing CNN-I 9 with CNN-I 5, we can see that the combination of Channels

15 with 7, 8 or 13 improves the model performance, while Channel 14 has little contribution to the improvement of the model because its wavelength is close to Channel 15. These two thermal infrared channels may provide redundant information for TC intensity estimation, resulting in data redundancy.

With the input of more (3 or 4) channels of data, the 4-channel model CNN-I 12 produces the best result with an accuracy of 82.9% and a RMSE of 10.64 kt. This is also consistent with some theoretical studies, indicating that the information of water vapor, cloud characteristics, the brightness temperature difference of cloud top and sea surface provided by the combination of channels 7, 8, 13 and 15, plays an important role in TC intensity estimation [29, 49, 53]. Through the combined input of multi-channel satellite images, the CNN-I model can learn the complex nonlinear relationship between various elements and TC intensity.

A variety of network configurations were tested further using the CNN-I 12 model, and the results are showed in Table 8. Four convolutional layers, four pooling layers,

Table 8 Results of the CNN based TC intensity estimation model with different parameters

Model number	Parameters	Accuracy	RMSE (kt)
CNN-I 13	C1(16@10 × 10), P1(3×3)	82.9%	10.64
	C2(32@5×5), P2(3×3)		
	C3(64@3×3) P3(3×3)		
	C4(128@3×3), P4(3×3)		
	Dropout = 0.5, FC1024, FC128		
CNN-I 12	C1(32@10× 10), P1(3×3)	84.8%	10.19
	C2(64@5× 5), P2(3×3)		
	C3(128@3×3) P3(3×3)		
	C4(256@3×3), P4(3×3)		
	Dropout = 0.5, FC1024, FC128		
CNN-I 14	C1(64@10×10), P1(3×3)	80.1%	11.48
	C2(128@5×5),P2(3×3)		
	C3(256@3×3) P3(3×3)		
	C4(512@3×3), P4(3×3)		
	Dropout = 0.5, FC1024, FC128		
CNN-I 15	C1(32@10×10), P1(3×3)	77.8%	12.59
	C2(64@5×5), P2(3×3)		
	C3(128@4×4) P3(3×3)		
	C4(256@3×3), P4(3×3)		
	Dropout = 0.5, FC1024, FC128		
CNN-I 16	C1(32@10×10), P1(3×3)	86.0%	10.06
	C2(64@5×5), P2(3×3)		
	C3(128@3×3), P3(3×3)		
	C4(256@3×3), P4(3×3)		
	Dropout = 0.5, FC1024, FC128		

and two fully connected layers make up CNN-I 13-16 models. For CNN-I 12 to CNN-I 15, the number of convolution kernel in the convolutional layer increases gradually. The model's accuracy improves slightly as the number of convolution kernels increases (CNN-I 13), but it decreases dramatically beyond a certain range of kernel numbers (CNN-I 15). Although more kernels allow for the extraction of more feature maps, these maps may not always have a favorable impact on the model's improvement. In addition, the decrease of the model performance with the increasing number of convolution kernel may also be associated with the number of training samples. More complex CNN models need to learn more parameters. It is difficult to obtain a high accuracy when the difference between the training sample size and the number of model parameters is too large. Compared with the CNN-I 12 model, CNN-I 14 changes the step size of the convolution kernel operation, but the accuracy is reduced.

Recently, Woo et al proposed the spatial attention and channel attention mechanism, which is based on the study of human vision [54]. As show in Fig. 8, after adding the spatial and channel attention layers, the CNN-I 17 model gives the highest accuracy (86.0%) and the lowest RMSE (10.06 kt) of the MWS. The CNN based TC intensity estimation model can focus on the key factors revealed by the attention mechanism, which helps to improve the accuracy of the model.

In general, the CNN-I 16 multi-category classification model does a good job at estimating TC intensity. However, as demonstrated in Tables 9, 10, the classification results for TC categories with little training samples are not particularly satisfactory. For example, there are 62 samples in the H3 category, but only 48 have been identified correctly. Only 77.0% of the H3 category is accurate. The degradation is caused by an imbalance in the training data across different types of TC datasets. Taking H3 as an example, the H3 category accounts for only 2.3% of total training numbers. Even if most of the H3 images are misclassified during the training step of the CNN-based TC intensity estimation model, the loss will only increase somewhat. Because the CNN adjusts the weight value of each layer in response to the loss, the network will struggle to learn the features of a category if there are few samples. As a result, the accuracy of the CNN-I model for H2 category is lower.

We use Focal_loss function to replace the original loss function in the CNN TC intensity estimation model. This function aids a model's learning of features by raising the category's weight with fewer data in loss, and has demonstrated excellence in the field of target recognition. In this way, In the event of limited samples, the model can better learn the TC category's relevant attributes. The Focal_loss function's definition is as follows: [27]:

$$FL(p_t) = -a_t(1 - p_t)^r \log(p_t) \quad (3)$$

where p_t is the output of model for NC to H5 category, a_t is the weight coefficient which is determined by the proportion of the number of NC to H5 category to the total data, and r is the empirical parameter. The values of a_t and r for each TC category used in this study are shown in Table 11. The accuracy of the TC intensity

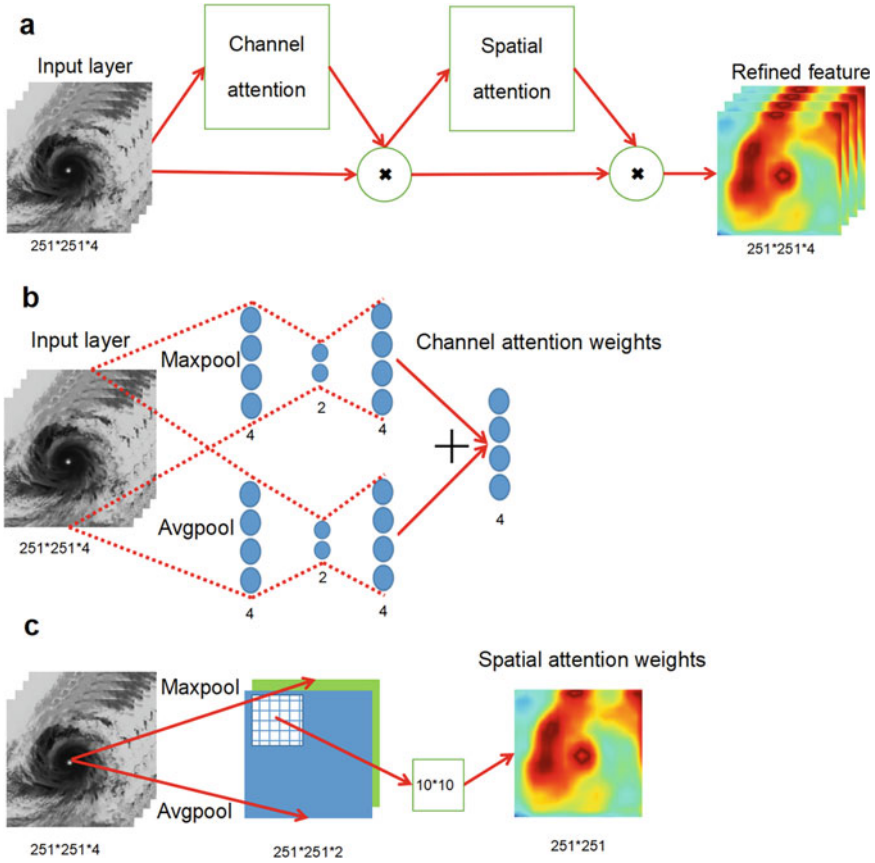


Fig. 8 Schematic diagram of spatial and channel attention layers in the CNN based TC intensity estimation model **a**, channel attention **b**, and spatial attention **c**. “+” and “×” are plus and multiply signs, respectively

estimation model (CNN-I 17) using the Focal_loss function is improved to 86.6%, and the RMSE is reduced by 2.1%.

In many domains, multi classification can be transformed into target recognition or binarization problem [18]. In this study, we further used eight binary models to replace the multi-classification model, and each binary model can learn the TC characteristics corresponding to each intensity category. Eight CNN based TC intensity estimation binary models were constructed to identify NC to H5 category, respectively.

The configuration of each model in Table 12 is the same as that of CNN-I 16. The Focal_loss with values of a_t and r listed in Table 11 was also used. We changed the classification label to “1” or “0”, which represents the intensity of whether a TC corresponds to a particular category. If the maximum sustained wind speed of a TC

Table 9 Confusion matrix of the CNN-I 16 multi-classification model

MODEL CATEGORY	ACTUAL CATEGORY								
	NC	TD	TS	H1	H2	H3	H4	H5	Total
NC	22	3	0	0	0	0	0	0	25
TD	11	349	29	1	0	0	0	0	390
TS	0	29	495	18	3	1	1	0	549
H1	0	4	27	145	14	2	2	0	194
H2	0	0	0	12	58	6	0	0	76
H3	0	1	1	2	8	48	4	0	64
H4	0	0	0	0	2	5	29	1	37
H5	0	0	0	0	0	0	0	3	3
Total	35	386	552	178	85	62	36	4	1338

Table 10 Classification report of the CNN-I 16 binary classification model

TC Category	P(%)	R(%)	F1(%)
NC	63	88	73
TD	90	89	89
TS	89	90	89
H1	81	74	78
H2	68	76	72
H3	77	75	76
H4	80	78	79
H5	75	100	85
Total	86	86	86

Table 11 Values of a_t and r in FOCAL_LOSS function

TC Category	NC	TD	TS	H1	H2	H3	H4	H5
r	2	2	2	2	2	2	2	2
a_t (%)	97	70	60	86	94	95	97	99.7

sample is 52.0 kt, which belongs to the TS category, the corresponding label of this image is "1" in the binary model which is responsible for judging the TS category, and "0" in the other binary models.

As shown in Table 12, the CNN-I 18 binary model has a much higher performance than that of the multi-classification model CNN-I 16. Compared with CNN-I 17, the accuracy of CNN-I 18 is improved to 88.9%, and the RMSE is reduced to 8.99 kt. The results show that the introduction of the Focal_Loss function and the transformation

Table 12 Results of the CNN based TC intensity estimation multi models with/without Focal_loss function and binary model

Model number	Method	Accuracy(%)	RMSE (kt)
CNN-I 16	CNN multi model	86.0	10.06
CNN-I 17	CNN multi model with Focal_loss	86.6	9.84
CNN-I 18	CNN binary model with Focal_loss	88.9	8.99

Table 13 Confusion matrix of the CNN-I 18 binary classification model

MODEL CATEGORY	ACTUAL CATEGORY								
	NC	TD	TS	H1	H2	H3	H4	H5	Total
NC	24	2	0	0	0	0	0	0	26
TD	11	354	24	2	0	0	0	0	391
TS	0	25	511	14	1	0	0	0	551
H1	0	2	14	152	12	3	0	0	183
H2	0	2	2	7	62	4	2	0	79
H3	0	1	1	3	9	52	4	0	70
H4	0	0	0	1	1	3	30	0	35
H5	0	0	0	0	0	0	0	4	4
Total	35	386	552	178	85	62	36	4	1338

of the multi-classification model into eight binary classification models helps to reduce the side effects caused by the imbalanced dataset.

The CNN-I 18 binary classification model’s confusion matrix and classification report are shown in Tables 13 and 14. The number of exact-hits for NC, TD, TS, H1, H2, H3, and H4 all rose when compared to the results from the multi-classification model (Tables 9 and 10). The precisions of H1-H4 classification have improved by 4.9%, 7.4%, 9.1%, and 3.8%, respectively.

Table15 compares the performance of the CNN model and other TC intensity estimation methods. The RMSE of the maximum wind speed estimated by the CNN based TC intensity estimation model proposed in this study is smaller than that of the DAVT technique or most CNN regression or classification models, which proves the potential of the CNN method in TC monitoring.

Table 14 Classification report of the CNN-I 18 binary classification model

TC Category	P(%)	R(%)	F1(%)
NC	69	92	79
TD	92	91	91
TS	93	93	93
H1	85	83	84
H2	73	78	76
H3	84	74	79
H4	83	86	85
H5	100	100	100
Total	89	89	89

Table 15 Comparison of our model and other methods

Literature	RMSE (kt)
Kossin et al. [25]	13.2
Ritchie et al. [39]	12.7
Fetanat et al. [13]	12.7
Pradhan et al. [36]	10.2
Chen et al. [3]	10.6
Tian et al. [45]	8.9
Our model	8.9

5 Summary

Accurately locating the TC center and estimating its intensity is an essential step for forecasters and emergency responders to make disaster warnings. In this chapter, a set of CNN-based model has been developed to automatically identify TC's center (CNN-L model) and intensity (CNN-I model) from H-8 geostationary satellite IR imagery, which can provide a reliable technical and information support for TC prediction and early warning systems.

Results show that the selection of satellite image channels has a significant impact on the performance of the TC intensity estimation model but hardly affects the TC center location model. Network parameters play an essential role in both models. The mean distance between the TC centers identified by the CNN-L model and by the Best Track dataset is 30 km for TCs in categories H1–H5. The accuracy of our CNN-L model is comparable to some techniques that locate a TC center based on its morphological features in IR images. Using four-channel (Channels 7, 8, 13, and 15) IR imagery, we found that the CNN-I 16 model has the best performance among the multi-classification models.

For TC categories with smaller training datasets, due to the unbalanced distributions of TC categories, the multi-classification model cannot produce a very

good result. By introducing the Focal_loss function in the CNN model and adopting eight binary classification networks, the side-effect of the unbalanced training data is reduced. In TC intensity estimate, the binary classification model CNN-I 18 gives a substantially lower RMSE (8.99 kt) of the maximum wind speed than the multi-classification model.

References

1. Bessho K, Date K, Hayashi M, Ikeda A, Imai T, Inoue H, Kumagai Y, Miyakawa T, Murata H, Ohno T et al (2016) An introduction to himawari-8/9-japan's new-generation geostationary meteorological satellites. *J Meteorol Soc Jpn Ser II* 94(2):151–183
2. Chaurasia S, Kishtawal C, Pal P (2010) An objective method of cyclone centre determination from geostationary satellite observations. *Int J Remote Sens* 31(9):2429–2440
3. Chen B, Chen BF, Lin HT (2018) Rotation-blended CNNs on a new open dataset for tropical cyclone image-to-intensity regression. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp 90–99
4. Chen BF, Chen B, Lin HT, Elsberry RL (2019) Estimating tropical cyclone intensity by satellite imagery utilizing convolutional neural networks. *Weather Forecast* 34(2):447–465
5. Chen G, Shen X, Yang Y (2010) β effect and vertical shear on typhoon asymmetrical structure and eyewall replacement. *Plateau Meteorol* 29(6):1474–1484
6. Combinido SJ, Mendoza RJ, Aborot J (2018) A convolutional neural network approach for estimating tropical cyclone intensity using satellite-based infrared images. *ICPR*, pp 1474–1480
7. DeMaria M, Sampson CR, Knaff JA, Musgrave KD (2014) Is tropical cyclone intensity guidance improving? *Bull Am Meteorol Soc* 95(3):387–398
8. Dvorak VF (1975) Tropical cyclone intensity analysis and forecasting from satellite imagery. *Mon Weather Rev* 103(5):420–430
9. Dvorak VF (1984) Tropical cyclone intensity analysis using satellite data, vol 11. US Department of Commerce, National Oceanic and Atmospheric Administration
10. Emanuel KA (1987) The dependence of hurricane intensity on climate. *Nat* 326(6112):483–485
11. Fang W, Shi X (2012) Review of stochastic simulation of tropical cyclone track and intensity for disaster risk assessment. *Prog Earth Sci* 27(8):866–875
12. Fangjie LXFHY, Zhibo H (1993) The technique for determining tropical cyclone intensity with enhanced satellite cloud imagery. *J Appl Meteorol Sci* 4(3):362–369
13. Fetanat G, Homaiifar A, Knapp KR (2013) Objective tropical cyclone intensity estimation using analogs of spatial features in satellite data. *Weather Forecast* 28(6):1446–1459
14. Fett RW, Brand S (1975) Tropical cyclone movement forecasts based on observations from satellites. *J Appl Meteorol Climatol* 14(4):452–465
15. Goodfellow I, Bengio Y, Courville A (2016) *Deep Learning*. MIT press
16. Goodfellow IJ, Shlens J, Szegedy C (2014) Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*
17. Hasler A, Palaniappan K, Kambhammetu C, Black P, Uhlhorn E, Chesters D (1998) High-resolution wind fields within the inner core and eye of a mature tropical cyclone from GOES 1-min images. *Bull Am Meteorol Soc* 79(11):2483–2496
18. Hsu CW, Lin CJ (2002) A comparison of methods for multiclass support vector machines. *IEEE Trans Neural Netw* 13(2):415–425
19. Hu T, Wu Y, Zheng G, Zhang D, Zhang Y, Li Y (2018) Tropical cyclone center automatic determination model based on HY-2 and QuikSCAT wind vector products. *IEEE Trans Geosci Remote Sens* 57(2):709–721

20. Hu Y, Song L, Liu A, Pan W (2008) Climatic characteristics of tropical cyclones landing in China in recent 58 years. *Journal of Sun Yat-sen University(Social Science Edition)* 47(5):115–121
21. Jaiswal N, Kishtawal CM (2010) Automatic determination of center of tropical cyclone in satellite-generated IR images. *IEEE Geosci Remote Sens Lett* 8(3):460–463
22. Jaiswal N, Kishtawal C, Pal P (2012) Cyclone intensity estimation using similarity of satellite IR images based on histogram matching approach. *Atmos Res* 118:215–221
23. Jin S, Wang S, Li X (2014) Typhoon eye extraction with an automatic SAR image segmentation method. *Int J Remote Sens* 35(11–12):3978–3993
24. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
25. Kossin J, Knapp K, Vimont D, Murnane R, Harper B (2007) A globally consistent reanalysis of hurricane variability and trends. *Geophys Res Lett* 34(4)
26. LeCun Y, Bengio Y, et al. (1995) Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks* 3361(10):1995
27. Lin TY, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp 2980–2988
28. Liu Z, Zou L, Wu B, Liu H (2003) Research on location algorithm of eye typhoon center in satellite cloud image. *Pattern Recognit Artif Intell* 3
29. Lu X, Yu H, Ying M, Zhao B, Zhang S, Lin L, Bai L, Wan R (2021) Western north pacific tropical cyclone database created by the china meteorological administration. *Adv Atmos Sci* 38(4):690–699
30. Olander TL, Velden CS (2007) The advanced dvorak technique: Continued development of an objective scheme to estimate tropical cyclone intensity using geostationary infrared satellite imagery. *Weather Forecast* 22(2):287–298
31. Ottenbacher A, Tomassini M, Holmlund K, Schmetz J (1997) Low-level cloud motion winds from meteosat high-resolution visible imagery. *Weather Forecast* 12(1):175–184
32. Pal P, Rao B, Kishtawal C, Narayanan M, Rajkumar G (1989) Cyclone track prediction using INSAT data. *Proc Indian Acad Sci-Earth Planet Sci* 98(4):353–364
33. Palaniappan K, Kambhamettu C, Hasler AF, Goldof DB (1995) Structure and semi-fluid motion analysis of stereoscopic satellite images for cloud tracking. In: *Proceedings of IEEE International Conference on Computer Vision*. IEEE, pp 659–665
34. Patadia F, Kishtawal C, Pal P, Joshi P (2004) Geolocation of indian ocean tropical cyclones using 85 GHz observations from TRMM microwave imager. *Curr Sci* pp 504–509
35. Pradhan R, Aygun RS, Maskey M, Ramachandran R, Cecil DJ (2017) Tropical cyclone intensity estimation using a deep convolutional neural network. *IEEE Trans Image Process* 27(2):692–702
36. Pradhan R, Aygun RS, Maskey M, Ramachandran R, Cecil DJ (2018) Tropical cyclone intensity estimation using a deep convolutional neural network. *IEEE Trans Image Process* (2):1–1
37. Rappaport EN, Jiing JG, Landsea CW, Murillo ST, Franklin JL (2012) The joint hurricane test bed: Its first decade of tropical cyclone research-to-operations activities reviewed. *Bull Am Meteorol Soc* 93(3):371–380
38. Rauber J, Brendel W, Bethge M (2017) Foolbox v0. 8.0: A python toolbox to benchmark the robustness of machine learning models. arXiv preprint [arXiv:1707.04131](https://arxiv.org/abs/1707.04131) 5
39. Ritchie EA, Wood KM, Rodríguez-Herrera OG, Piñeros MF, Tyo JS (2014) Satellite-derived tropical cyclone intensity in the North Pacific Ocean using the deviation-angle variance technique. *Weather Forecast* 29(3):505–516
40. Schmetz J, Holmlund K, Hoffman J, Strauss B, Mason B, Gaertner V, Koch A, Van De Berg L (1993) Operational cloud-motion winds from meteosat infrared images. *J Appl Meteorol Climatol* 32(7):1206–1225
41. sieron bs, zhang f, emanuel ak (2013) Feasibility of tropical cyclone intensity estimation using satellite-borne radiometer measurements: An observing system simulation experiment. *Geophys Res Lett*. 5332–5336

42. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15(1):1929–1958
43. Sun Y, Wang X, Tang X (2014) Deep learning face representation from predicting 10,000 classes. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp 1891–1898
44. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R (2013) Intriguing properties of neural networks. arXiv preprint [arXiv:1312.6199](https://arxiv.org/abs/1312.6199)
45. Tian W, Huang W, Yi L, Wu L, Wang C (2020) A CNN-based hybrid model for tropical cyclone intensity estimation in meteorological industry. *IEEE Access* 8:59158–59168
46. Velden C, Olander T, Herndon D, Kossin JP (2017) Reprocessing the most intense historical tropical cyclones in the satellite era using the advanced Dvorak technique. *Mon Weather Rev* 145(3):971–983
47. Velden CS, Olander TL, Zehr RM (1998) Development of an objective scheme to estimate tropical cyclone intensity from digital geostationary satellite infrared imagery. *Weather Forecast* 13(1):172–186
48. Wang G, Zhao B, Zhao C (2018) Study on the relationship between super typhoon intensity and upper ocean thermal structure in transit area. *Adv Mar Sci* 3
49. Wong V, Emanuel K (2007) Use of cloud radars and radiometers for tropical cyclone intensity estimation. *Geophys Res Lett* 34(12)
50. Xu J, Jiang Y, Wei F, Zhang G (2008) Analysis on the susceptibility of geological disasters in typhoon affected area of landfall in china. *Chin J Geol Hazard Control* 19(4):61–66
51. Xu Q, Zhang G, Li X, Cheng Y (2016) An automatic method for tropical cyclone center determination from SAR. In: *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, pp 2250–2252
52. Yan WK, Lap YC, Wah LP, Wan TW (2004) Automatic template matching method for tropical cyclone eye fix. In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., vol 3*. IEEE, pp 650–653
53. Ying M, Zhang W, Yu H, Lu X, Feng J, Fan Y, Zhu Y, Chen D (2014) An overview of the china meteorological administration tropical cyclone database. *J Atmos Ocean Technol* 31(2):287–301
54. Zhai AR, Jiang JH (2014) Dependence of US hurricane economic loss on maximum wind speed and storm size. *Environ Res Lett* 9(6):064019
55. Zhang W, Cui X (2013) A review of research on tropical cyclone formation. *J Trop Meteorol* 29(2):337–346
56. Zheng G, Yang J, Liu AK, Li X, Pichel WG, He S (2015) Comparison of typhoon centers from SAR and IR images and those from best track data sets. *IEEE Trans Geosci Remote Sens* 54(2):1000–1012
57. Zheng G, Liu J, Yang J, Li X (2019) Automatically locate tropical cyclone centers using top cloud motion data derived from geostationary satellite images. *IEEE Trans Geosci Remote Sens* 57(12):10175–10190

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits any noncommercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if you modified the licensed material. You do not have permission under this license to share adapted material derived from this chapter or parts of it.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Reconstruction of $p\text{CO}_2$ Data in the Southern Ocean Based on Feedforward Neural Network



Yanjun Wang, Xiaofeng Li, Jinming Song, Xuegang Li, Guorong Zhong, and Bin Zhang

1 Introduction

The ocean plays a vital role in regulating global climate change, About ~30% of total emissions since the pre-industrial period has been stored in the ocean, However, about 50% of the oceanic uptake of anthropogenic carbon takes place in the Southern Ocean. It dominates the global heat and carbon dioxide absorption, therefore, many scientists regard the Southern Ocean as the main research region. The “Southern Ocean” ($< 35^\circ\text{S}$) was proposed by scientists around 2000 and was determined to be the fifth largest ocean in the world. It is the only ocean that completely surrounds the earth but is not divided by continents. It has important differences from ocean currents in the Pacific, Indian and Atlantic oceans—Antarctic Circumpolar Current (ACC). Moreover, the Southern Ocean is also an important region for global carbon absorption and release. Before industrial time, due to the influence of upwelling in the Southern Ocean, it has become a major carbon source region [6]. With the influence of human activities, the atmospheric pressure gradient shifted and turned into a carbon sink region. In the following section, We use the SOCAT dataset to build a Feedfoward neural network (FFNN), based on this network we reconstruct the Southern Ocean $p\text{CO}_2$ data and calculate the CO_2 flux changes in the region, compare with other method, Our algorithm is compared with two neural network algorithms and has a smaller root mean square error.

Y. Wang · B. Zhang

Department of Marine Science Data Center, Institute of Oceanology,
Chinese Academy of Sciences, Qingdao 266071, China

X. Li (✉)

CAS Key Laboratory of Ocean Circulation and Waves, Institute of Oceanology,
Chinese Academy of Sciences, Qingdao 266071, China
e-mail: lixf@qdio.ac.cn

J. Song · X. Li · G. Zhong

Key Laboratory of Marine Ecology and Environmental Sciences, Institute of Oceanology,
and the Center for Ocean Mega-Science, Chinese Academy of Sciences, Qingdao 266071, China

© The Author(s) 2023

X. Li and F. Wang (eds.), *Artificial Intelligence Oceanography*,
https://doi.org/10.1007/978-981-19-6375-9_9

1.1 Observations of $p\text{CO}_2$ in Southern Ocean

Many data of the carbonate system can only be obtained by in-situ measurement. Due to the harsh environment of the Southern Ocean, the data collection is lacking. For sea surface data, through the continuous efforts of the scientists, the Surface Ocean CO_2 Atlas [13] has complies and quality control of ship data, fixed-point observation data, and drifting buoy data to formed a relatively complete observation data set (Fig. 1). This data set contains the $p\text{CO}_2$ data which can be used to calculate the sea-air carbon dioxide flux. We will use this database as the truth value to construct our neural network and reconstruct the $p\text{CO}_2$ gridded data of the entire Southern Ocean.

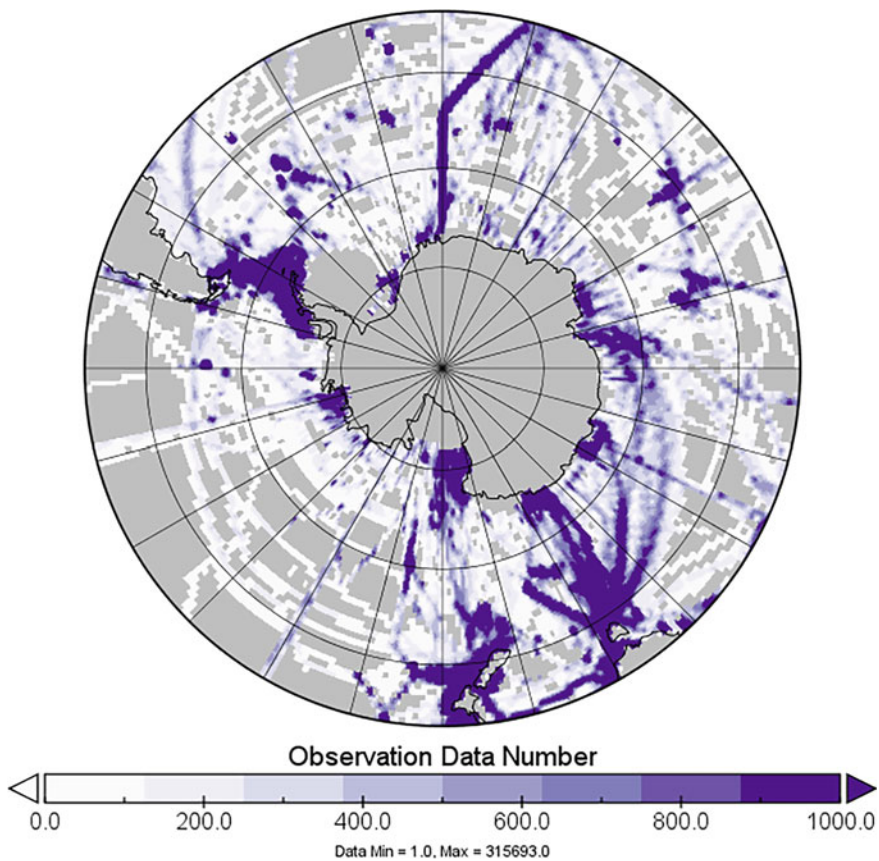


Fig. 1 1998-2018 SOCAT data observation heat map

1.2 Comparison of Reconstruction $p\text{CO}_2$ Data

The results obtained by some traditional atmospheric inversions algorithms are greatly affected by the amount of observational data [17, 20]. Some spatial and temporal interpolations are based on empirical relationships between carbon dioxide and alternative variables, and are mainly concentrated in areas with relatively rich observations.

Neural network approaches have been frequently used in the reconstruction of surface $p\text{CO}_2$ in recent years. To recreate the $p\text{CO}_2$ data of the Southern Ocean, Gregor et al. employed a support vector machine (SVM) and a random forest (RF). The root-mean-square errors (RMSEs) were $16.45 \mu\text{atm}$ and $24.04 \mu\text{atm}$, respectively. Meanwhile, Landschutzer et al. [11] created the SOM-FFNN method by combining a self-organizing map (SOM) with a feedforward neural network (FFNN) to recreate $p\text{CO}_2$ data from the Southern Ocean. Sea surface temperature (SST), sea surface salinity (SSS), Mixed Layer Depth (MLD), chlorophyll concentration (CHL), and other metrics are used as inputs. The study shows that during the period 1980-2000, the Southern Ocean carbon sink has remained stagnant or even weakened, and continued to increase after 2002. Both data products showed good interannual and seasonal cyclical changes, but compare with the traditional machine learning algorithm (SVM and RF), SOM-FFNN show better performance. Denvil-Sommer et al. [3] employed the Laboratory of Climate and Environmental Sciences (LSCE)-FFNN method to reconstruct global $p\text{CO}_2$ data, which maintained consistency with observational results. However, compared with the observed data, the Southern Ocean's reconstructed data has a larger error than other regions with more in situ observations.

In this chapter, we use the Surface Ocean CO_2 ATLAS (SOCAT V.6) data from 1998 to 2018 in the Southern Ocean, we applied the (CA)-FFNN method to reconstruct the monthly and $1^\circ \times 1^\circ$ $p\text{CO}_2$ data of the Southern Ocean. Due to FFNN produces more stable data in sparse areas [20], and interpolates the data with small deviation [12], we use this method to reconstruct the Southern Ocean regional data. The procedure is separated into two parts. First, each parameter's correlation index is calculated and arranged. Second, the $p\text{CO}_2$ data in the southern ocean blank area was interpolated using a relational model employing parameters with reasonably strong correlation coefficients as input variables of the FFNN. The current scenario, in which stations with less observation data have larger RMSE values, is improved by this strategy. As a result, this method might be used to recreate regional data. Finally, we looked at $p\text{CO}_2$ fluctuations in the Southern Ocean on a seasonal, interannual, and interdecadal scale.

2 Data and Methods

2.1 Data

The parameters used in the CA method included SST, SST anomaly (SSTA), SSS, and SSS anomaly (SSSA); these parameters were all from the gridded dataset of Global Ocean Heat Content Change [2], while anomaly data were obtained by subtracting the average data values from the climatic state data of each month. Chlorophyll concentration (Chl-a) were based on satellite remote sensing data from the European Space Agency's Global Color Project, while MLD data were obtained from the French Institute of Marine Development. The u- and v- components of the wind field at 10 meters above sea level (a.s.l.) were taken from the European Centre for Medium-Range Weather Forecasts. All these data except MLD are monthly averages over a $1^\circ \times 1^\circ$ Lat/Lon box. MLD data is monthly averages over $0.5^\circ \times 0.5^\circ$.

In this chapter, we convert the $f\text{CO}_2$ data in the SOCAT data set to $p\text{CO}_2$ data as the training set and test set of FFNN. Transformation relationship between $f\text{CO}_2$ and $p\text{CO}_2$ is as follows [10]:

$$f\text{CO}_2 = p\text{CO}_2 \cdot \exp\left(p \cdot \frac{B + 2\delta}{R \times T_{\text{subskin}}}\right) \quad (1)$$

where p is the atmospheric pressure (Pa), R is the gas constant ($8.314 \text{ J K}^{-1} \text{ mol}^{-1}$), SST is the sea surface temperature (K), T_{subskin} is the subskin temperature and B and δ are the correction coefficients, which are calculated as:

$$T_{\text{subskin}} = SST + 0.17 \quad (2)$$

$$B \left(\frac{\text{m}^3}{\text{mol}}\right) = (-1636.75 + 12.0408SST - 3.27957 \times 10^{-2}SST^2 + 3.16528 \times 10^{-5}SST^3) \times 10^{-6} \quad (3)$$

$$\delta \left(\frac{\text{m}^3}{\text{mol}}\right) = (57.7 - 0.118T_{\text{subskin}}) \times 10^{-6} \quad (4)$$

The partial pressure of atmospheric CO_2 was calculated by the following formula [14]:

$$p\text{CO}_{2a} = x\text{CO}_2 [P_{\text{eq}} - \text{VP}(\text{H}_2\text{O})] \quad (5)$$

where $x\text{CO}_2$ is the dry air mixing ratio of atmospheric CO_2 . The relevant data are collected from the reference data of marine boundary layer in the Earth System Research Laboratory of the National Oceanic and Atmospheric Administration (NOAA). Additionally, P_{eq} is the pressure at equilibrium, and $\text{VP}(\text{H}_2\text{O})$ is the steam of seawater at a given temperature [8]

$$\text{VP} = 0.61121 \times e^{\left(18.678 - \frac{T_{\text{subskin}}}{234.5}\right) \times \frac{T_{\text{subskin}}}{257.14 + T_{\text{subskin}}}} \quad (6)$$

where the T_{subskin} is subskin temperature.

In order to reduce the complexity of calculation of too large data set on neural network learning, we use Eq. 7 to normalize all data.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (7)$$

where x is actual value, $\min(x)$ is the minimum value of x , $\max(x)$ is the maximum value of x .

Since the Chl-a data in this study did not include relevant records before the launch of SeaWiFS in 1997, our research period was from 1998 to 2018. The spatial resolution of all parameter data was $1^\circ \times 1^\circ$. Longitude (Lon) and latitude (Lat) are in 360° and 180° coordinate systems, and trigonometric conversion functions were used to ensure continuity and normalization.

2.2 Nonlinear Neural Network Model for the $p\text{CO}_2$ Reconstruction in the Southern Ocean

We use Equations 8 and 9 to calculate the correlation coefficient, and build a covariance matrix between $p\text{CO}_2$ and other collected data, as shown in Fig. 3.

$$\text{Cov}(X, Y) = E[(X - u_x)(Y - u_y)] \quad (8)$$

$$\rho = \frac{\text{Cov}(X, Y)}{\beta_x \beta_y} \quad (9)$$

where u is the mean of the value, β is standard deviation of the value, $\text{Cov}(X, Y)$ is the calculated covariance matrix, and ρ is the correlation coefficient.

We use the parameters with correlation coefficients > 0.1 as the input parameters, considering the relevance of chemical effects between SST and $p\text{CO}_2$ [18], We still use SST as an input parameter. After correlation analysis, the selected parameters were the SST, SSSA, MLD, CHL, the u-component (U) of the sea surface wind field, and the partial pressure of atmospheric CO_2 ($p\text{CO}_{2a}$). The established correlation equations between $p\text{CO}_2$ and the main parameters are summarized in Eq. 10.

$$p\text{CO}_2 = f\left(\begin{matrix} \text{SST, SSSA, CHL, MLD,} \\ U, a\text{CO}_2, \text{Lon, Lat} \end{matrix}\right) \quad (10)$$

A nonlinear regression model was built using the FFNN. Although an FFNN's output data improves and becomes more accurate as the number of layers and neurons

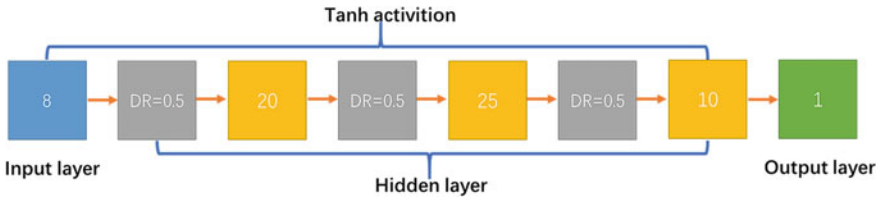


Fig. 2 The Structure of our FFNN, The gray square is the dropout layer and dropout rate is 0.5, blue square is input layer, yellow square is hidden layer, green square is output layer

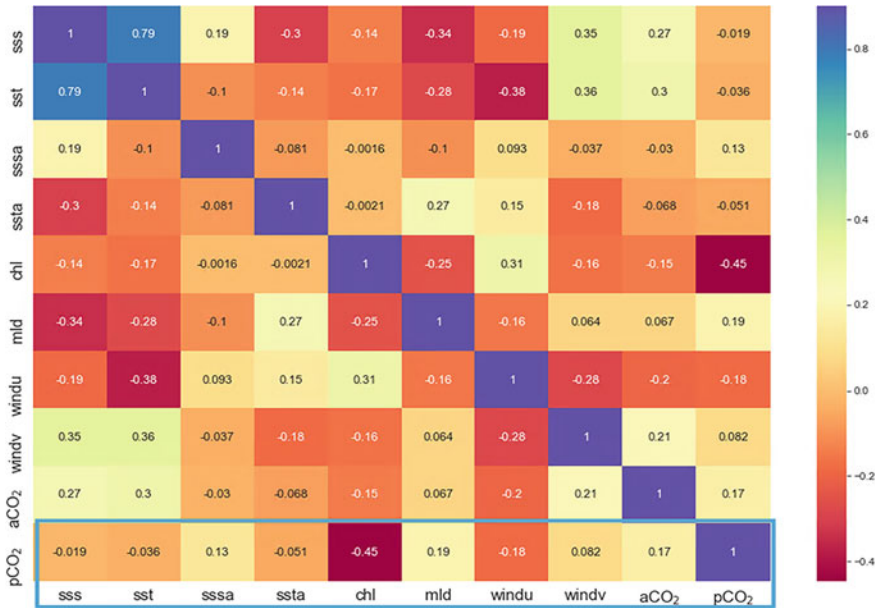


Fig. 3 Matrix of correlation coefficients. The correlation coefficient value of the x-axis and y-axis parameters is represented by each colored box. The value of the pCO₂ correlation coefficient with other parameters is contained within the blue box

in the FFNN grows, the model’s size is also determined by the amount of data utilized for model training. Because there is less observational data for the Southern Ocean than for other regions, we built a simple FFNN structure, the neural network structure of which is shown in Fig. 2. The final model at Step 2 has eight layers (six hidden layers), and the numbers on the figure represent the size of the tensor input to each layer. A gray square represents the dropout layer, and the dropout rate is 0.5. The hyperparameters of the neural network were determined using k-fold cross-validation (Fig. 4).

The data were divided into 75%/25% portions used for training/testing sets. The neural network consists of eight layers, and the middle layer had six completely connected hidden layers, we added three dropout layers and gave each layer’s dropout ratio 0.5 to prevent the FFNN from overfitting. Through many tests and detailed

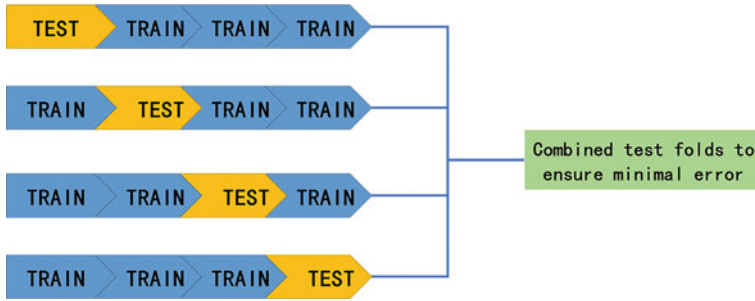


Fig. 4 k -Fold cross-validation, which was divided into four folds in this study, with 25% data for testing and the rest for training to create the best neural network. The yellow shape represents test data, whereas the blue shape represents train data

analyses, the hyperbolic tangent (Tanh) was selected as the activation function of the neuron, and the using the mean squared error (MSE) as the loss function:

$$MSE = \frac{1}{N} \sum_{i=1}^N (\text{observed}_i - \text{predicted}_i)^2 \quad (11)$$

where observed_i is the observation data, and predicted_i is the data predicted by the FFNN model, and we using RMSProp as the optimization function [21].

In order to control the amount of information, we adjusted the adaptive learning rate. The CA-FFNN was then formed by combining a main factor analysis and based on the parameters, we build a FFNN structure and get a nonlinear regression model through training.

2.3 Calculation of Carbon Dioxide Flux in the Southern Ocean

The formula for calculating the carbon dioxide flux at the air-sea interface is [29] :

$$F = K \cdot \Delta f\text{CO}_2 = K \cdot (a_{\text{subskin}} f\text{CO}_{2w} - a_{\text{skin}} f\text{CO}_{2a}) \quad (12)$$

where a is the solubility of CO_2 in seawater ($\text{mol kg}^{-1} \text{atm}^{-1}$), calculated by Weiss [10]:

$$\ln a = -60.2409 + 93.4517 \left(\frac{100}{T_{\text{subskin}}} \right) - 23.3585 \times \ln \left(\frac{T_{\text{subskin}}}{100} \right) - S \times \left[0.023517 - 0.023656 \times \left(\frac{T_{\text{subskin}}}{100} \right) + 0.0047036 \times \left(\frac{T_{\text{subskin}}}{100} \right)^2 \right] \quad (13)$$

In Equation 12, $a_{subskin}$ is calculated by the subskin temperature, a_{skin} is calculated by the skin temperature. fCO_{2w} is the fugacity of subskin seawater CO_2 , fCO_{2a} is the fugacity of atmospheric CO_2 , and K is the exchange rate, which is usually considered as a function of wind speed .

$$K = \Gamma(660/Sc)^{0.5}U^2 \quad (14)$$

Here, Sc is the Schmidt number of CO_2 in seawater at a given $T_{subskin}$ temperature, such that:

$$Sc = 2073.1 - 125.62 \times T_{subskin} + 3.6276 \times T_{subskin}^2 - 0.043219 \times T_{subskin}^3 \quad (15)$$

where U is the monthly mean wind speed (m/s) at 10 m height from the cross-calibrated multi-platform ocean surface wind vector analysis product and Γ is the scale factor which was evaluated based on different wind speed products (e.g., 0.39, 0.251, 0.31, etc.) and have been used in other studies [14, 24, 28]. Based on an average wind speed of 6.38 m s^{-1} in the ECMWF product the scale factor of 0.31 was used to reach a global mean transfer velocity of 16 cm h^{-1} , consistent with the new radiocarbon-based constraints.

2.4 Evaluation

Due to the limited observation data in the Southern Ocean, the data set used for verification will be very small, so the segmentation of the data set will lead to huge differences between RMSE and mean absolute error. In order to ensure reliable model verification, we used 100% data to train, test and verify the model, and continuously optimized the neural network model and the internal weight. Finally, the neural network was used to predict the observed area. RMSE is calculated to be $8.86 \mu\text{atm}$, while MAE is 5.01.

Figure 5 shows that the predicted values are very close to the observed values and $R^2 = 0.93$. In Table 1, we list the RMSE and MAE between the results of different algorithms and the actual values. SOM-FFNN merged a self-organizing map (SOM) and feedforward neural network, and the RMSE is 12.24. LSCE-FFNN employed the Laboratory of Climate and Environmental Sciences, and the RMSE is 17.40. We conclude that the CA-FFNN-based models outperform both the SOM-FFNN and LSCE-FFNN.

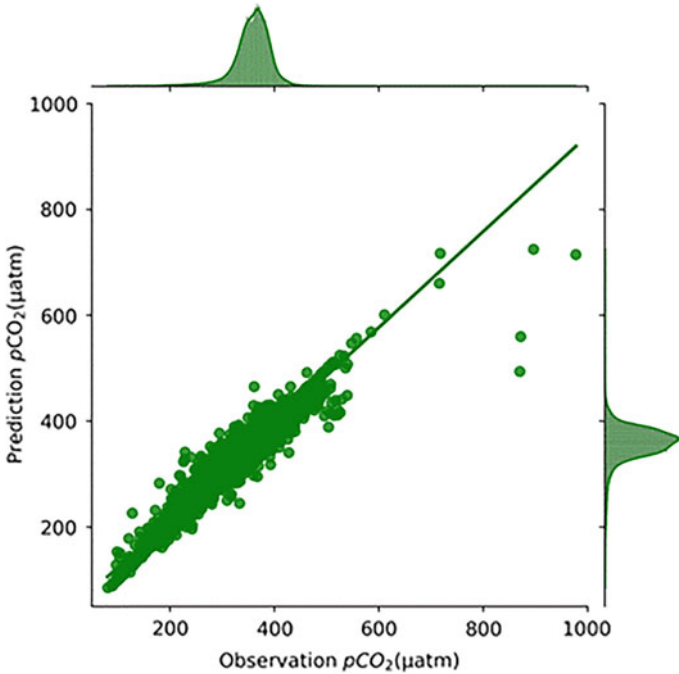


Fig. 5 Scatter fit of product data and observation data with same station

Table 1 Comparison of our Algorithms' Errors to LSCE-FFNN and SOM-FFNN

Artificial Intelligence Algorithm	RMSE	MAE
FFNN for Southern Ocean	8.86	5.01
LSCE-FFNN [3]	17.40	11.92
SOM-FFNN [12]	12.24	7.36

3 Results and Discussion

3.1 Seasonal Variation in Southern Ocean Sea Surface $p\text{CO}_2$

According to the new dataset, the $p\text{CO}_2$ data changes periodically with the seasons. This result is consistent with the seasonal changes in other studies [16, 25, 27]. The seasonal mean amplitude of ocean surface $p\text{CO}_2$ in the southern ocean was 13.02 μatm and our data has similar seasonal variation characteristics compared with the observational data of the Southern Ocean [15], the $p\text{CO}_2$ reaching its minimum in summer, and increase in winter (Fig. 6), and driven by both biological and physical factors, $p\text{CO}_2$ in the Southern Ocean shows obvious seasonal changes [22]. In winter,

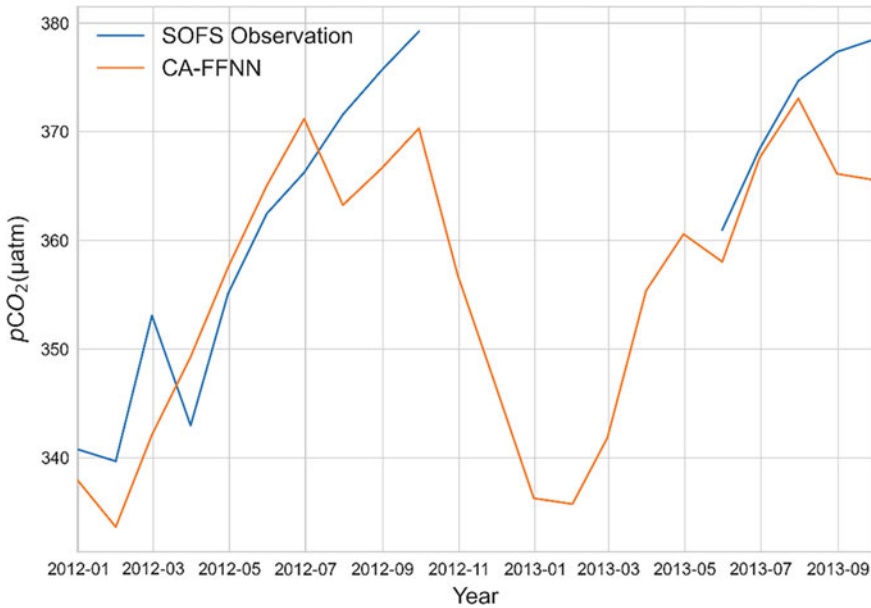


Fig. 6 In SOFS, the product data and real value

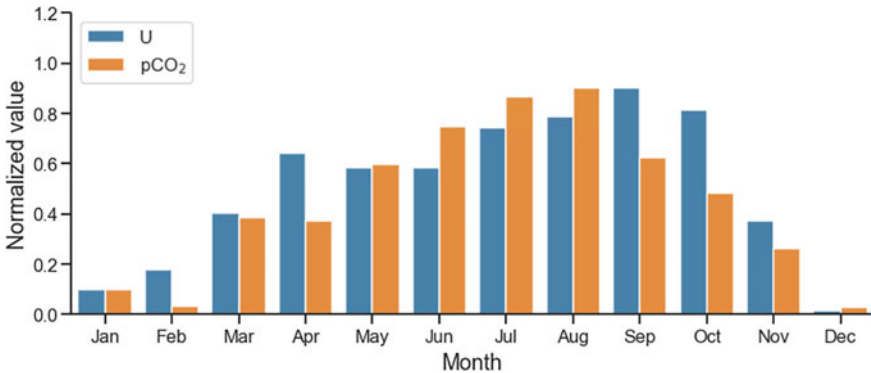


Fig. 7 From 1998 to 2018, the normalized mean monthly U-component of wind and pCO_2 was calculated

due to the enhancement of the wind field in the Southern Ocean, as shown in the Fig. 7, the Ekman transport caused by the wind field also intensifies [1, 7], strengthening upwelling and improving the efficiency of the biological pump.

The dissolved inorganic carbon in the bottom layer migrates to the surface layer under the influence of the upwelling, making the surface pCO_2 increase continuously. With the melting of sea ice in the Southern Ocean in summer, marine primary productivity gradually recovers, the Chl-a concentration increases, as shown in Fig. 8,

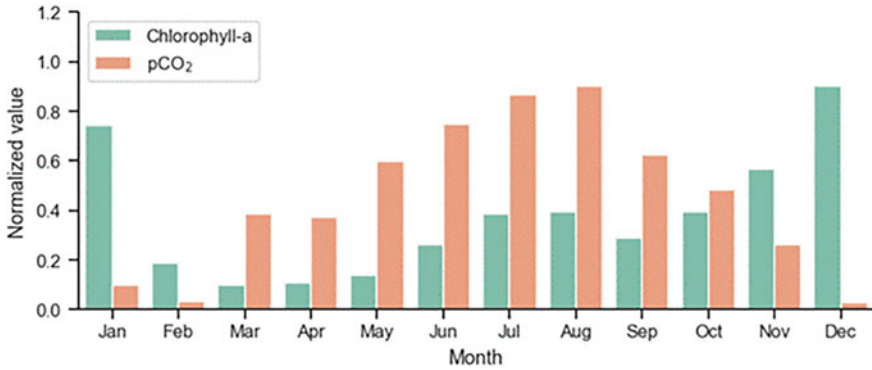


Fig. 8 From 1998 to 2018, the average monthly CHL and $p\text{CO}_2$ data were normalized

and CO_2 in sea water is absorbed through photosynthesis [26], which lead to surface $p\text{CO}_2$ decrease. This period is mainly due to biological factors.

3.2 Annual Variation in Southern Ocean Sea Surface $p\text{CO}_2$

Analyzing the inter-annual change of the reconstructed $p\text{CO}_2$ data from 1998 to 2018, the mean surface $p\text{CO}_2$ of the Southern Ocean increased from $351.88 \mu\text{atm}$ to $372.65 \mu\text{atm}$ —a total increase of $20.77 \mu\text{atm}$ in 21 years and an annual mean increase of $0.99 \mu\text{atm/yr}$. As shown in Fig. 9, the Southern Ocean $p\text{CO}_2$ has maintained a high growth rate.

By calculating the linear rate of change in the Southern Ocean spatial region over a 21-year period, it is found that the $p\text{CO}_2$ in most areas is gradually increasing, as shown in Fig. 10. The growth rate around 35° to 55° is faster than other regions.

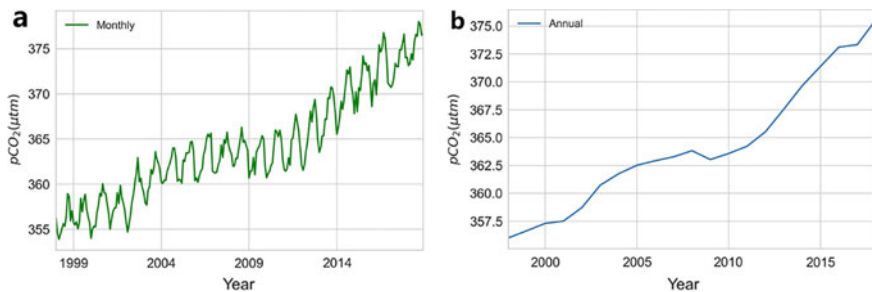


Fig. 9 **a** is monthly fluctuations in the Southern Ocean’s $p\text{CO}_2$ (atm) from 1998 to 2018; **b** is yearly fluctuations in the Southern Ocean’s $p\text{CO}_2$ (atm) from 1998 to 2018

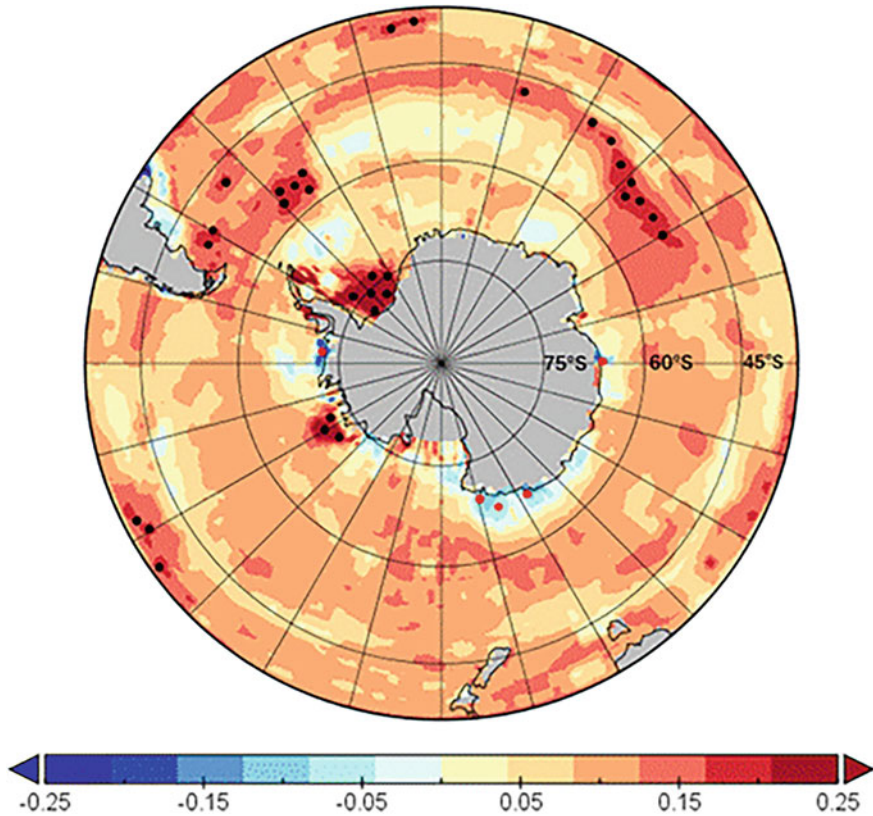


Fig. 10 The rate of change in the surface Southern Ocean's $p\text{CO}_2$ ($\mu\text{atm yr}^{-1}$) concentration

Since 2002, many study results have shown that $p\text{CO}_2$ in the Southern Ocean has maintained a high growth rate [23], and our data also shows this trend.

3.3 Variability in Sea—Air CO_2 Flux

As for the rate of change of $\Delta fC\text{O}_2$, Most of the Southern Ocean is transforming into a carbon sink area. The black/red dots in Fig. 11 represent $\Delta fC\text{O}_2$ regions toward positive/negative trends with high change rate. According to the distribution of $p\text{CO}_2$ in the Southern Ocean since 1998, the status of inner ring ($50 - 70^\circ\text{S}$) as a carbon source is changing, while the outer ring ($35 - 50^\circ\text{S}$) has always maintained a strong carbon sink state, and there is no tendency to weaken. The changes of CO_2 flux in the Southern Ocean calculated by our model are consistency with other models for the evolution of intensity [19]. Using Eq. 12 to calculate the CO_2 flux, the Southern

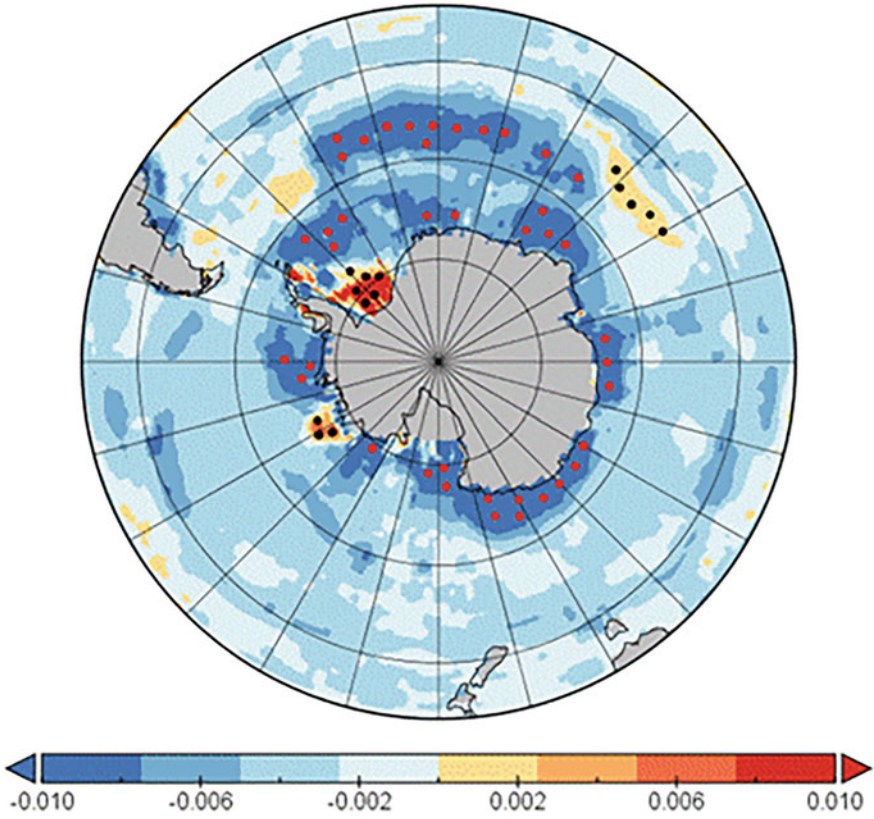


Fig. 11 Carbon sink when $\Delta f\text{CO}_2 < 0$, carbon source when $\Delta f\text{CO}_2 > 0$, Rate of change in the $\Delta f\text{CO}_2$ of the Southern Ocean

Ocean’s CO_2 flux was found to have changed substantially over the past two decades. The $\Delta f\text{CO}_2$ in the Southern Ocean also changes regularly with the seasons, with the strongest in early summer and get the weakest at the end of winter (Fig. 13). Many studies have shown that in early 1990s, the Southern Ocean was saturated with carbon and regained its vitality at the beginning of the 21st century [4]. The data products reproduces the strong increase of carbon sinks in the Southern Ocean since the 21st century (Fig. 14).

In terms of interannual changes, the carbon sink of the Southern Ocean increased from $-0.21 \text{ Pg C yr}^{-1}$ in 1998 to $-1.67 \text{ Pg C yr}^{-1}$ in 2018.

One standard deviation was used as an indicator of error:

$$\sigma_n = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n^2}} \tag{16}$$

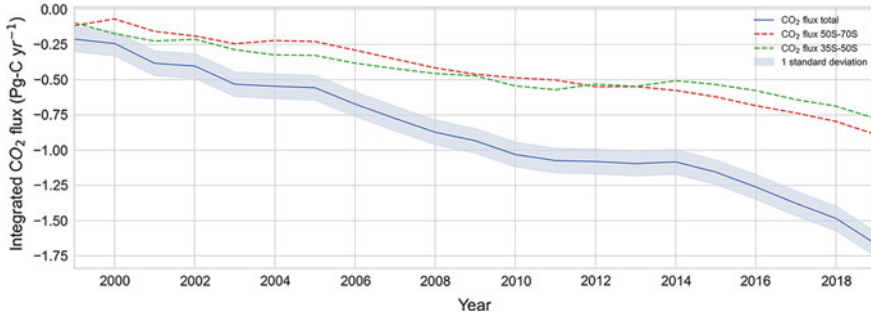


Fig. 12 CO₂ flux trends in of the Southern Ocean from 1998 to 2018

where x_i is the actual value, \bar{x} is the mean value of x , n is number of data, and the error range was within ± 0.0087 Pg C yr⁻¹.

We found that the carbon sinks in the Southern Ocean did not always maintain a trend of rapid growth. During 2010-2013, the carbon sinks stagnated. As shown in Fig. 12, we found the similar phenomenon in many other reconstructed data [5]. Many studies have shown that changes in the Southern Annular Mode (SAM) led to the stagnation of carbon sinks in the 1990s [5]. However, the stagnation was not strongly correlated with the SAM. Stability during this period was mainly due to the weakening of the carbon sink intensity from 35 – 50°S. Changes in this region have also been attributed to the barometric asymmetry of the Zontal Waves 3 (ZW3) model [9]. As for models that rely on observational data, it is difficult to capture such large and subtle inter-annual changes.

As shown in Fig. 16, there is an obvious double-ring structure before 2010, which is not always a carbon sink. The inner ring (50 – 70°S), change with the seasons. In April, May, June, July, August, and September, the region serves as a carbon source, emitting CO₂ into the atmosphere. In October, November, December, January, February, and March, it absorbs CO₂, as shown in Fig. 14. The outer ring (35 – 50°S) is the main carbon sink region (Fig. 15), and undertakes most CO₂ absorption. From the perspective of the inter-annual changes in the entire region, the Southern Ocean carbon dioxide flux changes to carbon sinks.

However, with the increase of carbon sink in the outer ring and the weakening of the carbon source in the inner ring, after 2010 this ring structure is gradually disappearing. As shown in Fig. 16, most Southern Ocean regions become carbon sinking regions, because the ΔfCO_2 in the Southern Ocean decrease significantly since 1998.

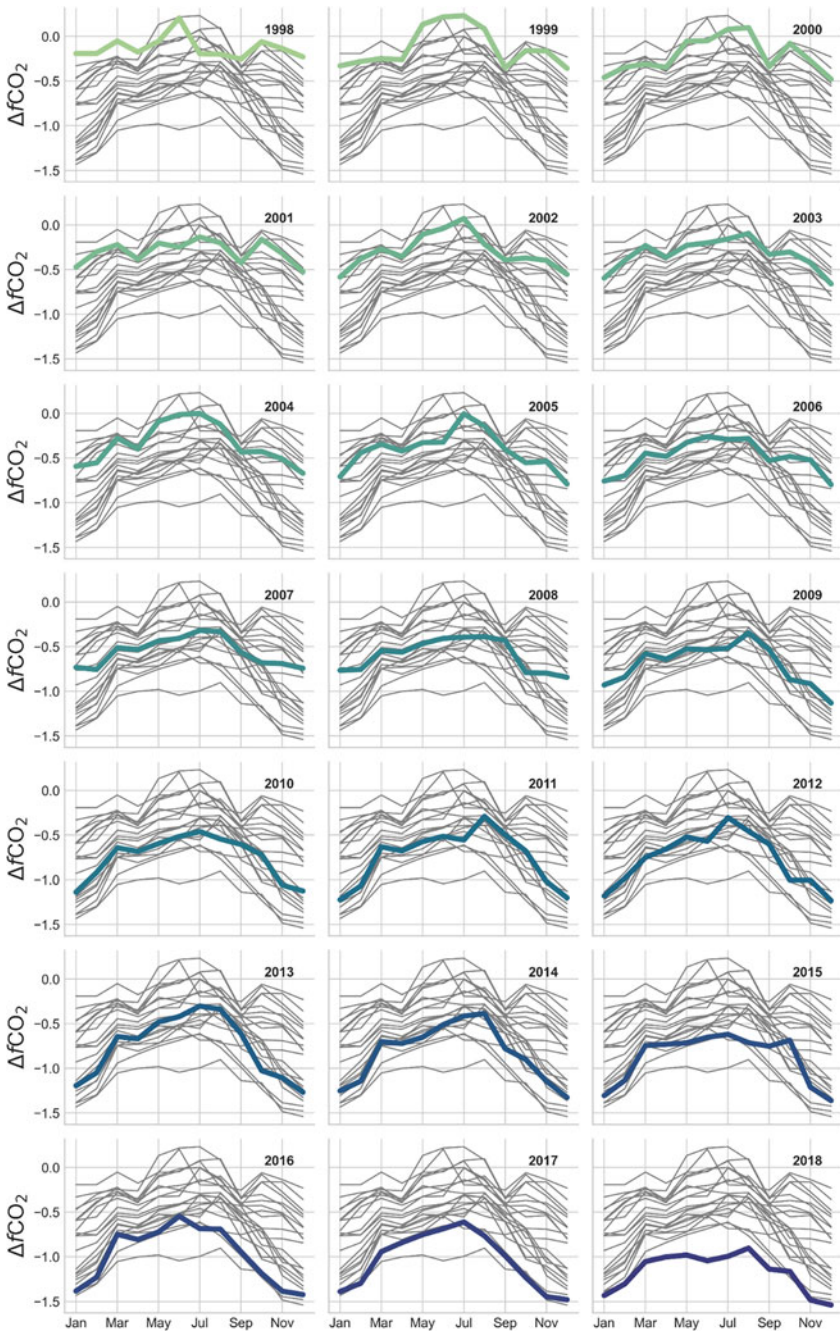


Fig. 13 Changes in $\Delta f\text{CO}_2$ values by month and year from 1998 to 2018 (μatm). The gray lines indicate fluctuations in previous years, whereas the colorful lines represent variations in the year under consideration

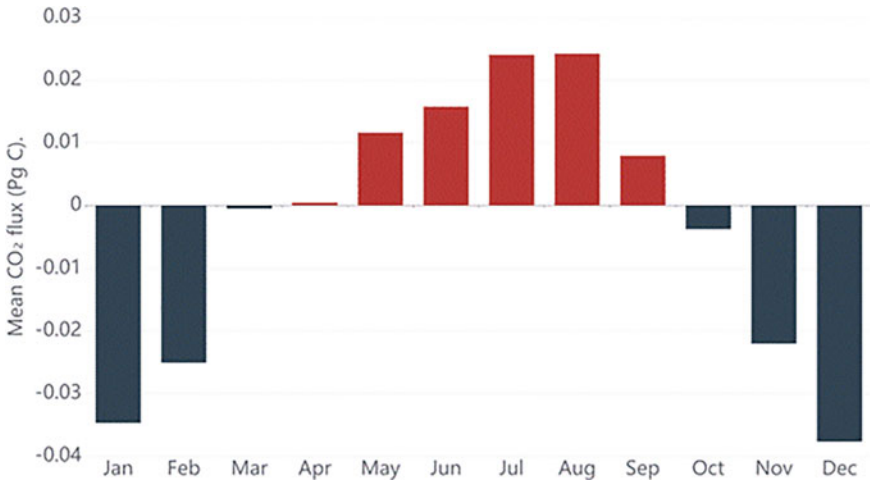


Fig. 14 Each month's average CO₂ flux in 50 – 70°S (Pg C). The Southern Ocean was a carbon supply in the red columns, whereas the Southern Ocean was a carbon sink in the black columns

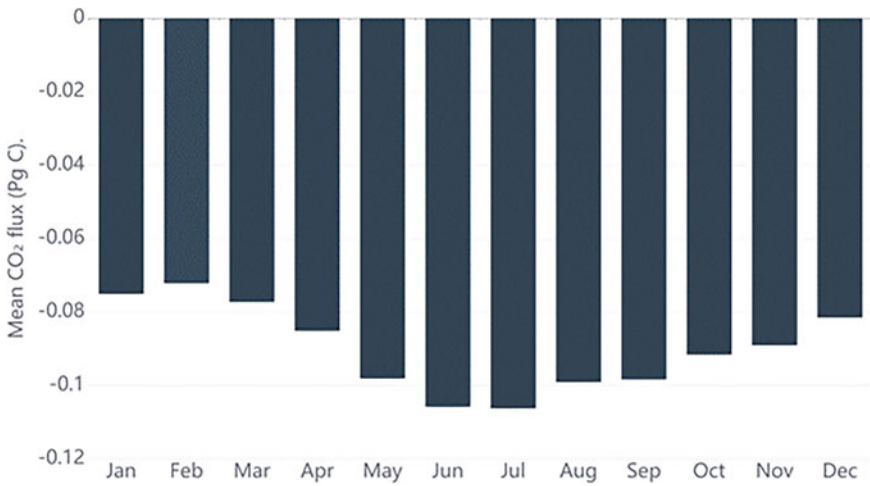


Fig. 15 Each month's average CO₂ flux in 35 – 50°S (Pg C). The Southern Ocean was a carbon sink, as indicated by the blue columns

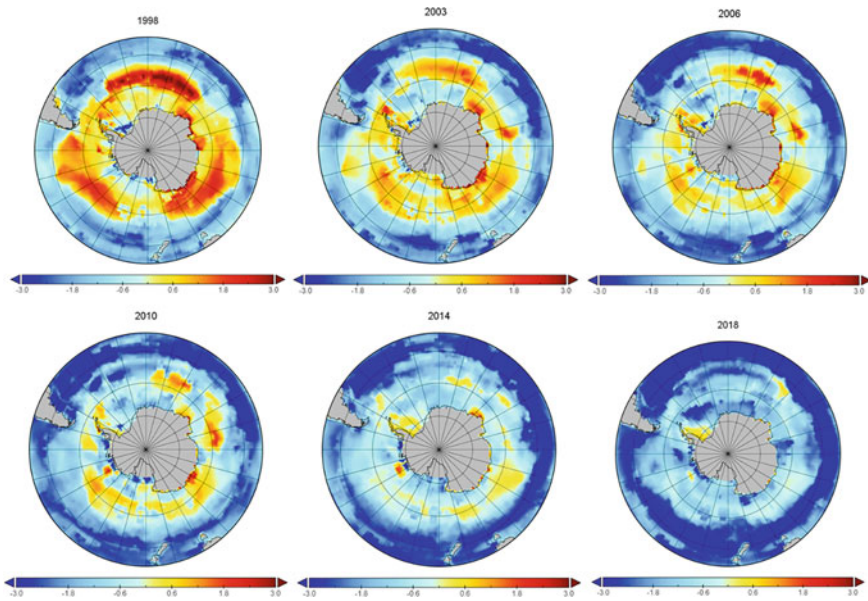


Fig. 16 In the Southern Ocean, mean sea surface CO_2 fluxes (Pg C) were measured in 1998, 2003, 2006, 2010, 2014, and 2018

4 Conclusion

In this chapter, we propose a feedforward neural network for reconstructing $p\text{CO}_2$ data in the Southern Ocean that is generalizable for reconstructing regional data. The reconstruction process consists of two steps. First, we collect all parameters that may have impact on $p\text{CO}_2$ from the literature and experimental data and obtain the covariance matrix of the variables by calculation. The parameters with higher correlation coefficient values and an effect on the process change of $p\text{CO}_2$ were kept as inputs FFNN, and the final model was constructed and used to reconstruct the $p\text{CO}_2$ data of the Southern Ocean with a monthly temporal resolution and a spatial resolution of $1^\circ \times 1^\circ$ in the second step after continuous and iterative calculation and optimization.

First of all, we find the key parameters that affect $p\text{CO}_2$ in the Southern Ocean changes. Secondly, use the advantages of neural network technology to interpolate in the data sparse area, and build a new model by filtering parameters. Finally, in the Southern Ocean, we compare the new data with the measured data and get the root mean square error with $8.86 \mu\text{atm}$ which is better than the data reconstructed from global data.

The results of our reconstruction demonstrate that $p\text{CO}_2$ in the Southern Ocean's surface layer varies seasonally and has risen since 2000. It did, however, reach a halt from 2010 and 2013, after which it resumed its upward trend. In the Southern Ocean,

carbon dioxide flux is distributed in a double ring shape. The primary carbon sink region is 35 – 50 °S; south of 50 °S, seasonal carbon sources and sinks alternated. Despite the fact that our findings are consistent with earlier studies, the reconstructed surface $p\text{CO}_2$ products require ongoing verification. Our model will improve as the frequency and range of observations in the Southern Ocean increase.

References

1. Anderson RF, Ali S, Bradtmiller LI, Nielsen SHH, Fleisher MQ, Anderson BE, Burckle LH (2009) Wind-driven upwelling in the southern ocean and the deglacial rise in atmospheric CO_2 . *Sci* 323(5920):1443–1448. <https://doi.org/10.1126/science.1167441>
2. Cheng L, Trenberth KE, Gruber N, Mann ME, Abraham JP, Fasullo J, Guancheng L, Zhu J (2020) Improved estimates of changes in upper ocean salinity and the water cycle. 2020:GC095–03
3. Denvil-Sommer A, Gehlen M, Vrac M, Mejia C (2019) LSCE-FFNN-v1: A two-step neural network model for the reconstruction of surface ocean $p\text{CO}_2$ over the global ocean. *Geosci Model Dev* 12(5):2091–2105. <https://doi.org/10.5194/gmd-12-2091-2019>
4. Frölicher TL, Sarmiento JL, Paynter DJ, Dunne JP, Krasting JP, Winton M (2015) Dominance of the southern ocean in anthropogenic carbon and heat uptake in cmip5 models. *J Clim* 28(2):862–886. <https://doi.org/10.1175/JCLI-D-14-00117.1>
5. Gregor L, Lebehod AD, Kok S, Scheel Monteiro PM (2019) A comparative assessment of the uncertainties of global surface ocean CO_2 estimates using a machine-learning ensemble (CSIR-ML6 version 2019a) - have we hit the wall? *Geosci Model Dev* 12(12):5113–5136. <https://doi.org/10.5194/gmd-12-5113-2019>
6. Gruber N, Gloor M, Mikaloff Fletcher SE, Doney SC, Dutkiewicz S, Follows MJ, Gerber M, Jacobson AR, Joos F, Lindsay K, Menemenlis D, Mouchet A, Müller SA, Sarmiento JL, Takahashi T (2009) Oceanic sources, sinks, and transport of atmospheric CO_2 . *Glob Biogeochem Cycles* 23(1).<https://doi.org/10.1029/2008GB003349>
7. Gruber N, Landschützer P, Lovenduski NS (2019) The variable southern ocean carbon sink. *Annu Rev Mar Sci* 11(1):159–186. <https://doi.org/10.1146/annurev-marine-121916-063407>
8. Junzeng X, Qi W, Shizhang P, Yanmei Y (2012) Error of saturation vapor pressure calculated by different formulas and its effect on calculation of reference evapotranspiration in high latitude cold region. *Procedia Eng* 28:43–48. <https://doi.org/10.1016/j.proeng.2012.01.680>
9. Keppler L, Landschützer P (2019) Regional wind variability modulates the southern ocean carbon sink. *Sci Rep* 9(1):7384. <https://doi.org/10.1038/s41598-019-43826-y>
10. Körtzinger A (1999) Determination of carbon dioxide partial pressure ($p(\text{CO}_2)$). *Methods Seawater Anal*, John Wiley & Sons Ltd, chap 9:149–158. <https://doi.org/10.1002/9783527613984.ch9>
11. Landschützer P, Gruber N, Bakker DCE, Schuster U, Nakaoka S, Payne MR, Sasse TP, Zeng J (2013) A neural network-based estimate of the seasonal to inter-annual variability of the atlantic ocean carbon sink. *Biogeosciences* 10(11):7793–7815. <https://doi.org/10.5194/bg-10-7793-2013>
12. Landschützer P, Gruber N, Haumann FA, Rödenbeck C, Bakker DCE, van Heuven S, Hoppema M, Metzl N, Sweeney C, Takahashi T, Tilbrook B, Wanninkhof R (2015) The reinvigoration of the southern ocean carbon sink. *Sci* 349(6253):1221–1224. <https://doi.org/10.1126/science.aab2620>
13. Landschützer P, Gruber N, Bakker DCE (2016) Decadal variations and trends of the global ocean carbon sink. *Glob Biogeochem Cycles* 30(10):1396–1417. <https://doi.org/10.1002/2015GB005359>

14. Landschützer P, Gruber N, Bakker DCE (2016) Decadal variations and trends of the global ocean carbon sink. *Glob Biogeochem Cycles* 30(10):1396–1417. <https://doi.org/10.1002/2015GB005359>
15. Landschützer P, Gruber N, Bakker DCE, Stemmler I, Six KD (2018) Strengthening seasonal marine CO_2 variations due to increasing atmospheric CO_2 . *Nat Clim Chang* 8(2):146–150. <https://doi.org/10.1038/s41558-017-0057-x>
16. Metzl N, Brunet C, Jabaud-Jan A, Poisson A, Schauer B (2006) Summer and winter air–sea CO_2 fluxes in the southern ocean. *Deep Sea Res Part I: Ocean Res Pap* 53(9):1548–1563. <https://doi.org/10.1016/j.dsr.2006.07.006>
17. Peylin P, Bousquet P, Le Quéré C, Sitch S, Friedlingstein P, McKinley G, Gruber N, Rayner P, Ciais P (2005) Multiple constraints on regional CO_2 flux variations over land and oceans. *Glob Biogeochem Cycles* 19(1). <https://doi.org/10.1029/2003GB002214>
18. Pinho L, Duarte CM, Marotta H, Enrich-Prast A (2016) Temperature dependence of the relationship between $p\text{CO}_2$ and dissolved organic carbon in lakes. *Biogeosciences* 13(3):865–871. <https://doi.org/10.5194/bg-13-865-2016>
19. Ritter R, Landschützer P, Gruber N, Fay AR, Iida Y, Jones S, Nakaoka S, Park GH, Peylin P, Rödenbeck C, Rodgers KB, Shutler JD, Zeng J (2017) Observation-based trends of the Southern Ocean Carbon Sink. *Geophysical Research Letters* 44(24):12,339–12,348. <https://doi.org/10.1002/2017GL074837>
20. Rödenbeck C, Bakker DCE, Gruber N, Iida Y, Jacobson AR, Jones S, Landschützer P, Metzl N, Nakaoka S, Olsen A, Park GH, Peylin P, Rodgers KB, Sasse TP, Schuster U, Shutler JD, Valsala V, Wanninkhof R, Zeng J (2015) Data-based estimates of the ocean carbon sink variability - first results of the Surface Ocean $p\text{CO}_2$ mapping intercomparison (SOCOM). *Biogeosciences* 12(23):7251–7278. <https://doi.org/10.5194/bg-12-7251-2015>
21. Ruder S (2017) An overview of gradient descent optimization algorithms. [arXiv:1609.04747](https://arxiv.org/abs/1609.04747) [cs] 1609.04747
22. Shadwick EH, Trull TW, Tilbrook B, Sutton AJ, Schulz E, Sabine CL (2015) Seasonality of biological and physical controls on surface ocean CO_2 from hourly observations at the southern ocean time series site south of australia. *Glob Biogeochem Cycles* 29(2):223–238. <https://doi.org/10.1002/2014GB004906>
23. Silvano A (2020) Changes in the southern ocean. *Nat Geosci* 13(1):4–5. <https://doi.org/10.1038/s41561-019-0516-2>
24. ...Takahashi T, Sutherland SC, Wanninkhof R, Sweeney C, Feely RA, Chipman DW, Hales B, Friederich G, Chavez F, Sabine C, Watson A, Bakker DCE, Schuster U, Metzl N, Yoshikawa-Inoue H, Ishii M, Midorikawa T, Nojiri Y, Körtzinger A, Steinhoff T, Hoppema M, Olafsson J, Arnarson TS, Tilbrook B, Johannessen T, Olsen A, Bellerby R, Wong CS, Delille B, Bates NR, de Baar HJW (2009) Climatological mean and decadal change in surface ocean $p\text{CO}_2$, and net sea–air CO_2 flux over the global oceans. *Deep Sea Res Part II: Top Stud Ocean* 56(8):554–577. <https://doi.org/10.1016/j.dsr2.2008.12.009>
25. Takahashi T, Sweeney C, Hales B, Chipman DW, Newberger T, Goddard JG, Iannuzzi RA, Sutherland SC (2012) The changing carbon cycle in the southern ocean. *Oceanography* 25(3):26–37
26. Takao S, Nakaoka SI, Hashihama F, Shimada K, Yoshikawa-Inoue H, Hirawake T, Kanda J, Hashida G, Suzuki K (2020) Effects of phytoplankton community composition and productivity on sea surface $p\text{CO}_2$ variations in the Southern Ocean. *Deep Sea Res Part I: Ocean Res Pap* 160:103263. <https://doi.org/10.1016/j.dsr.2020.103263>
27. Thomalla SJ, Fauchereau N, Swart S, Monteiro PMS (2011) Regional scale characteristics of the seasonal cycle of chlorophyll in the southern ocean. *Biogeosciences* 8(10):2849–2866. <https://doi.org/10.5194/bg-8-2849-2011>

28. Wanninkhof R, Castle RD, Shannahoff J (2016) A multi-decade record of high-quality fCO₂ data in version 3 of the Surface Ocean CO₂ Atlas (SOCAT). <https://doi.org/10.1594/PANGAEA.851406>
29. Watson AJ, Schuster U, Shutler JD, Holding T, Ashton IGC, Landschützer P, Woolf DK, Goddijn-Murphy L (2020) Revised estimates of ocean-atmosphere CO₂ flux are consistent with ocean carbon inventory. *Nat Commun* 11(1):4422. <https://doi.org/10.1038/s41467-020-18203-3>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits any noncommercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if you modified the licensed material. You do not have permission under this license to share adapted material derived from this chapter or parts of it.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Detection and Analysis of Mesoscale Eddies Based on Deep Learning



Yingjie Liu, Quanan Zheng, and Xiaofeng Li

1 Introduction

Mesoscale eddies are circular currents of water bodies with spatial scales from tens to hundreds of kilometers and temporal scales from days to years [7]. Mesoscale eddies play a significant role in the transport of momentum, mass, heat, nutrients, salt, and other seawater chemical elements across the ocean basins, effectively impacting the global ocean circulation, large-scale water distribution, air-sea coupling, and biological activities [2, 3, 7, 8, 13, 18]. Mesoscale eddies can be generally classified as either cyclonic eddies (CEs) if they rotate counterclockwise (in the Northern Hemisphere) or anticyclonic eddies (AEs) otherwise. CE (AE) drive local upwelling (downwelling), leading to negative (positive) sea surface height (SSH) anomalies and sea surface temperature (SST) anomalies. The changes in SSH, SST, chlorophyll concentration (CHL), and roughness caused by oceanic eddies can be recorded by altimeter, infrared, ocean color, and synthetic aperture radar (SAR) remote sensing, respectively. Accurate automatic eddy detection is crucial for monitoring the dynamics of mesoscale eddies on physical properties, transport, circulation, evolution, decay, and their impact on other ocean processes. Oceanic eddy detection based on a variety of remote sensing data has been widely studied.

Automatic eddy identification algorithms that developed based on altimeter SSH data can be divided into three categories: the physical-parameter-based method that includes the Okubo-Weiss parameter method [6, 27], the winding angle method [5, 46], and the 2D wavelet method [10]; the flow-direction-based method [39, 48]; and the SSH-based method [7, 18, 38]. Another modern method that is based on the instantaneous Lagrangian flow geometry [1, 22–25, 40] is proposed to identify eddies in turbulent flows. Several eddy detection methods were developed based on satellite SST data. E.g., edge detection method [29], neural network-based method

Y. Liu · X. Li (✉)

CAS Key Laboratory of Ocean Circulation and Waves, Institute of Oceanology,
Chinese Academy of Sciences, Qingdao 266071, China
e-mail: lixf@qdio.ac.cn

Q. Zheng

Department of Atmospheric and Oceanic Science, University of Maryland, College Park,
Maryland 20742, USA

© The Author(s) 2023

X. Li and F. Wang (eds.), *Artificial Intelligence Oceanography*,
https://doi.org/10.1007/978-981-19-6375-9_10

[4], SST contour-based method [17], and velocity-geometry method [12], etc. Compared to satellite SSH data, CHL and SAR images have a high spatial resolution, which makes them effective sources for gaining more comprehensive and detailed information on mesoscale eddies in the oceans [7, 14, 15, 36]. However, eddy detection based on CHL and SAR images is still in the stage of case study due to the low space-time coverage. In conclusion, existing eddy detection algorithms can detect major circular structures of mesoscale eddies, but more work is still to be done. On the one hand, eddy detection based on different remote sensing data has its own advantages and disadvantages. For instance, eddies may temporarily ‘disappear’ or cannot be detected due to noise and sampling errors of the altimeter SSH data, while eddy detection using SST is prone to false positives because many other ocean phenomena may impact SST. On the other hand, with the accumulation of remote sensing data, some algorithms lack computational efficiency due to contour iterations [49] or complex calculation processes [40].

Recently, deep learning (DL) [33] technology has exhibited state-of-the-art performance in mining the complicated rules hidden in multi-source ocean remote sensing images [26, 35, 52]. Moreover, in comparison with traditional statistical and machine learning methods, DL technology features a strong ability to learn and model complex relationships [28, 30, 43, 47, 51]. Therefore, it is natural to propose using the DL-based model to detect mesoscale eddies based on remote sensing images. Lguensat et al. [34] developed "EddyNet" that based on the encoder-decoder network U-Net to identify oceanic eddies in the southwest Atlantic. Franz et al. [20] also used the U-Net to detect and track oceanic eddies in Australia and the East Australia current regions. Du et al. [15] developed "DeepEddy" based on PCANet and spatial pyramid pooling to detect oceanic eddies based on SAR images. Xu et al. [50] applied the pyramid scene parsing network to detect eddies in the North Pacific Subtropical Countercurrent region. These regional studies proved that the DL-based model performed well in detecting mesoscale eddies in territorial seas. The DL-based model performance on the global mesoscale eddy detection remained unverified. Moreover, these works use one type of remote sensing data as input to detect mesoscale eddies.

In order to solve the above problems, we propose a DL-based global eddy detection model based on the fusion of SSH and SST data in this study. The remainder of the study is organized as follows. Section 2 firstly illustrates a DL-based model to identify global mesoscale eddies based on satellite SSH data. Furthermore, Sect. 3 shows a multi-model DL-based eddy detection model developed based on the fusion of SST and SSH data. Section 4 shows the characterization of global mesoscale eddies detected by the multi-model DL-based model. Finally, Sect. 5 summarizes the conclusions of our investigation.

2 DL-based Eddy Detection Model Based on SSHA Data

2.1 Data

The SSHA product is produced by Ssalto/Duacs and distributed by the Archiving, Validation, and Interpretation of Satellite Oceanographic (AVISO) and is available daily on 0.25° spatial resolution. The product is merged from all available altimeter missions, including TOPEX/Poseidon (TP), Jason-1&2, European Remote-Sensing Satellite (ERS)-1&2, Environmental Satellite (ENVISAT), Geosat Follow On (GFO), Cryosat-2, Saral/AltiKa, and Haiyang-2A, and covers the period from 1993 to the present. Since resolving oceanic mesoscale variability requires a minimum of three altimeter missions [32, 41, 42], only the period from 2000 onward meets the criterion.

2.2 Method

The DL-based eddy detection model is developed based on the U-Net architecture consisting of ResNet blocks, hereafter Res-UNet. Although developed initially for semantic segmentation of biomedical images [38], U-Net [19, 45] achieves successful applications in many fields. Fig. 1 shows the framework of the U-Net, which is consisted of the encoder-decoder module, bottleneck module, and concatenation module. The encoder module extracts information at different resolutions. The output module contains a convolutional layer and activation layer to yield class confidences at each pixel.

The ResNet block is designed to deepen the network while alleviating the problem of network degradation. The input to the ResNet block, x_r , is processed in two ways. A 3×3 convolution is used to obtain a direct linear mapping result; i.e., $w_r * x_r$, where w_r denotes the convolutional filter. Meanwhile, x_r is subjected to the following processes twice in sequence: batch normalization (BN), a rectified linear unit (ReLU), and Conv2D. The ReLU layer is used to increase the nonlinearity. By adding the direct and residual mapping, the ResNet block combines deep-learning and shallow-learning features, meaning that it can extract more valuable information. The original information is maintained and passed by the process of linear mapping with a 3×3 convolution, which reduces the possibility of degradation.

2.3 Experiment and Performance

The training and validation datasets of mesoscale eddies are generated automatically by using the SSH-based method [37], which is similar to the eddy identification method proposed by Chelton, et al. [7]. Mesoscale eddies from 2000-2013 and 2014-2015 are used as the training dataset and validation dataset. There are 5114 training

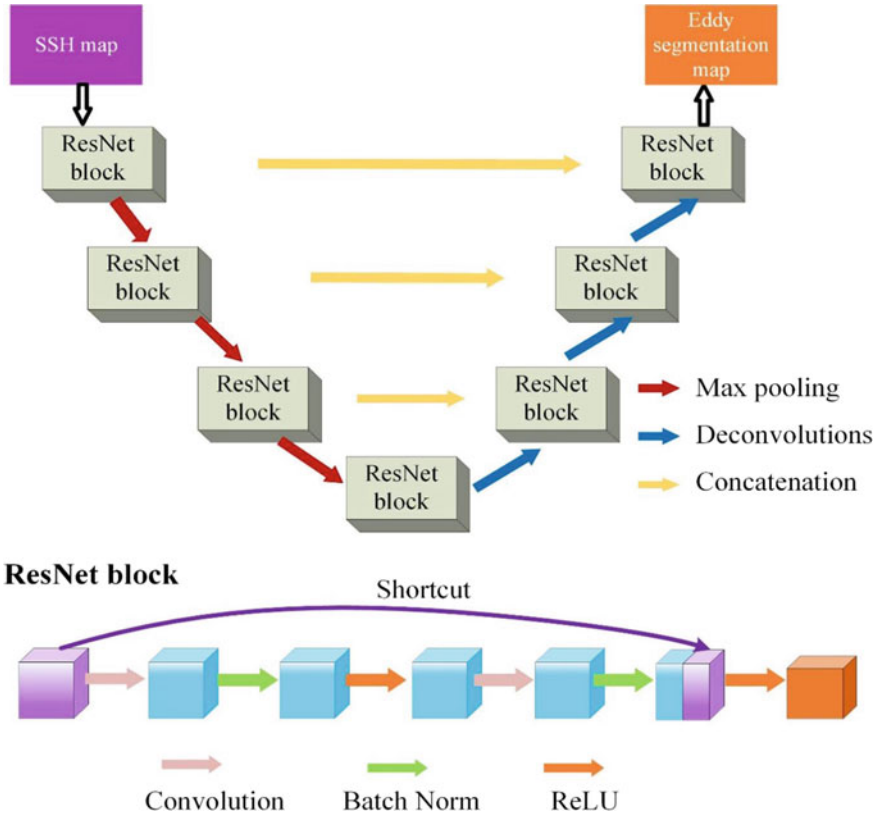


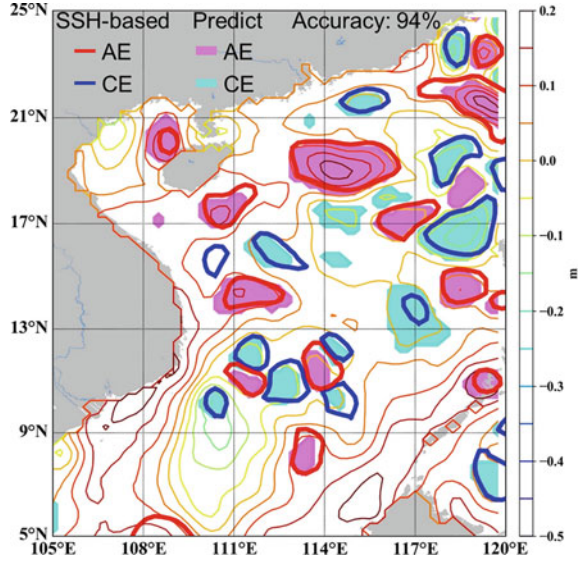
Fig. 1 Res-UNet based eddy detection model

samples and 730 testing samples. Pixels in each sample are labeled as ‘1’, ‘-1’, and ‘0’ inside anticyclonic eddies (AEs), cyclonic eddies (CEs), and background regions. The Res-UNet model is trained on an Nvidia GeForce RTX 2070 GPU card using ADAM optimizer [31] and mini-batches of 16 maps. An early-stopping strategy is used to stop the learning process when the validation dataset loss stops improving in five consecutive epochs. The implementation of our model is realized in Python. The Python interfaces are based on Keras framework [9] with TensorFlow backend. The dice loss function, which is widely used in segmentation problems, is the cost function. Given the predicted segmentation P and the ground truth region G , the dice coefficient is calculated as:

$$Dicecoef(P, G) = \frac{2|P \cap G|}{|P| + |G|} \tag{1}$$

where $|.$ is the sum of elements in the area. A good segmentation result is explained by a dice coefficient that is close to 1. By contrast, a low dice coefficient (near 0) indicates

Fig. 2 Mesoscale eddies detected by SSH-based method and Res-UNet in the SCS on January 1 2019



poor segmentation performance. A differentiable version of the above metric must be used to train deep neural networks. A soft Dice Coefficient was adopted in this work, and the output of the softmax layer was directly used to maximize loss calculations. The coefficient is given as:

$$softDicecoef(P, G) = \frac{\sum p_i \times g_i}{\sum p_i + \sum g_i} \tag{2}$$

p_i is the output of the softmax layer 1 for the correct class and otherwise set as 0. Finally, the Loss is calculated as:

$$Loss = 1 - softDicecoef(P, G) \tag{3}$$

The loss and accuracy of the Res-UNet model were about 14% and 94% when training using the ground truth dataset in the South China Sea (SCS) (Fig. 2).

Therefore, the Res-UNet model is accurate and reliable enough to obtain mesoscale eddies in the global ocean. The global SSH and SST maps were firstly partitioned into several regional maps of 80×60 pixels, respectively. Then, applying the Res-UNet model to SSHA maps in the same space-time until all the regions have been detected. Finally, all the regions' eddies were seamlessly merged to obtain a global eddy map. Figure 3 shows the mesoscale eddies identified by the Res-UNet model on January 1, 2019. There are 3314 (2963 ground truth) AEs and 3407 (3056 ground truth) CEs in the global ocean. Compared to the SSH-based method, the accuracy of the Res-UNet based global eddy detection method is 93.79%, and the mean IoU is 88.86%. Figure 3 clearly shows that the Res-UNet model identified many more small-scale

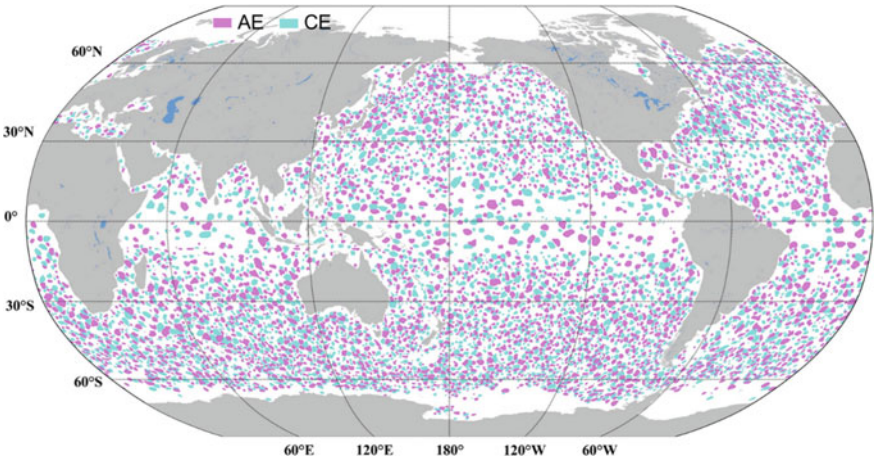


Fig. 3 Mesoscale eddies detected by the Res-UNet model in the global ocean on January 1, 2019

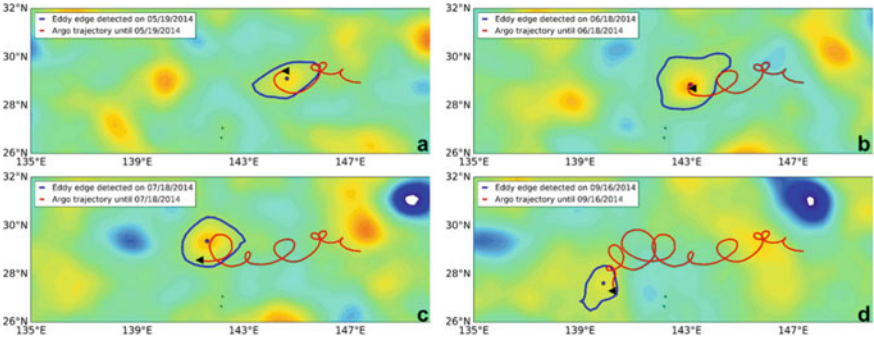


Fig. 4 An Argo float (red line) is captured by an AE (blue line) that detected by the Dense-UNet model in the KE region and rotated with the AE on **a** May 19, 2014, **b** June 18, 2014, **c** July 18, 2014, and **d** September 16, 2014 (the color denotes SSHA)

eddies. Besides, it takes less than 1 minute for the Res-UNet model costs to identify eddies in the global ocean, while the SSH-based method costs more than 16 hours [37]. In conclusion, the Res-UNet model can identify many more small-scale eddies and significantly improve computational efficiency.

Argo floats are associated with short repeating cycles, and they can observe mesoscale eddies in the global ocean. When trapped in an eddy, they show either a cyclonic or an anticyclonic trajectory. Therefore, the trajectory data of Argo floats are utilized to verify the accuracy of the Res-UNet model. In this chapter, the Argo float (2901556) is used to validate the results of the Res-UNet based eddy detection model. The Argo float was trapped in the AE and moved as a clockwise loop. Such a result is consistent with the concept that AEs rotate clockwise in the Northern Hemisphere (Fig. 4).

3 DL-based Eddy Detection Model Based on SSHA and SST Data

In order to solve the problem that eddies may temporarily ‘disappear’ or cannot be detected due to noise and sampling errors of the SSHA data, SST data that can finely delineate the eddy structure are added to the model, to detect mesoscale eddies more accurately.

3.1 Data

The SST dataset is the NOAA Optimum Interpolation (OI) SST product from Reynolds, et al. [44] on daily and 0.25° resolution. The OISST dataset is constructed from infrared satellite observations of the Advanced Very High Resolution Radiometer (AVHRR) with supplemental information provided by in situ observations and proxy SSTs computed from sea ice concentrations. Error fields were provided, showing an accuracy of about 0.1°C on daily basis. The OISST dataset is available from 1981 onward.

3.2 Method

The Dense-UNet model is comprised of a data fusion module and a feature extraction module (Fig. 5). Considering the complex nonlinear relationship between SST and SSHA within eddies, the layer-level fusion strategy is used to fusion SSH and SST data before the feature extraction. The layer-level fusion network can effectively integrate and fully leverage multi-modal images. Therefore, the data fusion model was developed based on the hyper-dense connectivity network [11] to integrate and fully leverage fused SSHA and SST images effectively. Satellite SST and SSHA data were imported into two streams, respectively. To better model relationships between SST and SSHA, dense connections, that use linear operations where every input is connected to every output by weight, were introduced into the model. Dense connections can relieve the vanishing gradient of networks, and reduce the parameters of deep networks [11]. Let x_l^1 and x_l^2 denote the outputs of the l^{th} layer in SST and SSHA streams and H_l is a mapping function composed of a convolution layer followed by a batch normalization and a ReLU activation function. The output of the l^{th} layer in a given stream s can then be defined as:

$$x_l^s = H_l^s ([x_{l-1}^1, x_{l-1}^2, x_{l-2}^1, x_{l-2}^2, \dots, x_0^1, x_0^2]) \quad (4)$$

Then, the fusion data x_l^s are used as input of the U-Net to detect mesoscale eddies.

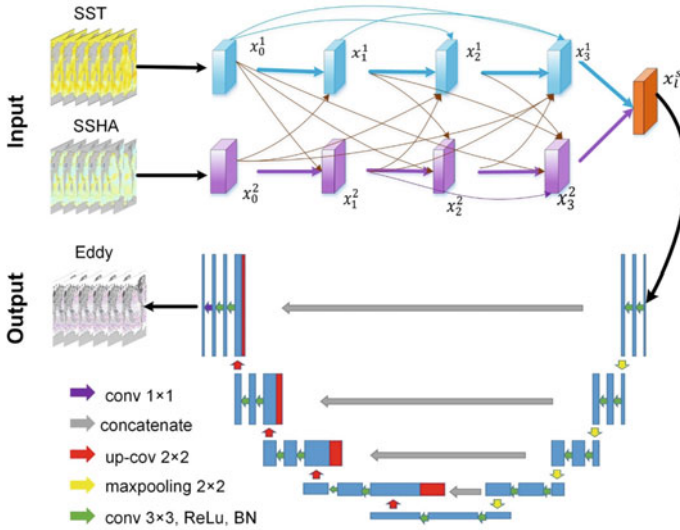


Fig. 5 Dense-UNet architecture based on the fusion of SSHA and SST data

3.3 Experiment and Performance

The training and validation datasets of mesoscale eddies are generated automatically using the SSH-based method [22]. Mesoscale eddies during 2000-2013 are used as the training dataset, and mesoscale eddies during 2014-2015 are used as the validation dataset. There are 5114 training samples and 730 testing samples. Pixels in each sample are labeled as ‘1’, ‘-1’, and ‘0’ inside AEs, CEs, and background regions. To evaluate the performance of the Dense-UNet model, we identify mesoscale eddies in the Kuroshio Extension (KE) and the SCS. The dice loss function is used as the cost function. As shown in Table 1, the loss based on the SSHA is larger than that based on the fusion of SSHA and SST. On the contrary, the accuracy based on the SSHA is lower than that based on the fusion of SSHA and SST.

The Dense-UNet model can be further verified by a case study of a CE in the KE (Fig. 6). On November 22 and 23, 2013, the CE identified by SSHA split into

Table 1 The loss and accuracy of the Dense-UNet model of different testing datasets in different ocean regions

Region	Dataset	Model	Dice loss	Accuracy
SCS	SSHA	Res-UNet	0.1455	0.9131
	Fusion dataset	Dense-UNet	0.0869	0.9490
KE	SSHA	Res-UNet	0.1637	0.9480
	Fusion dataset	Dense-UNet	0.1183	0.9640

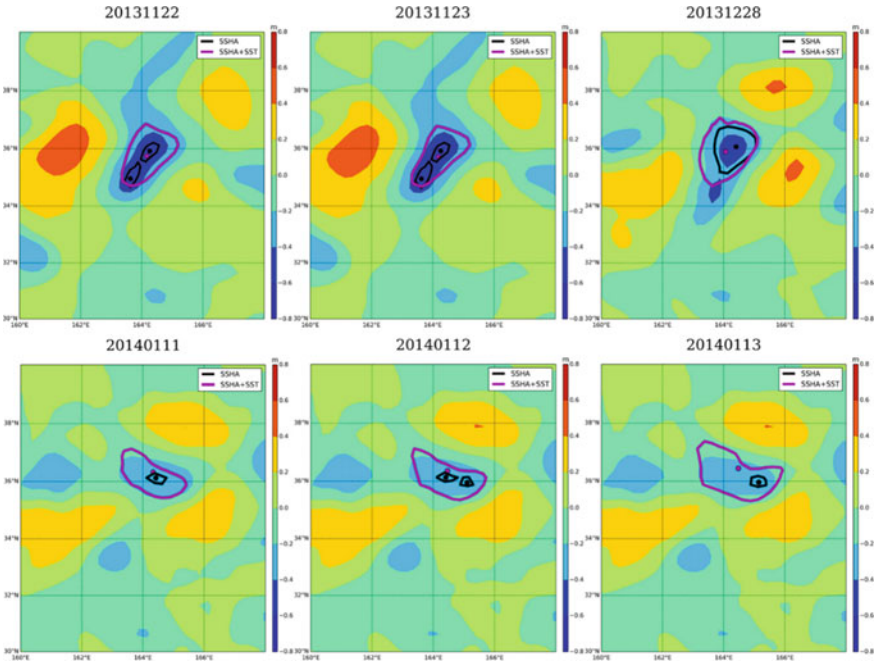


Fig. 6 Variations of a CE in the KE at different time during the evolution. The black line represents the eddy identified by SSHA, while the purple line represents the eddy identified by the fusion of SST and SSHA

two CEs, while the CE identified by the fusion of SST and SSHA was consistent with the negative area of SSHA. From December 28, 2013 to January 13, 2014, the CE identified by SSHA did not cover the negative area of SSHA, while the eddy boundary identified by the fusion of SST and SSHA completely covered the negative area of SSHA. Therefore, it can be indicated that the fusion of SSHA and SST data enhances the accuracy and robustness of eddy detection and can also ensure eddy tracking’s continuity and accuracy.

In this section, we propose the Dense-UNet method to identify oceanic mesoscale eddies. Compared to the methods that identify eddies based on one kind of remote sensing images, Dense-UNet detect eddies based on the fusion of SSHA and SST data. Using the Dense-UNet model, we perform a comparison experiment using SSHA data and fusion data in the SCS and KE regions, respectively. As a result, the Dense-UNet model achieves impressive detection performance based on the fusion data. The model not only improves eddy detection accuracy and efficiency but also gives a novel viewpoint on exploring the relationships between marine environmental variables and mesoscale eddies.

4 Characterization Analysis of Mesoscale Eddies in the Global Ocean

4.1 Spatiotemporal Distributions of Eddies in the Global Ocean

Based on the Dense-UNet model, mesoscale eddies were identified based on 23-year satellite SSHA and SST data in the global ocean during 1993–2015. In this study, the research focused on eddies with amplitudes greater than 2 cm and sea surface radii larger than 35 km, which was based on consideration of the resolution and precision of the SSH product [16]. Besides, we only consider eddies located in areas where water depths are greater than 200 m to minimize the impacts of data errors near the coastal shallow water region. An average of 4,100 mesoscale eddies were identified daily in the global ocean during the period 1993–2015. The frequency of eddies for a given geographic resolution (0.25° latitude by 0.25° longitude) was defined as an F -number for simplicity:

$$F(\%) = \frac{d_{eddy}}{d_{total}} \quad (5)$$

where d_{eddy} means the days that mesoscale eddies appeared, and d_{total} represents the total number of observation days. In other words, high F -numbers imply a high intensity of eddy activity and vice versa. The seasonal variability for AEs and CEs in the global ocean is similar (Fig. 7a-b). In the Southern Hemisphere (SH), eddy activity is weak in the austral summer (December–February, DJF) and fall (March–May, MAM), but intensive during the austral spring (September–November, SON) and winter (June–August, JJA), and vice versa in the Northern Hemisphere (NH).

Figure 7c-d shows the spatial distribution of mesoscale eddies in the global ocean. Mesoscale eddies with lower FF -number were distributed in tropical waters. On the contrary, mesoscale eddies with higher F -number were widely distributed in the middle latitude regions, including the Kuroshio Extension region, the Agulhas Current, the Gulf Stream, the Agulhas Return Current, the East Australia Current, and the Antarctic Circumpolar Current, etc. Besides, CE activities are more intensive than AEs in the Western Boundary Current regions. In general, the spatial distribution of global eddies detected in this study has good consistent with previous literature [7, 18, 49].

4.2 Long-term Variations in Derived Eddy Parameters

The long-term variations in annual mean eddy properties (eddy number, radius, amplitude, and rotational speed) are shown separately for AEs and CEs in the NH and the SH. The eddy number is the annual mean eddy census per day, and the percentage represents the ratio of the annual mean abnormal eddy census per day to the total

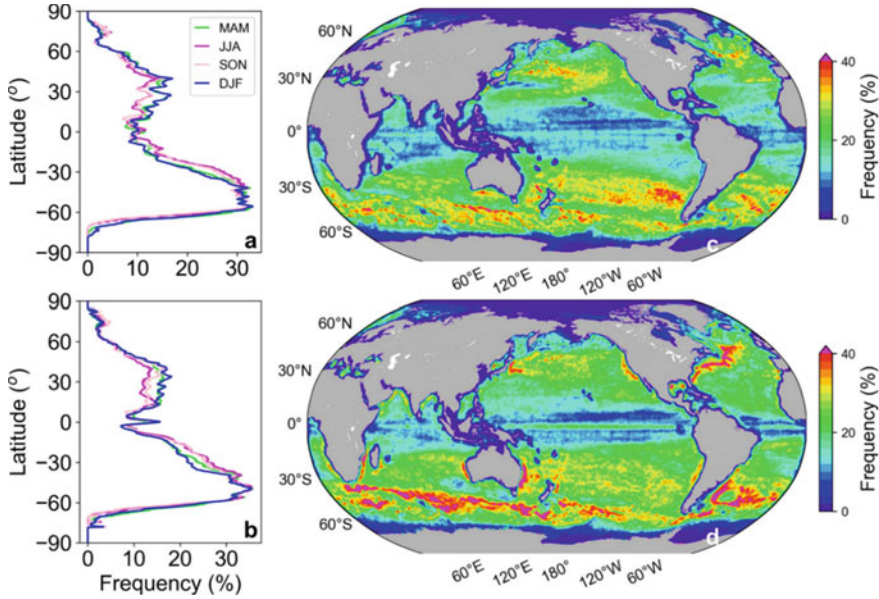


Fig. 7 Spatiotemporal distribution of the F -number of mesoscale eddies in the global ocean from 1993–2015. The graphs and maps show meridional variation and spatial distribution of the F -number of AEs (**a**, **c**), CE (**b**, **d**). MAM represents March–May, DJF represents December–February, JJA represents June–August, and SON represents September–November. The image resolution is 0.25° by 0.25°

number of eddies. The eddy radius is the distance from its center to the outermost SSH contour with the maximum average geostrophic speed (U). $U = \sqrt{u^2 + v^2}$, where u and v are the zonal and meridional components of the geostrophic velocity anomaly, which can be computed from the SSH gradients:

$$u = -\frac{g}{f} \frac{\partial SSH}{\partial y} \quad (6)$$

$$v = \frac{g}{f} \frac{\partial SSH}{\partial x} \quad (7)$$

where g is the acceleration due to gravity; ∂x and ∂y are the eastward and northward distances, respectively; and f is the Coriolis parameter. Eddy kinetic energy is given as $(EKE) = \frac{1}{2}(u^2 + v^2)$. The amplitude is the difference in SSHA between the eddy core and boundary. The rotational speed is the maximum of the average geostrophic speed around all of the eddy's closed SSHA contours.

About 2100 CE and 2000 AE formed per day as detected by the Dense-UNet eddy detection model for each global SSHA map. This is close to the result of Faghmous et al. [18], which identifies approximately 2300 CE and 2300 AE for

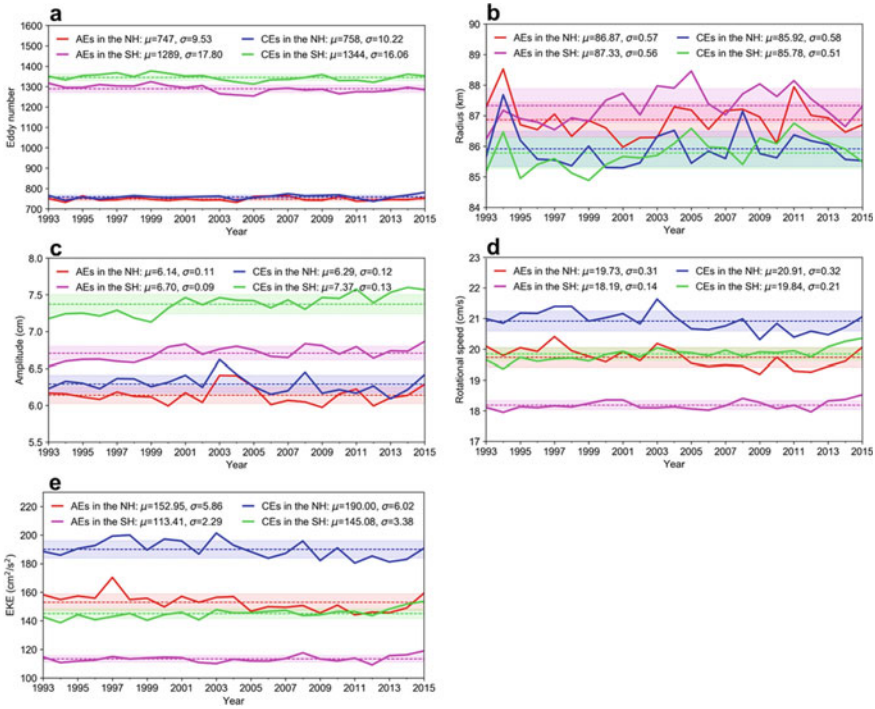


Fig. 8 Variations in annual mean parameters of mesoscale eddies in the global ocean from 1993 to 2015. Eddy number **a**, eddy radius **b**, eddy amplitude **c**, rotational speed **d**, and EKE **e**. μ (dotted line) and σ (shading) are mean values and one standard deviation of the annual mean eddy parameters

each daily SSHA snapshot. The slight difference in eddy number between the two eddy datasets is possible because there is no limit to the amplitude of eddies in Faghmous et al. [18]. There were no significant decreasing and increasing trends in the annual mean eddy parameters for both AEs and CE during the 1993–2015 period, and the annual mean eddy parameters for eddies in the NH and the SH are different (Fig. 8). Eddy numbers in the SH are twice as much as that in the NH, which is consistent with the result in Fig. 7a-b. The annual mean radius for AEs are slightly larger than that of CE, which is about 87.0 km and 86.0 km, respectively. The annual mean amplitude of the CE is larger than that of the AEs in both hemispheres, and annual mean eddy amplitude in the SH is larger than that in the NH. The annual mean eddy amplitude of AEs and CE in the NH (SH) is 6.14 (6.7) cm and 6.29 (7.37) cm, respectively. The difference between AEs and CE on amplitude is expected from the gradient wind effect of centrifugal force that pushes fluid outward in rotating eddies [21], thus intensifying the low pressure at the centers of CE and weakening the high pressure at the centers of AEs [7].

Similarly, the annual mean eddy rotational speed and EKE of CEs are also larger than that of AEs in their respective hemispheres since they were derived from eddy amplitude. However, the annual mean rotational speed and EKE of eddies in the NH are larger than in the SH. The annual mean eddy rotational speed of AEs and CEs in the NH (SH) is 19.73 (18.19) cm/s and 20.91 (19.84) cm/s, respectively. The annual mean EKE of AEs and CEs in the NH (SH) is 152.95 (113.41) cm/s and 190.00 (145.08) cm/s, respectively.

5 Conclusions

This chapter elaborated on how to apply deep learning technology to global mesoscale eddy detection. We first developed a deep learning-based eddy detection model based on SSHA data. The model consists of U-Net and ResNet blocks, called Res-UNet. The Res-UNet was applied to detect mesoscale eddies in the global ocean. Argo floats data are used to verify the Res-UNet model. The Argo float was trapped in the AE and moved as a clockwise loop. Such a result is consistent with the concept that AEs rotate clockwise in the Northern Hemisphere. Compared to the traditional eddy detection methods, the Res-UNet eddy detection model can accurately identify mesoscale eddies and significantly improve computational efficiency. Such a result proves that deep learning technology has strong learning abilities and can better use datasets for feature extraction.

Considering that eddies may temporarily ‘disappear’ or cannot be detected due to noise and sampling errors of the SSHA data, the study further develops a multi-modal deep learning model—Dense-UNet model to detect mesoscale eddies based on the fusion of SSHA and SST data. The Dense-UNet model extracts SSHA information for determining eddy locations and withdraws SST information to supplement and confirm eddy features embodied in SSHA data. The results show that the fusion of SSHA and SST data enhances the accuracy and robustness of eddy detection and can also ensure eddy tracking’s continuity and accuracy. Based on the Dense-UNet eddy detection model, mesoscale eddies are detected based on satellite SSHA and SST data in the global ocean from 1993–2015. The analysis of the spatiotemporal distribution of the 23-year global eddy dataset revealed that eddies were concentrated along western boundary currents. Mesoscale eddies are active in winter in the North Hemisphere and vice versa in the Southern Hemisphere. The spatiotemporal distribution of eddies detected by the Dense-UNet model is in good agreement with previous studies, thus further validating the model’s accuracy.

The long-term variations in annual mean eddy properties (eddy number, radius, amplitude, and rotational speed) are analyzed separately for AEs and CEs in the Northern and the Southern Hemisphere. There were no significant decreasing and increasing trends in the annual mean eddy parameters for both AEs and CEs during the 1993–2015 period, but the annual mean eddy parameters for eddies in the

Northern Hemisphere and the Southern Hemisphere are different. Eddy numbers in the Southern Hemisphere are twice as much as that in the Northern Hemisphere. The annual mean radius for AEs is slightly larger than that of CE in both hemispheres. The annual mean amplitude of the CE is larger than that of the AEs in both hemispheres, and the annual mean eddy amplitude in the Southern Hemisphere is larger than that in the Northern Hemisphere. The annual mean eddy rotational speed and EKE of CE are also larger than AEs in their respective hemispheres. However, the annual mean rotational speed and EKE of eddies in the Northern Hemisphere are larger than that in the Southern Hemisphere. The difference in eddy parameters between the two hemispheres is caused by the different generation mechanisms of mesoscale eddies, which deserves further study. In conclusion, the study extends the usage of satellite remote sensing big data, enriches the application of deep learning technology in oceanography, and promotes multidisciplinary research in this aspect.

References

1. Beron-Vera FJ, Yan W, Olascoaga MJ, Goni GJ, Haller G (2013) Objective detection of oceanic eddies and the agulhas leakage. *J Phys Ocean* 43(7):1426–1438
2. Bourras D (2004) Response of the atmospheric boundary layer to a mesoscale oceanic eddy in the northeast Atlantic. *J Geophys Res: Ocean* 109(D18):1480. <https://doi.org/10.1029/2004JD004799>
3. Byrne D, Münnich M, Frenger I, Gruber N (2016) Mesoscale atmosphere ocean coupling enhances the transfer of wind energy into the ocean. *Nat Commun* 7:ncomms11867. <https://doi.org/10.1038/ncomms11867>
4. Castellani M (2006) Identification of eddies from sea surface temperature maps with neural networks. *Int J Remote Sens* 27(8):1601–1618. <https://doi.org/10.1080/01431160500462170>
5. Chaigneau A, Gizolme A, Grados C (2008) Mesoscale eddies off peru in altimeter records: Identification algorithms and eddy spatio-temporal patterns. *Prog Ocean* 79(2–4):106–119. <https://doi.org/10.1016/j.pocean.2008.10.013>
6. Chelton DB, Schlax MG, Samelson RM, Szoeké RA (2007) Global observations of large oceanic eddies. *Geophys Res Lett* 34(15):2257. <https://doi.org/10.1029/2007gl030812>
7. Chelton DB, Schlax MG, Samelson RM (2011) Global observations of nonlinear mesoscale eddies. *Prog Ocean* 91(2):167–216. <https://doi.org/10.1016/j.pocean.2011.01.002>
8. Chen G, Gan J, Xie Q, Chu X, Wang D, Hou Y (2012) Eddy heat and salt transports in the South China Sea and their seasonal modulations. *J Geophys Res: Ocean* 117(C5). <https://doi.org/10.1029/2011jc007724>
9. Chollet F, et al. (2018) Keras: The python deep learning library. *Astrophysics Source Code Library*, pp ascl–1806
10. Doglioli AM, Blanke B, Speich S, Lapeyre G (2007) Tracking coherent structures in a regional ocean model with wavelet analysis: Application to cape basin eddies. *J Geophys Res: Ocean* 112(C5):20987. <https://doi.org/10.1029/2006jc003952>
11. Dolz J, Gopinath K, Yuan J, Lombaert H, Desrosiers C, Ayed IB (2018) HyperDense-Net: A hyper-densely connected CNN for multi-modal image segmentation. *IEEE Trans Med Imaging* 38(5):1116–1126
12. Dong C, Nencioli F, Liu Y, McWilliams JC (2011) An automated approach to detect oceanic eddies from satellite remotely sensed sea surface temperature data. *IEEE Geosci Remote Sens Lett* 8(6):1055–1059. <https://doi.org/10.1109/LGRS.2011.2155029>
13. Dong C, McWilliams JC, Liu Y, Chen D (2014) Global heat and salt transports by eddy movement. *Nat Commun* 5(1):3294. <https://doi.org/10.1038/ncomms4294>

14. Dong D, Yang X, Li X, Li Z (2016) Sar observation of eddy-induced mode-2 internal solitary waves in the south china sea. *IEEE Trans Geosci Remote Sens* 54(11):6674–6686
15. Du Y, Song W, He Q, Huang D, Liotta A, Su C (2019) Deep learning with multi-scale feature fusion in remote sensing for automatic oceanic eddy detection. *Inf Fusion* 49:89–99
16. Ducet N, Le Traon PY, Reverdin G (2000) Global high-resolution mapping of ocean circulation from TOPEX/Poseidon and ERS-1 and -2. *J Geophys Res: Ocean* 105(C8):19477–19498. <https://doi.org/10.1029/2000JC900063>
17. D’Alimonte D (2009) Detection of mesoscale eddy-related structures through Iso-SST patterns. *IEEE Geosci Remote Sens Lett* 6(2):189–193. <https://doi.org/10.1109/LGRS.2008.2009550>
18. Faghmous JH, Frenger I, Yao Y, Warmka R, Lindell A, Kumar V (2015) A daily global mesoscale ocean eddy dataset from satellite altimetry. *Sci Data* 2:150028. <https://doi.org/10.1038/sdata.2015.28>
19. Falk T, Mai D, Bensch R, Çiçek Ö, Abdulkadir A, Marrakchi Y, Böhm A, Deubner J, Jäckel Z, Seiwald K et al (2019) U-Net: deep learning for cell counting, detection, and morphometry. *Nat Methods* 16(1):67–70
20. Franz K, Roscher R, Milioto A, Wenzel S, Kusche J (2018) Ocean eddy identification and tracking using neural networks. In: *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, pp 6887–6890
21. Gill AE, Adrian E (1982) *Atmosphere-ocean dynamics*, vol 30. Academic press
22. Haller G (2005) An objective definition of a vortex. *J Fluid Mech* 525:1–26. <https://doi.org/10.1017/s0022112004002526>
23. Haller G (2015) Lagrangian coherent structures. *Annu Rev Fluid Mech* 47(1):137–162
24. Haller G, Beron-Vera FJ (2013) Coherent lagrangian vortices: The black holes of turbulence. *J Fluid Mech* 731:R4. <https://doi.org/10.1017/jfm.2013.391>
25. Haller G, Beronvera FJ (2014) Addendum to “coherent lagrangian vortices: The black holes of turbulence”. *J Fluid Mech* 755(3):134–140. <https://doi.org/10.1017/jfm.2014.441>
26. Ham YG, Kim JH, Luo JJ (2019) Deep learning for multi-year ENSO forecasts. *Nat* 573(7775):568–572. <https://doi.org/10.1038/s41586-019-1559-7>
27. Henson SA, Thomas AC (2008) A census of oceanic anticyclonic eddies in the gulf of alaska. *Deep Sea Res Part I: Ocean Res Pap* 55(2):163–176. <https://doi.org/10.1016/j.dsr.2007.11.005>
28. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. *Sci* 313(5786):504–507. <https://doi.org/10.1126/science.1127647>
29. Holyer RJ, Peckinpaugh SH (1989) Edge detection applied to satellite imagery of the oceans. *IEEE Trans Geosci Remote Sens* 27(1):46–56. <https://doi.org/10.1109/36.20274>
30. Jordan MI, Mitchell TM (2015) Machine learning: Trends, perspectives, and prospects. *Sci* 349(6245):255–260. <https://doi.org/10.1126/science.aaa8415>
31. Kingma D, Ba J (2014) Adam: A method for stochastic optimization. *Comput Sci*
32. Le Traon P, Dibarbouré G (1999) Mesoscale mapping capabilities of multiple-satellite altimeter missions. *J Atmos Ocean Technol* 16(9):1208–1223
33. Lecun Y, Bengio Y, Hinton GE (2015) Deep learning. *Nat* 521(7553):436–444. <https://doi.org/10.1038/nature14539>
34. Lguensat R, Sun M, Fablet R, Tandeo P, Mason E, Chen G (2018) EddyNet: A deep neural network for pixel-wise classification of oceanic eddies. In: *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, pp 1764–1767
35. Li X, Liu B, Zheng G, Ren Y, Zhang S, Liu Y, Gao L, Liu Y, Zhang B, Wang F (2020) Deep-learning-based information mining from ocean remote-sensing imagery. *Natl Sci Rev* 7(10):1584–1605. <https://doi.org/10.1093/nsr/nwaa047>
36. Li Y, Li X, Wang J, Peng S (2016) Dynamical analysis of a satellite-observed anticyclonic eddy in the northern bering sea. *J Geophys Res: Ocean* 121(5):3517–3531
37. Liu Y, Chen G, Sun M, Liu S, Tian F (2016) A parallel SLA-based algorithm for global mesoscale eddy identification. *J Atmos Ocean Technol* 33(12):2743–2754. <https://doi.org/10.1175/JTECH-D-16-0033.1>
38. Mason E, Pascual A, McWilliams JC (2014) A new sea surface height-based code for oceanic mesoscale eddy tracking. *J Atmos Ocean Technol* 31(5):1181–1188. <https://doi.org/10.1175/JTECH-D-14-00019.1>

39. Nencioli F, Dong C, Dickey T, Washburn L, McWilliams JC (2010) A vector geometry-based eddy detection algorithm and its application to a high-resolution numerical model product and high-frequency radar surface velocities in the southern california bight. *J Atmos Ocean Technol* 27(3):564–579. <https://doi.org/10.1175/2009jtecho725.1>
40. Onu K, Huhn F, Haller G (2015) LCS tool: A computational platform for lagrangian coherent structures. *J Comput Sci* 7:26–36. <https://doi.org/10.1016/j.jocs.2014.12.002>
41. Pascual A, Pujol MI, Larnicol G, Le Traon PY, Rio MH (2007) Mesoscale mapping capabilities of multisatellite altimeter missions: First results with real data in the mediterranean sea. *J Mar Syst* 65(1–4):190–211. <https://doi.org/10.1016/j.jmarsys.2004.12.004>
42. Pujol MI, Faugère Y, Taburet G, Dupuy S, Pelloquin C, Ablain M, Picot N (2016) DUACS DT2014: the new multi-mission altimeter data set reprocessed over 20 years. *Ocean Sci* 12(5):1067–1090. <https://doi.org/10.5194/os-12-1067-2016>
43. Reichstein M, Campsvalls G, Stevens B, Jung M, Denzler J, Carvalhais N, Prabhat, (2019) Deep learning and process understanding for data-driven earth system science. *Nat* 566(7743):195–204. <https://doi.org/10.1038/s41586-019-0912-1>
44. Reynolds RW, Smith TM, Liu C, Chelton DB, Casey KS, Schlax MG (2007) Daily high-resolution-blended analyses for sea surface temperature. *J Clim* 20(22):5473–5496. <https://doi.org/10.1175/2007JCLI1824.1>
45. Ronneberger O, Fischer P, Brox T (2015) U-Net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. Springer, pp 234–241
46. Sadarjoen IA, Post FH (2000) Detection, quantification, and tracking of vortices using streamline geometry. *Comput & Graph* 24(3):333–341. [https://doi.org/10.1016/s0097-8493\(00\)00029-7](https://doi.org/10.1016/s0097-8493(00)00029-7)
47. Schmidhuber J (2015) Deep learning in neural networks: An overview. *Neural Netw* 61:85–117
48. Williams S, Hecht M, Petersen M, Strelitz R, Maltrud M, Ahrens J, Hlawitschka M, Hamann B (2011) Visualization and analysis of eddies in a global ocean simulation. *Comput Graph Forum* 30(3):991–1000. <https://doi.org/10.1111/j.1467-8659.2011.01948.x>
49. Wu Q (2014) Region-shrinking: A hybrid segmentation technique for isolating continuous features, the case of oceanic eddy detection. *Remote Sens Environ* 153:90–98. <https://doi.org/10.1016/j.rse.2014.07.026>
50. Xu G, Cheng C, Yang W, Xie W, Kong L, Hang R, Ma F, Dong C, Yang J (2019) Oceanic eddy identification using an AI scheme. *Remote Sens* 11(11):1349
51. Zhang L, Zhang L, Du B (2016) Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geosci Remote Sens Mag* 4(2):22–40. <https://doi.org/10.1109/MGRS.2016.2540798>
52. Zheng G, Li X, Zhang RH, Liu B (2020) Purely satellite data-driven deep learning forecast of complicated tropical instability waves. *Sci Adv* 6(29):eaba1482. <https://doi.org/10.1126/sciadv.aba1482>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits any noncommercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if you modified the licensed material. You do not have permission under this license to share adapted material derived from this chapter or parts of it.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Deep Convolutional Neural Networks-Based Coastal Inundation Mapping from SAR Imagery: with One Application Case for Bangladesh, a UN-defined Least Developed Country



Bin Liu, Xiaofeng Li, and Gang Zheng

1 Introduction

Flooding is a severe natural disaster. It can be caused by various reasons. In the coastal areas, tropical cyclone-induced coastal flooding is the combined effect of storm surge-caused sea water inundation and rainfall-induced freshwater flooding. If tropical cyclone-induced flooding occurs at the same time as the rainy season, the consequences will be even more serious. If flooding occurs in locations with dense populations and large cities, it will result in huge loss of life and property [36]. For example, on August 26–28, 2017, Harvey lingered over the Houston area, a densely populated place, dumping massive amounts of rain. There were over 80 fatalities as a result of the extraordinary flooding [33]. Harvey produced over 125 billion dollars in damage, according to the National Hurricane Center.

Coastal flooding may become considerably severe in the future as a result of climate change and anthropogenic activities. For starters, greater temperatures may lead to more moisture in the atmosphere, enhancing the intensity of the flood [34]. Climate warming had increased the average and extreme rainfall of storms Katrina,

B. Liu

College of Marine Sciences, Shanghai Ocean University, Shanghai 201306, China

B. Liu · G. Zheng

State Key Laboratory of Satellite Ocean Environment Dynamics, Second Institute of Oceanography, Ministry of Natural Resources, Hangzhou 310012, Zhejiang, China

B. Liu

Key Laboratory of Marine Ecological Monitoring and Restoration Technologies, Ministry of Natural Resources, Shanghai 200137, China

X. Li (✉)

CAS Key Laboratory of Ocean Circulation and Waves, Institute of Oceanology, Chinese Academy of Sciences, Qingdao 266071, China

e-mail: lxf@qdio.ac.cn

Irma, and Maria, according to Patricola and Wehner's simulations [23]. Human activities, according to Bilskie et al. [2], can exacerbate the impact of coastal inundation on infrastructure. The study [38] found that urbanization worsened both the flood response and the total rainfall from hurricanes. These studies should raise our awareness of increased flooding in highly urbanized and densely populated coastal areas, both in developed and developing countries.

Accurate flood mapping can help emergency managers create more focused disaster response strategies, as well as researchers better understand flooding dynamics and study on more accurate forecasting methods. Ground surveys or information retrieval from remote sensing imagery can be used to map floods. Ground surveys are direct and exact, but they are expensive, and certain regions are inaccessible to humans after flooding. Flood mapping from remote sensing data is a means of low cost, and it could map areas human cannot access. The first remote sensing data source is optical data. The optical images are easy for human to interpret and then use. Optical sensors do not work at night and cannot see through cloud. This limits the applicability of optical remote sensing in information extraction during flooding. The second data source is radar remote sensing, especially the synthetic aperture radar (SAR) remote sensing with the ability of providing high-resolution images. SAR is an useful remote sensing tool for flood mapping since it can imaging floods at any time of day or night and in almost any weather condition. This ability is especially useful for mapping the dynamic flooding to understand flooding mechanisms and provide disaster relief plans.

Traditional flood mapping techniques using SAR data rely on image processing techniques that use backscattering, statistical, and polarimetric information. These methods include histogram thresholding [3], active contour segmentation [13], region growing [21], change detection [9, 20], statistical classification [10], neuro-fuzzy classification [6], multi-temporal statistics [4], pixel-based supervised [35], and object oriented rule-based classification [25]. Although traditional methods have achieved good results in some cases and some of them are even used in practical applications, they mine multi-dimensional SAR data using human-crafted features and rules to achieve flood mapping. It is difficult for human-crafted features and rules to guarantee stable performance under a variety of influences, including: (1) speckle; (2) temporal mis-registration; (3) imaging system parameters [22]; (4) meteorological factors [12, 18]; and (5) environmental conditions.

Deep learning (DL) technology, particularly deep convolutional neural network (DCNN) models, offers a promising route for reliable flood mapping. Instead of being pre-defined, the features for reliable flood classification in the DCNN models are mined from the multi-dimensional SAR data directly. These data-driven models are capable of offering reliable characteristics under a variety of influencing conditions, and they are optimized from data to information in an end-to-end style. This concept has been proven in a variety of communities, including computer vision [29], biomedical image processing [7] and geoscience [15, 26, 39]. DCNN-based methods for flood mapping have been proposed recently. Kang et al. [14] demonstrated that a fully convolutional network, which is one type of DCNN model, can produce more precise flooding mapping than previous approaches. Rudner et al. [28] presented a

DCNN-based method for retrieving flooded built-up areas that shows promise. We [18] presented an modified DCNN method for coastal flooding mapping from multi-temporal dual-polarimetric SAR data that offers reliable results, and this method is suitable for spatial and temporal investigation of storm-caused coastal flooding. For flooding mapping in built-up areas from high-resolution SAR imagery, Li et al. [16] presented an active self-learning DCNN model.

We believe the DCNN models can overcome the difficulties of robust flooding mapping, based on our past research [18, 19]. The DCNN-based SAR coastal flooding mapping network (SARCFMNet) is described in this chapter. SARCFMNet is a model designed specifically for coastal flooding mapping. It has two improvements that increase accuracy and robustness: (1) the physics-aware input information design fuses temporal and polarimetric information for more reliable mapping and integrates radar remote sensing mechanisms of flooding extraction into DCNN; (2) the regularization scheme useful for fully-convolutional networks enhance the model's reliability. The SARCFMNet was trained and tested using a dataset of coastal flooding in Houston, Texas, induced by Hurricane Harvey in 2017. The flooded regions, which cover around 4000 km², are delineated and studied in these images. The contributions of this study are listed as follows:

- Compared to the commonly used, benchmarking DCNN approach, the SARCFMNet performs better and is more stable. This demonstrates that the design of physics-aware input information and the regularization scheme can improve the performance and reliability.
- The spatial and multi-temporal characteristics of the Harvey-caused inundation are investigated using the mapping results.
- The wind influence is revealed, implying that DCNN models considering wind impact could improve reliability in practice.
- The cost-sensitive losses for DCNN models are investigated, which might be beneficial for more adaptive models that take performance costs into account.
- The trained and tested SARCFMNet model is applied to Bangladesh, which is one of the United Nations (UN)-defined least developed countries, to get nation-level, multi-year, high-temporal-resolution flooding maps. This can help us get deeper understanding of the flooding mechanism of this country.

The chapter is organized as follows. In Sect. 2, we will introduce the dataset used for the model training and testing. The model is described in Sect. 3. In Sect. 4, the model performances are presented. In Sect. 5, the multi-year flooding maps of Bangladesh are given with discussions. Section 6 concludes this chapter.

2 Dataset

2.1 Data Description

The dataset used for the training and testing of the SARCFMNet model is collected from Sentinel-1 SAR data during the Hurricane Harvey. Around the end of August 2017, Hurricane Harvey caused damage on the Houston region. Six pairs of Sentinel-1 SAR images were obtained in the research place during this time. The images are with VH and VV polarizations. One pair is from the Stripmap (SM) mode, while five pairs the Interferometric Wide (IW) swath mode. The products of Ground Range Detected are utilized. Table 1 lists the data parameters in the dataset. The IW01 pair's post-event image is impacted by strong wind. Harvey had degraded to a Tropical Storm by the time this image was taken, but it still delivered powerful winds to the scene, with the speed of around 20 m s^{-1} [31]. We labeled the flooded regions as the ground truth using land-cover categories from Google Earth and OpenStreetMap and Copernicus Emergency Management Service Rapid Mapping products [5].

In Fig. 1, we give a visual illustration of one pair from the data constructing the dataset, the SM01 pair. In this figure, the first and second rows show the images of the VV and VH. In these two rows, the first and second columns show pre- and post-event images respectively. The OpenStreetMap and Google Earth image of the region are in the third row. Houston's western and southern areas are covered by the SM01 pair.

Table 1 Descriptions of the image pairs for generating the dataset in this study

	Pre-event time (MM-DD, hh:mm, UTC)	Post-event time (MM-DD, hh:mm, UTC)	Coverage (Up left down right)	Sample number
IW01	08-05, 00:26	08-29, 00:26	N29.5°, W96.75° N28.75°, W95.0°	1973
IW02	08-05, 00:26	08-29, 00:26	N31.0°, W97.25° N30.0°, W95.0°	1384
IW03	07-31, 00:18	09-05, 00:18	N30.5°, W95.25° N29.5°, W92.75°	1769
IW04	08-24, 12:22	09-05, 12:22	N30.5°, W97.0° N29.0°, W94.75°	1746
IW05	08-18, 12:22	08-30, 12:22	N30.5°, W97.0° N29.25°, W94.75°	2184
SM01	08-23, 12:30	09-04, 12:30	N30.25°, W96.25° N29.0°, W95.25°	993

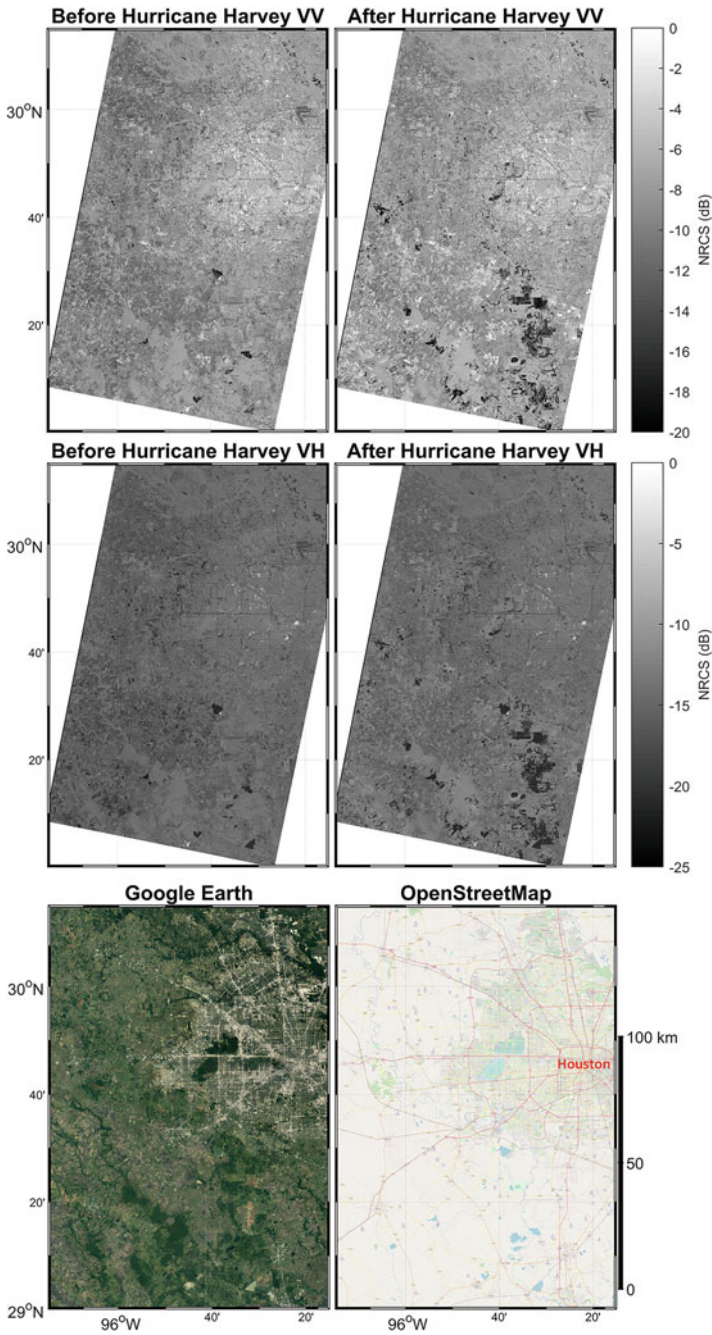


Fig. 1 Illustration of one image pair constructing the dataset

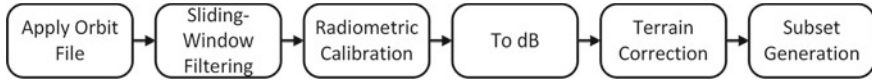


Fig. 2 Flowchart of data preprocessing

2.2 Data Preparation

The original SAR images are processed in the following steps, as illustrated in Fig. 2, to construct the dataset for model training and testing.

1. Application of orbit file: The accurate satellite orbit files are applied for the SAR products.
2. Filtering with sliding windows: To lessen the impact of speckle on SAR images, a filter is performed.
3. Radiometric calibration: After this calibration, the pixel values of the SAR images represent the back-scattering information (σ^0).
4. Conversion to dB: The linear scale σ^0 is converted to decibel (σ_{dB}^0). The normalized radar cross section (NRCS) images in dB are generated.
5. Terrain correction: The SAR images representing the σ_{dB}^0 information are geocoded into a geographical coordinate system, which is the World Geodetic System 84. The ocean is masked out with the digital elevation model (DEM) information. After this, each pixel occupies 8.9832×10^{-5} degrees.
6. Subset generation: The pre- and post-event images are transformed into the same coordinate system for each pair of data used to create the dataset. We trimmed the subsets from the pre- and post-event images by the same coverage.

The pre- and post-event images are geometrically matched after the preprocessing. We cut each pair into 256×256 , non-overlapping samples with pre- and post-multiple channels. For all the pairs, the sample numbers are shown in Table 1.

3 Model

The SARCFMNet model is specially tailored from the U-Net model [27] for its verified effectiveness. The U-Net model was proposed for biomedical image segmentation. Its architecture was created so that it could function with less training samples while still producing precise segmentations. The proposed SARCFMNet is shown in Fig. 3a. The network can be divided into two paths. The left path is an encoding path to extract abstracted features for accurate classification with down-sampling stage by stage. The right path is a decoding part to up-sample the feature maps. There are skip connections from the encoding to the decoding path to provide latter the high-resolution features via concatenation. As illustrated in Fig. 3a, the encoding phase consists of 3×3 convolutions activated by the rectified linear

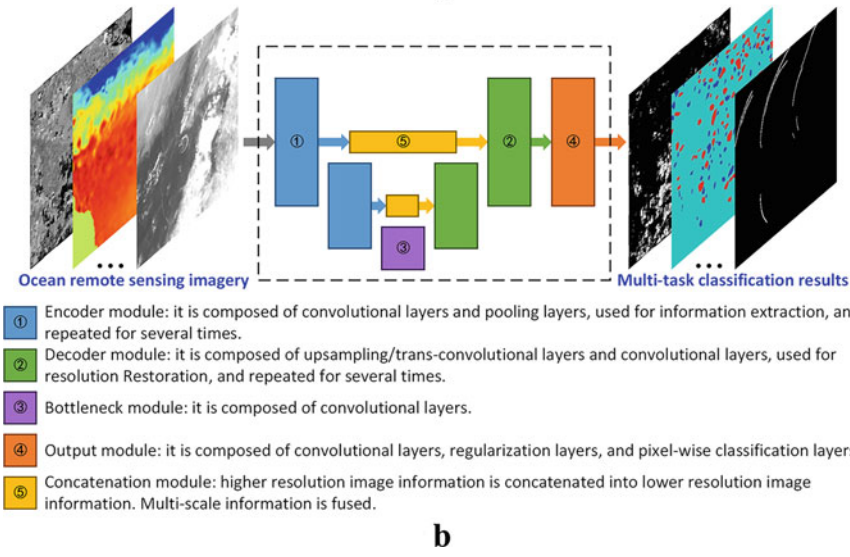
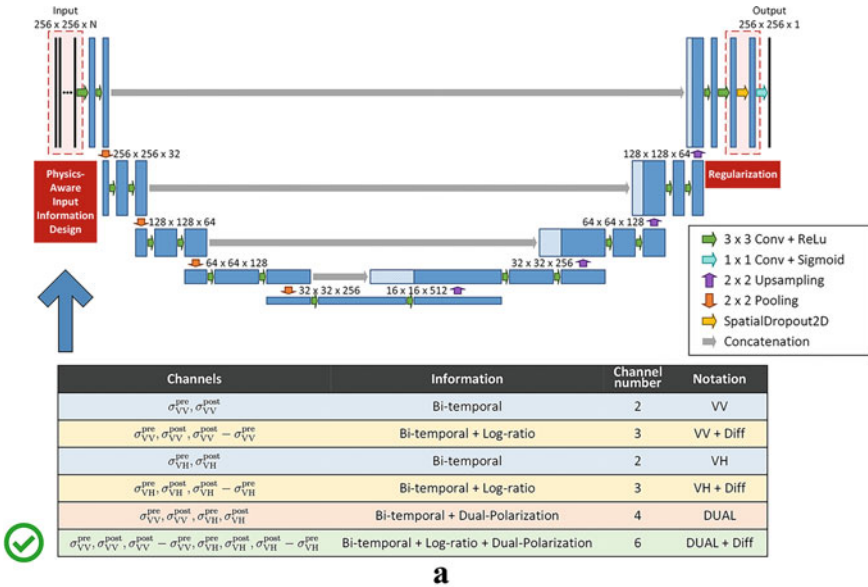


Fig. 3 The proposed model design. **a** The proposed SARCFMNet model structure. **b** The abstracted model design can be generalized to multiple ocean remote sensing image information mining problems

unit (ReLU) and 2×2 max-pooling. 3×3 convolutions with ReLU activation and 2×2 up-sampling operations constitute the decoding part. The output layer of the model is a Sigmoid-activated 1×1 convolution. Thus, this model can predict the probability of each pixel as flooding. The loss function of the model is the binary cross-entropy (BCE) loss [17]. Pixel-wise classification accuracy is used as metric to evaluate model performance.

In the SARCFMNet model, there are two specially-tailored modifications designed for coastal flooding mapping.

1. Physics-aware input information design—Defined by the problem, we design different input information combinations as in Fig. 3a. For the σ , its superscript indicates pre- or post-event information, and its subscript shows polarization. Bi-temporal information with pre- and post-event information from one single polarization is often used in flooding mapping [3, 10, 14]. This is a direct design. In this study, based on the radar remote sensing physics, we propose that the VV and VH polarization information should be fused, since the two polarizations can compensate each other. In addition, we propose the temporal difference images should also be used. From Sect. 2.2, we know that the preprocessed images represent the backscattering information in the log-scale. Therefore, the temporal difference images $\sigma_{VV}^{\text{post}} - \sigma_{VV}^{\text{pre}}$ and $\sigma_{VH}^{\text{post}} - \sigma_{VH}^{\text{pre}}$ represent the log-ratio information for VV and VH, respectively. From the previous studies [1], we know the log-ratio is useful for SAR image change detection. Based on radar remote sensing physics knowledge, the SARCFMNet model fuses temporal, log-ratio, and polarization together, denoted as DUAL+Diff. This approach can increase the accuracy and reliability of the DCNN model, making it appropriate for coastal flooding mapping from SAR remote sensing data. The fused input information sources are integrated as a data cube. This design can realize information fusion with little parameter increasing.
2. DCNN-suitable regularization design—For DCNN models, such as the proposed SARCFMNet, the models' ability to generalize is limited by model overfitting. When a model overfits, it might produce excellent results during the training phase but bad results when used in practice. Dropout is a suitable scheme to avoid overfitting for fully connected networks, although it is not so helpful for convolutional layers [8]. From the network design, we can find out there are no fully connected layers in the model. This is a fully convolutional model. In this case, we should use a dropout means which is effective for convolutional layers. Here, we include the SpatialDropout2D (SD2D) layer to leverage channel-level dropout to accomplish regularization and increase the model's generalization ability, as inspired by Tompson et al. [30].

The model can be generalized for multiple problems. The model can be abstracted as a design in Fig. 3b. There are five modules: (1) module 1 for encoding; (2) module 2 for decoding, module; (3) module 3 for generating high-level bottleneck features; (4) module 4 for outputting predictions with adaptive processes; (5) module 5 for fusing

feature information between skip connections. With suitable information input and specially-tailored modifications, this abstracted model can realize multi-task pixel-level ocean remote sensing image information mining [15].

4 Performance Evaluations and Discussions

4.1 Performance Evaluations

The six image pairs yield 10049 samples, as shown in Table 1. We randomly choose roughly 20% of the samples in each pair to create a sub-dataset with 2000 samples. This dataset is denoted as the S2000 dataset. The S2000 dataset is used for model training. During the training process, 70% (1400 samples) are randomly selected for training, and the other 30% (600 samples) are selected for validation. There is a hyperparameter for the SpatialDropout2D layer, that is the dropout rate. We set the dropout rate as 0.5. Thus, the results from the model with the SpatialDropout2D can be identified by `_SD2D0.5`. The model training and testing are implemented by the software framework Keras. The optimizer for the model fitting is Adam. The batch size is 32. The total number of epochs is 300. The validation set determines the model parameters. We use one Nvidia GeForce GTX 1080Ti graphics processing unit (GPU) card. The training time on the S2000 dataset is about 6.7 hours.

The losses and accuracies for the training and validation are documented and analyzed. The readers can find the details in [18]. The conclusions drawn from the analyses are listed here.

1. In all the settings, the models are fully trained. With the indication of validation loss, the models try not to overfit.
2. The usefulness of the log-ratio information—From the performance comparison, the inclusion of the log-ratio information can improve the model's performance for coastal inundation mapping.
3. The usefulness of the dual-polarization fusion—From the performance comparison, the fusion of the polarization information can improve the model's performance for coastal inundation mapping. In addition, the VH polarization can get better performance than the VV polarization. The possible reason is that VH is less sensitive to the wind condition during the flooding mapping. We will discuss this later.
4. The usefulness of the DCNN-suitable regularization design—With the regularization layer suitable for the fully convolutional model, although the performance decreases in the training processing, the performance increases in the validation process. This indicates the regularization design can make the model more reliable.

On the dataset created in Sect. 2.2, the SARCFMNet trained on the S2000 dataset is applied. The results are given in Table 2. The input data and regularization scheme

Table 2 The trained SARCFMNet model’s performance on the dataset

	VV+Diff (Accuracy Recall Precision F1 score)	VH+Diff (Accuracy Recall Precision F1 score)	DUAL+Diff (Accuracy Recall Precision F1 score)	DUAL+Diff_SD2D0.5 (Accuracy Recall Precision F1 score)
IW01	0.9677	0.9651	0.9754	<u>0.9810</u>
	0.8045	0.7226	0.8388	0.8849
	0.8922	0.9489	0.9313	0.9393
	0.8461	0.8204	0.8826	<u>0.9113</u>
IW02	0.9861	0.9897	<u>0.9916</u>	0.9907
	0.8947	0.7831	0.9087	0.9172
	0.7970	0.9647	0.8932	0.8678
	0.8430	0.8645	<u>0.9009</u>	0.8918
IW03	0.9350	0.9685	0.9684	<u>0.9741</u>
	0.6395	0.8996	0.9184	0.9541
	0.6507	0.7889	0.7793	0.8025
	0.6451	0.8406	0.8432	<u>0.8718</u>
IW04	0.9817	<u>0.9932</u>	0.9904	0.9912
	0.9143	0.8529	0.9301	0.8721
	0.6593	0.9324	0.8066	0.8598
	0.7661	<u>0.8909</u>	0.8640	0.8659
IW05	0.9647	0.9719	0.9728	<u>0.9824</u>
	0.8888	0.9151	0.9641	0.9397
	0.6193	0.6763	0.6718	0.7790
	0.7300	0.7778	0.7918	<u>0.8518</u>
SM01	0.9866	0.9823	0.9912	<u>0.9915</u>
	0.8580	0.6750	0.8816	0.8909
	0.8867	0.9907	0.9492	0.9467
	0.8721	0.8029	0.9142	<u>0.9180</u>
Weighted Average by sample number proportion	<u>0.9681</u>	<u>0.9771</u>	<u>0.9800</u>	<u>0.9842</u>
	0.8306	0.8219	0.9098	0.9118
	0.7363	0.8649	0.8230	0.8575
	0.7737	0.8313	0.8583	<u>0.8815</u>

are indicated by the column names. The subset names are indicated by the row names. There are four numbers in each block of the table. Classification accuracy, recall, precision, and F1 score are listed in that sequence. The ratio of true positives to the total number of true positives and false negatives is recall. A greater recall score indicates that the model misses fewer areas that are actually flooded. The ratio of true positives to the sum of true positives and false positives is precision. A greater precision score indicates that the model is less likely to produce incorrect flooding areas. The F1 score is the harmonic mean of precision and recall, and it leans to the lower value within precision and recall. The weighted average is shown in the last row. The number of samples in each subset determines the weights. The best accuracy and F1 are emphasized by underline. From observation, the block with the best accuracy has the best F1 score. From this table, we can draw the consistent conclusions as shown above:

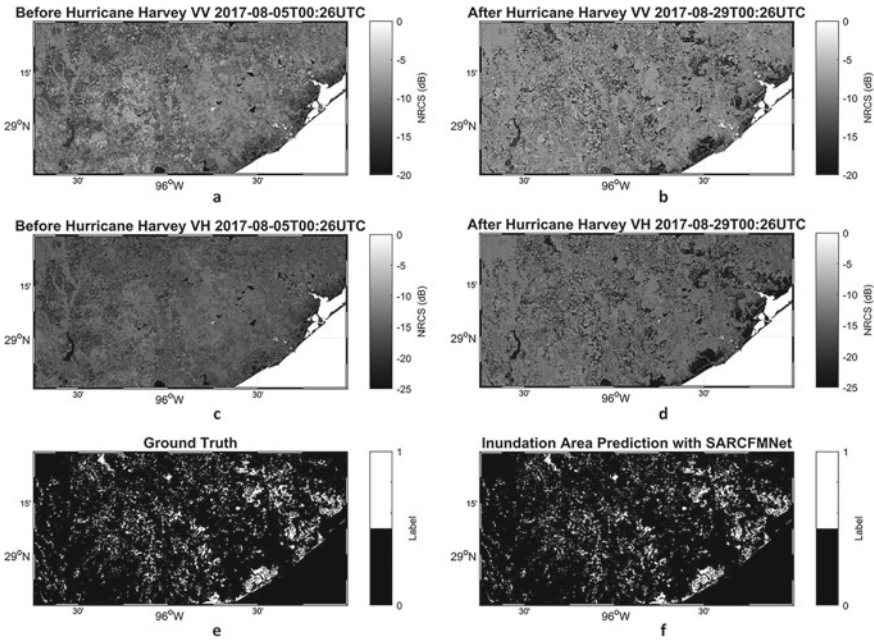


Fig. 4 Visual evaluation on the IW01 subset. **a** and **b** are the pre- and post-event NRCS images for the VV. **c** and **d** are the pre- and post-event NRCS images for the VH. **e** Ground truth. **f** Mapping prediction with the model (DUAL+Diff_SD2D0.5)

1. The fusion of dual-polarization information improves the model’s coastal inundation mapping performance.
2. The model gets better performance on VH polarization than VV polarization.
3. The DCNN-suitable regularization improves performance and robustness of the model.

The visual evaluation on the IW01 subset, used as an example for presentation, is shown in Fig. 4. In this figure, the first and second rows show the images of the VV and VH. In these two rows, the first and second columns show pre- and post-event images, respectively. The ground truth and flooding prediction using the model (DUAL+Diff_SD2D0.5) are shown in Fig. 4e, f. By comparing Fig. 4e, f, we can observe that the mapping prediction is very close to the ground truth, indicating that the presented model is effective.

4.2 Spatial and Temporal Characteristics

After we apply the trained SARCFMNet model on the dataset described in Sect. 2, we can analyze the spatial and temporal characteristics of 2017 Harvey-induced coastal inundation.

The image pair SM01 is selected as a case for the geospatial analysis. The predicted flooding mapping is shown in Fig. 5a. In order to perform the geospatial analysis, we collect useful supporting data. They are shown in Fig. 6. The supporting data include: (1) The elevation data of the scene, from the United States Geological Survey (USGS) National Elevation Dataset [32], and shown in Fig. 6a; (2) The land cover types of the scene, from the 2016 National Land Cover Database (NLCD) [37], and shown in Fig. 6b with legend; (3) the historical water occurrence data of the scene, from the Global Surface Water Mapping Dataset (1984–2015) [24], and shown in Fig. 6c.

We can derive certain geospatial analytic findings with the mapping predictions and supporting data:

1. General analysis—In this scene, the total flooding area is about 284 km² (about 3% of the scene). We use a disk-shape average filter (radius = 100 pixels) to process the flooding map, and create a flooding heat map for the scene. In Fig. 5b, the heat map is placed onto the pre-event image. In the southern part of Houston, severely flooded regions are densely scattered, as shown in the heat map.
2. Relation with elevation—The elevation distribution of the flooded and non-flooded areas in the scene is analyzed and shown in Fig. 7a. It shows the elevation distribution of the flooded areas is different from that of the non-flooded areas, and

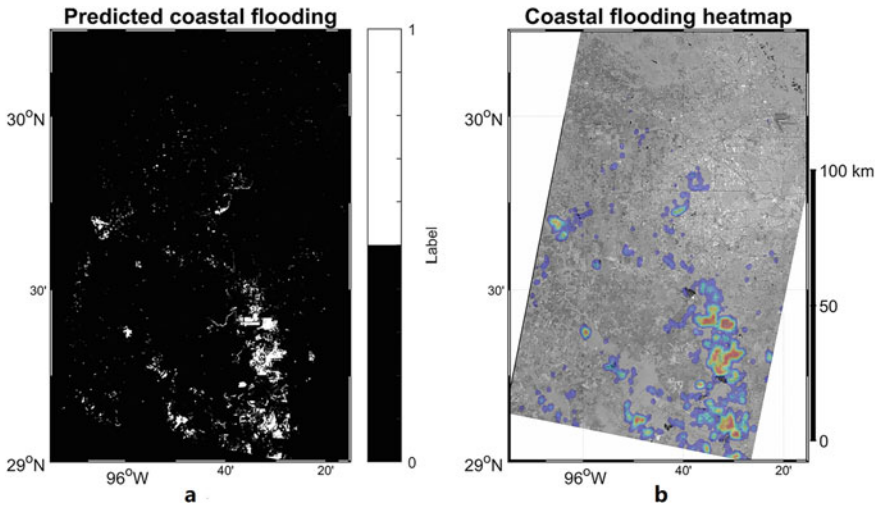


Fig. 5 Subset SM01 for the geospatial analysis. **a** Coastal inundation prediction from the SAR-CFMNet model. **b** Inundation heat map generated from the prediction

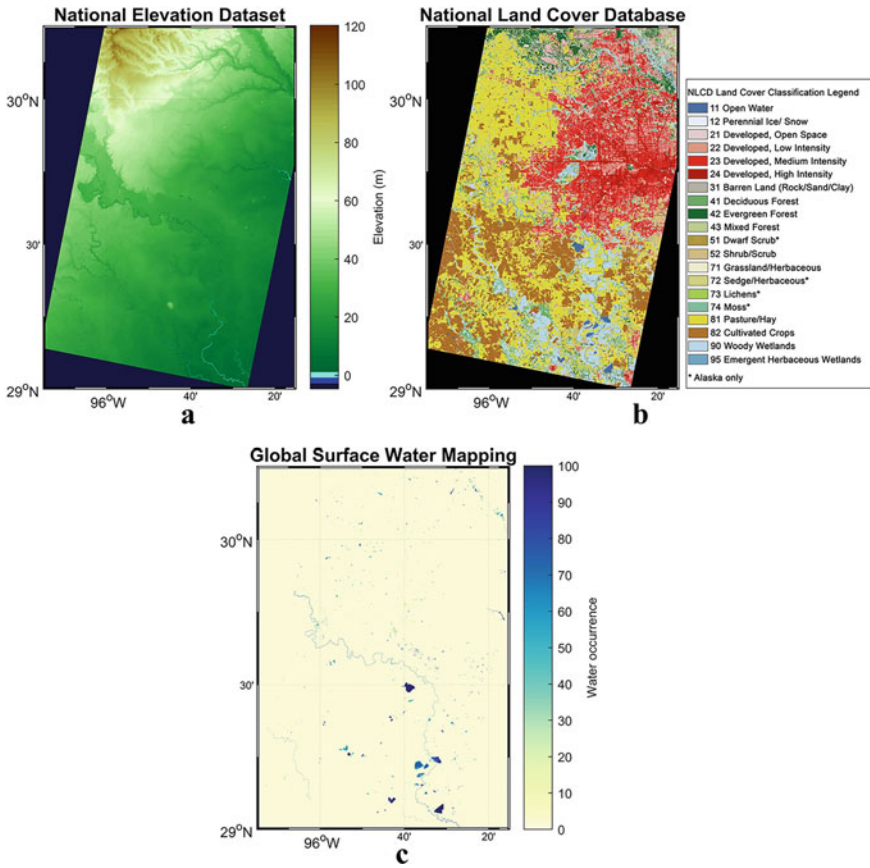


Fig. 6 Supporting data for the spatial analysis. **a** The elevation data of the scene, from the United States Geological Survey (USGS) National Elevation Dataset; **b** The land cover types of the scene, from the 2016 National Land Cover Database (NLCD); **c** the historical water occurrence data of the scene, from the Global Surface Water Mapping Dataset (1984–2015)

the former is obviously lower. It is more likely that flooding occurs and remains in lower regions, in this scene, the southern part.

3. Relation with land cover types—The proportion of land cover types affected by the flooding is illustrated in Fig. 7b. It demonstrates that pasture and cultivated crops are the dominant land cover types in flooded regions. They account for more than 76% of flooding. They are the main land cover types in the southern part which is severely flooded. The flooding may severely damage local agriculture. However, we have to realize that even if the hurricane caused severe flooding in the urban areas, the inner city flooding cannot be easily extracted by pure image-based analysis.

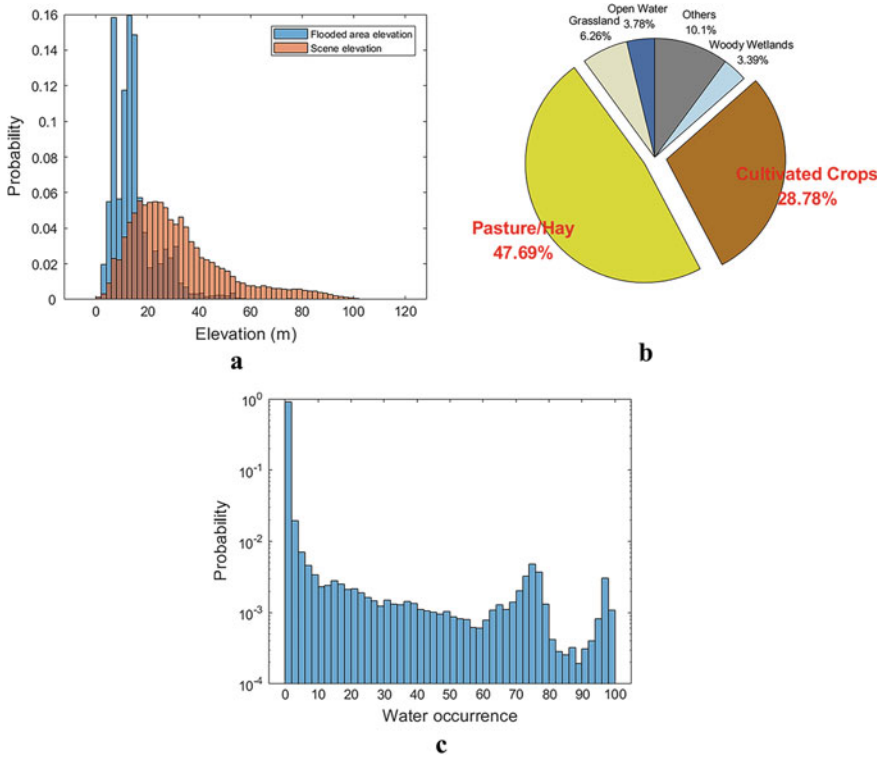


Fig. 7 Geospatial analysis for the SM01 subset. **a** The elevation distribution of the flooded and non-flooded areas. **b** The proportion of land cover types affected by the flooding. **c** The historical water occurrence of the flooded areas

4. Relation with historical water occurrence—The flooding is extracted from analyzing the pre- and post-event images. We have to be sure that the flooding is not caused by seasonal or periodic surface water increasing. For the flooded areas, the historical water occurrence is analyzed and shown in Fig. 7c. It reveals that, in flooded areas, the historical water occurrence is extremely close to zero. It signifies that the predicted flooding is abnormal, and people should be alert to it.

For SM01, IW01, IW04, and IW05, the mapping products have an overlapping region. The multi-temporal study of the mapping results will be performed in this region. Figure 8 depicts the temporal analysis. Figure 8a shows a Moderate Resolution Imaging Spectroradiometer (MODIS) image of Harvey in August 26, 2017. The region for temporal analysis is illustrated as the green rectangle. The flooding duration probability of the overlapping zone is shown in Fig. 8b. It can assist us in comprehending the temporal evolution of floods. The locations with the highest probability are likely to be the last to vanish. Pixels with a probability < 0 lack all of the mapping products needed for temporal analysis.

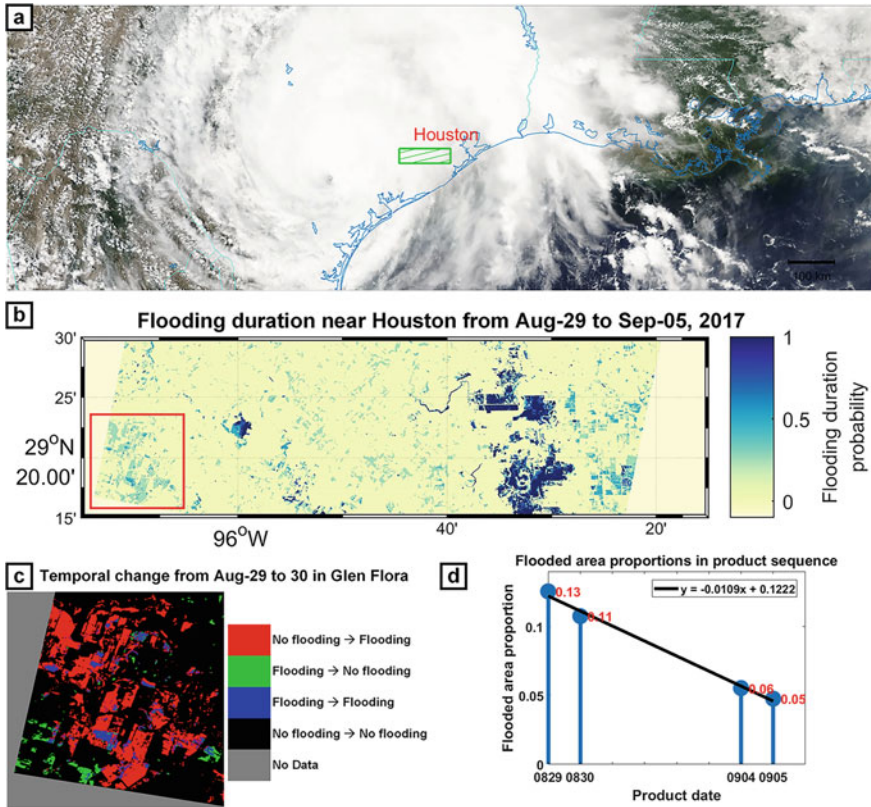


Fig. 8 Temporal analysis of SARCFMNet-generated flooding maps of IW01 (August 29), IW05 (August 30), SM01 (September 4), and IW04 (September 5). **a** A Moderate Resolution Imaging Spectroradiometer (MODIS) image shows Harvey in August 26, 2017. **b** The flooding duration probability of the region, which is illustrated as green rectangle in **a**. **c** Temporal flooding transition from IW01 to IW05 of the region, which is illustrated as red rectangle in **b**. **d** flooding area proportions

Figure 8d shows flooding area proportions of the product sequence, IW01 (August 29), IW05 (August 30), SM01 (September 4), and IW04 (September 5). It shows how the flooded areas in the region reduce over time as the product sequence progresses. We may calculate that the shrinkage rate is around 1% of the region area (roughly 23 km²) each day using regression analysis.

We discover a phenomenon of delayed flooding after the examination of the product sequence. One area does not show flooding in IW01 (August 29), but shows flooding in IW05 (August 30). In Fig. 8b, a region is marked in a red rectangle. In Fig. 8c, the temporal flooding transition from IW01 to IW05 is examined. The region is in Glen Flora, Texas. We check the news [11] and discover that the Colorado River (Texas) began flowing through and across the region on the evening of August 29

(local time). This is why the flooding is captured by IW05 (sensing time: 12:22 UTC, August 30), but not by IW01 (sensing time: 00:26 UTC, August 29). The reason for the delayed flooding deserves further studying.

4.3 Discussions of Performance

In this part, we discuss the performance of the proposed model in two aspects: first, the influence of wind; and second, cost-sensitive losses to adjust the performance.

The influence of wind is a factor seldom discussed in flooding mapping. However, in the case of storm-induced coastal flooding mapping, this is a practical issue. In order to map coastal flooding, we may encounter the following scenario: the storm has already generated coastal flooding, which is captured by remote sensing data; nevertheless, the storm has not yet left the scene and is still delivering strong winds. In this case, the wind can have adverse effects on inundation mapping, since the strong wind increases the water areas' backscattering. In this study, we also face this situation. Strong wind influences the IW01's post-event image, as described in Sect. 2.1. We use a toy example to demonstrate the impact of wind on the performance of the DCNN model.

400 samples are chosen from IW01 and IW03 to create IW01_selected and IW03_selected. Then, using the DUAL+Diff architecture, we train two models on IW01_selected and IW03_selected, and test them on IW01 and IW03. There are four scenarios here: (1) IW01_selected training, IW01 testing; (2) IW01_selected training, IW03 testing; (3) IW03_selected training, IW01 testing; (4) IW03_selected training, IW03 testing. Table 3 contains their results. The column names in this table denote the training subsets, whereas the row names denote the testing subsets. The numbers in each block indicate classification accuracy, recall, and precision. The numbers on the table's diagonal show training scene is the same as testing scene. The performances are excellent for obvious reasons. Outside of the table's diagonal,

Table 3 Toy experiment results of wind influence

	Trained on IW01_selected (Accuracy Recall Precision)	Trained on IW03_selected (Accuracy Recall Precision)
Tested on IW01	0.9875 0.9405 0.9459	0.9599 <u>0.6722</u> 0.9502
Tested on IW03	0.9596 0.9252 <u>0.7186</u>	0.9887 0.9441 0.9347

performance drops sharply. The model of IW01_selected training and IW03 testing presents low precision. We may deduce from the aforementioned information that the IW01_selected's post-event imaging is impacted by severe wind. In this situation, the subset IW01_selected convinces the model to find flooding areas with higher backscattering in the post-event image. This will cause false positive predictions and lower precision, if the model is tested on IW03. Based on the similar logics, we may understand the model of IW03_selected training and IW01 testing presenting low recall.

Based on the explanations from the toy example, we can get a better understanding of the total performance evaluation listed in Table 2. We can have two observations.

1. The model trained on VH polarization has better performance than that trained on VV polarization. The possible explanation is VH is less sensitive to the wind conditions.
2. The S2000 dataset is created from data of different wind conditions, the model trained on the S2000 dataset performs better in terms of balance. However, the results, particularly those tested on IW01 and IW03, still show the impact of wind. This tells us, in the future research, the DCNN models should be aware of the wind conditions. It is a direction to further improve the performances.
 - a. Since VH is less sensitive to wind conditions, the model can only use VH polarization. However, we can not deny that VV has its own advantages for flooding mapping. Maybe this is a design with much information loss.
 - b. The wind information can be input together with the image information, and the dual-polarization information fusion can be realized in a more flexible way.

In the deep learning-based paradigm for image understanding, the loss functions play an important role. They set the end rules for the models, making the predictions close to targets. The closeness is measured by losses. In this study, the BCE loss is useful and suitable for binary classification. The BCE loss can be adjusted according to user-defined costs. Accordingly, the performances will be adjusted. We use the toy experiments of two models with the DUAL+Diff design, one of IW01_selected training and IW01 testing, and one of IW03_selected training and IW03 testing.

The BCE loss is used first, and the results are shown in Table 4. The numbers in each block are classification accuracy, recall, and precision. From the first row, BCE is capable of balancing accuracy and recall.

In real applications, the users may have personalized needs, higher recall or higher precision. These personalized needs can be understood as cost-defined requests. If users believe that the cost of low recall is very great, the model must improve recall at the price of precision. Based on the similar logics, if users believe that the cost of poor precision is very great, the model must improve precision at the price of recall. To meet these requirements, cost-sensitive losses are utilized. The type-defined weighted α -balanced BCE (α BBCE) loss [17] is one technique to build cost-sensitive losses:

$$L_{\alpha\text{BBCE}} = -\frac{1}{N} \sum_{i=1}^N \{ \alpha y_i \log \hat{y}_i + (1 - \alpha)(1 - y_i) \log (1 - \hat{y}_i) \} \quad (1)$$

Table 4 Toy experiment results of cost-sensitive losses

	Trained on IW01_selected Tested on IW01 (Accuracy Recall Precision)	Trained on IW03_selected Tested on IW03 (Accuracy Recall Precision)
BCE	0.9875 0.9405 0.9459	0.9887 0.9441 0.9347
α BBCE ($\alpha = 0.8$) Weight introduced into BCE	0.9785 <u>0.9858</u> 0.8453	0.9846 <u>0.9848</u> 0.8670
$F\beta$ ($\beta = 2$) Directly use F1-score loss with weight	0.9741 <u>0.9724</u> 0.8242	0.9540 <u>0.9808</u> 0.6720
$F\beta$ ($\beta = 0.5$) Directly use F1-score loss with weight	0.9835 0.8838 <u>0.9635</u>	0.9798 0.8374 <u>0.9372</u>

where the i th pixel’s label is denoted as y_i ; the prediction is denoted as \hat{y}_i ; the pixel number is denoted as N ; the weight is denoted as $\alpha \in [0, 1]$. The accuracy of flooding is given more weight during training as α is higher. Given that the flooding pixel number is significantly less than non-flooding pixel number, a larger value for α is appropriate. The value of α in this experiment is 0.8. The α BBCE loss is effective, as seen in Table 4’s second row. Because the accuracy of flooding is given more weight during training, recall is increased at the price of precision.

Another technique to build a cost-sensitive loss is to utilize the $F\beta$ score directly:

$$L_{F\beta} = 1 - \underbrace{(1 + \beta^2) \frac{P \cdot R}{\beta^2 \cdot P + R}}_{F\beta \text{ score}} \tag{2}$$

where R and P is recall and precision, respectively. β is a positive real weight. Minimizing the $F\beta$ loss can increase the $F\beta$ score. If β is greater than 1, optimizing recall receives more attention during training. If β is less than 1, precision optimization is given more attention. This is clearly a more direct way of controlling the recall and precision in the results by their importance. The 3rd and 4th rows of Table 4 show the results of the $F\beta$ loss, $\beta = 2$ and $\beta = 0.5$, respectively. The results confirm that the $F\beta$ loss is an effective way for adjusting recall and precision in the results according to their importance: (1) For $\beta = 2$, recall is increased at the price of precision; (2) For $\beta = 0.5$, precision is increased at the price of recall.

Here, we use toy examples to show the design of cost-sensitive losses. There are two points should be aware of for designing these losses.

1. There is a performance tradeoff between recall and precision (there is no free lunch).
2. There is one more hyper-parameter should be pre-defined. This one more hyper-parameter gives us more control over the performance.

5 Application Case in Bangladesh

Bangladesh is a participating country in the Belt and Road Initiative, and it is one of the UN-defined least developed countries. Bangladesh is located on the coast of the Indian Ocean and has a low-lying terrain. Under the influence of the rainy season and tropical cyclones, severe flooding occurs every summer, especially from June to October. Flooding poses a huge threat to the safety of people's lives and property in the country, and has become an obstacle to the country's development. This chapter uses the SARCFMNet model to carry out a nation-level, multi-year, high-temporal-resolution flooding mapping of Bangladesh from 2016 to 2020. This can deepen our understanding of the flooding mechanism in Bangladesh, and provide powerful technology and data support for disaster mitigation and flood forecasting.

In order to provide the nation-level, multi-year, high-temporal-resolution flooding mapping products for Bangladesh from 2016 to 2020, we use the following processing for Sentinel-1 data based on the preprocessing introduced in Sect. 2.2.

1. For each year, we select images from February to March of that year to put together a nation-level pre-event image.
2. For each year, from a time window of each month from June to October, we select images to put together a nation-level post-event image.
3. The SARCFMNet model trained on the S2000 dataset is performed on the image pairs to get the nation-level flooding mapping result.

From the aforementioned steps, we provide 45 nation-level flooding mapping results:

1. Year 2016: From June to October, one nation-level flooding map is provided every month.
2. Year 2017: From June to October, two nation-level flooding maps are provided every month, the first and second halves of the month.
3. Year 2018: From June to October, two nation-level flooding maps are provided every month, the first and second halves of the month.
4. Year 2019: From June to October, two nation-level flooding maps are provided every month, the first and second halves of the month.
5. Year 2020: From June to October, two nation-level flooding maps are provided every month, the first and second halves of the month.

In 2016, since the temporal resolution of Sentinel-1 SAR data is relatively low, there is one nation-level flooding map per month. From 2017 to 2020, there are two nation-level flooding maps per month.

The flooding maps have the following characteristics:

- Spatial extent: Bangladesh
- Temporal extent: 2016–2020

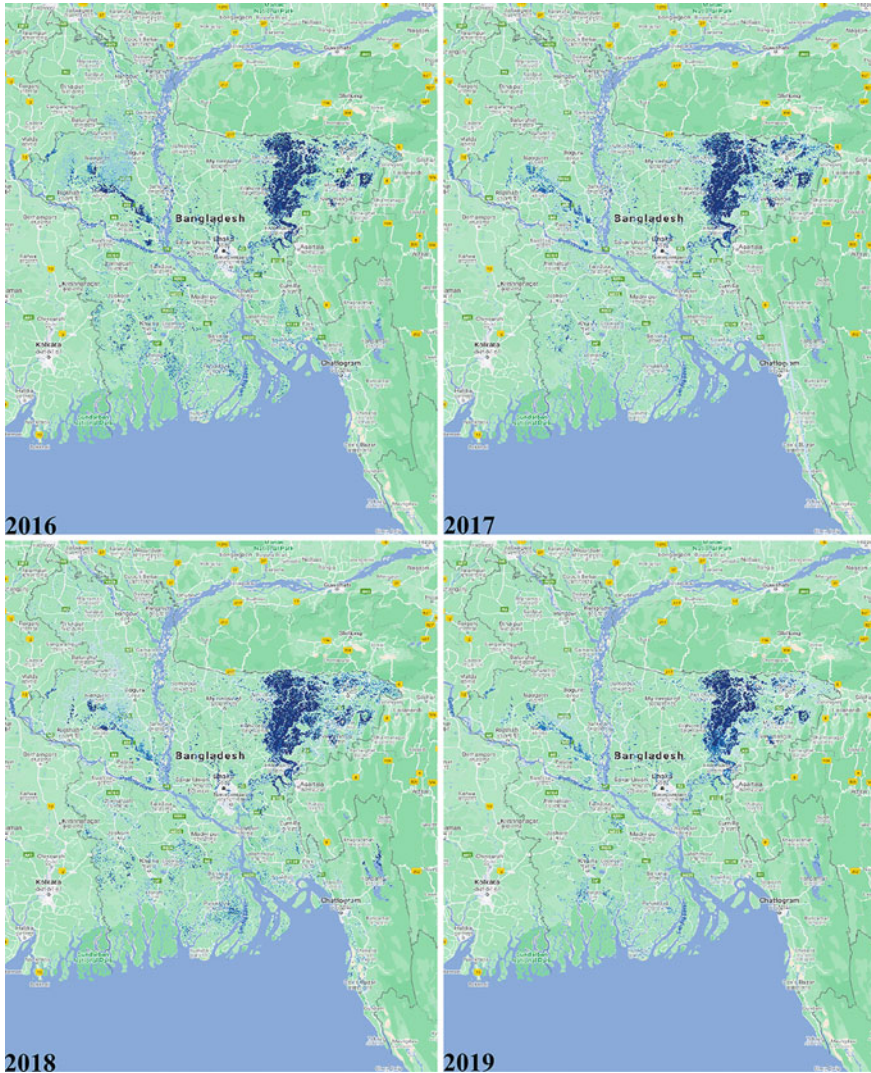
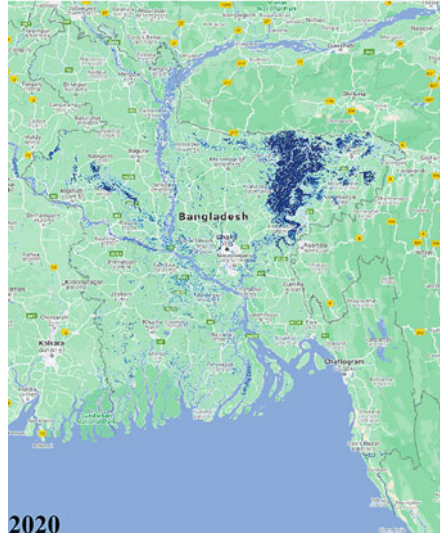


Fig. 9 Bangladesh nation-level flooding occurrence probability maps from 2016 to 2020. The flooding occurrence probability map is generated from the flooding maps of each year from June to October

Fig. 9 (continued)

- Spatial resolution: 3 arcsecond, consistent with main world-level DEM products
- Temporal resolution: half a month (For 2016, a month)

Based on the flooding maps, we first analyze the flooding occurrence probability each year, which is shown in Fig. 9. The flooding occurrence probability map is generated from the flooding maps of each year from June to October. This shows that, each year, the spatial distribution of high flooding occurrence probability is relatively stable. For each year, there are some flooded areas, which are not flooded areas for other years. Based on the products, these phenomena can be analyzed case by case.

Based on the flooding maps, we then analyze the flooding extent each year, which is shown in Fig. 10. From this analysis, we can get the following information:

1. We already know that the flooding mainly happens from June to October in Bangladesh due to rainy season and tropical cyclones. In the yearly flooding extent from 2016 to 2020, we can narrow down that the most severe flooded time window is from the second half month of July to the first half month of August.
2. For each year from 2016 to 2020, the peak flooding area is around $2e4 \text{ km}^2$.

The above two analyses just show the usefulness of the provided flooding maps. By using nation-level, multi-year, high-temporal-resolution flooding mapping products of Bangladesh from 2016 to 2020, we can perform more spatial and temporal, targeted analyses. Hopefully, this can deepen our understanding of the flooding mechanism in Bangladesh, and provide powerful information support for disaster mitigation and flood forecasting.

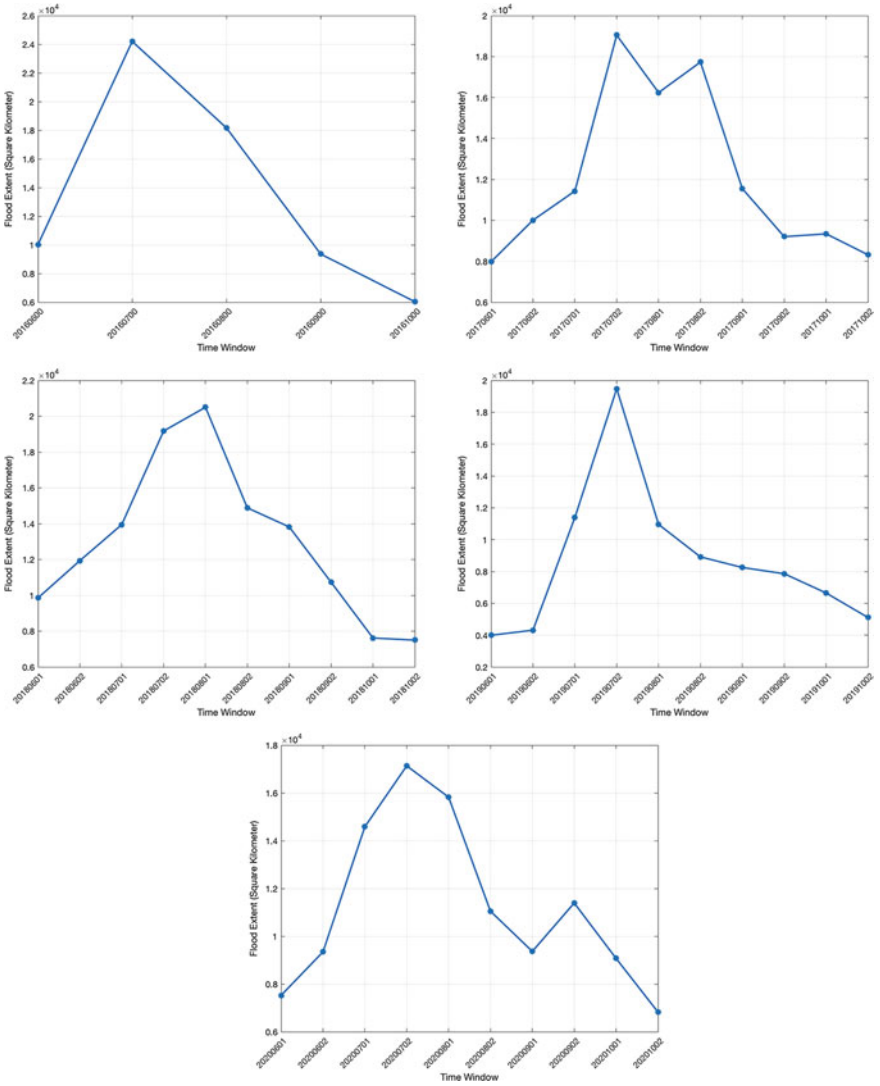


Fig. 10 Bangladesh nation-level flooding extent area from 2016 to 2020. In 2016, in the label of x-axis, the 00 after month means there is one flooding map product each month. From 2017 to 2020, in the label of x-axis, the 01 after month means the product of the first half of the month, and 02 means the second

6 Conclusions

The SARCFMNet model of mining multi-temporal and dual-polarimetric SAR data for coastal inundation classification is presented in this chapter. The SARCFMNet is built on U-Net, a benchmarking deep learning model for pixel-level classification that we have modified for the challenges of coastal inundation mapping from SAR imagery: 1) radar remote sensing physics-driven input information design; and 2) regularization suitable for fully convolutional networks. We present two study cases in this chapter. First, the SARCFMNet is trained and evaluated using a dataset derived from 2017 Hurricane Harvey-influenced Houston, Texas. Six image pairs, with ground truth delineated by human with the help of Google Earth and OpenStreetMap, are used to test the proposed SARCFMNet model. The average mapping accuracy and F1 score are 0.98 and 0.88, respectively. They are better than the benchmarking deep learning model for pixel-level classification. This verifies the usefulness of the proposed designs. The geospatial study of Harvey-caused floods is performed using the flooding predictions and indicates Harvey's massive impact on agriculture. The multi-temporal study estimates the flooding decreasing rate and uncovers a delayed-inundation phenomenon. Second, the trained and verified SARCFMNet model is applied to Bangladesh, which is one of the UN-defined least developed countries, to get nation-level, multi-year, high-temporal-resolution flooding maps. The flooding maps of Bangladesh are from 2016 to 2020, with spatial resolution of 3 arc second and temporal resolution of half a month (for 2016, a month). This can help us get deeper understanding of the flooding mechanism of this country. In addition, impact of meteorological factors in DCNN-based flooding mapping models and cost-sensitive losses are discussed. We propose that this model can be easily and readily generalized to other multi-temporal ocean remote sensing imagery information mining problems.

Acknowledgements This work is supported by the National Natural Science Foundation of China (No. 42006159), and the Open Fund of the State Key Laboratory of Satellite Ocean Environment Dynamics, Second Institute of Oceanography, Ministry of Natural Resources (No. QNHX2020 and No. QNHX2238). The authors thank the ESA for Sentinel-1 data and the SNAP software, GEE for the USGS National Elevation data, NLCD data, and Global Surface Water Mapping data, and Copernicus EMS for the Rapid Mapping products.

References

1. Bazi Y, Bruzzone L, Melgani F (2005) An unsupervised approach based on the generalized Gaussian model to automatic change detection in multitemporal SAR images. *IEEE Trans Geosci Remote Sens* 43(4):874–887
2. Bilskie MV, Hagen SC, Medeiros SC, Passeri DL (2014) Dynamics of sea level rise and coastal flooding on a changing landscape. *Geophys Res Lett* 41(3):927–934
3. Chini M, Hostache R, Giustarini L, Matgen P (2017) A hierarchical split-based approach for parametric thresholding of SAR images: flood inundation as a test case. *IEEE Trans Geosci Remote Sens* 55(12):6975–6988

4. Cian F, Marconcini M, Ceccato P (2018) Normalized difference flood index for rapid flood mapping: taking advantage of EO big data. *Remote Sens Environ* 209:712–730
5. Copernicus EMS Rapid Mapping (2019). <https://emergency.copernicus.eu/mapping/list-of-components/EMSR229>
6. Dasgupta A, Grimaldi S, Ramsankaran R, Pauwels V, Walker J (2018) Towards operational SAR-based flood mapping using neuro-fuzzy texture-based approaches. *Remote Sens Environ* 215:313–329
7. Falk T, Mai D, Bensch R, Çiçek Ö, Abdulkadir A, Marrakchi Y, Böhm A, Deubner J, Jäckel Z, Seiwald K et al (2019) U-Net: deep learning for cell counting, detection, and morphometry. *Nat Methods* 16(1):67–70
8. Ghiasi G, Lin TY, Le QV (2018) DropBlock: a regularization method for convolutional networks. [arXiv:1810.12890](https://arxiv.org/abs/1810.12890) [cs]
9. Giustarini L, Hostache R, Matgen P, Schumann GJP, Bates PD, Mason DC (2012) A change detection approach to flood mapping in urban areas using TerraSAR-X. *IEEE Trans Geosci Remote Sens* 51(4):2417–2430
10. Giustarini L, Hostache R, Kavetski D, Chini M, Corato G, Schläffer S, Matgen P (2016) Probabilistic flood mapping using synthetic aperture radar data. *IEEE Trans Geosci Remote Sens* 54(12):6958–6969
11. Glen Flora Emporium (2019). <https://heartofatexan.com/stories/hurricane-harvey-glen-flora-emporium>
12. Henry JB, Chastanet P, Fellah K, Desnos YL (2006) Envisat multi-polarized ASAR data for flood mapping. *Int J Remote Sens* 27:1921–1929
13. Horritt MS, Mason DC, Luckman AJ (2001) Flood boundary delineation from synthetic aperture radar imagery using a statistical active contour model. *Int J Remote Sens* 22(13):2489–2507
14. Kang W, Xiang Y, Wang F, Wan L, You H (2018) Flood detection in Gaofen-3 SAR images via fully convolutional networks. *Sensors* 18(9):2915
15. Li X, Liu B, Zheng G, Ren Y, Zhang S, Liu Y, Gao L, Liu Y, Zhang B, Wang F (2020) Deep learning-based information mining from ocean remote sensing imagery. *Nat Sci Rev* 7(10):1584–1605
16. Li Y, Martinis S, Wieland M (2019) Urban flood mapping with an active self-learning convolutional neural network based on TerraSAR-X intensity and interferometric coherence. *ISPRS J Photogramm Remote Sens* 152:178–191
17. Lin T, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: 2017 IEEE international conference on computer vision (ICCV), pp 2999–3007
18. Liu B, Li X, Zheng G (2019) Coastal inundation mapping from bitemporal and dual-polarization SAR imagery based on deep convolutional neural networks. *J Geophys Res: Ocean* 124(12):9101–9113
19. Liu B, Li X, Zheng G (2020) Automatic mapping of tropical cyclone-induced coastal inundation in SAR imagery based on clustering of deep features. In: 2020 IEEE international geoscience and remote sensing symposium (IGARSS), pp 5765–5768
20. Long S, Fatoyinbo TE, Policelli F (2014) Flood extent mapping for Namibia using change detection and thresholding with SAR. *Environ Res Lett* 9(3):035002
21. Matgen P, Hostache R, Schumann G, Pfister L, Hoffmann L, Savenije HHG (2011) Towards an automated SAR-based flood monitoring system: lessons learned from two case studies. *Phys Chem Earth Parts A/B/C* 36(7–8):241–252
22. O’Grady D, Leblanc M, Bass A (2014) The use of radar satellite data from multiple incidence angles improves surface water mapping. *Remote Sens Environ* 140:652–664
23. Patricola CM, Wehner MF (2018) Anthropogenic influences on major tropical cyclone events. *Nature* 563(7731):339–346
24. Pekel JF, Cottam A, Gorelick N, Belward AS (2016) High-resolution mapping of global surface water and its long-term changes. *Nature* 540(7633):418
25. Pradhan B, Tehrany MS, Jebur MN (2016) A new semiautomated detection mapping of flood extent from TerraSAR-X satellite image using rule-based classification and Taguchi optimization techniques. *IEEE Trans Geosci Remote Sens* 54(7):4331–4342

26. Reichstein M, Camps-Valls G, Stevens B, Jung M, Denzler J, Carvalhais N, Prabhat, (2019) Deep learning and process understanding for data-driven Earth system science. *Nature* 566(7743):195–204
27. Ronneberger O, Fischer P, Brox T (2015) U-Net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention. Springer, pp 234–241
28. Rudner TGJ, Rußwurm M, Fil J, Pelich R, Bischke B, Kopačková V, Biliński P (2019) Multi3Net: segmenting flooded buildings via fusion of multiresolution, multisensor, and multitemporal satellite imagery. *Proc AAAI Conf Artif Intell* 33:702–709
29. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M (2015) ImageNet large scale visual recognition challenge. *Int J Comput Vis* 115(3):211–252
30. Tompson J, Goroshin R, Jain A, LeCun Y, Bregler C (2014) Efficient object localization using convolutional networks. [arXiv:1411.4280](https://arxiv.org/abs/1411.4280) [cs]
31. Underground W (2019). <https://www.wunderground.com/hurricane/atlantic/2017/tropical-storm-harvey>
32. USGS National Map (2019). <https://www.usgs.gov/core-science-systems/national-geospatial-program/national-map>
33. van Oldenborgh GJ, van der Wiel K, Sebastian A, Singh R, Arrighi J, Otto F, Hausteijn K, Li S, Vecchi G, Cullen H (2017) Attribution of extreme rainfall from Hurricane Harvey, Aug 2017. *Environ Res Lett* 12(12):124009
34. Vahedifard F, AghaKouchak A, Jafari NH (2016) Compound hazards yield Louisiana flood. *Science* 353(6306):1374
35. Voormansik K, Praks J, Antropov O, Jagomägi J, Zalite K (2013) Flood mapping with TerraSAR-X in forested regions in estonia. *IEEE J Sel Top Appl Earth Obs Remote Sens* 7(2):562–577
36. Woodruff JD, Irish JL, Camargo SJ (2013) Coastal flooding by tropical cyclones and sea-level rise. *Nature* 504(7478):44–52
37. Yang L, Jin S, Danielson P, Homer C, Gass L, Bender SM, Case A, Costello C, Dewitz J, Fry J et al (2018) A new generation of the United States national land cover database: requirements, research priorities, design, and implementation strategies. *ISPRS J Photogramm Remote Sens* 146:108–123
38. Zhang W, Villarini G, Vecchi GA, Smith JA (2018) Urbanization exacerbated the rainfall and flooding caused by hurricane Harvey in Houston. *Nature* 563(7731):384–388
39. Zheng G, Li X, Zhang RH, Liu B (2020) Purely satellite data-driven deep learning forecast of complicated tropical instability waves. *Sci Adv* 6(29):eaba1482

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits any noncommercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if you modified the licensed material. You do not have permission under this license to share adapted material derived from this chapter or parts of it.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Sea Ice Detection from SAR Images Based on Deep Fully Convolutional Networks



Yibin Ren, Xiaofeng Li, Xiaofeng Yang, and Huan Xu

1 Introduction

The changes in global sea ice volume, distribution, and movement reflect the interaction of the atmosphere-cryosphere-hydrosphere and the global climate change [30]. Sea ice study is also significant because it causes marine navigation and transportation safety concerns. Since the classification of sea ice and open water provides valuable information for safe navigation, sea ice classification and monitoring draw extensive attention [8, 37, 39]. Satellite remote sensing, such as optical camera, microwave radiometer, and synthetic aperture radar (SAR), has been the most effective way to monitor sea ice in the polar regions [21, 40]. SAR images have been the primary source for sea ice classification and monitoring, due to its high spatial resolution, wide-coverage, and ability to penetrate clouds [7].

Series of studies have been devoted to classifying sea ice and open water on SAR images, including threshold-based methods, expert systems, and machine learning methods. Multi-Year Ice (MYI) Mapping System (MIMS) is a typical threshold-based model, and it can quickly map MYI in uncalibrated SAR images [13]. The representation of expert systems is the Advanced Reasoning using Knowledge for Typing Of Sea ice (ARKTOS) [38]. ARKTOS performs a fully automated analysis

Y. Ren · X. Li (✉)

CAS Key Laboratory of Ocean Circulation and Waves, Institute of Oceanology,
Chinese Academy of Sciences, Qingdao 266071, China
e-mail: lixf@qdio.ac.cn

X. Yang

State Key Laboratory of Remote Sensing Science, Aerospace Information Research Institute,
Chinese Academy of Sciences, Beijing 100101, China

Key Laboratory of Earth Observations of Hainan Province, Sanya 572029, China

H. Xu

School of Geomatics and Marine Information, Jiangsu Ocean University,
Lianyungang 222005, China

© The Author(s) 2023

X. Li and F. Wang (eds.), *Artificial Intelligence Oceanography*,
https://doi.org/10.1007/978-981-19-6375-9_12

of SAR sea ice images by mimicking the reasoning process of sea ice experts. For machine learning methods, the regression model is an early exploration. Lundhaug and Maria [29] proposed a multivariate regression method to model the relationship between the mean and standard deviation of the backscattering coefficients and air temperatures with sea ice types and water. Their experiments showed the correlation coefficients between predicted and actual values were higher than 0.90. Karvonen [20] developed a modified pulse-coupled neural network (PCNN) to classify the sea ice in the Baltic Sea. Zhang et al. [47] proposed a k-means-based model which combines microwave scatterometer and radiometer data to classify sea ice types. Zakhvatkina [45] extracted textural features from the gray-level co-occurrence matrix (GLCM) and input the features into an artificial neural network (ANN)-based model to classify sea ice and the open water. Similarly, researchers combined the GLCM with other machine learning algorithms, such as Markov random field (MRF) [6] and support vector machine (SVM) [25] to classify sea ice from SAR images.

Overall, the main drawback of the aforementioned traditional methods is that they need prior expert knowledge and sophisticated manual engineering to extract features for discriminating between sea ice and open water. This drawback has been a common challenge faced by the earth system science in the era of big data [33].

Deep learning (DL) technology addresses the mentioned challenge [19]. A typical DL model consists of deep neural networks (DNN), which accepts input data in a raw format and automatically discover the required features [24]. In recent years, DL has been successfully applied in oceanography, geography, and remote sensing, which has helped humans gain further process understanding of earth system science problems [27, 32–34, 43]. A deep convolution neural network (CNN) is a particular type of DNN composed of CNN layers. A CNN layer connects to the local patches of the previous layer through convolution kernels to extract local spatial features [22]. Since CNN-based methods have achieved great success in image classification, researchers employed CNN to extract features automatically to improve the accuracy and efficiency of sea ice classification. Yan and Scott [44] introduced an early CNN-based model AlexNet [2], and transfer learning to classify sea ice and open water. Li et al. [26] proposed a CNN-based model to classify sea ice and open water from Chinese Gaofen-3 SAR images. Wang et al. [42] constructed a CNN model consists of three CNN layers and two fully connected neural network layers to classify sea ice near the Bering Strait. [16] integrated transfer learning and dense CNN blocks to form a transferred multilevel fusion network (MLFN). The MLFN outperformed the PCAKM [5], the NBRELM [15], and the GaborPCANet [12] in classifying sea ice and open water.

More and more researchers are trying to construct DL-based models to achieve end-to-end classification between sea ice and open water. Though the aforementioned DL-based models deliver excellent performances, several issues still exist. First, classification accuracy needs to be further improved. Especially for the medium-high resolution SAR images, fine-grained objects such as small floes, sinuous ice-water boundaries, and ice channels need to be well classified. Second, the information of SAR images, such as dual-polarization information and incident angle (IA), are not fully utilized by most DL-based models. The benefit of fusing dual-polarized

information has been demonstrated in the conventional method [25], and the IA affects the radar backscattering intensity. All this information should be considered to improve classification accuracy. Third, most of the existing models are validated by independent images, and their applicability to more challenging tasks, such as classifying a series of images from freezing to melting, remains to be verified.

Aiming to solve the issues mentioned above, we propose a dual-attention U-Net model, DAU-Net, to classify sea ice and open water on SAR images. U-Net was initially developed for the semantic segmentation of biomedical images [35]. It is designed to work with fewer training samples but is still able to yield precise segmentations. The effectiveness of employing U-Net to solve classification or segmentation problems of geoscience has been demonstrated [11, 28, 46]. Therefore, we use the U-Net as the backbone of the classification model. The dual-polarized information and the IA of SAR images are utilized as the model inputs. To extract more characteristic features from the multiple input information, we integrate the dual-attention mechanism [14] to optimize the origin U-Net. Finally, we use SAR images near the Bering Sea to train and evaluate the model. We validate the applicability of DAU-Net by a series of SAR images of Bering Strait and compare the classification results with the sea ice products of the National Snow and Ice Data Center (NSIDC).

2 Data

2.1 Study Area

The study areas are the Bering Sea and Bering Strait, which locates near the out edge of the sea ice on the Pacific side of the Arctic (Fig. 1). The Bering Strait is the only channel for water exchange between the Pacific Ocean and the Arctic Ocean, showing strong atmosphere-sea-ice interactions and supports one of the world's most productive and valuable fisheries with ever increasing commercial vessel activities [9]. Therefore, sea ice detection and monitoring in this region are of great interest to scientific research communities and commercial fishing and transportation industries.

2.2 SAR Images

The SAR images are obtained from Sentinel-1A in the interferometric wide-swath (IW) mode with a swath width of 250 km. The images are the ground range detected (GRD) products with VV + VH (vertical emitting and vertical and horizontal receiving, respectively) polarizations. The IA is between 30.00-46.00 degrees. The range and azimuth resolutions are 5 and 20 m, respectively, with a sampling space of 10 m.

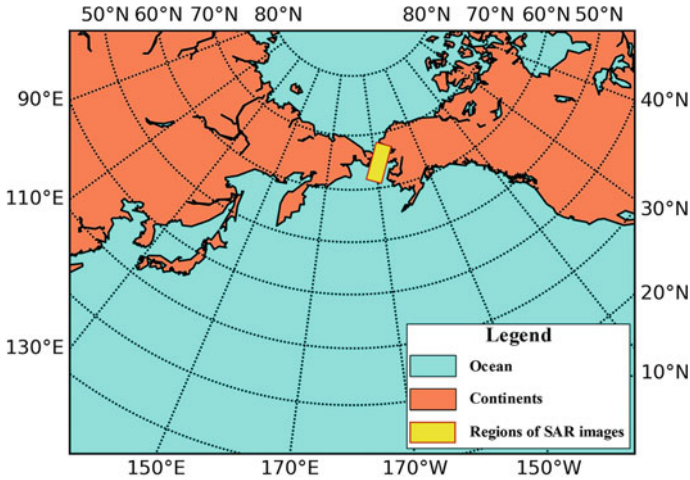


Fig. 1 The location of the study area

The data set consists of 34 SAR images as shown in Table 1, and is divided into three subsets: 1) the training set, 2) the testing set, and 3) the applicability validation set (Fig. 1). The model training set includes 15 images (No.1 – No.15 in Table 1). The testing set is the No.16 image in Table 1, and we used this image to evaluate the model performance by metrics. The applicability validation set is a series of images covering the Bering Strait. The series contains six images, each of which is mosaicked from three single Sentinel-1A images, a total of 18 Sentinel-1A images (No.17–No.34 in Table 1). The image series covers the whole ocean process from freezing to melting of the Bering Strait. Therefore, we could validate the applicability of the well-trained model by monitoring the entire cycle of sea ice in the Bering Strait.

2.3 NSIDC Sea Ice Products

The sea ice products of the NSIDC [41], named Multisensor Analyzed Sea Ice Extent - Northern Hemisphere (MASIE-NH), are employed as a reference for the applicability discussion. The product is based on the Interactive Multisensor Snow and Ice Mapping System (IMS) results produced by the National Ice Center (NIC). NIC utilizes visible imagery, passive microwave data, and NIC weekly analysis products to create their data product. MASIE-NH provides measurements of daily sea ice extent and sea ice edge boundary for the Northern Hemisphere and 16 Arctic regions in a polar stereographic projection at both 1 and 4 km grid cell sizes [41]. We choose the 1 km MASIE-NH products as the reference.

Table 1 Information of the SAR images

No.	Imaging Date	Center Location	Function
1	13/12/2018	169.13W, 63.21N	training
2	25/12/2018	169.78W, 61.74N	
3	25/12/2018	169.13W, 63.21N	
4	06/01/2019	169.78W, 61.74N	
5	18/01/2019	169.78W, 61.74N	
6	04/02/2019	171.77W, 61.87N	
7	06/02/2019	167.82W, 61.43N	
8	11/02/2019	169.78W, 61.74N	
9	14/03/2019	167.19W, 62.93N	
10	14/03/2019	167.84W, 61.48N	
11	19/03/2019	169.78W, 61.74N	
12	24/03/2019	171.77W, 61.87N	
13	26/03/2019	167.20W, 62.93N	
14	31/03/2019	170.35W, 60.25N	
15	24/04/2019	169.13W, 63.21N	
16	24/04/2019	169.78W, 61.74N	testing
17	13/12/2018	166.90W, 67.66N	applicability validating
18	13/12/2018	167.72W, 66.17N	
19	13/12/2018	168.42W, 64.70N	
20	25/12/2018	166.90W, 67.66N	
21	25/12/2018	167.72W, 66.17N	
22	25/12/2018	168.42W, 64.70N	
23	31/03/2019	166.90W, 67.66N	
24	31/03/2019	167.72W, 66.17N	
25	31/03/2019	168.42W, 64.70N	
26	12/04/2019	166.90W, 67.66N	
27	12/04/2019	167.72W, 66.17N	
28	12/04/2019	168.42W, 64.70N	
29	24/04/2019	166.90W, 67.66N	
30	24/04/2019	167.72W, 66.17N	
31	24/04/2019	168.42W, 64.70N	
32	06/05/2019	166.90W, 67.66N	
33	06/05/2019	167.72W, 66.17N	
34	06/05/2019	168.42W, 64.70N	

2.4 Data Preprocessing

We use SNAP 3.0 [10] to perform radiometric calibration and boxcar filtering on all SAR images. As the size of the source SAR image is too large, we downscale each image to 1/3 of the original image size, about $8,000 \times 5,000$ pixels. Although the spatial resolution is downscaled from 10 m to 30 m, it is still much higher than that of the MASIE-NH products (1 km). It is far more detailed than could be expected from existed manual or operational automatic classifiers [25]. We scale all pixel values to 0-1. All IA values are scaled to 0-1, referred by 0° - 90° .

The SAR images are labeled into two classes, 1 for sea ice and 0 for open water, by the annotation tool LabelMe [36] to obtain the ground truth labels. As the resolutions of existing sea ice products are much lower than that of the Sentinel-1A images [25], the labeling process is based on visual interpretation. For regions that are difficult to distinguish, we refer to the 1 km MASIE-NH products to label them. In this way, most of the pixels in the SAR images could be labeled correctly. Due to the limitations of SAR image noise and manual labeling, there are inevitably a few mislabeling pixels, and some small sea ice objects cannot be accurately labeled. This is a common problem in the supervised learning field. For most classification missions, such mislabeled pixels account for a small proportion of all pixels and do not affect the convergence of the model [17].

We divide all images (VV, VH, and IA) into 256×256 -pixel chips as the model inputs. Fig. 2 takes the VV channel as an example to show the SAR image chips and the corresponding ground truth labels.

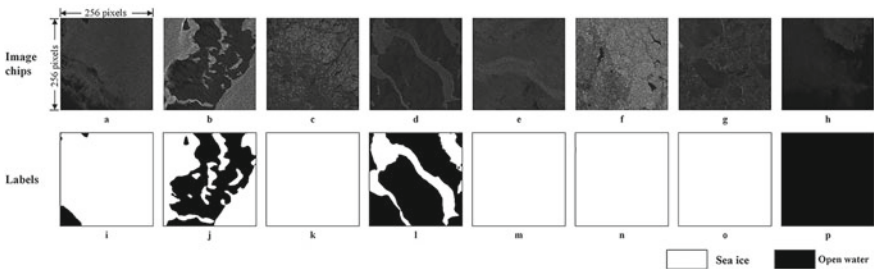


Fig. 2 Image chips (VV channel) and the corresponding labels. **a-h** SAR image chips with 256×256 -pixel. **i-p** Labels corresponding to the **a-h** SAR images

3 Method

3.1 Overall Structure of DAU—Net

The backbone of the proposed DAU-Net is a U-Net model. The U-Net is named for its almost symmetric encoder-decoder network architecture like a “U” shape and is designed to work with fewer training samples but still able to yield precise segmentations. The encoder extracts abstracted, downscaled high-level feature maps. The decoder restores the resolution of the high-level feature maps. The intermediate feature maps extracted by encoder and decoder are connected to form multi-scale feature maps for pixel-level classifications. The encoder can be a mature DNN model, such as VGG16, ResNet18, ResNet34, etc [4, 23].

Discriminant feature representations are essential for improving classification accuracy. To achieve a high accuracy classification between sea ice and open water in medium-high resolution SAR images, we need more characteristic features to discriminate fine-grained objects such as small floes, sinuous ice-water boundaries, and ice channels. Therefore, we integrate a dual-attention mechanism into the original U-Net and form a DAU-Net model to improve the feature representations of sea ice and open water. The dual self-attention mechanism means position attention module (PAM) and channel attention module (CAM), which could capture the long-range dependencies in spatial and channel dimensions. It has been demonstrated effective in classical image segmentation [14].

The PAM captures long-range dependencies in spatial dimension by a self-attention mechanism. For a feature map, the feature value at a specific position is updated by aggregating feature values at all positions with a weighted summation. The weights are determined by the feature similarities between the corresponding two positions. Any two positions with similar features can contribute to mutual improvement regardless of their distance in the spatial dimension. Similarly, the CAM employs the self-attention mechanism to capture the channel dependencies between any two-channel maps. Each channel map is updated by a weighted sum of all channel maps. Finally, the outputs of these two attention modules are fused to enhance the feature representations further.

Overall, as shown in Fig. 3, the DAU-Net consists of five parts: inputs, encoder, attention, decoder, and output. Each input unit consists of three channels of a 256×256 -pixel SAR image: VV, VH, and IA. The encoder is the ResNet-34, a mature model for image recognition, and it extracts abstracted, downscaled feature maps for accurate classification. The attention part performs position attention and channel attention on the extracted feature maps to capture long-range dependencies in spatial and channel dimensions. The outputs of the two attention modules are fused to form more characteristic features transmitted to the decoder. The decoder module rescales the downscaled feature maps to the original size. Skip connections link the encoder feature and decoder feature. Next, we will detail the encoder, attention, decoder, and output modules.

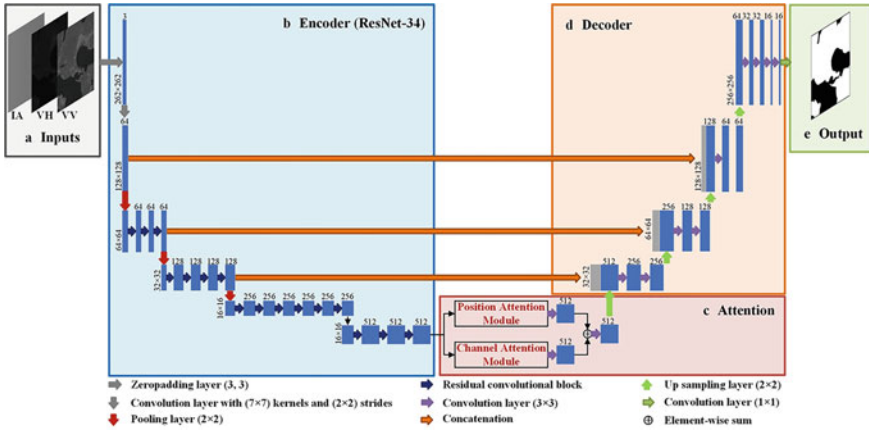


Fig. 3 Model design. **a** The model’s input: VV, VH, and IA channels. **b** The model’s encoder. **c** The attention modules. **d**. The model’s decoder. **e** The model’s output

3.2 Encoder

He et al. [18] proposed the residual network (ResNet) to increase the number of hidden CNN layers to more than one hundred. The ResNet family includes ResNet-18, ResNet-34, ResNet-50, and ResNet-101, where the number represents the number of CNN layers. Large numbers mean more CNN layers, more parameters, and more training complexity. The ResNet family has been widely used in semantic segmentation and object detection. Considering the depth of the model, the number of trainable parameters, and the complexity of sea ice texture, we choose ResNet-34 as the encoder for DAU-Net. The comparisons between the ResNet-34 and the other ResNet-based encoders are carried out in the F part of Section IV.

The encoder consists of 33 CNN layers of the ResNet-34, including five stages. The first stage is one CNN layer with 7×7 kernel size and 2×2 strides. After the first stage, the original image size is downscaled to 128×128 . The remaining four stages are composed of 3, 4, 6, and 3 ResNet blocks and a total of 16 ResNet blocks, Fig. 3. Each ResNet block contains two stacking CNN layers with a shortcut connection linking the input of the block and the output of the 2nd CNN layers [18]. The number of convolutional kernels in the five stages is 64, 64, 128, 256, and 512. The original ResNet34 model uses four 2×2 max-pooling layers that are stacked on four ResNet stages to downscale the feature map. Here, we discard the last max-pooling layer and retain the first three max-pooling layers. The activation function of each CNN layer is ReLU [1]. After encoding, the origin inputs are transformed into 512 16×16 feature maps. Following, these high-level features are transmitted to the attention part.

3.3 Attention

The 512 16×16 feature maps extracted by the encoder are fed into the PAM and CAM to capture spatial and channel dependencies. The outputs of these two attention modules are fused and transformed into the decoder.

3.3.1 PAM

Since CNN adopts local connection, the features captured by CNN are local. For semantic segmentation, local features generated by fully CNN are not representative enough, which could lead to misclassifications [31]. The PAM addresses this issue. The PAM updates the feature value at a specific position by aggregating feature values at all positions with a weighted summation. Thus, the global spatial dependencies of any two positions could be captured. These global features are fused with local features to form more characteristic features. Following, we will detail the calculation of PAM.

As shown in Fig. 4a., let H , W , and C represent the width, height, and channels, and $A \in \mathbb{R}^{H \times W \times C}$ is a local feature map extracted from the model inputs. The white/dark regions represent sea ice/water features. There are some inaccurate features in A , especially the regions marked by the red rectangle. Then A is fed into all three CNN layers to generate three feature

maps $B \in \mathbb{R}^{H \times W \times C}$, $C \in \mathbb{R}^{H \times W \times C}$, and $D \in \mathbb{R}^{H \times W \times C}$, as shown in Fig. 4b. B is reshaped to $B^1 \in \mathbb{R}^{N \times C}$, where $N = H \times W$ is the number of pixels. C is reshaped and transposed to $C^1 \in \mathbb{R}^{C \times N}$. Then, matrix multiplication is performed between B^1 and C^1 . Then, the multiplication result is activated by a softmax layer to calculate the spatial attention map $S \in \mathbb{R}^{N \times N}$. The softmax activation [3] normalizes S by row and makes the sum of each row is 1. The more similar feature representations of the two positions contribute to a higher correlation between them, generating a large value in S .

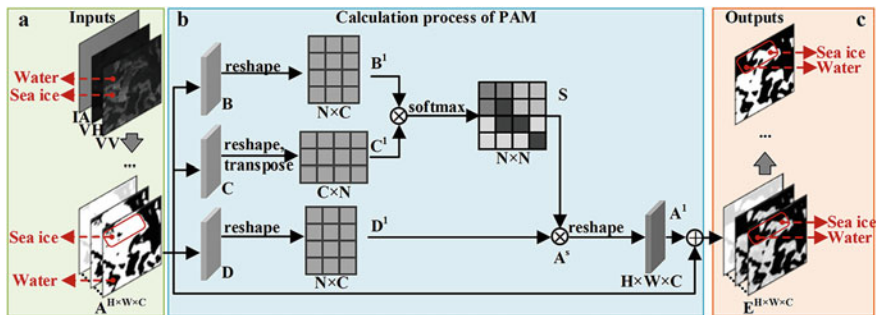


Fig. 4 Flow of PAM. **a** before PAM, some water pixels are misclassified. **b** PAM. **c** after PAM, the misclassified pixels are corrected

The global dependencies of any two positions in the feature map modeled by S . D is reshaped to $D^1 \in \mathbb{R}^{N \times C}$. S is multiplied by D^1 to generate $A^s \in \mathbb{R}^{N \times C}$:

$$a_{ij}^s = S_i \cdot D_j^1, j \in [1, C] \quad (1)$$

where a_{ij}^s is an element of A^s , S_i is the i_{th} row of S and D_j^1 is the j_{th} column of D^1 . A^s is reshaped to $A^1 \in \mathbb{R}^{H \times W \times C}$. For each channel of A^1 , the element of a position is the weighted sum of elements across all positions in the corresponding channel of D based on the weights in S . Therefore, A^1 has a global contextual view and selectively aggregates contexts according to the spatial attention map. A^1 is multiplied by a scale parameter α and added to the input feature map A in element-wise to obtain the output $E^{H \times W \times C}$:

$$E = \alpha A^1 + A \quad (2)$$

where α is initialized as 0 and gradually learns to assign more weight.

The pixel value of the output feature map E is a weighted sum of the features across all pixels and original features. E integrates the local features and the long-range global features. The similar semantic features achieve mutual gains, thus improving intra-class compact and semantic consistency. Intuitively, as shown in Fig. 4c, the inaccurate features in A are optimized by the PAM, which contributes to the final output.

3.3.2 CAM

Each channel map of high-level features can be regarded as a class-specific response, and different semantic responses are associated with each other. The CAM updates the feature value at a position by aggregating feature values of all channels in the same position with a weighted sum. The interdependencies between channels of feature maps are captured, which improves the feature representation of specific semantics.

The structure of CAM is illustrated in Fig. 5. As shown in Fig. 5a., let H , W , and C represent the width, height, and channels, and $A \in \mathbb{R}^{H \times W \times C}$ is a local feature map extracted from the model inputs, Fig. 5a. The channel attention map $X \in \mathbb{R}^{C \times C}$ is calculated from the original features $A \in \mathbb{R}^{H \times W \times C}$, Fig. 5b. A is reshaped to $A^1 \in \mathbb{R}^{N \times C}$, and is reshaped and transposed to $A^2 \in \mathbb{R}^{C \times N}$. Then, a matrix multiplication between A^2 and A^1 is performed. Then, a softmax layer is applied to obtain the channel attention map X . The more similar feature representations of the two channels contribute to a higher correlation between them, generating a larger value in X . The sum of each row in X is 1. A^1 is multiplied by the transpose of X to generate $A^x \in \mathbb{R}^{N \times C}$:

$$a_{ij}^x = A_i^1 \cdot X_j, j \in [1, C] \quad (3)$$

where a_{ij}^x is an element of A^x , A_i^1 is the i_{th} row of A^1 and X_j is the j_{th} column of X . A^x is reshaped to $A^3 \in \mathbb{R}^{H \times W \times C}$. For each position of A^3 , the element of a channel

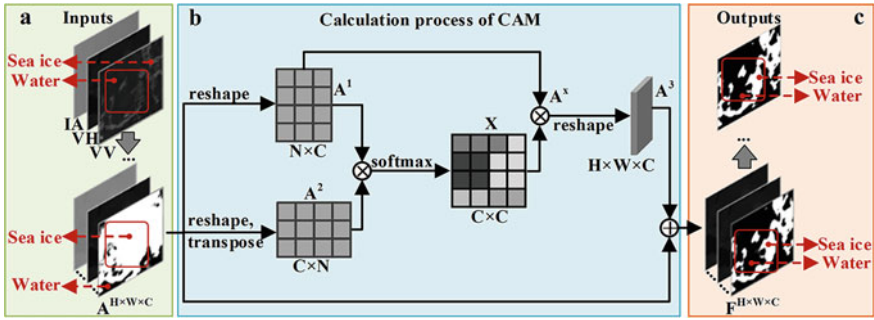


Fig. 5 The detailed calculation process of CAM in the DAU-Net. **a** Feature maps without CAM, some water pixels are inaccurately encoded as sea ice pixels, marked in the red rectangle. **b** The calculation process of CAM. **c** Feature maps after CAM. Some inaccurate sea ice pixels are modified as water pixels, improving the accuracy of outputs

is the weighted sum of elements across all channels in the corresponding position of A based on the weights in X . Therefore, A^3 has long-range contextual dependencies in channel dimensions. A^3 is multiplied by a scale parameter β and added to the input feature map A in element-wise to obtain the output $F^{H \times W \times C}$:

$$F = \beta A^3 + A \tag{4}$$

where β gradually learns a weight from 0.

The final feature of each channel is a weighted sum of the features of all channels and original features. The long-range semantic dependencies between different channels of the feature maps are modeled, which boosts feature discriminability. As shown in Fig. 5a, many open water regions are inaccurately represented as sea ice features in feature map A . After the channel attention procedure, most of the inaccurate regions in A are corrected, Fig. 5c. The outputted feature map F is more discriminating than A and helps to achieve a good classification result.

3.3.3 Fusion

The PAM output and CAM output are separately transformed by a CNN layer. An element-wise summation is performed on the two transformed results. A CNN layer executes convolutions on the summation to generate fusion features. Finally, the fusion features are transmitted to the decoding part.

3.4 Decoder

Five decoder modules are stacked upon the features outputted by attention modules, and each decoder module is composed of one up-sampling layer and two stacking CNN layers. Each CNN layer is followed by a batch normalization layer and a ReLU activation layer. The number of convolutional kernels in the four decoders is 256, 128, 64, 32, and 16, respectively. Three concatenations fuse the features generated from the same level encoder and decoder. The kernel size of all CNN layers in decoder modules is 3×3 . After decoding, the 16×16 feature maps are rescaled to the same size as the input image, 256×256 .

3.5 Output

The feature maps output by the decoder are fed into the output module that consists of one CNN layer with one 1×1 convolutional kernel. One sigmoid layer performs non-linear activation on the convolutional outputs to predict the value of each pixel. The activation value is between $[0,1]$. If it is larger than 0.5, the pixel is sea ice; otherwise, it is open water. The loss function is binary cross-entropy.

4 Experiments

4.1 Experiments Setting

There are 4,684 SAR chips in the training set. We split 30% samples from the training set as the validation set. We choose a typical image with rough sea surface and various sea ice textures as the testing image. We divided the testing image into 672 256×256 chips. The developed model runs on a GPU workstation with one NVIDIA TESLA V100 32 GB GPU. Its batch size is 16, and the initial learning rate is 0.0001. We use Keras as the DL packages, and the ReduceLROnPlateau and early stopping strategies in Keras are employed to accelerate convergence and avoid overfitting.

4.2 Evaluation Metrics

Accuracy, precision, recall and mean intersection over union (IoU) are employed to evaluate the performance of the classification methods. The definition of these metrics is shown in Fig. 6. Precision refers to the proportion of correctly predicted pixels, both sea ice, and water, among all predicted pixels. Precision refers to the proportion of pixels that are true sea ice and predicted as sea ice to all predicted

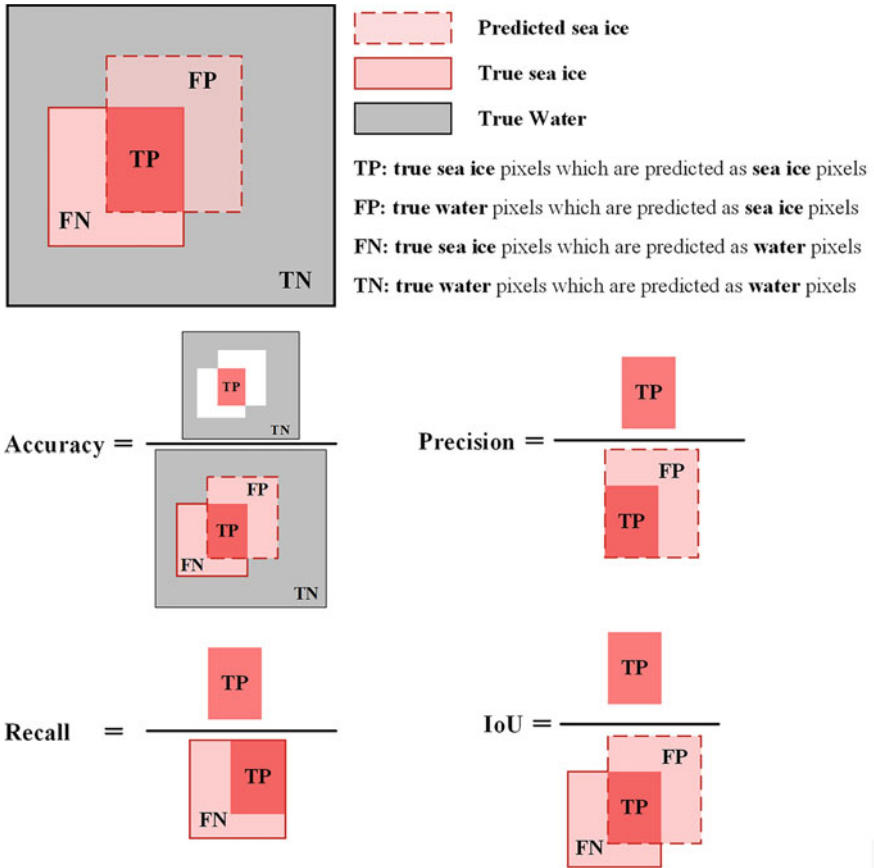


Fig. 6 Definitions of accuracy ((TP+TN)/(TP+TN+FP+FN)), precision (TP/(TP +FP)), recall (TP/(TP +FN)), and IoU (TP/(TP +FP+FN))

sea ice pixels. A higher precision value means the model extracts less false alarms. Recall refers to the proportion of pixels that are true sea ice and predicted as sea ice to all true sea ice pixels. A higher recall value means the model misses fewer sea ice pixels. IoU means the proportion of pixels that are true sea ice and predicted as sea ice to the union of true sea ice and predicted sea ice pixels. When the predicted sea ice pixels coincide with the true sea ice pixels completely, the IoU is the maximum value of 1.

4.3 Comparison Experiments Against Other Models Performances

To validate the performance of the proposed DAU-Net, two recently proposed DL-based sea ice classification models are selected for comparison: 1) CNN_{wang} , which is the CNN-based detection model proposed by Wang et al. [42] in 2018. It consists of five CNN layers and three max-pooling layers; 2) $DenseNet_{FCN}$, which has a similar structure with the MLFN model proposed in 2019 [16]. To satisfy the pixel-level segmentation and make a fair comparison, $DenseNet_{FCN}$ replaces the fully connected layers in the original MLFN with fully convolutional layers and adds upsampling blocks, forming a “U” shape segmentation model.

We also compare our model performance against the classic U-Net model that has a similar structure with DAU-Net except that the CAM and PAM are removed. It is worth noting that the CNN layers after two attention modules and the CNN layer of the fusion part are retained to ensure a fair comparison. $U-Net_{CAM}$ means the U-Net model with CAM but no PAM. $U-Net_{PAM}$ is the U-Net model with PAM but no CAM. Similarly, the CNN layers are retained in these two models. We tune the hyper parameters of all compared models and record the results with the best accuracy.

The evaluation metrics of all models are shown in Table 2, and the corresponding classification results are shown in Fig. 7. The accuracy, IoU, and precision of CNN_{wang} are lower than those of the other five models. However, the recall of CNN_{wang} is the largest one. The precision and the recall are very unbalanced, which means CNN_{wang} misses fewer sea ice pixels but misclassifies many open water pixels as sea ice (high false alarms). As shown in Fig. 7d, the classification results of CNN_{wang} , such as sea ice edges and ice blocks, are coarse-grained. Limited by the model complexity, it is difficult for CNN_{wang} to extract enough representative features to achieve fine-grained classification, thus generate many false alarms. Compared with CNN_{wang} , the accuracy, IoU, and precision of $DenseNet_{FCN}$ are improved obviously, and recall is reduced. The gap between precision and recall is narrowed. Fig. 7e shows that the classification results are much more refined than those of CNN_{wang} . However, there are still some false alarms in the region marked by the

Table 2 Evaluation results of all compared models

Model	Accuracy(%)	IoU	Precision	Recall
CNN_{wang}	92.94	0.8428	0.8801	0.9520
$DenseNet_{FCN}$	93.22	0.8451	0.9027	0.9298
U-Net	93.89	0.8573	0.9241	0.9222
$U-Net_{CAM}$	94.03	0.8611	0.9203	0.9305
$U-Net_{PAM}$	94.01	0.8606	0.9295	0.9207
DAU-Net	94.39	0.8673	0.9355	0.9225

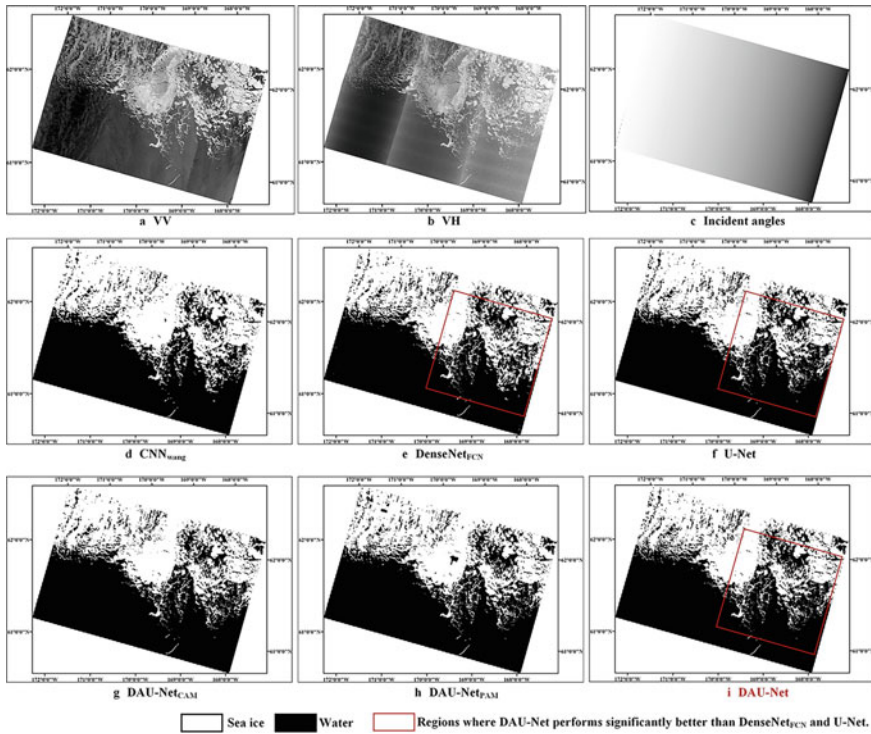


Fig. 7 a–c Inputs of the test SAR image, VV channel, and VH channel are scaled to 0-255 for better visualization. d–i classification results of different models

red rectangle. Although DenseNet_{FCN} is more complicated than CNN, it is still not enough to extract sufficiently characteristic features to accurately distinguish sea ice and water, especially in areas where sea ice and water are mixed under complex sea conditions.

The U-Net model outperforms CNN_{wang} and DenseNet_{FCN} in both accuracy and IoU. Its recall and precision are also more balanced. Fig. 7f shows that the U-Net obviously reduces the false alarms generated by DenseNet_{FCN} (marked by the red rectangle). By introducing attention modules, U-Net_{CAM} and U-Net_{PAM} show improvements in accuracy and IoU. The precisions and recalls do not show significant improvements. However, as shown in Fig. 7g-h, the classification results of U-Net_{CAM} and U-Net_{PAM} are more refined, and the boundary between sea ice and open water is more smoother. Finally, the DAU-Net, integrated with CAM and PAM, obtains the most considerable accuracy, IoU, and precision (Table 2). Compared with the original U-Net model, the accuracy, IoU, and precision of the DAU-Net increased by 0.50%, 1.00%, and 1.14%, respectively. The accuracy and recall are in balance. By comparing Fig. 7i and f, it can be found that the false alarms generated by U-Net are reduced significantly, and the classification results of DAU-Net are more

refined. The fine-grained objects such as small floes, sinuous ice-water boundaries, and ice channels are classified more smoothly by DAU-Net. Therefore, the CAM and the PAM can improve the representative ability of extracted features to promote the classification results of sea ice and open water.

4.4 Effectiveness of IA

As the IA is ignored in existed DL-based models [16, 42], we design an experiment to evaluate the effectiveness of employing the IA of SAR images as one input. Table 3 shows the experiment results. DAU-Net is the model with IA, and DAU-Net_{NIA} is the model without IA. The other experiment settings are unchanged. The accuracy and IoU of DAU-Net_{NIA} are less than those of the DAU-Net. The precision is much larger than the recall, which means DAU-Net_{NIA} misses many sea ice pixels. As shown in Fig. 8c, some sea ice pixels are misclassified as open water in the upper left part of the image. Thus, the IA is essential to obtain better classification results.

Table 3 Evaluation results of using IA

Model	Accuracy(%)	IoU	Precision	Recall
DAU-Net _{NIA}	92.48	0.8183	0.9554	0.8508
DAU-Net	94.39	0.8673	0.9355	0.9225

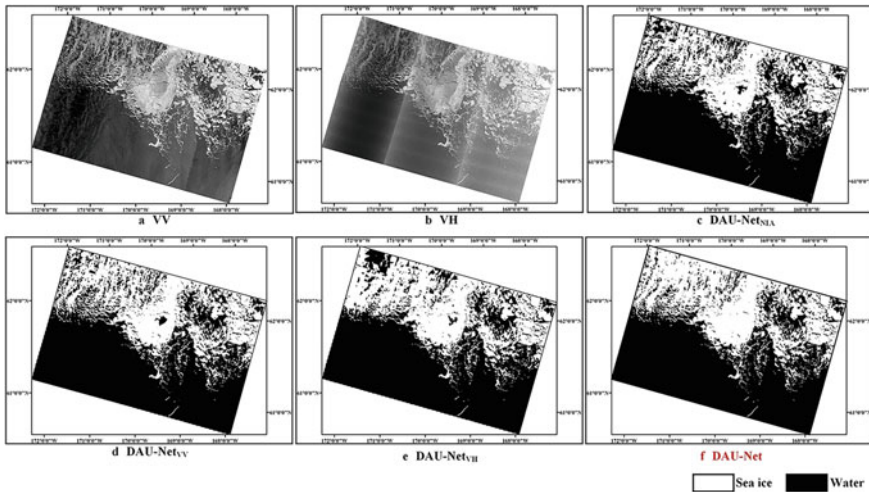


Fig. 8 a and b, VV channel and VH channel of the testing set; c–f, classification results of the model without IA, VH, VV as inputs, separately

4.5 Effectiveness of Dual—Polarization Information

We design an experiment to evaluate the effectiveness of dual-polarization inputs. DAU-Net uses the VV channel, VH channel, and IA as the inputs. DAU-Net_{VV} uses VV channel and IA as the inputs, and DAU-Net_{VH} uses VH channel and IA as inputs. The other experiment settings are unchanged, as shown in Table 4. The four metrics of DAU-Net_{VH} are smaller than those of the other two models. As Fig. 8e shown, DAU-Net_{VH} misclassifies many sea ice pixels as open water, mainly the pixels in the upper left part of the image. DAU-Net_{VV} performs better than DAU-Net_{VH}, but it still misses some sea ice pixels in the middle part of the image, Fig. 8d. Finally, by combining VV and VH as inputs, DAU-Net achieves the best performance. Thus, the dual-polarization information of SAR image is helpful to obtain better classification results.

Table 4 Evaluation results of using Dual-Polarization Information

Model	Accuracy(%)	IoU	Precision	Recall
DAU-Net _{VH}	91.21	0.7956	0.9136	0.8603
DAU-Net _{VV}	93.25	0.8375	0.9523	0.8742
DAU-Net	94.39	0.8673	0.9355	0.9225

4.6 Performances of Different ResNet-Based Encoders

The encoder in DAU-Net is ResNet-34. We design an experiment to evaluate the performances of the other two ResNet-based encoders. DAU-Net₁₈ is the model using ResNet-18 as the encoder, and DAU-Net₅₀ is the model using ResNet-50 as the encoder. The other parts of these two models are the same as those of the DAU-Net. As shown in Table 5, the performances of the three models do not show much difference. DAU-Net with ResNet-34 as encoder slightly outperforms the other two ResNet-based encoders. For our classification mission, ResNet-34 is a more suitable encoder than the other two ResNet models.

Table 5 Evaluation of different ResNet encoders

Model	Encoder	Accuracy (%)	IoU	Precision	Recall
DAU-Net ₁₈	ResNet-18	94.08	0.8598	0.9374	0.9121
DAU-Net ₅₀	ResNet-50	93.72	0.8523	0.9292	0.9115
DAU-Net	ResNet-34	94.39	0.8673	0.9355	0.9225

5 Discussions

To validate the robustness of the proposed model, we employ the DAU-Net to classify sea ice and open water from a series of SAR images in the Bering Strait and compare the classification results with the sea ice products provided by NSIDC. As the DenseNet_{FCN} represents the existing DL-based classification model for sea ice, we take the results of DenseNet_{FCN} as comparison targets. The image series consists of six images, each of which is mosaiced from three Sentinel-1A images, and a total of 18 Sentinel-1A images. Their details are shown in Table 1. The image series covers the process from freezing to melting of the Bering Strait, including a variety of sea ice textures and sea surface conditions. As shown in Fig. 9a-f, sea ice partially appeared in the Bering Strait on Dec 13, 2018, and it covered the entire region until Mar 19, 2019. Then, on Mar 31, 2019, the sea ice started to melt, and by May 6, 2019, most of it had receded. The most recent data (generally from the previous day) of the 1 km products appear in the archive at approximately 10:00 p.m. (Greenwich Mean Time, GMT). The 18 Sentinel-1A images in the Bering Strait are acquired around 06:00 p.m. (GMT). Due to the time difference, the date of the MASIE-NH products we employed is one day later than the date of the Sentinel-1A images. The cell size of the DAU-Net result is 30 m. The spatial resolution of the two data is too different, so it is unreasonable to compare their evaluation metrics quantitatively. Here, we discuss the performance of DAU-Net through the visual comparison of classification results.

Figure 9g-l show the classification results of DAU-Net and Fig. 9m-r are the corresponding MASIE-NH products. Overall, the DAU-Net results are consistent with the MASIE-NH products. The sea surface in Fig. 9a, d, and f is very rough and bright, mixing with the sea ice, especially the regions marked as red rectangles. As shown in Fig. 9g, j, and l, the DAU-Net classifies the sea ice and open water well, which demonstrates that the proposed model can deal with the complex sea surface. There are many water gaps, small sea ice floes, and sinuous ice-water boundaries in Fig. 9c and f, which are finely classified by the DAU-Net, as shown in Fig. 9i and l. The separate water channels in Fig. 9e are also successfully classified by DAU-Net, as shown in Fig. 9k. As the spatial resolution of the MASIE-NH products is 33.3 times lower than that of DAU-Net results. Many fine-grained objects cannot be classified in the MASIE-NH products. As shown in Fig. 9i, k, and l, the classification results of DAU-Net are more consistent with the SAR images than the MASIE-NH products, especially in the regions marked by the yellow rectangles in Fig. 9c, e, and f. Taking the region marked by the yellow rectangle in Fig. 9f as an example, we show the detailed comparisons between the classification results of DAU-Net and 1km MASIE-NH products in Fig. 10. Our classification results show obvious advantages over MASIE-NH products in spatial resolution, Fig. 10b-d.

However, DAU-Net performs not very well in some regions. As marked by the green rectangles in Fig. 9a, some sea ice pixels with dark textures are misclassified as open water. Some open water pixels with extremely rough surfaces are misclassified as sea ice. The misclassifications may be due to the lack of these two types of

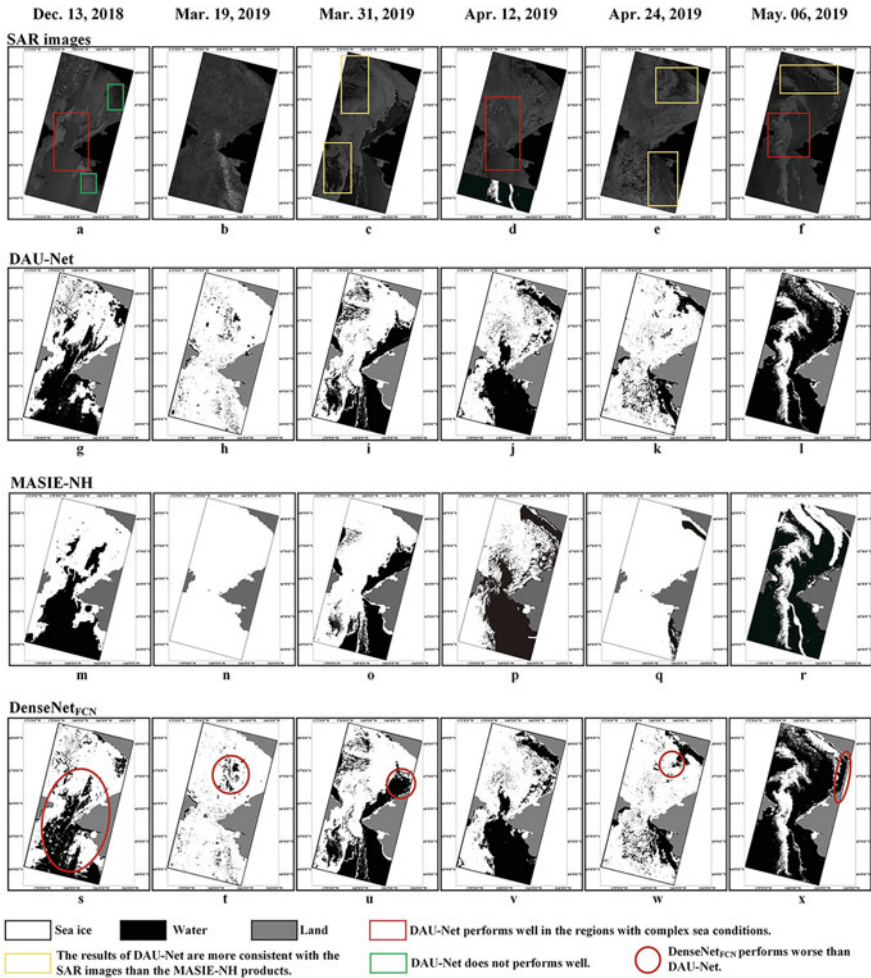


Fig. 9 Comparison between results of DAU-Net, MASIE-NH products, and results of DenseNet_{FCN} of a time series (Dec 13, 2018-May. 06, 2019) in Bering Strait. **a-f** SAR images, VV channel. **g-l** classification results of DAU-Net. **m-r** 1km MASIE-NH products. **s-x** classification results of DenseNet_{FCN}

samples in the training set. The misclassifications mainly exist in the SAR image on Dec 13, 2018, the early stage of sea ice in the Bering Strait, with some very dark sea ice textures. These textures are rare during the freezing and melting stages. In addition, the extremely rough sea surfaces are also rare in the training set, resulting in misclassifications. As shown in Fig. 9s-x, the results of DenseNet_{FCN} are generally consistent with the MASIE-NH products. However, DenseNet_{FCN} performs worse than DAU-Net, especially in the regions marked by red circles. Some rough sea surface pixels are misclassified as sea ice pixels.

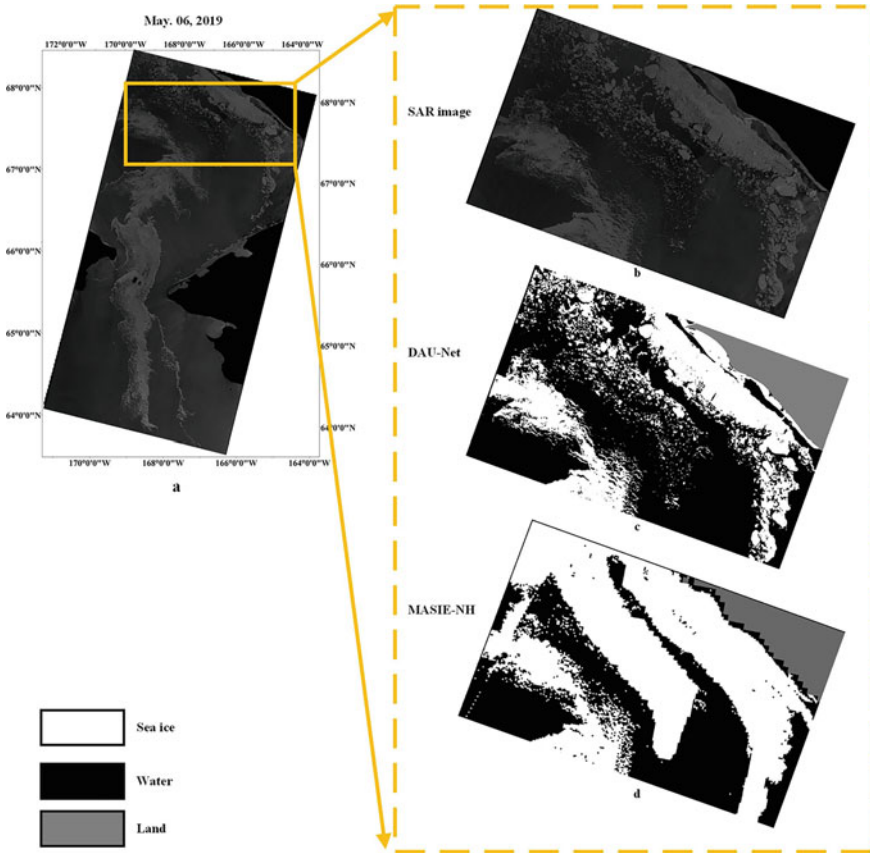


Fig. 10 A detailed comparison between results of DAU-Net and MASIE-NH products in a representative region marked in Fig. 9f. **a** The SAR image on May 6, 2019. **b–d** the detailed SAR image, classification results of DAU-Net, and 1km MASIE-NH products corresponding to the marked region

In summary, by validating the applicability of DAU-Net through a series of SAR images in the Bering Strait, we demonstrated that the DAU-Net performs well in most sea conditions. The proposed is capable of dealing with various sea ice textures. Due to the advantages of SAR image resolution and model performance, the results of DAU-Net are more refined than MASIE-NH products. DAU-Net also outperforms the existing DL-based sea ice classification model, DenseNet_{FCN}. However, the DAU-Net performs not well on some unusual textures. To further improve the model applicability, we will collect more training samples to supplement the rare texture types.

6 Conclusions

This study proposes a DAU-Net model to classify the sea ice and open water from SAR images. We combine the ResNet34 with the U-Net to form the model backbone. SAR images are obtained from Sentinel 1A. The dual-polarized information and the IA of SAR images are utilized as the model inputs. We integrate the dual-attention mechanism, PAM and CAM, into the original U-Net model to extract more characteristic features, which helps to achieve more accurate classifications. We use 15 Sentinel-1A SAR images acquired near the Bering Sea to train the model. We evaluate the model performance by one SAR image and compare the DAU-Net with the typical DL-based ice classification models. Further, we use the well-trained model to classify a series of SAR images of Bering Strait, which covers the process from freezing to melting. We make a comparison between the classification results of DAU-Net and the 1km MASIE-NH products of NSIDC. Experiments show that: 1) the dual-attention mechanism enhances the representative ability of features and help the DAU-Net outperforms the origin U-Net and typical existing DL-based ice classification models, especially in the classification of fine-grained targets; 2) the three-channel inputs, dual-polarized information (VV and VH) and IA, contribute to high accuracy classifications; and 3) the DAU-Net is capable of dealing with complex sea state conditions from freezing to melting, showing good robustness and applicability.

In the future, to address the misclassifications on unusual sea ice textures, we will collect more training samples from a wide range of space and time. We will also explore the possibility of integrating few-shot learning to solve the mentioned problem. Besides, the multi-category classification models to discriminate MYI, sea ice, and open water will be will become a follow-up work.

Acknowledgements The authors would like to thank the European Space Agency for providing the Sentinel-1 data, the Sentinel Application Platform (SNAP) software. Ground truth labels are annotated by LabelMe (<http://labelme.csail.mit.edu>).

References

1. Agarap AF (2018) Deep learning using rectified linear units (ReLU). arXiv preprint [arXiv:1803.08375](https://arxiv.org/abs/1803.08375)
2. Alom MZ, Taha TM, Yakopcic C, Westberg S, Sidike P, Nasrin MS, Van Eesn BC, Awwal AAS, ASARi VK (2018) The history began from AlexNet: A comprehensive survey on deep learning approaches. arXiv preprint [arXiv:1803.01164](https://arxiv.org/abs/1803.01164)
3. Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. *Comput Sci*
4. Balakrishna C, Dadashzadeh S, Soltaninejad S (2018) Automatic detection of lumen and media in the IVUS images using U-Net with VGG16 encoder. arXiv preprint [arXiv:1806.07554](https://arxiv.org/abs/1806.07554)
5. Celik T (2009) Unsupervised change detection in satellite images using principal component analysis and k-Means clustering. *IEEE Geosci & Remote Sens Lett* 6(4):772–776

6. Clausi D, Deng H (2003) Operational segmentation and classification of SAR sea ice imagery. In: *Advances in Techniques for Analysis of Remotely Sensed Data*, 2003 IEEE Workshop on
7. Clausi DA, Member S, Deng H (2005) Operational map-guided classification of SAR sea ice imagery. *IEEE Trans Geosci & Remote Sensin* 43:2940–2951
8. Dabboor M, Geldsetzer T (2014) Towards sea ice classification using simulated RADARSAT constellation mission compact polarimetric sar imagery. *Remote Sens Environ* 140:189–195
9. Danielson S, Curchitser E, Hedstrom K, Weingartner T, Stabeno P (2011) On ocean and sea ice modes of variability in the bering sea. *J Geophys Res: Ocean* 116(C12)
10. ESA (2021) Step science toolbox exploitation platform. <http://step.esa.int/main/toolboxes/snap/>
11. Falk T, Mai D, Bensch R, Çiçek Ö, Abdulkadir A, Marrakchi Y, Böhm A, Deubner J, Jäckel Z, Seiwald K et al (2019) U-Net: deep learning for cell counting, detection, and morphometry. *Nat Methods* 16(1):67–70
12. Feng G, Dong J, Bo L, Xu Q (2017) Automatic change detection in synthetic aperture radar images based on PCANet. *IEEE Geosci Remote Sens Lett* 13(12):1792–1796
13. Fetterer F, Bertoin C, Jing PY (2002) Multi-year ice concentration from RADARSAT. In: *Geoscience & Remote Sensing, IGARSS 97 Remote Sensing—a Scientific Vision for Sustainable Development*, IEEE International
14. Fu J, Liu J, Tian H, Li Y, Bao Y, Fang Z, Lu H (2020) Dual attention network for scene segmentation. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*
15. Gao F, Dong J, Li B, Xu Q, Xie C (2016) Change detection from synthetic aperture radar images based on neighborhood-based ratio and extreme learning machine. *J Appl Remote Sens* 10(4):046019
16. Gao Y, Gao F, Dong J, Wang S (2019) Transferred deep learning for sea ice change detection from synthetic aperture radar images. *IEEE Geosci Remote Sens Lett* 16(10):1655–1659
17. Garcia LP, de Carvalho AC, Lorena AC (2015) Effect of label noise in the complexity of classification problems. *Neurocomputing* 160:108–119
18. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp 770–778. <https://doi.org/10.1109/CVPR.2016.90>
19. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. *sci* 313(5786):504–507
20. Karvonen JA (2004) Baltic sea ice SAR segmentation and classification using modified pulse coupled neural networks. *IEEE Trans Geosci & Remote Sens*
21. Komarov AS, Buehner M (2017) Automated detection of ice and open water from dual-polarization RADARSAT-2 images for data assimilation. *IEEE Trans Geosci & Remote Sens* 55(10):1–15
22. Krizhevsky A, Sutskever I, Hinton G (2012) Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25(2)
23. Lau S, Wang X, Xu Y, Chong E (2020) Automated pavement crack segmentation using fully convolutional U-Net with a pretrained resnet-34 encoder. *arXiv preprint arXiv:2001.01912*
24. Lecun Y, Bengio Y, Hinton G (2015) Deep learning. *Nat* 521(7553):436–444. <https://doi.org/10.1038/nature14539>
25. Leigh S, Wang Z, Clausi, DA (2014) Automated ice-water classification using dual polarization SAR satellite imagery. *IEEE Trans Geosci & Remote Sens*
26. Li J, Wang C, Wang S, Zhang H, Wang Y (2017) Gaofen-3 sea ice detection based on deep learning. In: *2017 Progress in Electromagnetics Research Symposium - Fall (PIERS - FALL)*
27. Li X, Liu B, Zheng G, Ren Y, Zhang S, Liu Y, Gao L, Liu Y, Zhang B, Wang F (2020) Deep learning-based information mining from ocean remote sensing imagery. *Natl Sci Rev*
28. Liu B, Li X, Zheng G (2019) Coastal inundation mapping from bitemporal and dualpolarization SAR imagery based on deep convolutional neural networks. *J Geophys Res: Ocean* 124(12)
29. Lundhaug Maria (2002) ERS SAR studies of sea ice signatures in the pechora sea and kara sea region. *Can J Remote Sens* 28(2):114–127

30. Olonscheck D, Mauritsen T, Notz D (2019) Arctic sea-ice variability is primarily driven by atmospheric temperature fluctuations. *Nat Geosci* 12(6):430–434
31. Peng C, Zhang X, Yu G, Luo G, Sun J (2017) Large kernel matters—improve semantic segmentation by global convolutional network. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 4353–4361
32. Petrou ZI, Tian Y (2019) Prediction of sea ice motion with convolutional long short-term memory networks. *IEEE Trans Geosci Remote Sens* 57(99):1–12
33. Reichstein M, Camps-Valls G, Stevens B, Jung M, Denzler J, Carvalhais N, Prabhat (2019) Deep learning and process understanding for data-driven earth system science. *Nat* 566(7743):195
34. Ren Y, Chen H, Han Y, Cheng T, Chen G (2019) A hybrid integrated deep learning model for the prediction of citywide spatio-temporal flow volumes. *Int J Geogr Inf Sci*. 1–22
35. Ronneberger O, Fischer P, Brox T (2015) U-Net: Convolutional networks for biomedical image segmentation. Springer International Publishing
36. Russell BC, Torralba A, Murphy KP, Freeman WT (2008) LabelMe: A database and web-based tool for image annotation. *Int J Comput Vis* 77(1-3)
37. Shokr ME (1991) Evaluation of second-order texture parameters for sea ice classification from radar images. *J Geophys Res: Ocean* 96
38. Soh LK (2002) Arktos: An intelligent system for satellite sea ice image analysis. *Cse Technical Reports*
39. Soh LK, Tsatsoulis C (1999) Texture analysis of sar sea ice imagery using gray level co-occurrence matrices. *IEEE Trans Geosci & Remote Sens* 37(2):780–795
40. Su H, Wang Y, Xiao J, Yan XH (2015) Classification of MODIS images combining surface temperature and texture features using the support vector machine method for estimation of the extent of sea ice in the frozen bohai bay, China. *Int J Remote Sens* 36(9–10):2734–2750
41. US National Ice Center (2008) IMS Daily Northern Hemisphere Snow and Ice Analysis at 1 km, 4 km, and 24 km Resolutions, Version 1e. Tech. rep., NSIDC: National Snow and Ice Data Center, Boulder, Colorado USA., <https://doi.org/10.7265/N52R3PMC>, <https://nsidc.org/data/g02156>
42. Wang C, Zhang H, Wang Y, Zhang B (2018) Sea ice classification with convolutional neural networks using Sentinel-L scanSAR images. In: *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*
43. Wang L, Scott KA, Xu L, Clausi DA (2016) Sea ice concentration estimation during melt from dual-pol SAR scenes using deep convolutional neural networks: A case study. *IEEE Trans Geosci Remote Sens* 54(8):4524–4533. <https://doi.org/10.1109/TGRS.2016.2543660>
44. Yan X, Scott KA (2017) Sea ice and open water classification of SAR imagery using cnn-based transfer learning. In: *IGARSS 2017 - 2017 IEEE International Geoscience and Remote Sensing Symposium*
45. Zakhvatkina NY, Alexandrov VY, Johannessen OM, Sandven S, Frolov IY (2013) Classification of sea ice types in ENVISAT synthetic aperture radar images. *IEEE Trans Geosci & Remote Sens* 51(5):2587–2600
46. Zhang Z, Liu Q, Wang Y (2017) Road extraction by deep residual U-Net. *IEEE Geosci Remote Sens Lett* 15(99):1–5
47. Zhang Z, Yu Y, Li X, Hui F, Cheng X, Chen Z (2019) Arctic sea ice classification using microwave scatterometer and radiometer data during 2002-2017. *IEEE Trans Geosci & Remote Sens*. 1–10

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits any noncommercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if you modified the licensed material. You do not have permission under this license to share adapted material derived from this chapter or parts of it.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Detection and Analysis of Marine Green Algae Based on Artificial Intelligence



Le Gao, Xiaofeng Li, Yuan Guo, Fanzhou Kong, and Rencheng Yu

1 Introduction

Harmful algal blooms (HAB), e.g., Yellow sea green algae, are disastrous ecological events in coastal oceans. During the blooming period, the rapid biomass increase severely impacts the coastal ecosystems and even the Olympic regatta games in 2008 [6, 13–15, 17, 20, 22, 23].

Satellite remote sensing is a suitable means for green algae (*U. prolifera*) observation and analysis because of the frequent data acquisition and broad coverage area [5, 7]. Existing studies mostly use passive optical-sensor images of 250-1,000 m resolution, e.g., Moderate Resolution Imaging Spectroradiometer (MODIS). The floating *U. prolifera* modulate the ocean color properties to make sea surface appear the prominent algae features in optical images [2, 8, 10, 21]. Active Synthetic aperture radar (SAR) images provide sea surface roughness with a resolution of tens of meters. The floating algae on the sea surface behave like a volume-scattering hard object, and the algae patch area's reflected signal is much stronger than that backscattered one from the surrounding water, which appears as brighter regions in SAR images. SAR has become another option for detecting algae because some SAR images have become free and open, e.g., the European Space Agency (ESA) Sentinel-1 and Chinese Gaofen-3 data. For optical-sensor images, biological index methods, e.g., NDVI (Normalized Difference Vegetation Index) and FAI (Floating Algae Index), are commonly used [1, 5]. For SAR-sensor images, previous studies usually use grey, roughness or backscatter coefficient difference to identify the target [4, 12]. However, these methods cannot effectively fuse the information from the optical and SAR images since the physical mechanisms of optical- and SAR-sensors for *U. prolifera* detection are very different. Based on the algae's characteristics in the two sensors' images, deep-learning (DL) offers a possibility to perform data fusion

L. Gao · X. Li (✉) · Y. Guo

CAS Key Laboratory of Ocean Circulation and Waves, Institute of Oceanology,
Chinese Academy of Sciences, Qingdao 266071, China
e-mail: lixf@qdio.ac.cn

F. Kong · R. Yu

Key Laboratory of Marine Ecology and Environmental Sciences, Institute of Oceanology,
Chinese Academy of Sciences, Qingdao 266071, China

© The Author(s) 2023

X. Li and F. Wang (eds.), *Artificial Intelligence Oceanography*,
https://doi.org/10.1007/978-981-19-6375-9_13

277

[12]. *U. prolifera* algae have the thalli's hollow tubular structure. During its blooms, some parts of the algae body are exposed above the sea surface, while other parts are submerged below the surface. An optical sensor can collect spectral information at a certain sea depth to effectively capture the floating and underwater part of the algae [7, 11]. The SAR sensor captures only the floating part on the sea surface. Thus, we can define the floating and submerged algae ratio (FS ratio), i.e., part of SAR-sensor detection/part of optical-sensor detection. The objectives of this research include 1) proposing a DL network to detect *U. prolifera* from optical and SAR images better, and 2) using the defined FS ratio to represent algae life stages.

2 Data and Methodology

2.1 Satellite Images and Labels

We collected geometrically and radiometrically corrected 250 m spatial resolution MODIS true-color imagery (Bands: 1/4/3) containing algae patches in the Yellow Sea, and these MODIS images are under clear sky conditions from 2008 to 2021. Compared to the surrounding seawater, the *U. prolifera* algae show more prominent green slick/patch features (Fig. 1a–d). Using the Labelme software [16], we can label sample image slices containing different algae shapes for DL algorithm development. Finally, 1,055 pairs of MODIS labelled samples were obtained, and 680/292/83 pairs were used as training/validation/testing sets.

We also collected Sentinel-1 Level-1 GRD (Ground Range Detected) dual-polarization (VV, VH) interferometric wide images with 10 m spatial resolution and 250 km swath and the Chinese GaoFen-3 SAR Fine Stripe Mode II (FSII) dual-polarization (HH, HV) image with 10 m resolution and 100 km swath between 2015 and 2019. All SAR images were processed with speckle filtering and geometric, radiometric, orthometric, and terrain corrections to improve image quality using the Sentinel Application Platform (SNAP) 7.0 software. The algae patches show bright spots/slicks in SAR images (Fig. 1e). We marked 4,071 pairs of the algae labelled samples; 2,086/895/1,090 pairs were used as training/validation/testing sets.

2.2 UNet-Based Algae Detection Network (AlgaeNet)

We propose a DL-based model, AlgaeNet, to detect the algae patches better. Figure 2 shows the model's system diagram based on the U-Net framework [12, 16]. Optical and SAR images are input to the DL model separately, and the corresponding detection result of the optical (SAR) image is Algae coverage-1 (2) through the improved model. Then the model can perform data fusion based on the two sensors' detection results, and the FS ratio can be estimated by $\text{Algae coverage-2}/\text{Algae coverage-1}$.

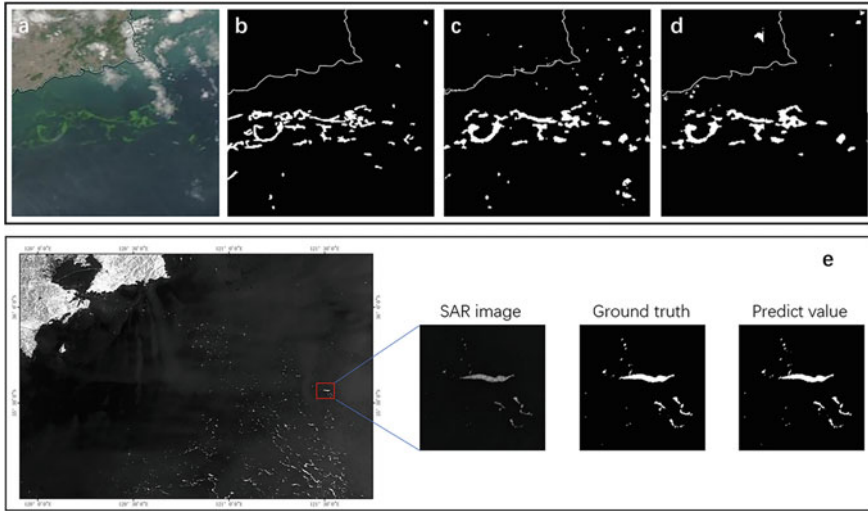


Fig. 1 U. prolifera algae blooms in Yellow sea and algae detection examples at the pixel level in green tide bloom period. **a** the random selected MODIS true-color images on June 25, 2008; **b** is the marked ground truth by manual; **c** is the predicted value of the classic U-Net model; **d** is the corresponding predicted value of the AlgaeNet model; the white dots are algae pixels, and the black are the background ocean; **e** is an algae detection example based on the AlgaeNet model in Sentinel-1 SAR images

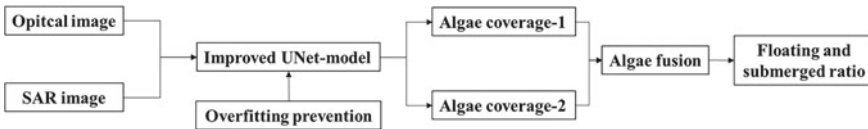


Fig. 2 AlgaeNet model design. Algae coverage-1 (2) is based on MODIS (SAR) images

During the DL architecture design, we should pay particular attention to maintaining the tradeoff between optimization and generalization of the network. The overfitting of the algae detection model is usually prevented mainly through the following three methods: dropout, weight regularization, and batch normalization. We found that batch normalization (BN) and weight regularization were beneficial for the network of the three technologies. BN provided any layer with zero mean/unit variance in the DL model [9]. The initialization type of weights could cause a digression to gradients, meaning the gradients have to compensate for the outliers. BN regularizes the gradient by normalizing activations throughout the network. It prevents small parameter changes from amplifying into more significant and suboptimal changes in gradients' activations. L2 weight regularization is also added to each hidden layer. During optimization, L2 regularization adds penalty items to model parameters or activation values in the hidden layer, limiting the model parameters too much/too large to avoid the network being too complicated. These penalty terms will be used as the network's ultimate optimization goal.

Table 1 Performance of AlgaeNet model

Input data	AI-Model	Output (%)				
		Accuracy	Precision	Recall	F1 score	IoU
MODIS	Classic U-Net	96.37	62.96	53.84	58.04	37.89
	AlgaeNet	97.51	66.61	55.41	60.50	42.62
Sentinel-1/GF-3 SAR	AlgaeNet	99.83	95.35	92.04	93.67	88.09
	Random Forest	99.39	72.95	87.96	79.96	66.60

2.3 Model's Performance

The performance evaluation of the AlgaeNet model includes the assessments of the algae detection performance for MODIS and SAR images, respectively. For evaluating the AlgaeNet-MODIS model, Table I shows that the performance of the AlgaeNet-MODIS model is better than the original U-Net model; the AlgaeNet-MODIS (U-Net) model reached 97.51 (96.37)%, 66.61 (62.96)%, 55.41 (53.84)%, 60.50 (58.04)%, and 42.62 (37.89)% in the five commonly used indicators of Accuracy, Precision, Recall, F1_Score, and Mean Intersection over Union (IoU). For the evaluation of AlgaeNet-SAR, the model reached 99.83, 95.35, 92.04, 93.67, and 88.09% in the five indicators, which are significantly better than AlgaeNet-MODIS. Figure 1 also gives a visual presentation of the algae detection performance in the *U. prolifera* blooming period. Finally, we compared the model's further with the recent neural networks: Random Forest (RF) models. Table 1 shows that our model has significantly higher performance than the RF model and indicates the excellent portability of the particular improvement strategy in the networks.

3 Results and Discussion

The AlgaeNet model was used in MODIS and SAR images to examine the algae coverage changes in 2020 and 2021. Figure 3 shows that the maximum biological coverage in 2021 is nearly four times that of 2020. This significant difference has attracted widespread attention, and it is related to nutrients, sea surface temperature, sea surface salinity, seaweed planting valve area and valve frame recovery time, species competition, etc. [3, 18, 21].

We used the AlgaeNet model to process the collected MODIS and SAR images and acquired twelve pairs of spatiotemporally matching MODIS and SAR images/slicks. Figure 4 shows that the algae patches captured by MODIS and SAR sensors have a highly consistent spatial distribution pattern (Fig. 4b-c). In addition, we also found one interesting detail: for the big algae patches/slicks with a high aggregation degree,

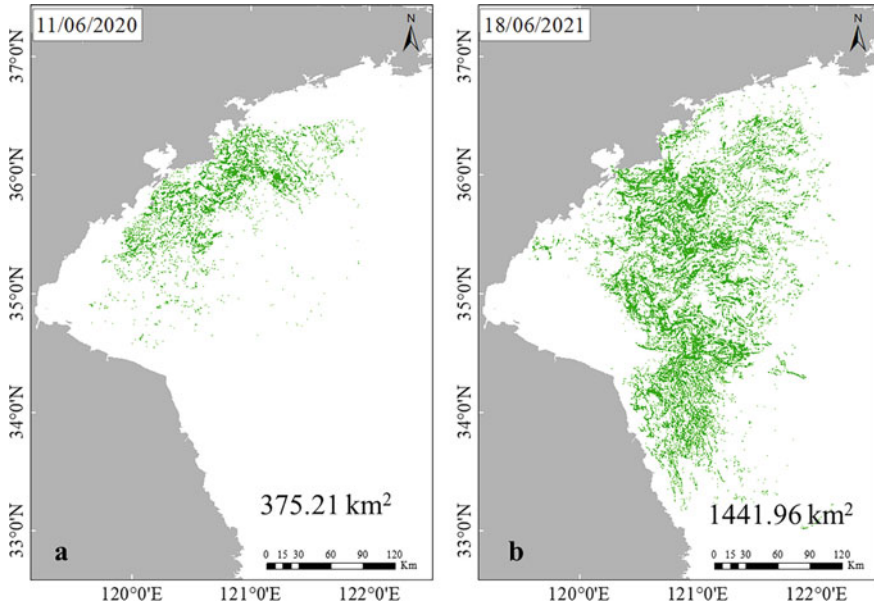


Fig. 3 The detected maximum algae coverage in 2020 **a** and 2021 **b** in Yellow Sea

the margin of the algae patches observed by the MODIS sensor is broader than that observed by the SAR sensor. That is due to the unique floating mechanism of the algae body. The *U. prolifera* algae has the thalli's hollow tubular structure and floats on the sea surface; some parts of the algae body are exposed above the sea surface, while others are submerged below the surface. Therefore, optical sensors can collect spectral information at a certain sea depth to effectively capture the underwater part of the algae [7, 11]. On the other hand, the SAR sensors capture only the floating part on the sea surface. Thus, we can estimate the floating and submerged algae ratio (FS ratio) of *U. prolifera* algae.

As shown in Fig. 5, the FS ratio reflects the changes in the floating status of *U. prolifera*. Based on the algae distribution, coverage, and biomass results of the collected MODIS and SAR images from 2008-2021/2015-2019, the *U. prolifera* bloom originated from the Subei Shoal and drifted northward experienced different phases from initiation, development, maintenance, and decline. At the various stages of the *U. prolifera* bloom, the floating *U. prolifera* underwent morphological changes. At the initiation phase in the Subei Shoal, the *U. prolifera* algae had a large proportion submerged in seawater [19] and rare algae biomass. Based on the two matching MODIS AND SAR image pairs, the FS ratio of the algae body was less than 5% (Fig. 4). During the development phase, the biomass of *U. prolifera* rapidly increased. A large proportion of *U. prolifera* became floating due to the optimal illumination and temperature, and therefore FS ratio quickly increased to 24.75%, and some local areas even reached more than 40%. During the maintenance phase of the *U.*

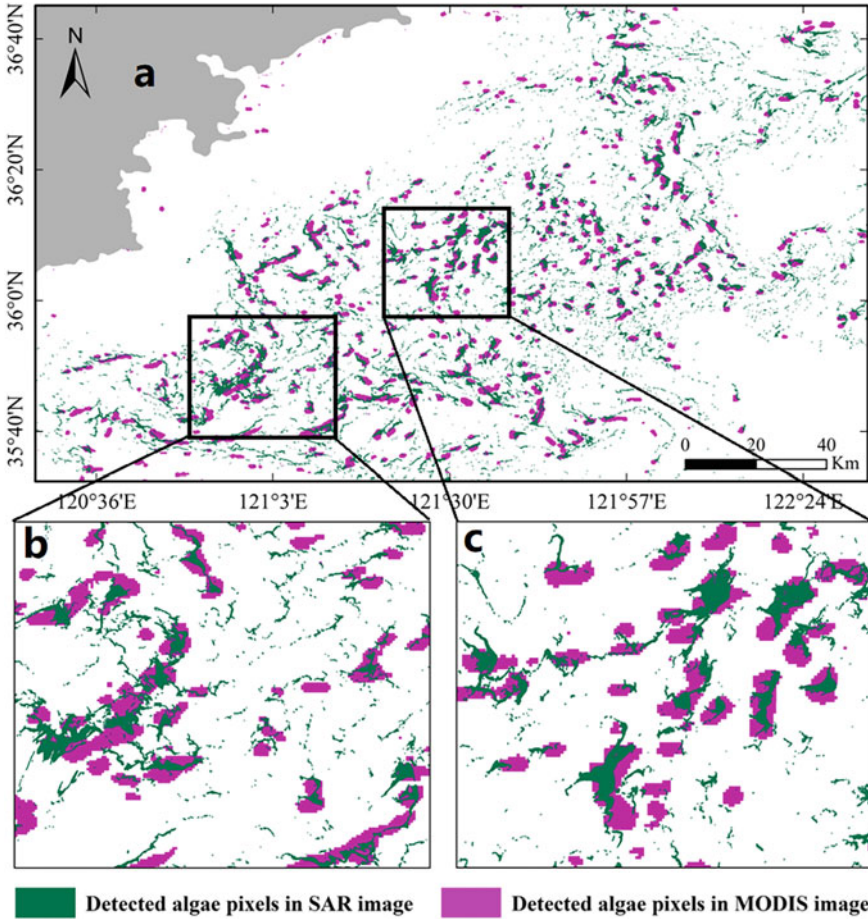


Fig. 4 Algae FS ratio estimation. **a** detected algae pixels between MODIS and SAR images; **b** and **c** corresponds to the enlarged view of two randomly selected sub-areas

prolifera bloom, the *U. prolifera* algae moved northward, and the biomass and FS ratio remained at a high level, basically unchanged, shown as the dotted box of ~21.35%. During the decline phase of the bloom, there were almost no algae near Subei Shoal, and the FS ratio of algae patches in the Yellow Sea decreased rapidly to 14.33%. Therefore, in the entire life phase, the FS ratio of the *U. prolifera* had a parabolic process from increasing, maintaining, and then decreasing. The rates of increasing (initiation phase) and decreasing (decline phase) were high-speed compared to the development and maintenance phases.

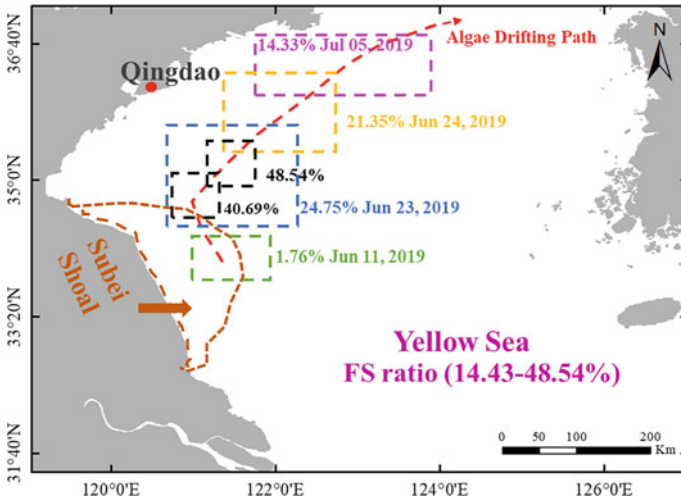


Fig. 5 FS ratio: a life status indicator of *U. prolifera* algae

4 Conclusions

This chapter establishes an improved DL model for detecting *U. prolifera* algae in MODIS and SAR images, and the model has a high detection accuracy, i.e., 97.51%, and Mean IoU to 42.62% for MODIS images and 99.83% and 88.09% for SAR images. The detection results show that the maximum biological coverage in 2021 is almost four times that of 2020 due to various natural and manufactured reasons. Besides, we can take the FS ratio as an excellent indicator to reflect the life status of floating *U. prolifera* algae.

References

1. Blondeau-Patissier D, Gower JF, Dekker AG, Phinn SR, Brando VE (2014) A review of ocean color remote sensing methods and statistical techniques for the detection, mapping and analysis of phytoplankton blooms in coastal and open oceans. *Progress Oceanogr* 123:123–144
2. Cao Y, Wu Y, Fang Z, Cui X, Liang J, Song X (2019) Spatiotemporal patterns and morphological characteristics of *Ulva prolifera* distribution in the Yellow Sea, China in 2016–2018. *Remote Sens* 11(4):445
3. Fan S, Fu M, Wang Z, Zhang X, Song W, Li Y, Liu G, Shi X, Wang X, Zhu M (2015) Temporal variation of green macroalgal assemblage on porphyra aquaculture rafts in the Subei Shoal, China. *Estuar Coast Shelf Sci* 163:23–28
4. Geng X, Li P, Yang J, Shi L, Li Xm, Zhao J (2020) *Ulva prolifera* detection with dual-polarization GF-3 SAR data. In: IOP conference series: earth and environmental science, vol 502, p 012026. IOP Publishing
5. Hu C (2009) A novel ocean color index to detect floating algae in the global oceans. *Remote Sens Environ* 113(10):2118–2129

6. Hu C, Li D, Chen C, Ge J, Muller-Karger FE, Liu J, Yu F, He MX (2010) On the recurrent *Ulva prolifera* blooms in the Yellow Sea and East China Sea. *J Geophys Res: Oceans* 115(C5)
7. Hu C, Feng L, Hardy RF, Hochberg EJ (2015) Spectral and spatial requirements of remote measurements of pelagic sargassum macroalgae. *Remote Sens Environ* 167:229–246
8. Hu L, Zeng K, Hu C, He MX (2019) On the remote estimation of *Ulva prolifera* areal coverage and biomass. *Remote Sens Environ* 223:194–207
9. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning, PMLR, pp 448–456
10. Lee JH, Pang IC, Moon IJ, Ryu JH (2011) On physical factors that controlled the massive green tide occurrence along the southern coast of the Shandong Peninsula in 2008: a numerical study using a particle-tracking experiment. *J Geophys Res: Oceans* 116(C12)
11. Lee ZP, Du KP, Arnone R (2005) A model for the diffuse attenuation coefficient of downwelling irradiance. *J Geophys Res: Oceans* 110(C2)
12. Li X, Liu B, Zheng G, Ren Y, Zhang S, Liu Y, Gao L, Liu Y, Zhang B, Wang F (2020) Deep-learning-based information mining from ocean remote-sensing imagery. *Nat Sci Rev* 7(10):1584–1605
13. Liu D, Keesing JK, He P, Wang Z, Shi Y, Wang Y (2013a) The world's largest macroalgal bloom in the Yellow Sea, China: formation and implications. *Estuar Coast Shelf Sci* 129:2–10
14. Liu F, Pang S, Chopin T, Gao S, Shan T, Zhao X, Li J (2013b) Understanding the recurrent large-scale green tide in the Yellow Sea: temporal and spatial correlations between multiple geographical, aquacultural and biological factors. *Marine Environ Res* 83:38–47
15. Lü X, Qiao F (2008) Distribution of sunken macroalgae against the background of tidal circulation in the coastal waters of Qingdao, China, in summer 2008. *Geophys Res Lett* 35(23)
16. Ronneberger O, Fischer P, Brox T (2015) U-Net: Convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention, pp 234–241. Springer
17. Smetacek V, Zingone A (2013) Green and golden seaweed tides on the rise. *Nature* 504(7478):84–88
18. Song W, Peng K, Xiao J, Li Y, Wang Z, Liu X, Fu M, Fan S, Zhu M, Li R (2015) Effects of temperature on the germination of green algae micro-propagules in coastal waters of the Subei Shoal, China. *Estuar Coast Shelf Sci* 163:63–68
19. Xiao J, Fan S, Wang Z, Fu M, Song H, Wang X, Yuan C, Pang M, Miao X, Zhang X (2020) Decadal characteristics of the floating *Ulva* and sargassum in the Subei Shoal, Yellow Sea. *Acta Oceanologica Sinica* 39(10):1–10
20. Xiao X, Agusti S, Lin F, Li K, Pan Y, Yu Y, Zheng Y, Wu J, Duarte CM (2017) Nutrient removal from Chinese coastal waters by large-scale seaweed aquaculture. *Sci Rep* 7(1):1–6
21. Xing Q, An D, Zheng X, Wei Z, Wang X, Li L, Tian L, Chen J (2019) Monitoring seaweed aquaculture in the Yellow Sea with multiple sensors for managing the disaster of macroalgal blooms. *Remote Sens Environ* 231:111279
22. Zhang J, Zhao P, Huo Y, Yu K, He P (2017) The fast expansion of *Pyropia* aquaculture in “Sansha” regions should be mainly responsible for the *Ulva* blooms in Yellow Sea. *Estuar Coast Shelf Sci* 189:58–65
23. Zhang Y, He P, Li H, Li G, Liu J, Jiao F, Zhang J, Huo Y, Shi X, Su R et al (2019) *Ulva prolifera* green-tide outbreaks and their environmental impact in the Yellow Sea. *China, National Science Review*

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits any noncommercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if you modified the licensed material. You do not have permission under this license to share adapted material derived from this chapter or parts of it.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Automatic Waterline Extraction of Large-Scale Tidal Flats from SAR Images Based on Deep Convolutional Neural Networks



Shuangshang Zhang, Qing Xu, and Xiaofeng Li

1 Introduction

Coastal zones are ecologically essential and exceptionally dynamic. Monitoring these regions is essential for coastal environmental protection and development. The waterline, also called shoreline or coastline in the coastal zones, is defined as contact between land and the water body. It plays an essential role in analyzing land/water resources, monitoring coastal erosion [3], as well as global sea-level rise.

Clouds easily contaminate optical remote sensing waterline detection. Waterline extraction from synthetic aperture radar (SAR) imagery is becoming more common due to the radar's all-weather and all-day capability. However, distinguishing the waterline in SAR images is not as simple a procedure for visible-band sensors. The wind-roughed and wave-modulated water return can frequently equal or exceed the return from a nearby land area, resulting in an inadequate contrast for unambiguous land-sea separation. In addition, affected by the moisture of the sandy sediments [6], this phenomenon is more evident in some tidal flat areas. Besides, the speckle noise generated by the coherent signal-scattering complicates the waterline extraction problem for SAR images.

Since the remote sensing data has been growing exponentially and the manual delineation is labor-intensive and subjective, several automatic or semi-automatic waterline extraction methods for SAR images have been proposed based on two conventional approaches: edge detection [10, 13, 16, 19, 28] and image segmentation [9, 14, 22, 24]. However, no matter which one they are based on, these methods more

S. Zhang · X. Li (✉)

CAS Key Laboratory of Ocean Circulation and Waves, Institute of Oceanology, Chinese Academy of Sciences, Qingdao 266071, China

e-mail: lixf@qdio.ac.cn

Q. Xu

College of Marine Technology, Faculty of Information Science and Technology, Ocean University of China, Qingdao 266100, China

© The Author(s) 2023

X. Li and F. Wang (eds.), *Artificial Intelligence Oceanography*,
https://doi.org/10.1007/978-981-19-6375-9_14

or less require preprocessing and postprocessing for an accurate extraction result from SAR images [9, 24].

Recently, deep convolutional neural networks (DCNN) have widely been employed to extract information from remote sensing images [11]. Several machine-learning-based methods have been proposed for waterline or coastline extraction from SAR images, which all show far better results than the conventional edge detectors [1, 8, 29]. However, unlike regular land or ice regions, the SAR imaging of tidal flat areas shows dramatic brightness changes under different sea conditions.

In this chapter, a modified U-Net has been used to create a framework for automatic waterline extraction from Sentinel-1 SAR images of a large-scale tidal flat at Subei Bank in the Southern Yellow Sea. The extracted waterlines are continued to be applied to construct the digital elevation model (DEM) series in different years for evolution analysis of tidal flat using the waterline method. In this chapter, we first describe our study area, the unique palm-like Radial Sand Ridges along the Jiangsu coast, and the various sandbanks' SAR imaging features under different sea conditions. Afterward, we introduce our input data and the DCNN-based method. Finally, after testing the trained model's performance, we developed a processing chain for constructing the tidal flats DEM with the automatically extracted waterlines and an assimilative ocean tide model.

2 Study Area and Data

The Jiangsu coast is located in the western part of the South Yellow Sea, and its offshore area is characterized by palm-shaped radial sand ridges (RSRs). The RSRs consist of more than ten prominent submarine sand ridges and have a unique radial palm shape with the central apex near Jianggang. This giant system is well-developed owing to the active tidal processes and abundant sediment supply from the river runoff [4]. It has a length of 200 km in the north-south direction and a width of 90 km in the east-west direction, with the water depth ranging from 0 to 25 m [27]. The complex hydrodynamic system [20, 30] makes the area's topography changeable. As shown in Fig. 1b, there are several large-scale tidal flats distributed in the study area.

Compared to optical imaging systems, the active microwave sensor acquires data independent from night and cloud cover, ensuring continuous study area acquisitions. The Sentinel-1 mission comprises a constellation of two polar-orbiting satellites, operating day and night performing C-band SAR imaging, enabling them to acquire imagery regardless of the weather [26]. The two satellites, Sentinel-1A (launch on 3 April 2014) and Sentinel-1B (launch on 25 April 2016), complement each other allowing six days revisit times or even less (in polar regions). With the support of Google Earth Engine [7], we collect 140 pre-processed Ground Range Detected (GRD) IW (interferometric wide-swath) mode with dual-band cross-polarization (VV and VH) and 10m spatial resolution Sentinel-1 SAR imagery from 2015 to 2019 for the waterline extraction analysis in this chapter. Among the 140 images, 52

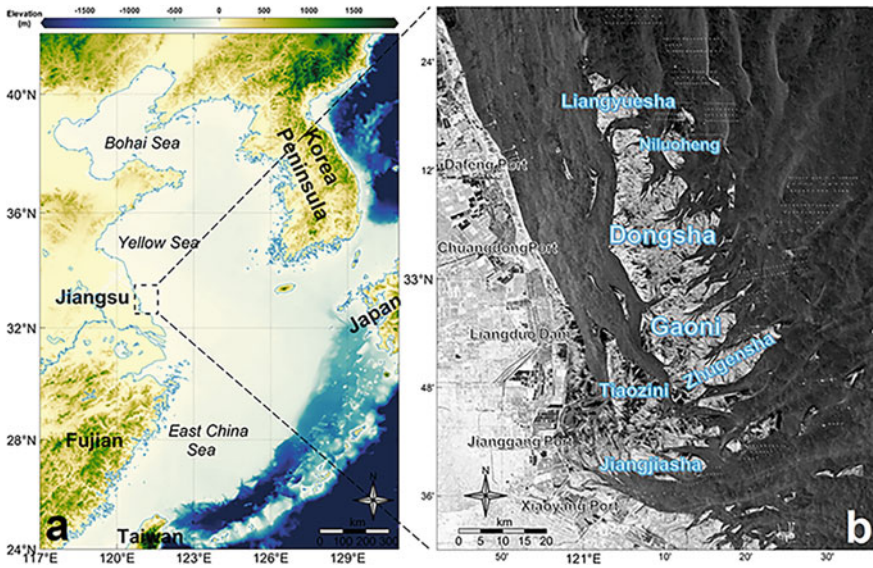


Fig. 1 **a** Overview of the study area. The elevation data are from the ETOPO1 (NOAA National Geophysical Data Center, 2009). **b** Sentinel-1B SAR image of the study area at low tide, imaged at Greenwich Mean Time (GMT) 09:54, 26 November 2019

acquired in 2019 are used for training and testing our DCNN model and the remains for constructing the large-scale tidal flats' DEM in Subei Bank.

Besides the speckle noise, the accuracy and efficiency of the automatic extraction of waterlines in the study area are mainly interfered with by two other factors: the rapid local brightness changes in seawater and tidal flats. The SAR image represents a two-dimensional radar backscatter map of the ocean surface roughness. Therefore, some related processes (such as winds, internal solitary waves, currents, underwater topography, oil spill, rainfall, and eddies) that cause local roughness changes will drive apparent brightness or darkness in imaging. According to Zhang et al. [31], affected by wind and tidal currents, the imaging features of shallow water topography in our study area can often be captured by SAR. As shown in the northeast corner of Fig. 1b, the three underwater sand ridges are shown as narrow bright stripes (1 km wide) in this SAR image. The non-uniform SAR imaging of the sea surface is more evident in Fig. 2. These four sub-images are acquired under different sea conditions and show a considerable imaging difference from each other both on seawater and the tidal flats. The uncertain changes bring great difficulties to the automatic extraction of waterlines for these large-scale tidal flats.

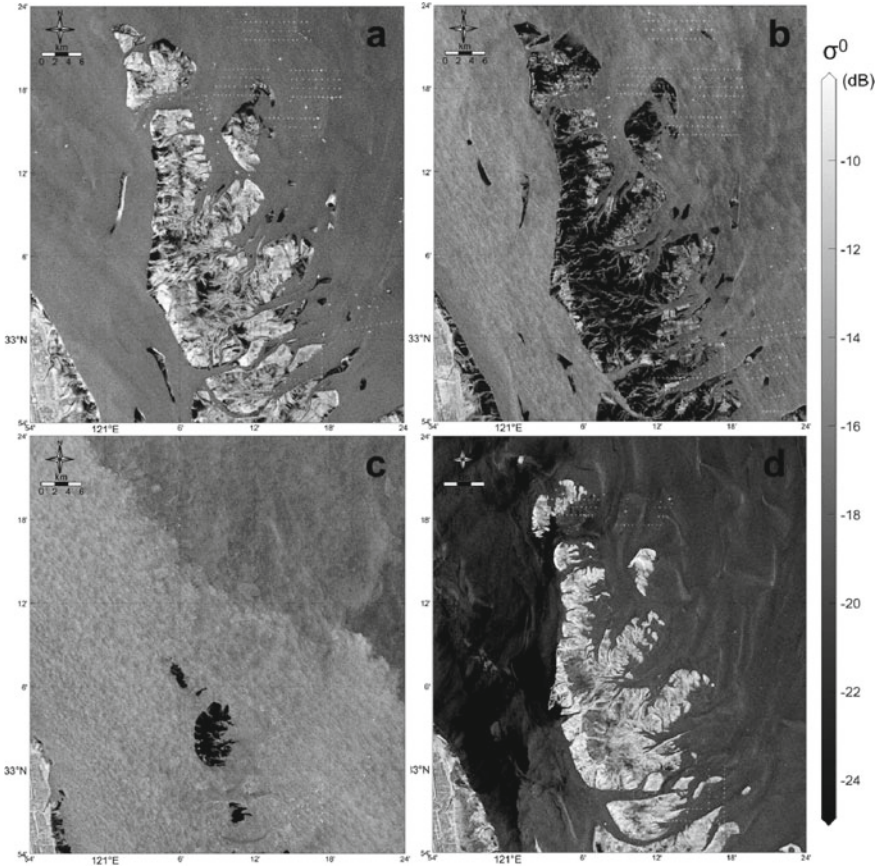


Fig. 2 Four typical Sentinel-1 VV-polarized SAR image examples acquired at the different tidal level: **a** Sentinel-1B image acquired at GMT 09:54, 19 March 2019; **b** Sentinel-1A image acquired at GMT 09:55, 26 December 2019; **c** Sentinel-1A image acquired at GMT 09:54, 23 November 2016; **d** Sentinel-1A image acquired at GMT 09:55, 21 July 2017

3 Methodology

3.1 U-Net

The DCNN to extract pivotal information from remote sensing images has been successfully applied in oceanography. Recently, Li et al. [11] established an improved U-Net network to efficiently and automatically extract different ocean process signatures in optical and radar images. The U-Net [23] is a modified fully convolutional network [15] initially developed for biomedical image segmentation. The network is based on the Fully Convolutional Network but extended to work with fewer training

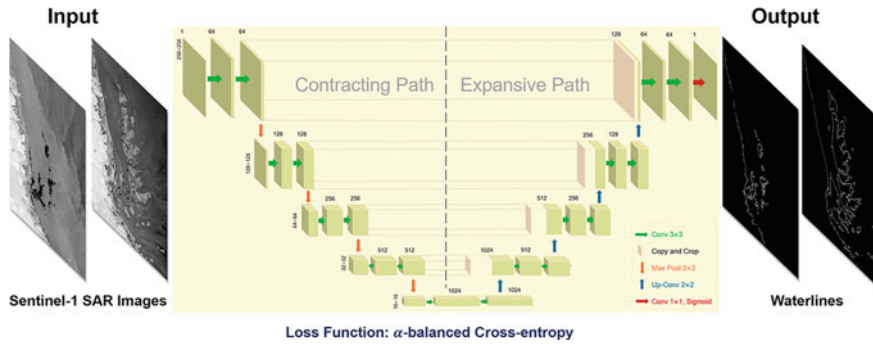


Fig. 3 The U-Net architecture specially tailored for this chapter

images to yield more precise segmentation. The network consists of a contracting path and an expansive path, giving it a U-shaped architecture. As shown in Fig. 3, the left contracting path is a typical convolutional network that consists of repeated application of convolutions, which are followed by a rectified linear unit (ReLU) and a max-pooling operation. During the contraction processing, the spatial information is reduced while the image feature is increased. The right expansive pathway combines a sequence of up-convolutions and concatenations with high-resolution features from the contracting path. One crucial modification in U-Net is that there are many feature channels in the upsampling part, allowing this network to propagate context information to higher resolution layers. Consequently, the expansive path is more or less symmetric to the contracting path, yielding a U-shaped architecture of this network. The main idea is to supplement a usual contracting network by successive layers, where upsampling operators replace pooling operations. Hence these layers increase the resolution of the output. A successive convolutional layer is able to learn to assemble precise output based on this information.

As shown in Fig. 3, the U-Net's last layer is 1×1 convolution with the Sigmoid activation. Traditionally, the loss function of the original U-Net is the cross-entropy. However, in the task of waterline extraction, the samples are highly unbalanced, i.e., the background samples' numbers are much higher than those of waterline samples (less than 1% points in whole SAR images). Motivated by Lin et al. [12], we adopt the α -balanced cross-entropy in this task.

3.2 Data Preparation

The original spatial resolution of the dual-polarized Sentinel-1 SAR imagery downloaded from Google Earth Engine is 10m. After statistical analysis, we found that the boundary lines between land and water on the VV polarization images are more apparent than those of the VH images. To save computing resources and training

time, we only use the VV polarization images and downsample them to a resolution of 50m. Finally, a full SAR image of the study area is 2229 pixels high and 2005 pixels wide. We further crop the images and their corresponding ground truth into sub-images with 256×256 pixels size to keep memory consumption low during training (the edge is filled with black when it is less than 256 pixels). In the end, we acquired a total of 3024 pairs of images for training the U-Net. In addition, before training the network, data augmentations are performed to compensate for a limited number of images in the training dataset. Data augmentation is a technique to increase the amount of data by adding slightly modified copies of already existing data, including random contrast, brightness change, image rotation/cropping, noise injection, etc. It may help the network learn more tidal flat waterline features in the SAR imagery with protean brightness and shapes.

Ground truth labels are necessary when we train a machine learning classifier. Since there is no corresponding waterline product and a method that can automatically extract these edges, we use manual drawing to obtain the ground truth value of the waterline of the 52 Sentinel-1 SAR images acquired in 2019 (the depicted result is shown as the output in Fig. 3). In practice, we use a stylus and touch screen to represent the position of the waterlines accurately. We first randomly select 1/5 of 52 pairs of images, that is, ten pairs as the testing set, to examine the accuracy of extracting the waterlines in the independent data of the trained model. The remaining 42 SAR images with their labels are used for U-Net model training.

3.3 Training

The cropped sub-images from 42 Sentinel-1 SAR imagery are divided into 80% for training and 20% for validation in the training process. The training and testing of the network are implemented by Keras/Tensorflow framework (on NVIDIA Tesla V100 GPU, 32 GB). As mentioned above, we adopt the α -balanced cross-entropy as the loss function (α is set to 0.99) and the classification accuracy as the performance metric. Furthermore, the batch size is set to 16, and the number of epochs is 4000. Finally, the classification accuracy of the 20% validation images is 94.45% after nearly ten hours of training.

4 Results

4.1 Model Performance

The binary classification accuracy is estimated by calculating the precision and recall of the automatically extracted waterlines to manual ones. The mean precision and recall of the ten testing images are 0.92 and 0.77, respectively (see Table 1 for details).

Table 1 Details of ten testing Sentinel-1 SAR images

Image ID	Satellite	Imaging time (MM/DD HH:mm, GMT)	Tidal level(m)	Precision	Recall
1	Sentinel-1A	02/17 09:55	-3.32	0.96	0.76
2	Sentinel-1A	04/30 09:55	-1.39	0.93	0.78
3	Sentinel-1B	05/30 09:54	-1.40	0.93	0.75
4	Sentinel-1A	06/05 09:55	-1.28	0.91	0.69
5	Sentinel-1A	06/17 09:55	-2.40	0.92	0.83
6	Sentinel-1A	06/29 09:55	-1.52	0.89	0.78
7	Sentinel-1A	07/11 09:55	0.97	0.93	0.84
8	Sentinel-1B	07/29 09:54	-1.95	0.94	0.79
9	Sentinel-1A	09/09 09:55	-0.47	0.95	0.81
10	Sentinel-1A	12/26 09:55	-3.02	0.93	0.74
Mean				0.92	0.77

Four examples of the ten testing results under different sea conditions are shown in Fig. 4. We use three-color lines to compare the difference between the model results and the true values. Yellow represents the waterlines accurately extracted by our DCNN-based model. Red indicates the missing parts of the model, while blue means the false detected lines that shouldn't be there. The fluctuation of the tides causes drastic changes in the shape and distribution of the waterlines. Figures 4a-c show the results under three typical tidal levels: high, medium, and low, which can also be judged from the exposed area of the tidal flats. What's interesting here is that Fig. 4c captures a small amount of Enteromorpha information shown as the little bright spot in the northern sea. As shown by the yellow lines in Fig. 4, most of the obtained extraction results from the DCNN-based model correspond well to the manually annotated ground truth waterlines.

4.2 Automatic Topographic Mapping of Tidal Flats

Knowledge of a waterline's orientation, position, and outline is essential in sea autonomous pilot, verification of coastal platform's attitude and place, the geolocation of ships, geographic mapping, etc. It also has a specific application for constructing a digital elevation model (DEM) of an intertidal zone by the waterline method [16]. This method is first introduced by Mason et al. [17]. The waterline can be regarded as a quasi-contour line of the topography. This method was proved to be one of the best methods that provide an excellent trade-off between accuracy and cost-effectiveness for the DEM generation of tidal flats [18, 24, 32].

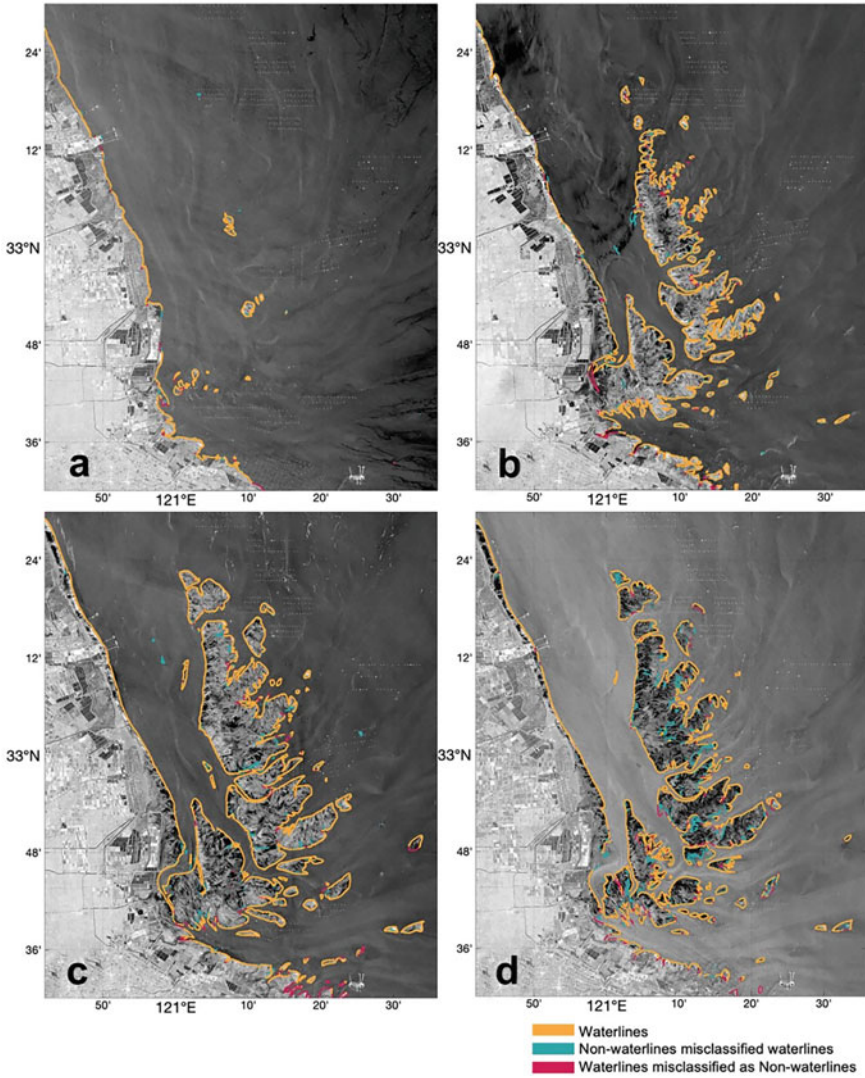


Fig. 4 Four examples of the ten testing images overlaid with their corresponding trained model extraction results and ground truth waterlines: **a** Sentinel-1A image acquired at GMT 09:55, 11 July 2019; **b** Sentinel-1B at 09:54, 29 July 2019; **c** Sentinel-1A at 09:55, 17 June 2019; **d** Sentinel-1A at 09:55, 05 June 2019. The mean precision and recall of the ten testing images are 0.92 and 0.77, respectively

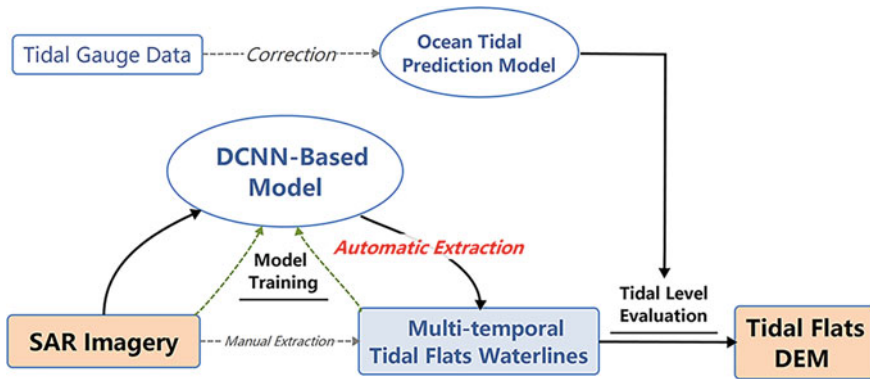


Fig. 5 The flowchart of the method for automatic topographic mapping of the tidal flats developed in this study

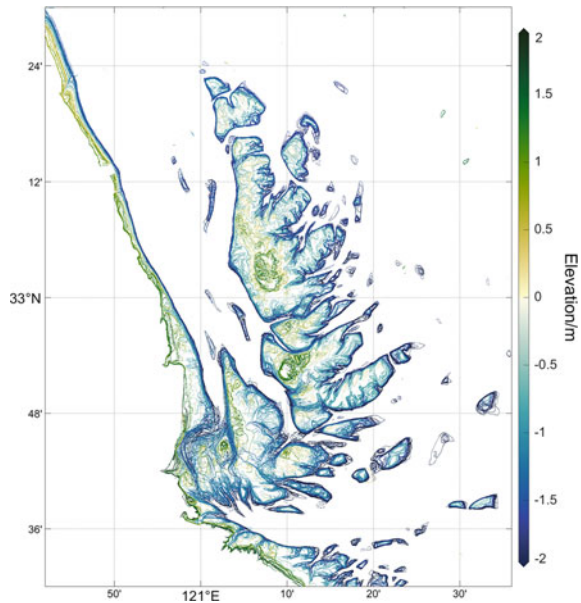
This study further attempts to establish a method for automatic topographic mapping of tidal flats based on the waterline method and the DCNN-based waterline extraction model for SAR images. The flowchart of this method is shown in Fig. 5. The elevation generation process can be divided into four steps:

1. Gaining the waterline information in a series of Sentinel-1 SAR images showing different tidal levels automatically by the trained DCNN-based model;
2. Discretizing the lines into points and estimating their Lon/Lat position from original images;
3. Evaluating the water level of each point by the ocean tidal prediction model at the SAR imaging time;
4. At last, interpolating the resulting grid of quasi-contour lines to a DEM map.

According to the previous subsection, the DCNN-based model performs well, with little or no postprocessing required to obtain accurate waterlines, even for large-scale tidal flats like the Subei Bank. In addition, our method has extremely high extraction efficiency, with an average of two seconds per SAR image (2229×2005 pixels, based on the NVIDIA Tesla V100 32GB GPU). According to Zhang et al. [31], the TPXO tide model [5] perform well in the tidal phase in our study area. However, this tidal model presents a systematic underestimation of tidal amplitude. Then, the in-situ water level data from two tidal gauge stations in our study area were used to calibrate this tide model (see [31] for details). The corrected TPXO tide models with Tidal Model Driver software are employed as the ocean tidal prediction model to evaluate the tidal level for each point of each waterline on this method.

We first used the waterlines of 2019 to verify the accuracy of the waterline method in measuring tidal flats elevation in our study area. We eliminated five scenes with wind speed greater than 10 m/s, which have a large offset from their original location caused only by tidal fluctuation [25]. Then the remaining waterlines from 47 SAR images was assigned with the tidal level value using the corrected ocean tidal model (see Fig. 6).

Fig. 6 Assembled tidal level evaluated waterlines extracted manually from all SAR images acquired in 2019



Finally, as shown in Fig. 7, these points/lines were interpolated to obtain gridded DEM of the large-scale tidal flats in our study area. One transect line of measured topographic data, which were acquired by an in-situ survey in May 2019, was used to test the accuracy of the derived DEM. The mean absolute error along this transect line is about 0.3 m (see Fig. 8).

Among the waterline method steps, the most time-consuming one is to extract the waterline, especially for SAR images. With the support of the DCNN-based automatic waterline extraction model, the efficiency of implementing this method can be significantly improved. We took the generation of the tidal flats' DEM for 2018 as an example. A total of 29 pre-processed Sentinel-1 SAR images throughout the year were collected with Google Earth Engine and used as inputs to the DCNN-based model to obtain the geolocation of waterlines quickly. The final gridded DEM result for 2018 is shown in Fig. 9a after the same subsequent processes such as tidal level evaluation and spatial interpolation. In addition, interannual topographic changes can be analyzed by subtracting these two waterline-derived DEMs. As shown in Fig. 9b, the topography of these large-scale tidal flats changes significantly in two years under the action of strong tidal currents [2]. The erosion-deposition balance showed a net deposition of 0.12 km^3 from 2018 to 2019. It implies our presented methodologies are also suitable for rapid monitoring the morphological and sedimentary changes of large-scale intertidal areas.

Fig. 7 The derived DEM result of Subei Bank for 2019 using the waterline method (overlaid on the Sentinel-1A SAR image acquired at GMT 09:55, 20 November 2019)

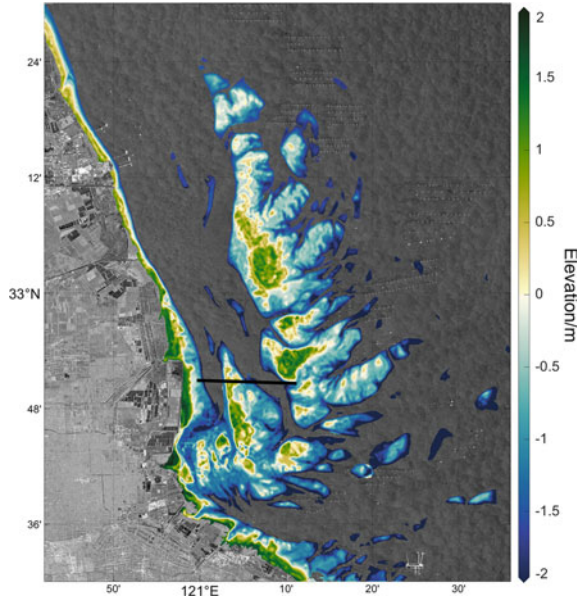
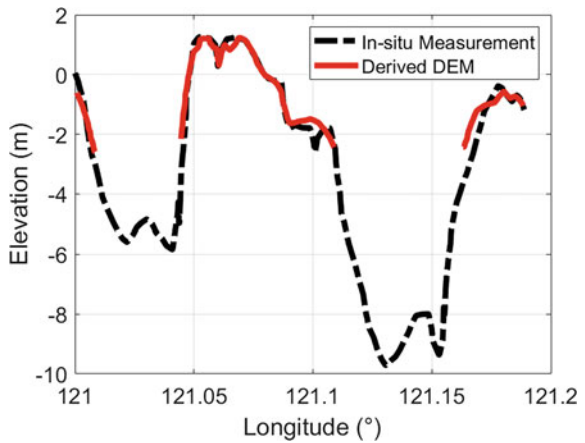


Fig. 8 Topography profile comparison between the derived DEM and the in-situ transect (*solid black line* in Fig. 7)



5 Discussions

Because of the frequent lack of consistent, sufficient intensity contrast between land and water regions and the complications of distinguishing waterline from other object boundaries, waterline extraction is harrowing with most general-purpose edge detectors or image segmentation techniques, especially for radar images in the intertidal areas. Previous studies used edge detection methods where a thresholding process was necessary at some point under relatively complex imaging conditions (such as

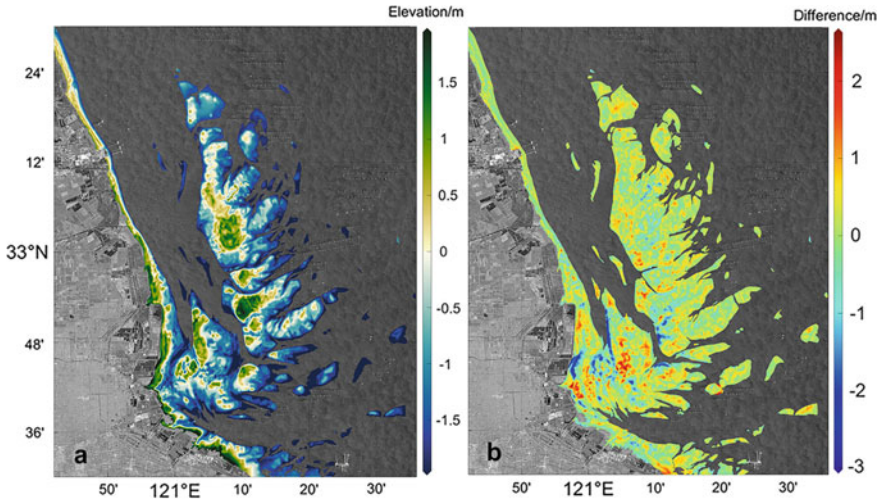


Fig. 9 **a** The derived DEM result of Subei Bank for 2018 using the waterline method based on the waterlines automatically extracted by our DCNN model; **b** Tidal flats DEM of difference map between 2019 and 2018 (both overlaid on the Sentinel-1A SAR image acquired at GMT 09:55, 20 November 2019)

the methods developed by [16, 21], and [9]). In addition, with the unprecedented amount of data containing waterline information available, an automatic extraction method should be prioritized. The DCNN-based method developed in this study performed well for automatic waterline extraction from SAR imagery in large-scale tidal flats area under changeable imaging conditions.

With the support of big data platforms such as Google Earth Engine and the ocean tidal prediction model, we developed a waterline method-based workflow that can quickly obtain relatively accurate DEM of tidal flats after extracting multi-temporal waterlines from SAR images under different tidal levels. This technique provides an efficient method for the rapid analysis of large-scale tidal flat topography evolution, which is of great significance for applying SAR images to monitoring coastal terrains.

6 Conclusions

This chapter proposes a DCNN-based method to extract waterlines automatically from SAR images. Our approach shows a relatively high extraction accuracy for the waterlines in complicated large-scale tidal flats (the mean precision and recall are 0.92 and 0.77, respectively) and efficiency (several seconds per image) simultaneously. This chapter also presents the first attempt for intertidal DEM generation of the Subei Bank using the waterline method by analyzing high spatial resolution SAR images. The DEM results show that, in general, there is a good agreement between the derived

elevation and in-situ topographic data, implying that the waterline method based on SAR images can be used for large-scale tidal flats such as the Subei Bank area. Furthermore, based on the waterline extraction model and the waterline method, we developed a novel workflow for automatic topographic mapping of large-scale tidal flats, which has excellent potential for rapid analysis of intertidal topography evolution.

References

1. Baumhoer CA, Dietz AJ, Kneisel C, Kuenzer C (2019) Automated extraction of Antarctic glacier and ice shelf fronts from Sentinel-1 imagery using deep learning. *Remote Sens* 11(21):2529. <https://doi.org/10.3390/rs11212529>
2. Chen J, Wang Y, Zhang R, Lin X (2007) Stability study on the Dongsha Sandbanks in submarine radial sand ridges field off jiangsu coast (in Chinese). *Ocean Eng* 25(1):105–113
3. Ding X, Li X (2014) Shoreline movement monitoring based on SAR images in Shanghai, China. *International Journal of Remote Sensing* 35(11–12):3994–4008. <https://doi.org/10.1080/01431161.2014.916480>
4. Ding X, Kang Y, Ge X, Li Q, Zhang T (2011) Tidal flat evolution analysis using remote sensing on Tiaozini flat of the radial sand ridges (in Chinese). *Journal of Hohai University: Natural Sciences* 39(2):231–236
5. Egbert GD, Erofeeva SY (2002) Efficient inverse modeling of barotropic ocean tides. *J Atmos Oceanic Tech* 19:22
6. Gade M, Wang W, Kemme L (2018) On the imaging of exposed intertidal flats by single- and dual-co-polarization Synthetic Aperture Radar. *Remote Sens Environ* 205:315–328. <https://doi.org/10.1016/j.rse.2017.12.004>
7. Gorelick N, Hancher M, Dixon M, Ilyushchenko S, Thau D, Moore R (2017) Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sens Environ* 202:18–27. <https://doi.org/10.1016/j.rse.2017.06.031>
8. Heidler K, Mou L, Baumhoer C, Dietz A, Zhu XX (2021) HED-UNet: Combined segmentation and edge detection for monitoring the Antarctic coastline. *IEEE Trans Geosci Remote Sens*, 1–14. <https://doi.org/10.1109/TGRS.2021.3064606>
9. Heygster G, Dannenberg J, Notholt J (2010) Topographic mapping of the German tidal flats analyzing SAR images with the waterline method. *IEEE Trans Geosci Remote Sens* 48(3):1019–1030. <https://doi.org/10.1109/TGRS.2009.2031843>
10. Lee J, Jurkevich I (1990) Coastline detection and tracing in SAR images. *IEEE Trans Geosci Remote Sens* 28(4):662–668. <https://doi.org/10.1109/TGRS.1990.572976>
11. Li X, Liu B, Zheng G, Ren Y, Zhang S, Liu Y, Gao L, Liu Y, Zhang B, Wang F (2020) Deep-learning-based information mining from ocean remote-sensing imagery. *Natl Sci Rev* 7(10):1584–1605. <https://doi.org/10.1093/nsr/nwaa047>
12. Lin TY, Goyal P, Girshick R, He K, Dollár P (2018) Focal loss for dense object detection. [arXiv:1708.02002](https://arxiv.org/abs/1708.02002) [cs] 1708.02002
13. Liu C, Xiao Y, Yang J (2017) A coastline detection method in polarimetric SAR images mixing the region-based and edge-based active contour models. *IEEE Trans Geosci Remote Sens* 55(7):3735–3747. <https://doi.org/10.1109/TGRS.2017.2679112>
14. Liu H, Jezek KC (2004) Automated extraction of coastline from satellite imagery by integrating Canny edge detection and locally adaptive thresholding methods. *Int J Remote Sens* 25(5):937–958. <https://doi.org/10.1080/0143116031000139890>
15. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 3431–3440

16. Mason D, Davenport I (1996) Accurate and efficient determination of the shoreline in ERS-1 SAR images. *IEEE Trans Geosci Remote Sens* 34(5):1243–1253. <https://doi.org/10.1109/36.536540>
17. Mason DC, Davenport IJ, Robinson GJ, Flather RA, McCartney BS (1995) Construction of an inter-tidal digital elevation model by the ‘Water-Line’ Method. *Geophys Res Lett* 22(23):3187–3190. <https://doi.org/10.1029/95GL03168>
18. Mason DC, Gurney C, Kennett M (2000) Beach topography mapping—a comparison of techniques. *J Coast Conserv* 6(1):113–124. <https://doi.org/10.1007/BF02730475>
19. Modava M, Akbarizadeh G (2017) Coastline extraction from SAR images using spatial fuzzy clustering and the active contour method. *Int J Remote Sens* 38(2):355–370. <https://doi.org/10.1080/01431161.2016.1266104>
20. Ni W, Wang Y, Zou X, Zhang J, Gao J (2014) Sediment dynamics in an offshore tidal channel in the southern Yellow Sea. *Int J Sedim Res* 29(2):246–259. [https://doi.org/10.1016/S1001-6279\(14\)60040-8](https://doi.org/10.1016/S1001-6279(14)60040-8)
21. Niedermeier A, Hoja D, Lehner S (2005) Topography and morphodynamics in the German Bight using SAR and optical remote sensing data. *Ocean Dyn* 55(2):100–109. <https://doi.org/10.1007/s10236-005-0114-2>
22. Nunziata F, Buono A, Migliaccio M, Benassai G (2016) Dual-polarimetric C- and X-band SAR data for coastline extraction. *IEEE J Sel Topics Applied Earth Obs Remote Sens* 9(11):4921–4928. <https://doi.org/10.1109/JSTARS.2016.2560342>
23. Ronneberger O, Fischer P, Brox T (2015) U-Net: Convolutional networks for biomedical image segmentation. [arXiv:1505.04597](https://arxiv.org/abs/1505.04597) [cs] 1505.04597
24. Salameh E, Frappart F, Turki I, Laignel B (2020) Intertidal topography mapping using the waterline method from Sentinel-1 & -2 images: The examples of Arcachon and Veys Bays in France. *ISPRS J Photogramm Remote Sens* 163:98–120. <https://doi.org/10.1016/j.isprsjprs.2020.03.003>
25. Tong SS, Deroin JP, Pham TL (2020) An optimal waterline approach for studying tidal flat morphological changes using remote sensing data: A case of the northern coast of Vietnam. *Estuar Coast Shelf Sci* 236:106613. <https://doi.org/10.1016/j.ecss.2020.106613>
26. Torres R, Snoeij P, Geudtner D, Bibby D, Davidson M, Attema E, Potin P, Rommen B, Floury N, Brown M, Traver IN, Deghaye P, Duesmann B, Rosich B, Miranda N, Bruno C, L’Abbate M, Croci R, Pietropaolo A, Huchler M, Rostan F (2012) GMES Sentinel-1 mission. *Remote Sens Environ* 120:9–24. <https://doi.org/10.1016/j.rse.2011.05.028>
27. Wang Y, Liu Y, Jin S, Sun C, Wei X (2019) Evolution of the topography of tidal flats and sandbanks along the Jiangsu coast from 1973 to 2016 observed from satellites. *ISPRS J Photogramm Remote Sens* 150:27–43. <https://doi.org/10.1016/j.isprsjprs.2019.02.001>
28. Wei X, Zheng W, Xi C, Shang S (2021) Shoreline extraction in SAR image based on advanced geometric active contour model. *Remote Sens* 13(4):642. <https://doi.org/10.3390/rs13040642>
29. Xie S, Tu Z (2015) Holistically-Nested edge detection. [arXiv:1504.06375](https://arxiv.org/abs/1504.06375) [cs] 1504.06375
30. Xu F, Tao J, Zhou Z, Coco G, Zhang C (2016) Mechanisms underlying the regional morphological differences between the northern and southern radial sand ridges along the Jiangsu Coast, China. *Mar Geol* 371:1–17. <https://doi.org/10.1016/j.margeo.2015.10.019>
31. Zhang S, Xu Q, Zheng Q, Li X (2017) Mechanisms of SAR imaging of shallow water topography of the Subei Bank. *Remote Sens* 9(11):1203. <https://doi.org/10.3390/rs9111203>
32. Zhang S, Xu Q, Wang H, Kang Y, Li X (2022) Automatic waterline extraction and topographic mapping of tidal flats from SAR images based on deep learning. *Geophys Res Lett*. <https://doi.org/10.1029/2021GL096007>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits any noncommercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if you modified the licensed material. You do not have permission under this license to share adapted material derived from this chapter or parts of it.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Extracting Ship's Size from SAR Images by Deep Learning



Yibin Ren, Xiaofeng Li, and Huan Xu

1 Introduction

Ship detection is very important to marine transportation [5]. Space borne Synthetic Aperture Radar (SAR) has been one of the most critical data source for ship detection because it can penetrate the clouds and track objects in all kinds of weather [28]. In marine applications, ship recognition from SAR imagery has long been a hotspot [4, 9, 19]. With the advancement of image analysis technology, SAR images can be used to derive more detailed ship information [8]. The size of a ship provides basic information for ship classification [11]. And the size information can provide useful information for ship classification. The intricate geometric parameter estimate is also a part of the interpretation of SAR image. A method for extracting ship size that is both efficient and precise will bring a new concept for SAR image interpretation.

Ships, in general, are metallic objects that may reflect SAR sensor electromagnetic radiation significantly more strongly than the surrounding ocean. On SAR images, one ship can be identified as a bright back scattering intensity target with high normalized radar cross-section (NRCS) values. The minimum bounding rectangle (MBR) is a geometric characteristic of the ship's NRCS that offers a preliminary size for determining a ship's ground size. In the meantime, the ship's superstructure, sea-ship interaction, and imaging conditions all have an effect on the NRCS. Li et al. [11]. These factors lead to a large gap between the initial size and the ground size. Figure 1 shows several examples of ship's signature on SAR images, the size of the MBR,

Y. Ren · X. Li (✉)

CAS Key Laboratory of Ocean Circulation and Waves, Institute of Oceanology,
Chinese Academy of Sciences, Qingdao 266071, China
e-mail: lixf@qdio.ac.cn

Key Laboratory of Earth Observations of Hainan Province, Sanya 572029, China

H. Xu

School of Geomatics and Marine Information, Jiangsu Ocean University,
Lianyungang 222005, China

© The Author(s) 2023

X. Li and F. Wang (eds.), *Artificial Intelligence Oceanography*,
https://doi.org/10.1007/978-981-19-6375-9_15

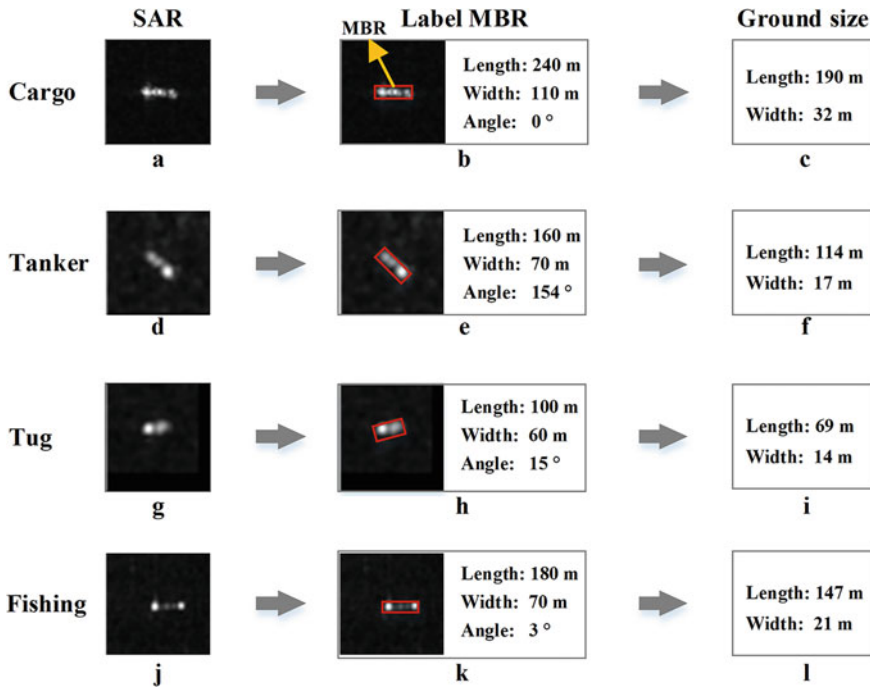


Fig. 1 Examples of ships on SAR images. **a/d/g/j** ship signature on SAR image; **b/e/h/k** labeled MBR of ship signature and the MBR's size; **c/f/i/l** the ship's ground size

and the ground size of the ship. The MBR is labeled by visual interpretation. The difference between the MBRs and the ground size appears to be clear. As a result, precisely extracting ship size from SAR images is difficult.

2 Traditional Methods

2.1 Typical Procedure of Traditional Methods

The majority of classic techniques for extracting ship size from SAR images have three stages (Fig. 2): (1) binarization, (2) initial size extraction, and (3) accurate size estimation. Binarization divides the pixels in the SAR image into two groups: ship signatures and non-ship signatures. The binary result is then converted into an MBR in the second phase. The length and width of the created MBR are used to determine the ship's starting size. Finally, a regression model is used to determine the accurate ship size using the initial size and other relevant factors such as the maximum and minimum NRCS of the ship signature. Statistical/machine learning

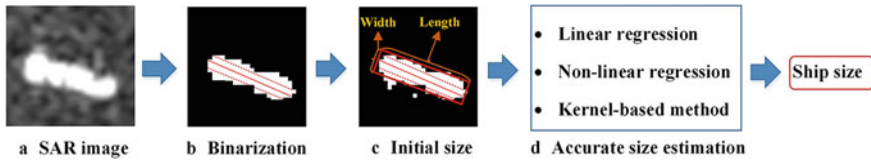


Fig. 2 The procedure of the traditional algorithm for ship size extraction from SAR images

(ML) methods, such as linear regression, non-linear regression, and kernel-based methods, are commonly used in regression models.

2.2 Representative Traditional Methods

Stasolla and Greidanus [26] used Constant False Alarm Rate (CFAR) to binary the SAR image. CFAR is a common method [21, 29, 30] that separates ship signatures and backgrounds. Further, to extract the ship's MBR, they used the mathematical morphology method to refine the signature. They adopted the MBR's length and width as the ship's final length and width without a third step. They tested their model with 127 available ship samples from Sentinel-1 images. The mean absolute error (MAE) of length is 30 m (relative error 16%), and the MAE of width is 11 m (relative error 37%). In 2018, Li et al. [11] estimated the ship's size of the OpenSARShip [7]. The ship signature was obtained using a threshold-based approach. They use an image segmentation procedure to refine the ship signature and determine the original ship size. Finally, the gradient boosting model is employed to estimate the accurate ship size. The MAE of the length and width, according to experiments, is 8.80 m (relative error 4.66 percent) and 2.17 m (relative error 7.01%), respectively.

2.3 Issue to be Further Addressed

The accuracy of ship size extraction is improving as years roll on. The standard three-step procedure is quite complicated. Binarization and initial size extraction need advanced image processes in order to meet the next estimation stage [11]. The third stage is similarly difficult [20]. The inaccuracies caused in each stage will add up and eventually compromise the accuracy of the final size extraction. It is possible to build new approaches to increase ship size extraction accuracy and efficiency in the era of big data.

Deep learning (DL), as the cutting-edge AI technology, has made great achievements in computer vision [10]. Multiple neural network layers make up a typical DL model. It accepts raw data as input and learns the essential characteristics automatically to perform classification or prediction [25]. End-to-end learning is the term

for this process. DL simplifies feature engineering and is well suited to modeling massive data and complex interactions when compared to traditional machine learning. DL has been successfully employed in oceanography, geography, and remote sensing in recent years [12, 13, 22, 24, 31, 32]. DL proposes novel approaches to the problem of estimating the size of a ship.

3 Deep Learning Method

3.1 Ship Detection Based on DL

A deep convolution neural network (CNN) is a subtype of DNN that is made up of CNN layers. CNN-based models have had a lot of success in target detection. Researchers proposed CNN-based ship detection models, such as models based on faster region-based convolutional network (Faster-RCNN) [23], single-shot multi-box detector (SSD) [15], and you only look once (YOLO) [2]. Orientation is an important characteristic of a ship. Several researchers suggested a rotatable bounding box (RBB) to replace the usual non-rotating RBB, such as DRBox [14] and DRBox-v2 [1].

For the ship detection task, DL has become the first choice. DL-based models achieve end-to-end detections with higher accuracy and robustness over conventional models. However, for ship size extraction, there is almost no application of deep learning. Therefore, developing an end-to-end DL model is necessary.

3.2 SSENet: A Deep Learning Model to Extract Ship Size from SAR Images

SSENet is a new end-to-end DL model that replaces the previous three-step process for extracting ship size from SAR data. The model uses DRBox-v2 to create the ship's RBB from the SAR image and a DNN-based regression model to estimate the accurate ship size. The DNN-based regression model is proposed using a hybrid input and a loss function termed mean scaled square error (MSSE), which considerably increases ship size estimation accuracy.

3.2.1 Overall Structure of SSENet

SSENet's overall structure consists of three phases (Fig. 3): (1) RBB generation; (2) accurate ship size estimation; (3) MSSE loss calculation and overall model optimization. The SAR chip is used as input in the first stage, which uses a deep CNN model called DRBox-V2 to automatically detect the ship's RBBs. The RBB with

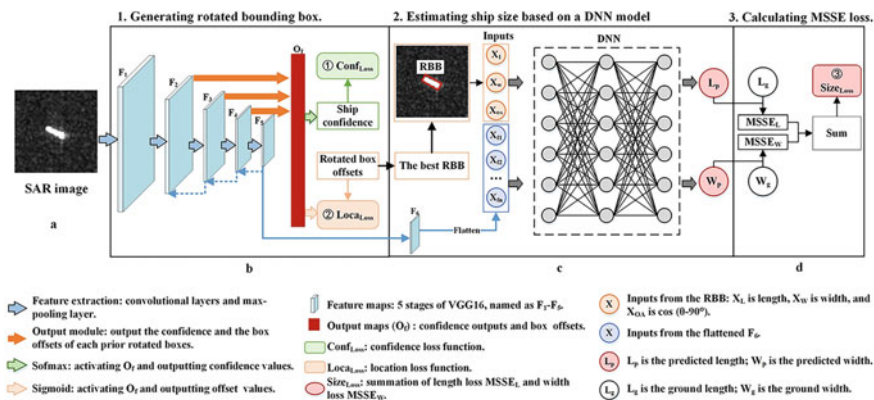


Fig. 3 Structure of SSENNet

the highest confidence is chosen as the initial RBB. A DNN model is used in the second stage to estimate ship size. The DNN model takes two types of data as inputs: (1) the initial length, width, and orientation angle, and (2) the SSD feature map. The accurate ship's length and width are generated using the DNN model.

3.2.2 Generating RBB for the Ship

The DRBox-v2 is used to generate RBB for the ship [1]. Its input is a 300×300 pixels SAR image, and its output is a series of RBBs. DRBox-v2 contains two sub-modules: a feature extraction module and an output module. The feature extraction module extracted abstracted features. Here, the VGG16 is employed as the feature extraction module. The VGG16 consists of five feature extraction units. Two stacking CNN layers make up the first feature extraction unit, while a max-pooling layer and two stacking CNN layers make up the others. Each feature extraction unit produces a three-dimensional feature map as its output. Five feature maps named F_1, F_2, \dots, F_5 are generated. The number of channels in the F_1 - F_5 feature maps is 64, 128, 256, and 512. The pooling kernel is 2×2 . After on max-pooling layer, the spatial size of a feature map is downscaled as $1/2$ size of its original size. As the input SAR image is 300×300 pixels, the spatial size of F_1 - F_5 feature maps is $300 \times 300, 150 \times 150, 75 \times 75, 38 \times 38,$ and 19×19 pixels.

The output module generates output maps by convolutioning feature maps O_f , Fig.2b. There are two outputs for one SAR image: the confidence of being a ship, as well as the geographic offsets of prior RBBs. A softmax function activates the O_f to obtain the confidence output. A sigmoid function activates the O_f to obtain the location offsets. Three feature maps ($F_2, F_3,$ and F_4) are fused to generate O_f .FPN is used to combine different feature maps. The cross-entropy and the smooth L1 loss [15] are used as the confidences loss and geographic loss for DRBox-v2.

Following the first process, a ship's candidate RBBs are collected, providing beginning references for the future exact size estimation.

3.2.3 Estimating Ship Size Based on a DNN Model

There are two elements to the DNN model's inputs, as shown in Fig. 3c. The initial ship size and orientation angle, which are determined from the best RBB and give primary and direct information for correct ship size regression, are the first part. The DRBox-v2 generates a sequence of ship RBBs. As the best RBB, the RBB with the highest confidence value is chosen. The initial ship size is the length and width of the best RBB. Furthermore, the best RBB's orientation angle is the ship's orientation angle, as shown in Fig. 4. It has an impact on the SAR image's ship signature [7, 11]. As the orientation does not distinguish between the bow and the stern of one ship, we transform the angle's range to $(-90^\circ, 90^\circ]$.

The other component of the inputs is the feature map derived from the input SAR image. In typical environmental conditions, the ship's signature in the SAR image reflects the sea clutter. It indicates whether the ship is moving or stationary. During the SAR integration time, a moving target is frequently found in several resolution cells. Smearing and brightness loss in the SAR image are caused by the dispersion of backscattered energy. A moving ship's signature reveals an azimuth displacement. The SAR system receives the Doppler signal from the scatter in the azimuth direction. A stationary ship's azimuth position is identical to the azimuth position of a SAR platform. The Doppler shift, on the other hand, has an extra component for a moving ship, resulting in an azimuth change in the ship signature. The environmental conditions during satellite imaging, such as wind fronts, ocean waves, and rain cells, alter the ship's signature on the SAR image. Under typical conditions, the sea-ship interaction produces a complicated ship motion in the real world and a polarimetric

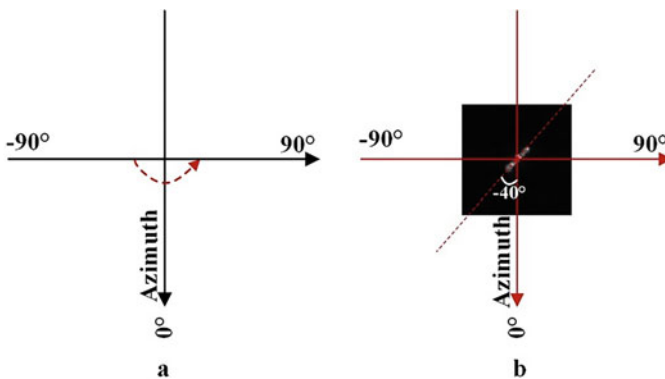


Fig. 4 Illustration of the ship orientation. **a** Coordinate system; **b** An example of a ship chip

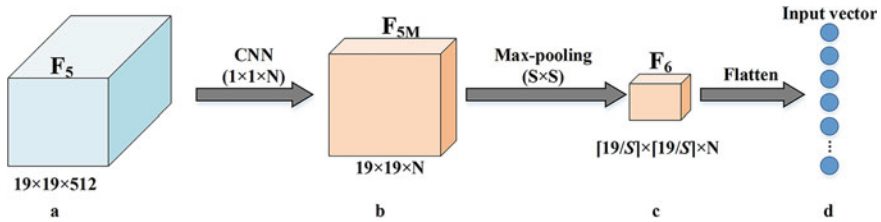


Fig. 5 Transforming the feature map F_5 as inputs. **a** F_5 feature map. **b** Compressing F_5 in the channel dimension and obtain the F_{5M} . **c** Compressing F_{5M} in the spatial dimension and obtain F_6 . **d** Flattening F_6 as one-dimensional input vector

scattering signature with a wide range of polarimetric scattering processes [14, 16, 17]. In reference [11], the relationship between the status of the ship, the surroundings, and the ship's size has been demonstrated. The abstracted feature map derived from the input SAR image contains the factors stated above. Therefore, the feature map F_5 in Fig. 3b is employed as the other component of the input.

F_5 is a three-dimensional feature map with 512 19×19 pixels channels. The input vector contains 184,832 ($512 \times 19 \times 19$) elements, which brings training difficulties for the fully connected DNN regression model. It is necessary to make some transformations to reduce the dimension of F_5 .

As shown in Fig. 5a, b, we transform F_5 by a CNN layer with $1 \times 1 \times N$ convolutional kernels, obtaining F_{5M} . Compared with F_5 , the channel number of F_{5M} is reduced from 512 to N , Fig. 4b. F_5 is compressed in channel dimension. Then, an S size max-pooling is performed on the new feature map F_{5M} , and a new feature map F_6 is obtained, Fig. 5c. The spatial size of the F_6 is $\lceil 19/S \rceil$. The values of N and S are defined by experiments. Finally, F_6 is flattened as a one-dimensional feature vector. The flattened vector is concatenated with the initial width, length, and orientation to form the inputs of the DNN model, Fig. 3c.

As shown in Fig. 3c, to perform regression, three hidden NN layers are used. There are 256 neurons in each NN layer. The parameter-tuning experiment produces the number of hidden NN layers and the number of neurons. The rectified linear unit is the activation function of each layer. Two neurons are stacked on the last hidden NN layer to form an output layer. A sigmoid function is stacked one the output layer to transform the estimated values to 0–1 and output the estimated width W_p and the estimated length L_p , Fig. 3c.

3.2.4 Calculating MSSE Loss and Optimizing SSENet

The MSSE loss function is used in the DNN regression model. For most regression issues, the mean square error (MSE) is a commonly used loss function. The definition of MSE is shown in Equation (1): y_i represents the ground truth, y'_i represents the prediction value, and N means the number of values to be predicted. The loss value calculated by MSE and the ground truth value have no relation. Assume a ship's

ground length and width are 100 and 50 m, respectively, and the predicted length and width are 80 and 30 m, respectively. Both the length and width MSE values are 400. Because the model is optimized based on loss values, both the length and width losses contribute equally to the model's optimization. In practice, a ship's length is much greater than its width. In most cases, the length is more concerning than the width. In order to increase the length estimate accuracy, we hope that the length loss helps to optimize the model more than the width loss.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - y'_i)^2 \quad (1)$$

$$MSSE = \frac{1}{N} \sum_{i=1}^N y_i \cdot (y_i - y'_i)^2 \quad (2)$$

$$Size_{Loss} = MSSE_L + MSSE_W \quad (3)$$

MSSE loss function solves the mentioned issue. MSSE incorporates the ship length and width ground truth into the traditional MSE. The ground truth is utilized as a dynamic parameter to scale the square error. The definition of MSSE is shown in Eq. (2): y_i , y'_i and N is the number of all samples. The MSSE length and width losses in the example are 40,000 and 20,000, respectively. The loss in length is substantially greater than the loss in width. As a result, the penalty for the model's length will be increased during the training phase. Therefore, the optimization procedure is more conducive to length estimation. Based on Eq. (2), the loss of length $MSSE_L$ and the loss of width $MSSE_W$ are calculated. The size loss ($Size_{Loss}$) is the summation of $MSSE_L$ and $MSSE_W$, Eq. (3).

Besides $Size_{Loss}$, the confidence loss ($Conf_{Loss}$) and the location loss ($Loca_{Loss}$) are another two losses calculated in the first stage, Fig. 3b. $Conf_{Loss}$ is the cross-entropy loss, and $Loca_{Loss}$ is the smooth L1 loss [1, 23]. Their definitions are as follow:

$$Conf_{Loss} = \sum_{i=1}^N c_i \log c'_i + (1 - c_i) \log (1 - c'_i) \quad (4)$$

$$Loca_{Loss} = \frac{1}{N} \sum_{i=1}^N \text{smooth}_{L1}(x_i) = \begin{cases} 0.5x_i^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \quad (5)$$

where N is the number of predicted targets, c_i is the ground confidence of a sample, c'_i is the predicted confidence of a sample, and x_i is the element-wise difference between the ground RBB and the predicted RBB. The three losses, $Size_{Loss}$, $Conf_{Loss}$, and $Loca_{Loss}$, are added to form the final loss that optimizes SSENNet integrally.

3.3 Experiments on SSENet

3.3.1 Experiments Data

The OpenSARShip dataset (<http://opensar.sjtu.edu.cn/>) is a Sentinel-1 ship interpretation dataset that includes 11,346 SAR ship chips and automatic identification system (AIS) messages. The ground size for each ship is provided via the AIS. The ground range detected (GRD) of IW is the picture mode of Sentinel-1. The spatial resolution of the SAR image is around 20m, with a pixel spacing of 10m. SNAP 3.0 performs radiometric calibration and terrain correction. The amplitude values of pixels for VH (vertical emitting and horizontal receiving) and VV (vertical emitting and vertical receiving) polarizations are stored on each SAR chip, which has one ship and two channels. The experiment set for SSENet includes 1,890 samples in the VV mode. Figure 6 shows the distributions of ground ship's length and width. The length ranges from 28 to 399 m. The width ranges from 6 m to 65 m. Each SAR chip is 300×300 pixels in size. We transform the values of SAR images to $[0, 255]$. The training set consisted of 1,500 SAR chips chosen at random. The remaining 390 chips will be used for testing.

The ground truths for the experimental set include two parts: the ground ship size and the RBB for each ship. The ground size is obtained from the OpenSARShip. The RBB for each ship is labeled manually by a Matlab tool shared in DRBox-v2. The DRBox-v2 is trained to generate accurate RBB based on the ground RBB.

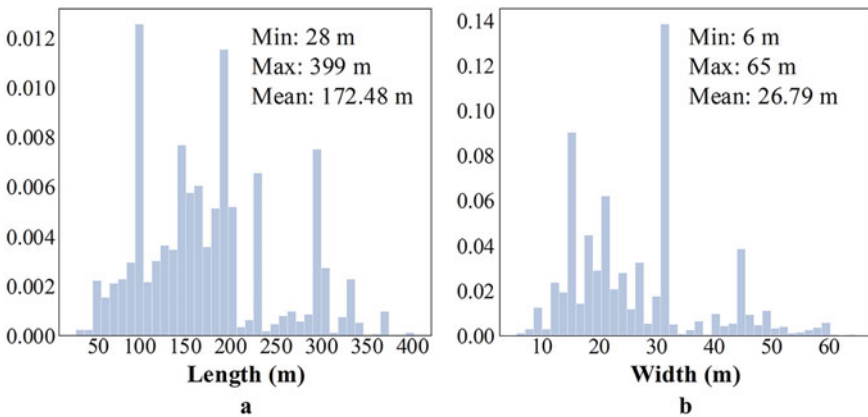


Fig. 6 The range of length and width of the testing set

3.3.2 Experiments Setting

A workstation with one GeForce RTX 2070 8GB GPU is used in the experiment. Python 3.6 is the programming language used. TensorFlow is a deep learning package. For training, the batch size is six. 0.0002 is the initial learning rate. The learning rate reduces by half every 5,000 training epochs during the training procedure. When the $Size_{Loss} < 0.001$, the $Loca_{Loss} < 0.005$, and the composite loss < 0.01 , the training procedure stops.

MAE and the mean absolute percentage error (MAPE) are employed as metrics. MAE is a typical absolute error, and MAPE is a widely used relative error. Assuming y_i is the ground truth, y'_i is the estimation value, and N is the number of samples, the definitions of MAE and MAPE are as follow:

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - y'_i| \quad (6)$$

$$MAPE (\%) = \frac{100}{N} \sum_{i=1}^N \left| \frac{y_i - y'_i}{y_i} \right| \quad (7)$$

3.3.3 Performance of SSENet

The hyper-parameters of SSENet are determined by parameter tuning, and a well-trained model is picked up to be evaluated. The 390 samples of the testing set are fed into the well-trained SSENet. The outputs are the scaled lengths and widths estimated by the model. The scaled values are rescaled to normal values.

The estimated ship sizes are shown in Fig. 7a, b. The length and width MAEs are 7.88 and 2.23 m, respectively. The MAEs of the estimated length and width are pushed under 0.8-pixel spacing. The MAPE of estimated length and width are 5.53 and 8.93%, respectively. The R^2 score are 0.9773 and 0.9093. This indicates that the estimated ship length/width is quite close to the ground length/width. The R^2 score of widths is smaller than that of length, which means the width is difficult to estimate than the length. There are two factors that contribute to this phenomena. A ship's width is far smaller than its length. The width of the ship's signature on the SAR image is more ambiguous than the length [26], which causes random errors in the width of the labeled RBB. Second, the MSSE loss function makes the model fit the length better.

We plot the relationship between the labeled RBB's size and the ship's ground size, as shown in Fig. 7c, d. The labeled RBB is treated as the RBB closest to the ship's signature for visual interpretation. As shown in Fig. 7c, the MAE of length is nearly 40 m, and the MAE of width is more than 50 m. The gap between the labeled RBB's size and the ground size is large. By adding the regression model, SSENet pushes the MAEs under 8 m. Therefore, the proposed regression model based on

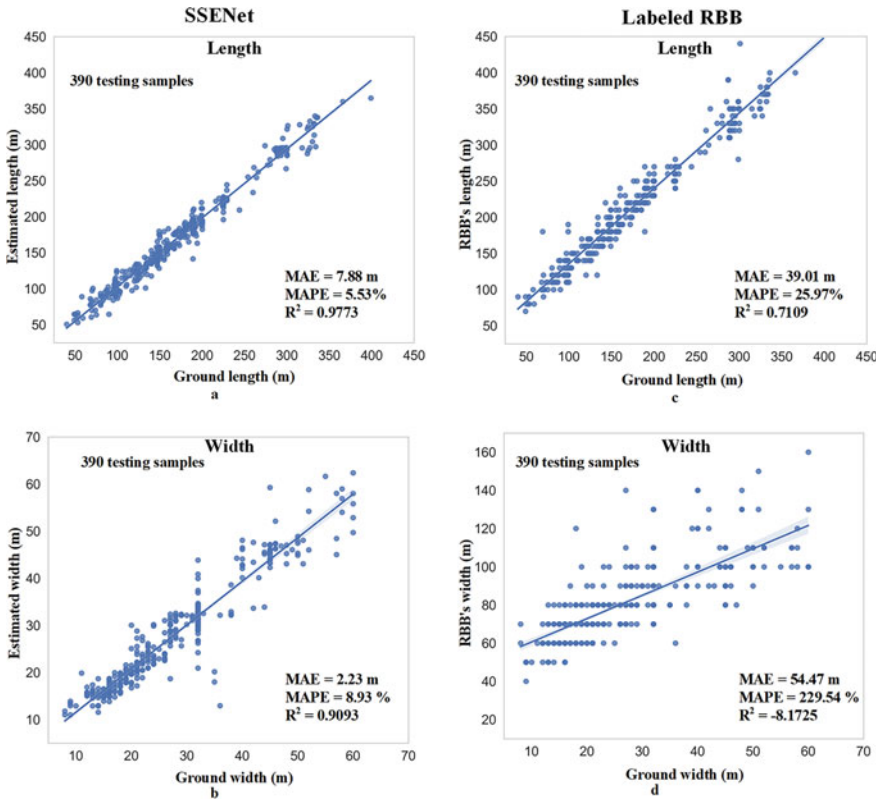


Fig. 7 Relationships between the SSENet’s estimated size, and the size of the labeled RBB. **a** and **b** The relationships between the ground size and the SSENet’s size. **c** and **d** The relationships between the ground size and the labeled RBB’s size

DNN is necessary and effective. Figure 8 shows some examples of SSENet’s results. The outputs of one sample include the detected RBB, the confidence score to be a ship, and the estimated ship size. For most ship samples, the estimated sizes are consistent with the ground sizes.

3.3.4 Effectiveness of the Inputs

The efficiency of the inputs for the DNN regression model is tested. The results are shown in Table 1. Three compared models employ different inputs. The inputs for SSENet₁ include initial ship size, without feature map F_6 . For SSENet₂, the inputs are initial ship size and F_6 . Based on the three inputs, SSENet₃ adds the initial orientation as another input.

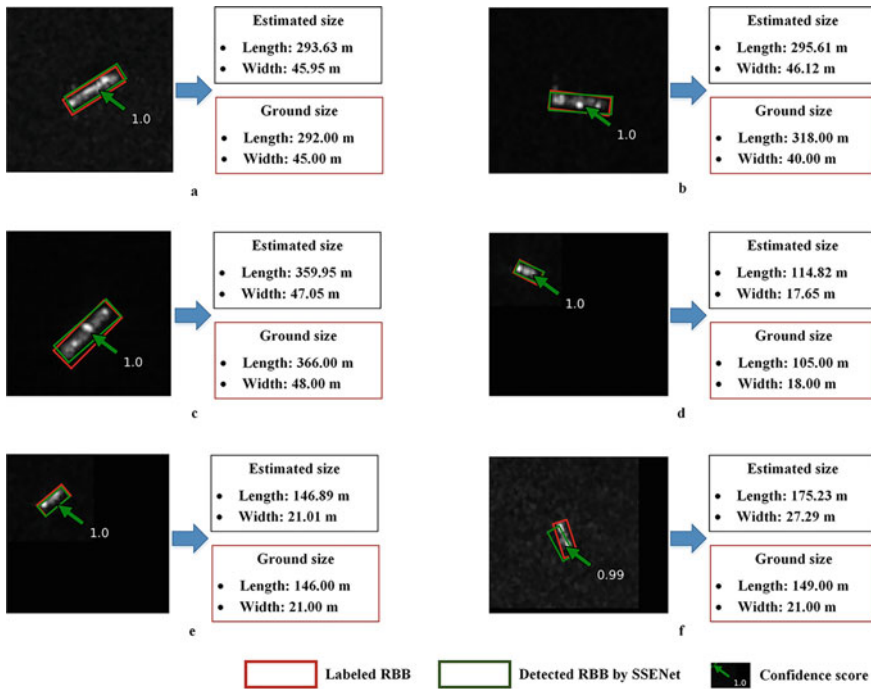


Fig. 8 Some examples of SSENNet, the outputs include the detected RBB, the confidence score to be a ship, and the estimated ship size

Table 1 Model performance with different inputs

Models	Inputs of DNN	MAE (m)		MAPE (%)	
		Length	Width	Length	Width
SSENNet ₁	X_l, X_w	10.01	2.65	6.87	9.89
SSENNet ₂	X_l, X_w, F_6	8.14	2.27	5.82	9.22
SSENNet ₃	$X_l, X_w, \cos \theta, F_6$	7.88	2.23	5.53	8.93

The results are displayed in Table 1. SSENNet₁ obtains the largest MAE and MAPE among the three models. By adding F_6 , SSENNet₂ reduces the length’s MAE about 2 m compared with SSENNet₁. This finding illustrates that the feature map of a SAR image is an important input for estimating ship size. Adding the feature map as an input improves the accuracy of size estimation. Finally, by explicitly including the ship’s initial orientation as another input, the estimation errors are significantly minimized. Therefore, each element of the inputs for SSENNet shows contributions to the final size estimation. Figure 8 shows several results of SSENNet₃, and the red/green rectangle is the labeled/detected RBB. The estimated confidence score to be a ship and estimated the size by SSENNet are also displayed.

Table 2 Performance of with MSE or MSSE

Models	Loss Function	MAE (m)		MAPE (%)	
		Length	Width	Length	Width
SSENet _{MSE}	MSE	8.85	2.20	5.99	8.81
SSENet _{MSSE}	MSSE	7.88	2.23	5.53	8.93

3.3.5 Effectiveness of MSSE Loss

An experiment is conducted to test the effectiveness of the new loss function, MSSE. The results are shown in Table 2. SSENet_{MSE} is the model with MSE loss. SSENet_{MSSE} is the model with MSSE loss. The other parts of the two models are the same.

The results are shown in Table 2. The length MAE of SSENet_{MSSE} is nearly 1m less than that of SSENet_{MSE}, reducing by 11%. For the width, SSENet_{MSSE} performs slightly worse than SSENet_{MSE}. The reason for this is that MSSE emphasizes a significant loss and drives the model to focus on length rather than width. The difference in width between the two values, however, is only a few centimeters. The disadvantages of MSSE are not overshadowed by the aforementioned constraint. As a result, our MSSE loss is helpful, particularly when evaluating the ship’s length.

4 Discussions

4.1 ML versus DL

SSENet’s regression model is a DNN model. We choose three typical ML models, Gradient Boosting Regression (GBR) [6], Support Vector Regression (SVR) [3], and Linear Regression (LR) [18] to discuss their performances. GBR and SVR are applied in ship size extraction [8, 11]. LR is a baseline model [27]. Because these three ML models aren’t NN-based, they can’t be combined with the SSD to create an end-to-end model. The SAR images cannot be fed into the three ML models. The inputs for these three models are the initial ship size and orientation of the labeled RBB. The parameters of GBR, SVR, LR are tuned and the estimation results with the best metrics are recorded. The DNN model is used by SSENet.

The results are shown in Table 3. The result of SSENet is in the last row. GBR performs the best among four models (LR, SVR, GBR, and DNN). GBR is an ensemble learning model with good performance in the three-stage procedure [26]. However, GBR is unable to extract features from SAR images automatically. GBR cannot be combined with a DL-based ship detection model, such as DRBox-v2, to create an integrated ship size extraction model. The premise of using GBR is that the SAR image should be binarized accurately, and the initial RBB is well extracted by tra-

Table 3 MAE and MAPE of different models

Models	Inputs	MAE (m)		MAPE (%)	
		Length	Width	Length	Width
LR	RBB	10.60	2.86	6.97	10.47
SVR	RBB	10.16	2.48	6.61	9.50
GBR	RBB	9.69	2.39	6.93	9.27
DNN	RBB	10.79	2.83	7.27	10.22
SSENet	Images	7.88	2.23	5.53	8.93

ditional methods. As stated in Sect. 2, the traditional method faces big challenges. Practically, GBR is not an end-to-end model: feeding the SAR image and obtaining the ship size.

The error of DNN model is large. However, a DNN model can be combined with any deep learning models based on CNN or NN to extract size from the SAR image from beginning to end. In contrast to traditional techniques, the DL model optimizes all parameters globally. The DNN regression model can use the feature maps extracted by the DL model to increase the accuracy of the estimated ship size. As shown in Table 3, the SSENet reduces the MAE of length by nearly 2 m compared with the GBR, about 18.68 %. Therefore, compared with traditional methods, the ship size extraction model based on deep learning is more practical.

4.2 Errors's Sources

This section delves into the details of estimation errors and attempts to determine what causes large inaccuracies. The ship's direction and transit speed are two elements that need to be investigated, according to previous research [10, 26].

4.2.1 Ship Orientation

The estimated errors with respect to the ship's orientation angle is displayed. Figure 9a and b show the results of the length. Fig. 9c, d show the results of the width. The MAEs vary with the ship orientation variation. Large MAEs occur when the orientation angles are closer to 0° (0° means the azimuth direction) in the range of (−45°, 45°). The reasons for the above phenomenon include two aspects. The first one is the ship motion. When the ship moves in a direction that is near to the azimuth direction, the azimuth direction's speed component is large. Because of the large component, the ship signature appears to be smeared, increasing the estimation error. The other reason is the unequal resolution during imaging, 5 m × 20 m for range and azimuth directions, respectively. The low resolution in the azimuth enlarges the errors [26].

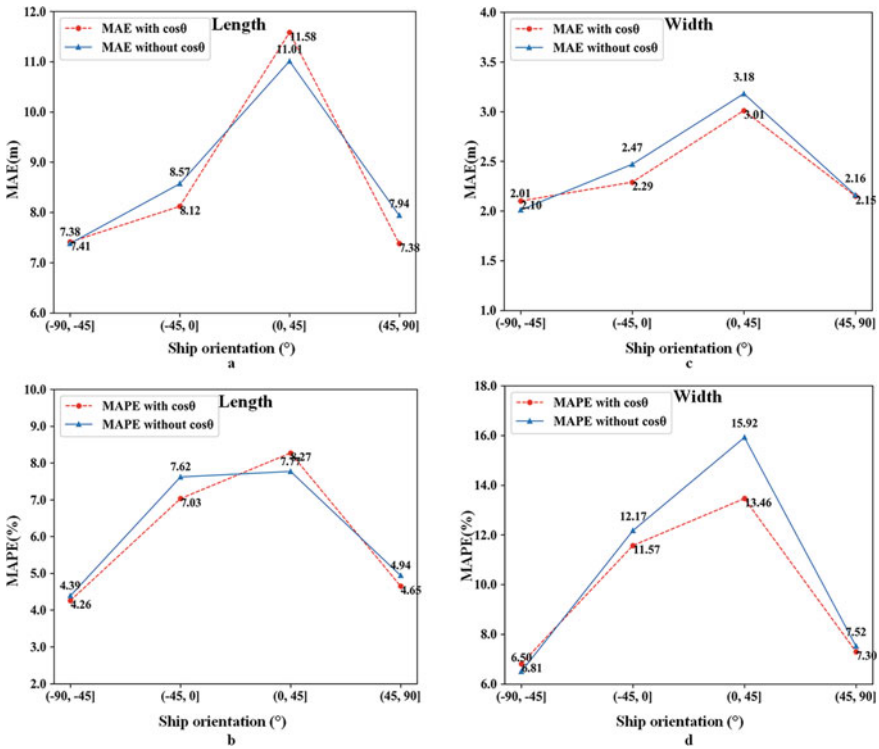


Fig. 9 The trend of errors with respect to the orientation angle. **a** Trend of length's MAE; **b** Trend of length's MAPE; **c** Trend of width's MAE; **d** Trend of width's MAE

As shown in Fig. 9, when the initial orientation angle ($\cos\theta$) is added to the DNN model, the errors are reduced. This finding also proves that using the original orientation angle as an input is valid.

4.2.2 Ship Speed

Figure 10 shows the errors corresponding to the ship's speed. Because the OpenSARShip's SAR images are mostly from ports, around 83% of the ships are still there. Figure 10a shows that the MAEs are small in the range of (0, 1) knot (1.852 km/h). With the increase of ship speed, the MAE fluctuates slightly. When the speed is greater than 15 kn (27.780 km/h), the MAEs increase apparently: 19.04 and 4.71 m. These two values are far greater than those of other speed intervals. The ship's speed cannot be derived from the SAR image signature. Therefore, it is difficult to refine the estimated sizes of ships by pre-input the ship's speed parameter.

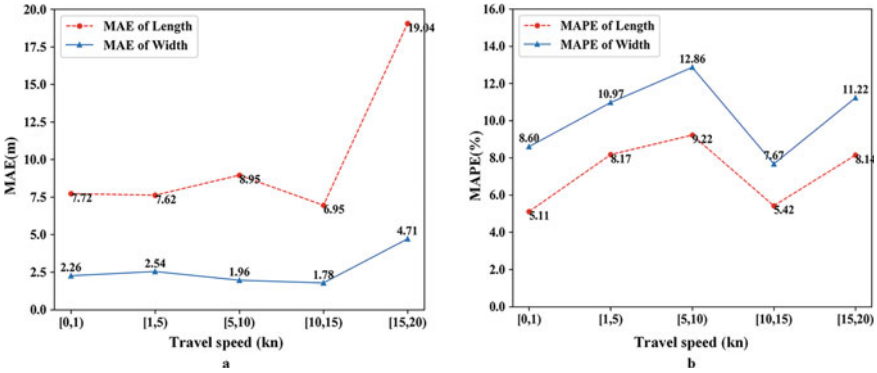


Fig. 10 Figure 10. The trend of errors with respect to the travel speed. **a** The trend of MAE; **b** Trend of MAPE

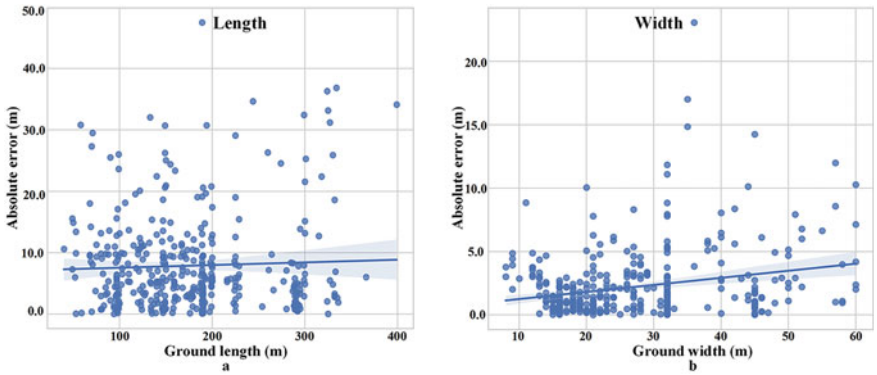


Fig. 11 The distribution of AE with respect to the ground size. **a** The AE of length; **b** The AE of width

4.2.3 Ship Size

Figure 11 shows the absolute error (AE) of each estimated and the ground size. The AE of a estimated size takes the absolute value of the difference between the predicted value and the true value. As shown in Figs 11a and b, there are no obvious relationships between AE and ground size. Therefore, the ship size is not a source of errors.

5 Conclusions

SSENet, a DL-based model for extracting ship size from SAR data, is proposed in this chapter. A DNN-based regression model and an SSD-based model make up the SSENet. The DNN model is fed the initial ship size and orientation angle derived from the RBB, as well as the high-level features extracted from the input SAR image. The OpenSARShip trains and validates SSENet. Experiments show that: (1) the SSENet can straight extract ship size from SAR images with MAE less than 0.8 pixels; (2) the new MSSE loss reduces the length's MAE nearly 1 m than the old MSE loss; (3) SSENet shows obvious advantage over the GBR model; (4) SSNet exhibits robustness over four separate data sets.

Acknowledgements The OpenSARShip is downloaded from “OpenSAR Platform” (<http://opensar.sjtu.edu.cn>).

References

1. An Q, Pan Z, Liu L, You H (2019) DRBox-v2: an improved detector with rotatable boxes for target detection in SAR images. *IEEE Trans Geosci Remote Sens* 57(99):8333–8349
2. Chang YL, Anagaw A, Chang L, Wang YC, Hsiao CY, Lee WH (2019) Ship detection based on YOLOv2 for SAR imagery. *Remote Sens* 11(7):786
3. Drucker H, Burges C, Kaufman L, Chris JC, Kaufman BL, Smola A, Vapnik V (1997) Support vector regression machines. *Adv Neural Inf Process Syst* 28(7):779–784
4. Eldhuset K (1996) An automatic ship and ship wake detection system for spaceborne SAR images in coastal regions. *IEEE Trans Geosci Remote Sens* 34(4):1010–1019. <https://doi.org/10.1109/36.508418>
5. Fingas M, Brown C (2001) Review of ship detection from airborne platforms. *Canadian J Remote Sens* 27(4):379–385. <https://doi.org/10.1080/07038992.2001.10854880>
6. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. *Ann Stat* 29(5):1189–1232
7. Huang L, Liu B, Li B, Guo W, Yu W, Zhang Z, Yu W (2018) OpenSARShip: a dataset dedicated to Sentinel-1 ship interpretation. *IEEE J Sel Top Appl Earth Obser Remote Sens* 11(1):195–208. <https://doi.org/10.1109/JSTARS.2017.2755672>
8. Lang H, Wu S (2017) Ship classification in moderate-resolution SAR image by naive geometric features-combined multiple kernel learning. *IEEE Geosci Remote Sens Lett* 14(10):1765–1769. <https://doi.org/10.1109/LGRS.2017.2734889>
9. Lang H, Xi Y, Zhang X (2019) Ship detection in high-resolution SAR images by clustering spatially enhanced pixel descriptor. *IEEE Trans Geosci Remote Sens* 57(8):5407–5423. <https://doi.org/10.1109/TGRS.2019.2899337>
10. Lecun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444. <https://doi.org/10.1038/nature14539>
11. Li B, Liu B, Guo W, Zhang Z, Yu W (2018) Ship size extraction for Sentinel-1 images based on dual-polarization fusion and nonlinear regression: Push error under one pixel. *IEEE Trans Geosci Remote Sens* 56(8):4887–4905. <https://doi.org/10.1109/TGRS.2018.2841882>
12. Li X, Liu B, Zheng G, Ren Y, Zhang S, Liu Y, Gao L, Liu Y, Zhang B, Wang F (2020) Deep learning-based information mining from ocean remote sensing imagery. *Nat Sci Rev*
13. Liu B, Li X, Zheng G (2019) Coastal inundation mapping from bitemporal and dualclarization SAR imagery based on deep convolutional neural networks. *J Geophys Res: Oceans* 124(12)

14. Liu P, Jin YQ (2017) A study of ship rotation effects on SAR image. *IEEE Trans Geosci Remote Sens* 1–13
15. Liu W, Angelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) SSD: single shot multibox detector. Springer, Cham
16. Margarit G, Mallorquí J, Fàbregas X (2006) Study of the influence of vessel motions and sea-ship interaction on classification algorithms based on single-pass polarimetric SAR interferometry. In: *IEEE international conference on geoscience and remote sensing symposium*
17. Margarit G, Mallorquí JJ, Fortuny-Guasch J, Lopez-Martinez C (2009) Exploitation of ship scattering in polarimetric SAR for an improved classification under high clutter conditions. *IEEE Trans Geosci Remote Sens* 47(4):1224–1235
18. Montgomery DC, Peck EA, Vining GG (2021) *Introduction to linear regression analysis*. Wiley
19. Ouchi K, Tamaki S, Yaguchi H, Iehara M (2004) Ship detection based on coherence images derived from cross correlation of multilook SAR images. *IEEE Geosci Remote Sens Lett* 1(3):184–187. <https://doi.org/10.1109/LGRS.2004.827462>
20. Pelich R, Longépé N, Mercier G, Hajduch G, Garello R (2015) Performance evaluation of Sentinel-1 data in SAR ship detection. In: *IEEE international geoscience and remote sensing symposium (IGARSS)*, pp 2103–2106 (2015). <https://doi.org/10.1109/IGARSS.2015.7326217>
21. Qin X, Zhou S, Zou H, Gao G (2013) A CFAR detection algorithm for generalized gamma distributed background in high-resolution SAR images. *IEEE Geosci Remote Sens Lett* 10(4):806–810. <https://doi.org/10.1109/LGRS.2012.2224317>
22. Reichstein M, Camps-Valls G, Stevens B, Jung M, Denzler J, Carvalhais N, Prabhat (2019) Deep learning and process understanding for data-driven Earth system science. *Nature* 566(7743):195
23. Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell* 39(6):1137–1149
24. Ren Y, Cheng T, Zhang Y (2019) Deep spatio-temporal residual neural networks for road-network-based data modeling. *Int J Geogr Inf Sci* 1–19
25. Schmidhuber J (2015) Deep learning in neural networks: an overview. *Neural Netw* 61:85–117 <https://doi.org/10.1016/j.neunet.2014.09.003>, <http://dx.doi.org/10.1016/j.neunet.2014.09.003>
26. Stasolla M, Greidanus H (2016) The exploitation of Sentinel-1 images for vessel size estimation. *Remote Sens Lett* 7(12):1219–1228. <https://doi.org/10.1080/2150704X.2016.1226522>, <https://doi.org/10.1080/2150704X.2016.1226522>, <https://doi.org/10.1080/2150704X.2016.1226522>
27. Tings B, Da S, Bentes CA, Lehner S (2016) Dynamically adapted ship parameter estimation using TerraSAR-X images. *Int J Remote Sens*
28. Torres R, Snoeijs P, Geudtner D, Bibby D, Davidson M, Attema E, Potin P, Rommen B, Floury N, Brown M, Traver IN, Deghaye P, Duesmann B, Rosich B, Miranda N, Bruno C, L'Abbate M, Croci R, Pietropaolo A, Huchler M, Rostan F (2012) GMES Sentinel-1 mission. *Remote Sens Environ* 120:9–24. <https://doi.org/10.1016/j.rse.2011.05.028>, <https://www.sciencedirect.com/science/article/pii/S0034425712000600>, the Sentinel Missions—New Opportunities for Science
29. Wackerman C, Friedman K, Pichel W, Clemente-Colón P, Li X (2001) Automatic detection of ships in RADARSAT-1 SAR imagery. *Canadian J Remote Sens* 27(5):568–577. <https://doi.org/10.1080/07038992.2001.10854896>, <https://doi.org/10.1080/07038992.2001.10854896>, <https://doi.org/10.1080/07038992.2001.10854896>
30. Wang C, Liao M, Li X (2008) Ship detection in SAR image based on the Alpha-stable distribution. *Sensors* 8(8):4948–4960. <https://doi.org/10.3390/s8084948>, <https://www.mdpi.com/1424-8220/8/8/4948>
31. Zhang X, Li X (2020) Combination of satellite observations and machine learning method for internal wave forecast in the Sulu and Celebes Seas. *IEEE Trans Geosci Remote Sens* (99):1–11
32. Zheng G, Li X, Zhang RH, Liu B (2020) Purely satellite data-driven deep learning forecast of complicated tropical instability waves. *Sci Adv* 6(29):eaba1482

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits any noncommercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if you modified the licensed material. You do not have permission under this license to share adapted material derived from this chapter or parts of it.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Benthic Organism Detection, Quantification and Seamount Biology Detection Based on Deep Learning



Yuhai Liu, Yu Xu, Haining Wang, and Xiaofeng Li

1 Overview

1.1 Backgrounds

Deep-sea organisms are those living below the ocean belt, and they can be divided into three categories according to their living styles, including plankton, swimming organisms and benthos. Deep-sea biological resources are an essential part of the marine ecosystem and play a vital role in the formation, maintenance, and development of marine ecosystem. The deep-sea biological resources are the foundation of marine ranch construction and aquatic development [34]. The problems such as the

Y. Liu

Dawning International Information Industry Co., Ltd., Qingdao 266101, China

Sugon Nanjing Institute, Co., Ltd., Nanjing 211100, China

Y. Xu

Laboratory of Marine Organism Taxonomy and Phylogeny, Shandong Province Key Laboratory of Experimental Marine Biology, Center for Ocean Mega-Science, Institute of Oceanology, Chinese Academy of Sciences, Qingdao 266071, China

Y. Xu · H. Wang

University of Chinese Academy of Sciences, Beijing 100049, China

H. Wang

Key Laboratory of Marine Ecology and Environmental Sciences, Institute of Oceanology, Chinese Academy of Sciences, Qingdao 266071, China

Deep Sea Research Center, Institute of Oceanology, Chinese Academy of Sciences, Qingdao 266071, China

X. Li (✉)

CAS Key Laboratory of Ocean Circulation and Waves, Institute of Oceanology, Chinese Academy of Sciences, Qingdao 266071, China

e-mail: lixf@qdio.ac.cn

© The Author(s) 2023

X. Li and F. Wang (eds.), *Artificial Intelligence Oceanography*,
https://doi.org/10.1007/978-981-19-6375-9_16

establishment of deep-sea protection areas, the sustainable utilization of resources, and the maintenance of vulnerable marine ecosystems based on species diversity have become the hot spots in global deep-sea research. The research on the distribution and diversity of deep-sea organisms is helpful to promote human cognition of the ecosystem and plays a vital role in the maintenance of the marine ecosystem. Due to the year-round darkness of the deep-sea area where the sunlight is difficult to penetrate, the high salinity, the considerable pressure, the low water temperature, the number of biological species is relatively small. In contrast, the biological quantity is numerous in some intensive biological areas. Therefore, it is crucial to solving the practical problems by using modern technology.

Species discovery and identification are crucial ways to explore deep-sea biodiversity. To better protect marine ecology, we can monitor the health status and biodiversity of the benthos ecosystem by analyzing the species, quantity, and growth. Traditional methods of marine biological identification are based on morphology and molecular genetics and sometimes even need to use the advanced DNA sequencing technology supported by electron microscope. Although this method is accurate, there are still two main problems for marine species classification. On the one hand, it costs a large amount of human and financial resources to cultivate professional taxonomy experts for marine species, and artificial identification has low efficiency. On the other hand, the special ocean environment is unsuitable for in-situ detection during scientific research using molecular and electron microscopy methods, and heterotopic detection can lead to biological inactivation and species death. To solve the problems above, the application of the target detection technology based on the deep neural network in marine species identification and quantitative analysis emerged.

Considering the problems, including the difficulties in underwater target recognition caused by complex marine imaging environment, brutal penetration of sunlight, high salinity, the high similarity of some detected targets, and uneven distribution of biological density [26], the static counting of dense marine biological communities, and automatic real-time dynamic detection and counting algorithm of marine benthos were explored and studied in this paper. It is significant in helping marine biologists identify marine species, evaluate the population density, improve the operational performance of underwater autonomous robots and promote the underwater operation and the development and ecological protection of marine resources.

Seamounts are relatively isolated conical peaks or groups of peaks in the various oceans and are also an essential part of the marine environmental system. Seamounts rise from the seafloor but do not protrude from sea level. There are an estimated 30,000 seamounts worldwide, but only a few have been studied. However, seamounts have become one of the most popular systems in deep-sea research in recent years because of their unique topographic and hydrological features and their unique ecosystems, rich biodiversity, and excellent resource value.

Recently, our country has successfully carried out a series of seamount explorations represented by 'Jiaolong' manned HOV and 'FaXian' ROV, and obtained many first-hand submarine image data and samples in the South China Sea, Western and Central Pacific. It not only significantly improves the level of deep-sea detection

of our country but also provides data support for automatic detection of benthos in seamounts. Section 4 will use deep learning technology to identify and detect the giant benthos in seamounts.

1.2 Related Works

Due to the influence of the medium, the propagation distance of light and radio waves is very limited in seawater. In contrast, the propagation performance of sound waves in water is much better, covering a wider sea area. However, the acoustic signal will propagate along different paths because of the reflection, refraction, and other phenomena, so the underwater target recognition based on sonar echo technology has many interferences and low accuracy. The target recognition method of sonar image has the characteristics of high resolution and real-time performance. Therefore, in underwater target recognition and detection, the current research mainly focuses on the underwater target detection of sonar images for a long time.

Traditional sonar image detection algorithm mainly extracts features from sonar images and then classifies and locates the target. Extracting enough information features is the key to detect the underwater target. In this way, researchers proposed a series of hand-designed feature extraction methods, such as Scale Invariant Feature Transform (SIFT) [24, 25], Histogram of Oriented Gradient (HOG) [5]. The features are extracted effectively, and then recognized by algorithms like Morphology, Fuzzy Clustering and Markov Random Field [29]. The manual feature extraction, classification, and detection methods have a good recognition effect in specific application scenarios. Still, these algorithms have poor scalability and low generalization ability, which different features need to be designed for different problems. Therefore, the application value of the algorithm is limited.

With the development of underwater high-definition imaging technology such as ROVs (Remote Operated Vehicles) and AUVs (Autonomous Underwater Vehicles), the data of close-range targets collected by optical imaging equipment can be analyzed by computer vision algorithm without sonar images. It makes the feature information of the target more fully retained and used, and the accuracy and efficiency of target detection are greatly improved. In this trend, Fish4Knowledge [7] project has collected 115 TB of underwater high-definition image/video data and proposed many methods to detect fish in the underwater video for assessing fish biodiversity [39]. In fish detection, SIFT [1, 28, 37] or SC (Shape Context) [35] algorithms have been widely used to calculate marker features. But in the reference [47], the author concludes that HOG algorithm is better than SIFT and SC algorithm. Marcos et al. [27] used Normalized Chromaticity Coordinates (NCC) histogram to extract color features, and Local Binary Pattern (LBP) feature descriptor to extract texture features of the coral image. Stokes and Deane [40] proposed coral classification Discrete Cosine Transform and K-Nearest Neighbor classifier algorithm. Although the upgrades of underwater acquisition devices improve the quality of data, the analysis algorithm continues to use SIFT and HOG to extract features and then use Support

Vector Machine (SVM) [4] and Adaptive Boosting (AdaBoost) [42] to classify. The problem of poor robustness of manual feature extraction has not been effectively solved.

Due to the continuous development of deep learning in recent years, all aspects of computer vision are have been promoted. Especially, Convolutional Neural Network (CNN) algorithm, Fast R-CNN algorithm, and Feature Pyramid Network (FPN) algorithm, which are widely used in image classification, image annotation, and multi-target detection, enable people to obtain rich deep semantic information of images and improve the accuracy of image classification and recognition significantly. The target detection and recognition method based on deep learning benefits from CNN's strong feature autonomous learning ability on large-scale data sets and can effectively solve feature extraction in the above method. Therefore, it has been successfully applied to many underwater target detection and recognition scenes [18]. Kratzert and Mader [16] used the marine fish channel monitoring platform based on CNN algorithm to detect targets without using any artificial features, and the final fish classification accuracy reached 93%. Huang et al. [15] applied Faster R-CNN to detect and identify marine organisms, expanded a small number of samples through three data enhancement methods and verified the effectiveness of Faster R-CNN in biological detection in different marine turbulent environments. Xia et al. [43] proposed a sea cucumber detection scheme based on YOLOv2 model, which has a good detection effect on sea cucumbers with a regular shape or simple natural scene coverage. Although these methods have achieved some success, they are applied to specific target scenarios and do not include the study of target quantity statistics.

1.3 Research Content and Innovation

With the change of economic and marine environment, the value of marine biological resources is enormous. To further improve the ability of marine resources utilization and marine ecological protection, advanced information technology, and data analysis ability are needed to provide accurate data support and decision support for relevant personnel. After in-depth research on big data technology, artificial intelligence, and other technologies, combined with the existing business needs, the deep-sea biological identification quantitative model was designed and realized in this chapter.

The key to the success of the deep-sea biometric quantitative system is the extraction and application of data. The fast extraction of data and big data requires a stable and reliable algorithm basis. Data acquisition, extraction, conversion, cleaning, and data loading are used to enter the data storage layer. A deep-sea biometric quantitative database is formed by deep learning technologies such as data analysis. The computing layer can provide robust image classification and recognition, realize deep-seated analysis of deep-sea biological data, fully excavate the hidden value of data, and provide support for quantitative recognition of deep-sea organisms. In this

chapter, Faster R-CNN and SSD are adopted respectively to achieve marine biometric identification and quantification for different scenarios.

Considering the current situation of deep-sea biological resources, and in order to realize the requirements of automatic identification and quantitative analysis of deep-sea organisms and detection of giant benthic organisms in seamounts, the following functions are studied and realized mainly in this chapter:

1. The deep-sea biological recognition and quantitative analysis system is constructed to process a large number of deep-sea biological image data, analyze the deep-sea biological data, extract biological features, classify and quantitatively analyze the deep-sea biological recognition by using deep learning and other artificial intelligence technology.
2. According to the high-definition seamount image data taken by the research ship during the investigation of a seamount in the Western Pacific Ocean, the seamount biological training library is constructed. On this basis, the SSD target detection model is trained. The feasibility of automatic real-time seamount species detection and counting was studied under the condition of the trade-off between speed and accuracy. 63 high-quality images of seamount macrobenthos in the Western Pacific are constructed and manually labeled. They can be used to train various deep learning models, which alleviates the lack of training data for marine species to a certain extent, and is helpful for other people in the same field.

2 The Target Detection Techniques

2.1 Introduction on Target Detection

In computer vision and image processing, Target Detection is an image segmentation technology that scans and searches for specific semantic targets (such as people, buildings or cars) in digital images or videos and marks them. Generally speaking, it is not only to identify which category the target belongs to, but also to get its specific position in the picture. Target Detection is widely used in computer vision tasks, such as automatic image annotation, behavior recognition, face recognition and video target segmentation. It can also be used for target trackings, such as the ball in a football match or the players on the court.

Traditional target detection is usually based on the traditional machine learning method, which is generally divided into two stages: firstly, SIFT, HOG, and other methods are used to extract features, and then, SVM, AdaBoost, and other algorithms are used for classification. However, there are two main problems in traditional target detection methods: (a) feature extraction is not targeted, and time complexity is high; (b) the features designed manually are not robust to the change of diversity. Therefore, when the detection task changes, the features need to be redesigned.

In recent years, with the help of Deep Neural Networks (DNN), the target detection algorithm based on DNN has gradually replaced the traditional target detection

algorithm. In computer vision tasks, DNN based target detection and recognition algorithms are mainly divided into two categories: one is a region proposal-based target detection algorithm, that is, a two-stage detection algorithm. In the first step, a series of sparse candidate regions are generated by a certain method, and in the second step, the candidate regions are further classified and regressed. Typical representatives of such algorithms are R-CNN [9], Fast R-CNN [8], Faster R-CNN [33], Mask R-CNN [14], etc. Due to the low recognition error rate and missing recognition rate, the two-stage target detection algorithm has achieved excellent performance on several challenging benchmarks including Pascal VOC [6] and MS COCO [21]; The other is the single-stage target detection algorithm, which skips the stage of generating candidate regions and directly generates the class probability and position coordinate value of targets. The final detection result can be obtained through a single detection. Therefore, compared with the two-stage algorithm, it has a faster detection speed. There are many typical algorithms, such as YOLOv1 [32], YOLOv2 [30], YOLOv3 [31], YOLOv4 [2], SSD(Single Shot Multibox Detector) [23], RetinaNet [22], RefineDet [46], CornerNet [17], etc. The advantage of a single-stage target detection algorithm is high detection efficiency, but its detection proficiency often lags behind a two-stage algorithm.

2.2 *The Single-Stage Target Detection*

The core idea of the single-stage target detection algorithm is to take the whole image as the input of the network, and apply regression on the position and category of Bbox in the output layer directly. The primary representative is SSD [23] and YOLO (You Only Look Once). In this paper, we use the SSD to complete the detection of seamount macrobenthos. As a result, the model has a simple structure and fast speed. The following focuses on the SSD algorithm and illustrates the principle of single-stage target detection.

SSD (single shot multibox detector) [23] is the first single-stage detector of a single shot. It abandons the practice of Faster R-CNN using RPN to generate boundary boxes and classify them and puts forward the ideas of multi-scale features and default boxes. Similar to other single-stage detectors, its speed is better than two-stage detectors. SSD algorithm is an algorithm with high speed, high accuracy, and high robustness to scale change. Its main feature is to use multi-layer convolution features with different scales and receptive fields for target detection and recognition.

SSD algorithm is based on a feedforward convolutional neural network. The algorithm first generates a series of fixed number of default boxes. It then uses the corresponding feature graphs of different levels to predict the location and category based on these default boxes. For all the predicted bounding boxes of each category, the redundant and low probability bounding boxes are removed by the non-maximum suppression algorithm. Finally, the detection results are generated. This method is a target detection algorithm based on regression, which imultaneously predicts the location and category within a network framework. Compared with R-CNN series

algorithms, SSD is a single-stage, end-to-end target detection algorithm, and the detection speed is greatly improved. Moreover, multi-layer convolution layers with different scales are used for target detection and recognition due to their unique design. As a result, the detection performance has been improved to a certain extent.

SSD network framework is divided into the base net and extra feature layers. The basic network is a truncated VGG network. The additional layer is the CNN layer with a gradually decreasing scale, and the detection of targets is carried out simultaneously on these characteristic maps with different scales. Feature maps of different scales are used to predict targets of different scales.

The input of the SSD is a 3 channel RGB image. Firstly, the algorithm will map a series of default bounding boxes (default boxes), according to the size of the feature map, and then convolute through a series of convolution cores. Each layer will produce a fixed number of predictions, including 4 position predictions and several category predictions. The default box mechanism is similar to the anchor boxes mechanism in the Region Proposal Network (RPN) in Faster R-CNN. For a p -channel feature map with $m \times n$ size, the convolution kernel with scale $3 \times 3 \times p$ is used to predict the category and location information at each location $m \times n$. Category prediction will predict a score value for each category, representing the category target's possibility in the corresponding box. The position prediction will predict the scale scaling and displacement change based on the corresponding default box, which is the position adjustment based on the default box according to the characteristics of CNN. The default box is a series of rectangular default boxes corresponding to each position $m \times n$ on the original map according to the scale of different levels of the feature map. These default boxes have different sizes and aspect ratios to adapt to the scale transformation of the target to be detected.

For the K default boxes of each position, the SSD algorithm uses convolution operation to predict $c + 1$ category scores (including C target category and a background category) and 4 coordinate positions. That is $(c + 1 + 4) \times K$, each position needs a convolution kernel, so for a scale of $m \times n$. The characteristic graph of n needs a convolution kernel corresponding to $(c + 1 + 4) \times Kmn$ prediction output. Each location corresponds to a fixed number of default boxes, which have different sizes and aspect ratios according to the location and scale of the layer.

During training, you need to match the truth value with the default box to produce positive and negative samples. SSD matches the positive and negative samples by calculating the Jaccard overlap of the default box and the truth box. The threshold is 0.5. If the Jaccard overlap in the truth box is greater than 0.5, it is set as a positive sample, otherwise it is a negative sample. A truth box can match multiple default boxes.

SSD has the following main features:

1. Inherit the idea of transforming detection into expression from Yolo to complete target positioning and classification at one time.

2. Based on anchor in Fast RCNN, a similar prior box is proposed.
3. Add the detection method based on the feature pyramid, that is, predict the target on the feature map of different receptive fields.

2.3 The Two-Stage Target Detection

Among the two-stage target detection algorithms, the R-CNN series is the most famous. This chapter mainly focuses on Faster R-CNN, and its predecessor is Fast R-CNN and R-CNN. We first briefly introduce R-CNN and Fast R-CNN target detection principles, and then focuses on the Faster R-CNN target detection algorithm.

Given the two problems existing in traditional target detection algorithms (see Sect. 2.2), Girshick proposed the R-CNN algorithm in 2014 [9]. Its principle is elementary, mainly by extracting multiple candidate regions to determine the target's position. The R-CNN target detection process is shown in Fig. 1.

Because the traditional algorithm for detecting each sliding window is a way of wasting resources, the R-CNN model uses SS (selective search) image segmentation algorithm [41] to extract 1k-2k candidate regions from the bottom to up. These regions are converted into fixed-size images and sent to CNN respectively to extract the features of each candidate area. Then, the SVM classifier is used to classify the feature vectors extracted by CNN. Then the regression of the coordinates of the upper left and right lower corner of the candidate region is made to modify the location of the candidate region to achieve the target classification and get the boundary. R-CNN uses SS algorithm to generate higher quality ROI and CNN instead of the sliding window used in traditional target detection as ROI and manual feature design. It makes the target detection field achieve a significant breakthrough and open the upsurge of deep learning target detection.

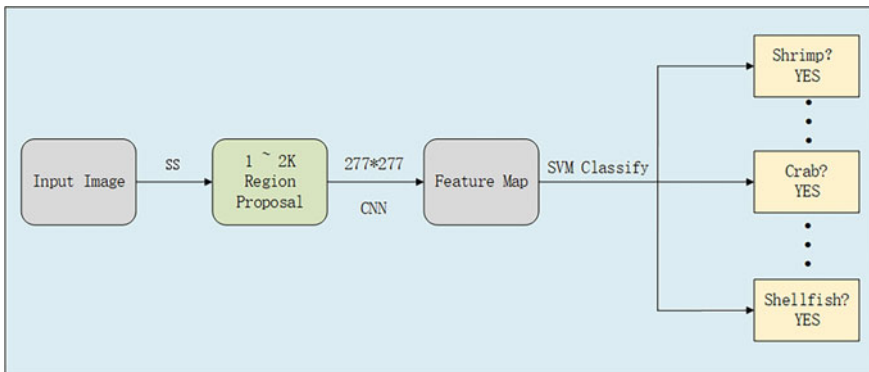


Fig. 1 The target detection process of R-CNN

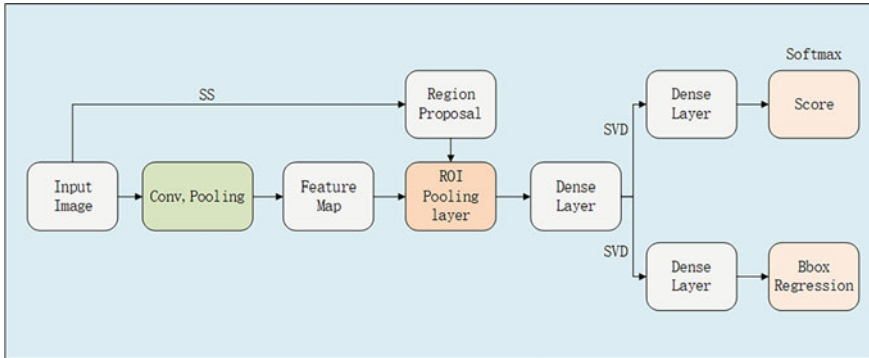


Fig. 2 Fast R-CNN model structure

But the classical R-CNN has the following problems:

1. Due to the need of calculating features for each candidate region, the amount of calculation is very tremendous.
2. The candidate regions are highly overlapped and there are too many repeated calculations.
3. Not end-to-end.
4. Strict size requirement for the input image.

In this case, SPPNet proposed by He et al. [12] successfully solves the problem of repeated convolution in R-CNN. However, the problems of multi-step training and large memory consumption still exist. Therefore, Girshick proposed the Fast R-CNN target detection algorithm in 2015 [8], and the target detection process is shown in Fig. 2.

Fast R-CNN can input any size of pictures into CNN and get the feature map by convolution and pooling operation, which avoid the time-consuming operation of generating candidate regions before convolution in R-CNN. Like R-CNN, Fast R-CNN also uses an SS algorithm to obtain about 2K candidate regions, and then find the corresponding feature boxes of each candidate region in the feature map. However, different from that, Fast R-CNN introduces ROI (Region of Interest) pooling operation. Its input is the feature map and the frame of candidate regions with different sizes obtained by CNN. The size of the output is fixed. The role of the ROI pooling layer is to pool the corresponding region into a fixed-size feature vector in the feature map according to the position coordinates of candidate regions, to carry out the following softmax classification and Bbox (Bounding box) regression.

Fast R-CNN abandons multiple SVM classifiers and Bbox regressors in RCNN and combines classification and regression in one network using a multi-task loss function. It also trains the whole network end-to-end and outputs the target's Bbox value and category label, which improves the model's accuracy. In addition, Fast R-CNN solves the problem of repeatedly extracting features by R-CNN, so the training speed has been significantly improved.

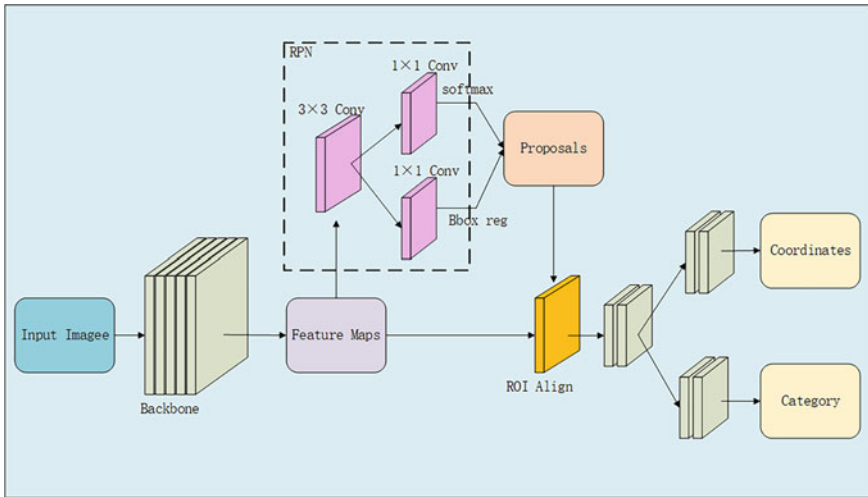


Fig. 3 Faster R-CNN model structure

In 2016, Ren et al. [33] proposed the Faster R-CNN target detection algorithm based on Fast R-CNN. Compared with Fast R-CNN, the most critical point of Faster R-CNN is using RPN (Region Proposal Network) instead of the SS segmentation algorithm to generate candidate frames, which significantly improves the speed of detection frame generation. In addition, Faster R-CNN integrates feature extraction, candidate region extraction, Bbox regression, and softmax classification into one network, which significantly improves speed and accuracy. Generally speaking, the improvement of Faster R-CNN to Fast R-CNN is that the speed of obtaining candidate regions is much faster. The Faster R-CNN target detection process is shown in Fig. 3.

The network structure of Faster R-CNN is similar to that of Fast R-CNN. Firstly, the backbone network is used to extract the features of the input image. The backbone network can use ResNet [13], VGG16, etc. Then, the RPN network is used to obtain the offset of the candidate box relative to the anchor box and the probability of containing targets. The specific operation is: the RPN takes the output feature map of the backbone network as the input and convolutes it using the kernel of 3×3 , and then performs 2 times of 1×1 convolution. The number of output channel is $2 \times k$ and $4 \times k$ respectively. Among them, k represents the number of prior frames anchor on each grid point, and RPN uses this k anchor to make k predictions; The output $2 \times k$ is the target score, which represents whether the predicted candidate box on each grid point contains the target and the probability of containing the target; The output $4 \times k$ is coordinate information, which represents the offset of the predicted candidate frame on each grid point relative to the anchor frame; In Faster R-CNN, k is usually taken as 9. Finally, neural network and maximum pooling are used to calculate the pooled ROI feature map, and the result is reshaped into a vector $1 \times n$. Two fully connected layers are used for classification and regression to obtain the target location and classification information.

2.4 Summary

This section introduces the theory of target detection firstly, then focuses on two-stage R-CNN series target detection algorithm and single-stage YOLO series target detection algorithm, especially Faster R-CNN algorithm and SSD algorithm, and introduces the advantages and disadvantages of single-stage algorithm and two-stage algorithm. Combining the different characteristics of the two algorithms, this chapter provides the basis for the subsequent discussion on target detection counting models. Specifically, in Sect. 3, the Detection and Quantification of Benthic Organisms (DQBO) is introduced. Section 4 presents Detection of Macrobenthos in Seamounts (DMS).

3 DQBO Based on Faster R-CNN with FPN

3.1 Introduction on DQBO

Benthic density has always been an indispensable part of benthic target detection. By analyzing the images of marine benthic density, we can understand the social habits of organisms, help estimate the number of organisms and carry out a series of applications such as aquaculture and biotope protection. With the development of artificial intelligence technology and the depth of computer vision theory, intelligent image processing has become a critical research area. Although CNN-based target detection algorithms are widely used in many scenarios, the detection results do not meet all requirements and usually require more in-depth exploration. As shown in Fig. 4, the number of organisms is dense and numerous. Counting the number is cumbersome and has a high labor cost, so it is of great practical significance to automatically count the image targets.

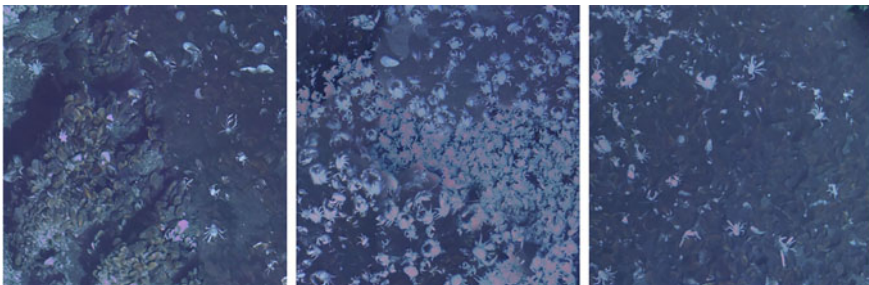


Fig. 4 Benthic organisms density images

3.2 The Faster R-CNN with FPN Framework for DQBO

How to deal with the large-scale change of objects is a fundamental problem in applying target detection. Whether the RPN in Faster R-CNN or Fast R-CNN, it is both based on a single high-dimensional feature, which generally has a poor effect on small object detection. FPN mainly solves the problem of detecting small and medium-sized objects in object detection scenes. Connecting high-dimensional features with low resolution and high semantic information and low dimensional features with high resolution dramatically improves the performance of small object detection.

This section embeds the FPN structure into the Faster R-CNN, combining it with the high-dimensional and low-dimensional feature extraction. Without increasing the amount of calculation of the original model, we successfully solve large-scale change and small object missing detection problems. The FPN network structure is shown in Fig. 5.

Figure 5 ① shows the forward propagation process of the neural network from bottom to top. After convolution operation, the size of the feature map becomes smaller and smaller, and more and more abstract. A pyramid level is defined for each stage of FPN. The output of the last layer of each stage is selected as the reference set of the feature graph because the deepest layer of each stage has more robust semantic information. ② is a top-down process, making the higher-level feature graph more abstract and more semantic to enhance the higher-level feature. Because the feature maps used in each layer are fused with features of different resolutions and semantic intensities, it can detect objects with corresponding resolutions and

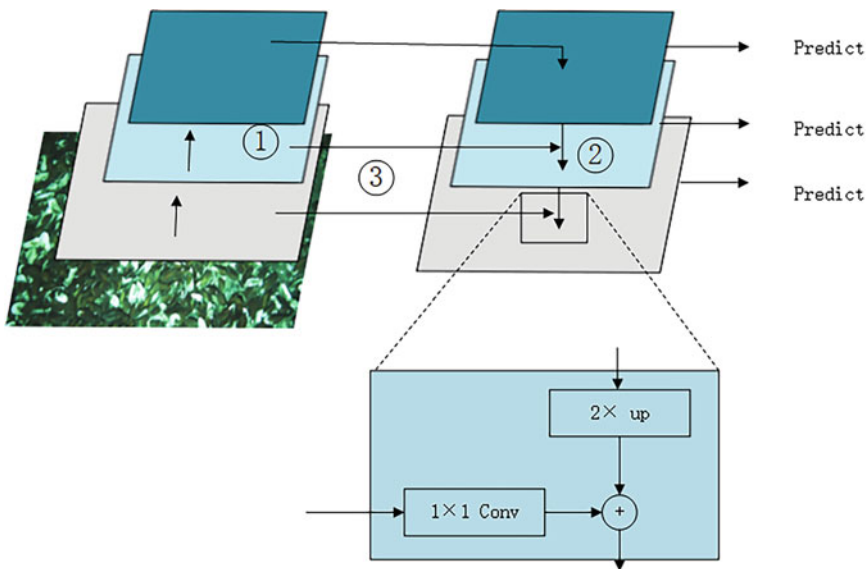


Fig. 5 The structure of FPN

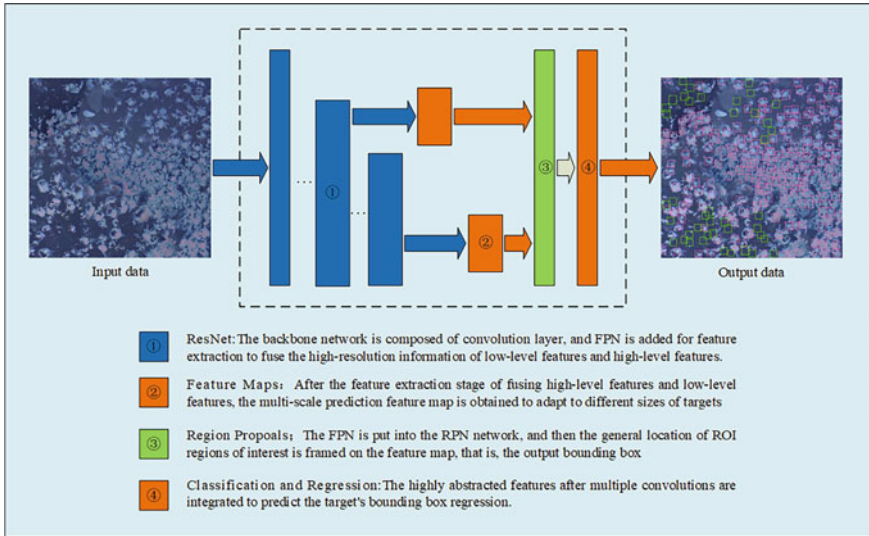


Fig. 6 The structure of Faster R-CNN with FPN

ensure that each layer has appropriate resolution and solid semantic features. ③ is a horizontal connection process, which uses a convolution kernel 1×1 to fuse the result of ② with the output feature graph of ① without changing the size of the feature graph.

In the detection process, the FPN structure is embedded in the Faster R-CNN feature extraction part. The framework of the target detection and counting model based on Faster R-CNN consists of the following three parts: feature extraction, candidate region generation and classification, and Bbox regression. The network structure of Faster R-CNN with FPN is shown in Fig. 6.

1. FPN: Feature extraction

To improve the recognition accuracy of different sizes of organisms in the image, the backbone in Faster R-CNN is replaced by ResNet50 which combines FPN instead of VGG16. Feature maps of different scales are obtained by the FPN and then sent to the RPN to generate candidate regions. The fusion of deep and shallow features makes the FPN structure effectively improve the detection rate of small targets. The multi-resolution feature map detection design makes Faster R-CNN have a better detection effect for different scale targets.

2. RPN: Get candidate region

RPN is a complete convolution network, which can be trained end-to-end, to generate the suggestion bounding box, which can predict not only the boundary of the object but also the probability score of the object. The network structure contains two types of outputs: Softmax classifier and Bounding box, a multi-task model. The core of RPN is the anchor. RPN is mainly used to generate candidate regions. However, the different sizes and aspect ratios of targets make it necessary

to set multiple-scale windows. RPN generates a large number of anchor boxes firstly. After clipping and filtering, Softmax further determines whether these anchors belong to the foreground or background, whether there are objects in the box. The foreground represents containing objects and the background represents not containing objects. At the same time, the other branch begins to modify the anchor frame to form more accurate candidate regions.

The implementation process of RPN is as follows: firstly, a small network is used to perform sliding scanning operation on the feature image obtained by convolution, and it is connected fully with the window on the feature image, then it is mapped into a low dimensional vector, and finally, the vector is fed into the Bbox regression layer (reg) and Bbox classification layer (cls). The reg layer is mainly used to estimate the candidate output (x, y, w, h) corresponding to the candidate anchor. The cls layer is used to judge whether the candidate region is foreground or background.

3. Target classification and Bbox regression

Before the target classification and bounding-box regression, we need to carry out the pooling operation. This layer uses the candidate regions generated by RPN and the feature maps of different scales generated by the backbone network to get the fixed-sized candidate feature maps and inputs them into the subsequent network. We can use the full connection operation to identify and locate the target. In the classification process, Softmax is used as the classification function to classify the fixed-size feature image formed by the ROI pooling layer according to the specific category. At the same time, the L1 loss is used to complete the candidate regression operation on the bounding box for position verification to obtain the accurate position of the object. The loss function equations of the whole network is shown in Eq. 1.

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (1)$$

P_i is the probability of the category of anchor calculated by the Softmax. When the IOU between the anchor and the target window is greater than 0.7, the value of p_i^* is 1, and when the IOU is less than 0.3, the value is 0. t_i^* is a scaling parameter, which is the real scaling value for regression, including coordinate scaling and size scaling. t_i is used to represent the scaling value predicted by the network in the training process. Faster R-CNN completes the regression task by learning the scaling value. The loss function consists of two parts: classification loss and regression loss. See Eq. 2 for the calculation of classified loss:

$$L_{cls}(t_i, t_i^*) = -\log(p_i^* p_i + (1 - p_i^*)(i - p_i)) \quad (2)$$

See Eqs. 3 for the calculation of regression loss:

$$L_{reg}(t_i, t_i^*) = R(t_i - t_i^*) \quad (3)$$

Where is the loss value is calculated by smooth L1 function, see Eqs.4:

$$Smooth_{L1} = \begin{cases} 0.5x^2 & |x| < 1 \\ |x| - 0.5 & \text{others} \end{cases} \tag{4}$$

The general implementation process of the whole network is as follows: First, the input image is represented as $Height \times Width \times Depth$. The tensor is processed by the backbone network with an FPN structure to obtain feature maps of different scales. Then, the RPN is used to extract candidate regions. After obtaining the possible related objects and their corresponding positions in the original image, the features extracted from the backbone network and the bounding box containing the related objects are pooled by ROI, and the features of the related objects are extracted to obtain new vectors. Then it is sent to the subsequent classification and regression network to complete the target recognition and positioning.

3.3 Experimental Results and Discussions

In this experiment, the iterations are 120 and the batch size is 1. We set the learning rate and weight decay to 0.0001. Set the size and scale of anchor to (8, 16, 32) and (0.5, 1, 2). The impulse gradient descent method is used to reduce overfitting, and the impulse is set to 0.9.

For the static image data, a total of 630 samples were obtained. Labeling image annotation tool is used to calibrate these samples manually, and then we divide the training, validation and test sets according to the ratio of 7:2:1. We used the test set to evaluate our model.

In this paper, Recall, Precision and AP were used to evaluate the results of this experiment. The results are shown in Table 1.

The definition of recall and precision is as follows: Eq. 5:

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN} \tag{5}$$

Among them, the definition of TP , TN , FP , FN is shown in Table 2, which respectively represents true positive, true negative, false positive, and false negative recognition, *Precision* represents the correct proportion of all predicted targets, *Recall*

Table 1 The experimental result

Class	Recall	Precision	AP
Mussel	0.745	0.719	0.720
Shinkaia	0.876	0.755	0.756

Table 2 Obfuscation matrix for specified categories

Ground truth	Predictive	
	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

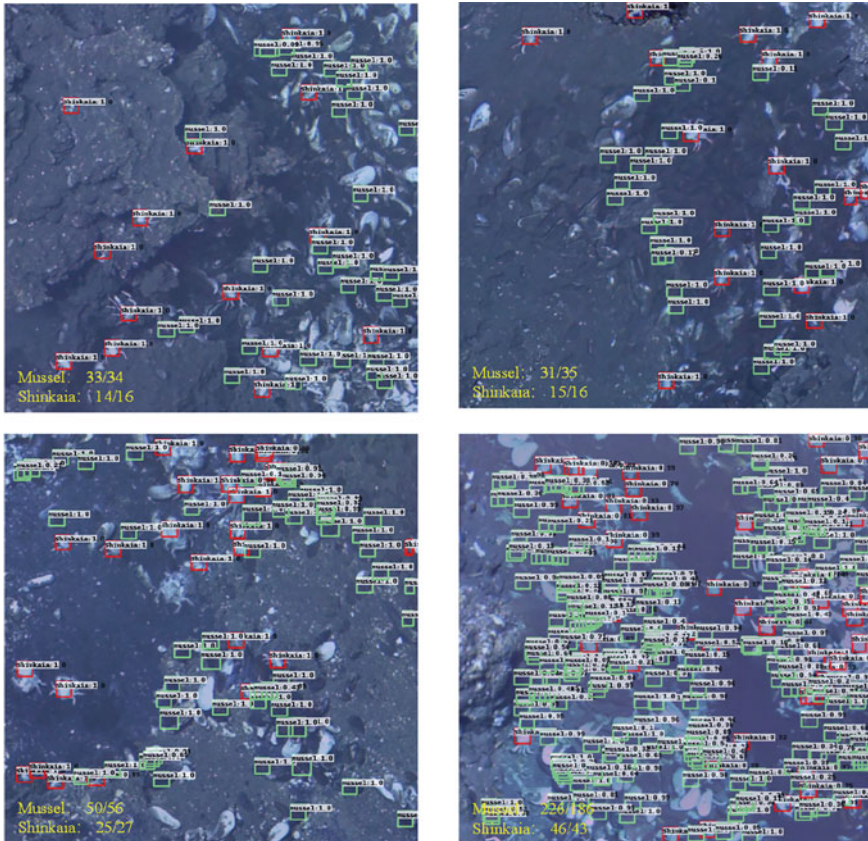


Fig. 7 Marine benthos detection and quantification results. The tag 33/34 indicates that the real quantity of mussel is 34 and the detection quantity is 33

represents the proportion of correctly located and recognized targets in the total number of targets.

Finally, the mean average precision (mAP) of 73.8% is obtained on the marine biological data set. Among them, the accuracy of FPN method for mussels recognition is 72.0%, and the accuracy of shinkaia is 75.6%. The visualization of experimental results is shown in Fig. 7. It can be seen that FPN is an excellent static image counting model for marine organisms.

3.4 Summary

This section first introduces the overall structure of the Faster R-CNN model and describes the network structure in detail. Next, the structure of the convolution neural network used for feature extraction is introduced, and 441 images are trained by this network. The experimental results show that the Faster R-CNN recognition has good effect, and can be applied to the quantitative analysis of marine biological recognition.

4 DMS Based on SSD

4.1 Introduction on DMS

In a narrow sense, seamount refers to the submarine uplift with a height of more than 1000 m below sea level. In a broad sense, the sea-knolls with a height of 500–1000 meters and hills below 500 m are called seamounts. Seamounts are the significant ecological landscapes in the deep ocean. It is estimated that the global seamounts account for 21% of the global seabed area [3, 45]. With the unique topography and hydrological characteristics, as well as the unique ecosystem, abundant biodiversity, and colossal resource value, seamounts have become one of the most concerning areas in deep-sea research. Compared with the surrounding deep-sea area, seamounts have high productivity, high biomass, and high biodiversity.

With the change of water depth and sediment types, the biological communities of seamounts show obvious biota replacement, and different sediment types often distribute different biota. For example, in soft bottom sediments, sea gills, starfish, sea urchins, and sea cucumbers are more common, while in hard rock bottom sediments, sponges, black corals, gorgonians, and sea anemones are dominant. The research on seamounts primarily focuses on the macrobenthos, whose individual is more than 2 cm and can be identified through the seabed image.

With their unique biological communities, rich biodiversity, and huge resource value, seamounts have become the focus of deep-sea biodiversity protection. At present, the protection of marine Biodiversity Beyond National Jurisdiction (BBNJ) has become an issue of global concern. Scientific understanding of the biological composition and distribution of seamounts is the key to the development, utilization, and protection of this fragile deep-sea ecosystem. It is the most concentrated among the seamounts globally and has the most significant number in the Western Pacific Ocean. The Western Pacific is the area with the most densely distributed seamounts and the most developed trench-arc-basin system globally. The cross-linking area of the Yapu Trench, Mariana Trench, and Caroline Ridge is the most representative. It is also one of the areas with minor research on seamounts in the world.

The most considerable difficulty in studying deep-sea biodiversity lies in the acquisition of deep-sea specimens and data. Due to the complex topography of seamounts, biological sampling is more complicated than the general deep-sea. Among the more

than 30000 seamounts globally, only 1% of them have been carried out in biological sampling, and only about 50 seamounts have been sampled comprehensively. As a result, seamounts are still one of the “least known biological habitats” for humans. The research on the biodiversity of seamounts is limited to the macrobenthos, and most of the research only focuses on the species composition, while a few focus on community structure and distribution. Due to the limitation of sample acquisition and insufficient sampling, a considerable part of the classification and identification of seamount organisms is based on the analysis of video and image of benthic organisms. Many novel organisms cannot be identified due to the lack of samples.

In recent years, our country has successfully carried out some seamount explorations represented by ‘Jiaolong’ HOV and ‘FaXian’ ROV and obtained many first-hand submarine image data in the South China Sea, Western and Central Pacific. It significantly improves the deep-sea detection level of our country and provides data support for automatic detection of benthos in seamounts.

4.2 Seamount Macrobenthos Dataset

Supported by the strategic leading science and technology project of the Chinese Academy of Sciences(A) “material and energy exchange and its impact on the tropical western Pacific Ocean system”, the Institute of Oceanology, Chinese Academy of Sciences has established a research system for the detection of marine biodiversity in seamounts through the construction of technical platform and team. A comprehensive survey of the deep-sea environment, biodiversity, and ecosystem structure of three seamounts in the cross-linking area of Yapu and Mariana Trench and Caroline Ridge in the Western Pacific Ocean was carried out (as shown in Fig. 8). More than 1000 giant and large biological samples were collected through the sampling of seamount detection by using “FaXian” ROV, and more than 880 GB in situ imaging data of seabed organisms were obtained. In Fig. 8, the peak of Yapu seamount (Y3) is located in $8^{\circ}51'N$, $137^{\circ}47'E$; the peak of the Mariana seamount (M2) is located in $11^{\circ}19'N$, $139^{\circ}20'E$; the peak of the Caroline seamount (M4) is located in $10^{\circ}29'N$, $140^{\circ}8'E$.

Based on the in-situ image data of macrobenthos obtained from the above three seamounts’ surveys, the 63 in-situ image data were labeled as Paskal VOC format data by LabelImg, an image annotation tool. This images data include *Pheronemoides fungosus* Gong & Li, 2017 [10], *Paragorgia rubra* Li, Zhan & Xu, 2017 [20], *Chrysogorgia ramificans* Xu et al., 2019 [44], *Paraphelliactis tangi* Li & Xu, 2016 [19], *Poloipogon distortus* Gong & Li, 2018 [11] and *Chrysogorgia binata* Xu et al., 2019 [44]. These six species are newly discovered in recent years. Then we check all data manually to ensure that all image resolutions are 1920×1080 .

In computer vision, the typical annotation method is annotating the object on the image with a rectangular bounding box. In seamount data, the bounding box is labeled with $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$, and (x_{\min}, y_{\min}) and (x_{\max}, y_{\max}) are the two vertices on the diagonal of the rectangular label box. The whole macrobenthos data set of seamounts are collected, and each image corresponds to an XML annotation

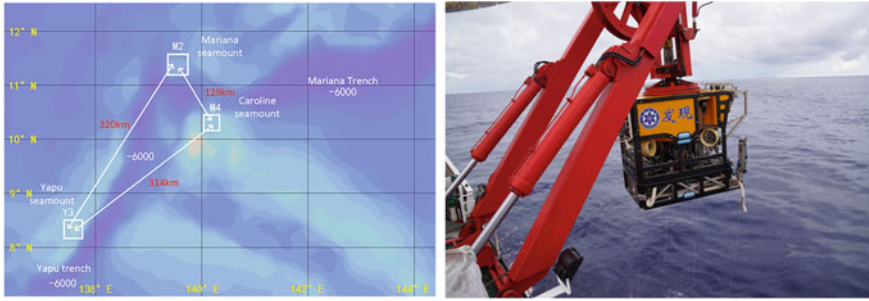


Fig. 8 Seamount data acquisition location and discovery ROV

file. Finally, stratified random sampling is used to divide the labeled data into the training set and test set according to the ratio of 8:2.

4.3 The SSD Framework for DMS

The experimental framework of macrobenthos detection in seamounts with the SSD model is shown in Fig. 9.

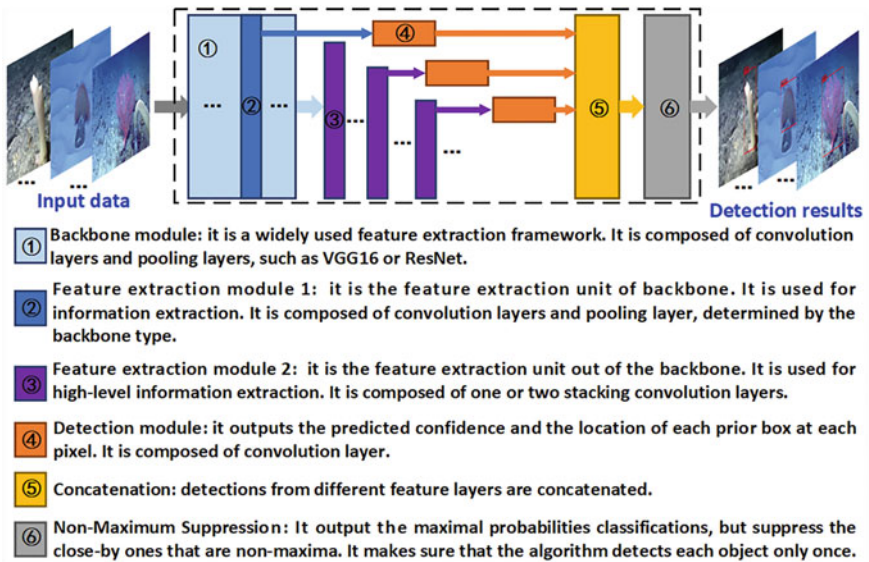


Fig. 9 The entire process of underwater species detection by SSD

SSD is based on VGG16 [38], which is pre-trained on the ILSVRC CLS-LOC dataset [36]. We convert FC6 (sixth fully connected layer) and FC7 to convolutional layers, subsample parameters from FC6 and FC7, after that remove all the dropout layers and the FC8 layer using SSD Weighted Loss Function [23]. We adjust the outcome model using SGD with initial learning rate 10^{-4} , 0.9 momentum, 0.0005 weight decay, and batch size 32. The entire process can be seen in Fig. 9.

4.4 Experimental Results and Discussions

Part of the output of our SSD model is shown in Fig. 10. Among all the six different marine species, our SSD model achieved 98.04% mAP (mean Average Precision) and the average value of IOU (Intersection-over-Union) over 0.8 on the test dataset with 63 images.

In our experiment, although we have verified that the implementation of SSD on our marine species data is feasible, SSD still often fails to detect small objects. Besides, our sample size and marine species categories are not enough. In the future, we will improve the SSD, and improve the ability to detect small objects in the camera further. Our ultimate vision is to build an AI system that can identify hundreds of thousands of marine species in real-time.

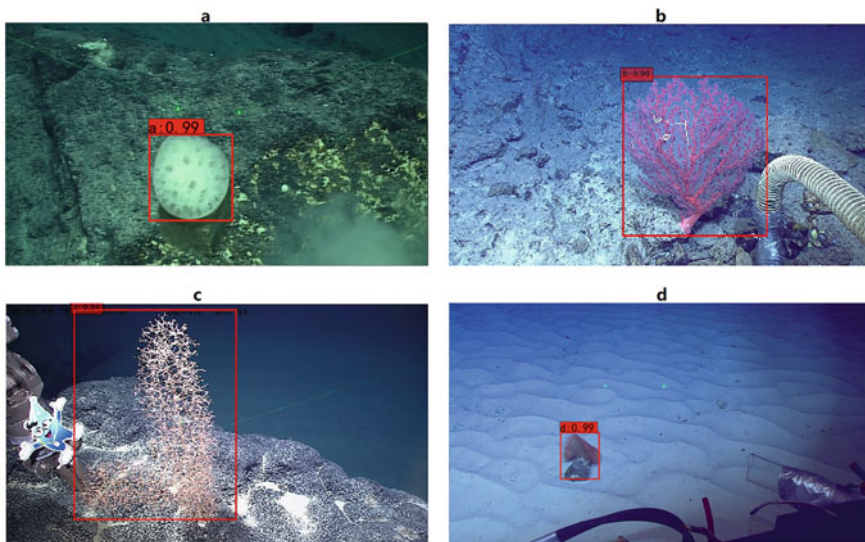


Fig. 10 **a** *Pheronemoides fungosus* Gong & Li, 2017, **b** *Paragorgia rubra* Li, Zhan & Xu, 2017, **c** *Chrysogorgia ramificans* Xu et al., 2019, **d** *Paraphelliactis tangi* Li & Xu, 2016

4.5 Summary

This section first introduces the overall structure of the SSD model applied and describes the network structure in detail. Next, the structure of the convolution neural network used for feature extraction is introduced, and 63 images are trained by this network. The experimental results show that the SSD recognition effect is good and it can be applied to the detection macrobenthos in seamounts.

5 Conclusions and Future Works

Target detection is one of the three major tasks in the field of computer vision. Computer vision algorithms are gradually applied to underwater scenes with the continuous development of deep learning technology and its wide application on land. In this chapter, the application of deep learning algorithm in target detection is extended to detecting marine organism. The detection and counting of marine organisms are studied in detail. First of all, in marine static image biological detection and counting, we explore the network architecture based on Faster R-CNN with FPN. Then, we verify the feasibility of SSD in the detection of giant benthic organisms in seamounts.

This chapter has completed the critical technologies of quantitative analysis system of artificial intelligence for marine benthos, realized the integrated development of marine big data, marine artificial intelligence and marine Internet of things, promoted the comprehensive development of marine artificial intelligence application, and filled the lack of artificial intelligence application in the deep-sea field partly.

Based on the above research, our subsequent work includes the following aspects:

1. Further expand the species richness. In addition, manual tagging is time-consuming and labor-consuming. The active learning method will be used for tagging in the subsequent expansion of the deep-sea biology training database.
2. The following research will focus on the dynamic video object detection and counting algorithm based on the static image.
3. Promote the AI algorithm model's landing and start developing the deep-sea macro-organism recognition and quantitative analysis system.

Acknowledgements The work was supported by the Key Program of National Natural Science Foundation of China (No. 41930533), the Senior User Project of RV KEXUE, managed by the Center for Ocean Mega-Science, Chinese Academy of Sciences (KEXUE2019GZ04), and the Key R & D Project of Shan-dong Province (2019JZZY010102).

References

1. Blanc K, Lingrand D, Precioso F (2014) Fish species recognition from video using SVM classifier. In: Proceedings of the 3rd ACM international workshop on multimedia analysis for ecological data, pp 1–6
2. Bochkovskiy A, Wang CY, Liao HYM (2020) Yolov4: optimal speed and accuracy of object detection. [arXiv:2004.10934](https://arxiv.org/abs/2004.10934)
3. Clark MR, Rowden AA, Schlacher T, Williams A, Consalvey M, Stocks KI, Rogers AD, O'Hara TD, White M, Shank TM et al (2010) The ecology of seamounts: structure, function, and human impacts. *Ann Rev Marine Sci* 2:253–278
4. Cortes C, Vapnik V (1995) Support-vector networks. *Mach Learn* 20(3):273–297
5. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), vol 1. IEEE, pp 886–893
6. Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A (2010) The pascal Visual Object Classes (VOC) challenge. *Int. J. Comput. Vis.* 88(2):303–338
7. Fisher RB, Chen-Burger YH, Giordano D, Hardman L, Lin FP et al (2016) Fish4Knowledge: collecting and analyzing massive coral reef fish video data, vol 104. Springer
8. Girshick R (2015) Fast R-CNN. *Int J Comput Vis* 1440–1448
9. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587
10. Gong L, Li X (2017) A new genus and species of Phoronematidae (Porifera: Hexactinellida: Amphidiscosida) from the western pacific ocean. *Zootaxa* 4337(1):132–140
11. Gong L, Li X (2018) A new species of Phoronematidae (Porifera: Hexactinellida: Amphidiscosida) from the Northwest Pacific Ocean. *Acta Oceanologica Sinica* 37(10):175–179
12. He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans Pattern Anal Mach Intell* 37(9):1904–1916
13. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
14. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask R-CNN. *Int J Comput Vis* 2961–2969
15. Huang H, Zhou H, Yang X, Zhang L, Qi L, Zang AY (2019) Faster R-CNN for marine organisms detection and recognition using data augmentation. *Neurocomputing* 337:372–384
16. Kratzert F, Mader H (2017) Advances of FishNet towards a fully automatic monitoring system for fish migration. In: EGU general assembly conference abstracts, p 7932
17. Law H, Deng J (2018) Cornernet: Detecting objects as paired keypoints. In: Proceedings of the European conference on computer vision (ECCV), pp 734–750
18. Li X, Liu B, Zheng G, Ren Y, Zhang S, Liu Y, Gao L, Liu Y, Zhang B, Wang F (2020) Deep-learning-based information mining from ocean remote-sensing imagery. *Nat Sci Rev* 7(10):1584–1605
19. Li Y, Xu K (2016) *Paraphelliactis tangi* n. sp. and *Phelliactis yapensis* n. sp. two new deep-sea species of Hormathiidae (Cnidaria: Anthozoa: Actiniaria) from a seamount in the tropical Western Pacific. *Zootaxa* 4072(3):358–372
20. Li Y, Zhan Z, Xu K (2017) Morphology and molecular phylogeny of *Paragorgia rubra* sp. nov. (Cnidaria: Octocorallia), a new bubblegum coral species from a seamount in the tropical Western Pacific. *Chinese J Oceanol Limnol* 35(4):803–814
21. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: common objects in context. In: European conference on computer vision. Springer, pp 740–755
22. Lin TY, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision, pp 2980–2988
23. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) SSD: single shot multibox detector. In: European conference on computer vision. Springer, pp 21–37

24. Lowe DG (1999) Object recognition from local scale-invariant features. In: Proceedings of the seventh IEEE international conference on computer vision, vol 2. IEEE, pp 1150–1157
25. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *Int J Comput Vis* 60(2):91–110
26. Lu H, Li Y, Uemura T, Kim H, Serikawa S (2018) Low illumination underwater light field images reconstruction using deep convolutional neural networks. *Futur Gener Comput Syst* 82:142–148
27. Marcos MSA, David L, Peñafior E, Ticzon V, Soriano M (2008) Automated benthic counting of living and non-living components in Ngedarrak Reef, Palau via subsurface underwater video. *Environ Monit Assess* 145(1):177–184
28. Matai J, Kastner R, Cutter Jr G, Demer D (2010) Automated techniques for detection and recognition of fishes using computer vision algorithms. In: Williams, K., Rooper, C., Harms, J., (eds.), NOAA technical memorandum NMFS-F/SPO-121, report of the national marine fisheries service automated image processing workshop, Seattle, Washington, 4–7 Sept 2010
29. Mignotte M, Collet C, Pérez P, Bouthemy P (2000) Markov random field and fuzzy logic modeling in sonar imagery: application to the classification of underwater floor. *Comput Vis Image Underst* 79(1):4–24
30. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7263–7271
31. Redmon J, Farhadi A (2018) Yolov3: an incremental improvement. [arXiv:1804.02767](https://arxiv.org/abs/1804.02767)
32. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
33. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: towards real-time object detection with region proposal networks. [arXiv:1506.01497](https://arxiv.org/abs/1506.01497)
34. Rohwer F, Youle M, Vosten D (2010) Coral reefs in the microbial seas, vol 1. Plaid Press Granada Hills
35. Rova A, Mori G, Dill LM (2007) One fish, two fish, butterflyfish, trumpeter: recognizing fish in underwater video. In: MVA
36. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M et al (2015) ImageNet large scale visual recognition challenge. *Int J Comput Vis* 115(3):211–252
37. Shiau YH, Lin SI, Chen YH, Lo SW, Chen CC (2012) Fish observation, detection, recognition and verification in the real world. In: Proceedings of the international conference on image processing, computer vision, and pattern recognition (IPCV). The Steering Committee of The World Congress in Computer Science, Computer, p 1
38. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)
39. Spampinato C, Giordano D, Di Salvo R, Chen-Burger YHJ, Fisher RB, Nadarajan G (2010) Automatic fish classification for underwater species behavior understanding. In: Proceedings of the first ACM international workshop on analysis and retrieval of tracked events and motion in imagery streams, pp 45–50
40. Stokes MD, Deane GB (2009) Automated processing of coral reef benthic images. *Limnol Oceanogr: Methods* 7(2):157–168
41. Uijlings JR, Van De Sande KE, Gevers T, Smeulders AW (2013) Selective search for object recognition. *Int J Comput Vis* 104(2):154–171
42. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition, vol 1. IEEE, pp I–I
43. Xia C, Fu L, Liu H, Chen L (2018) In situ sea cucumber detection based on deep learning approach. In: 2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO). IEEE, pp 1–4
44. Xu Y, Li Y, Zhan Z, Xu K (2019) Morphology and phylogenetic analysis of two new deep-sea species of *Chrysogorgia* (Cnidaria, Octocorallia, Chrysogorgiidae) from Kocebu Guyot (Magellan seamounts) in the Pacific Ocean. *Zookeys* 881:91

45. Yesson C, Clark MR, Taylor ML, Rogers AD (2011) The global distribution of seamounts based on 30 arc seconds bathymetry data. *Deep Sea Res Part I: Oceanogr Res Pap* 58(4):442–453
46. Zhang S, Wen L, Bian X, Lei Z, Li SZ (2018) Single-shot refinement neural network for object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 4203–4212
47. Zhu Q, Yeh MC, Cheng KT, Avidan S (2006) Fast human detection using a cascade of histograms of oriented gradients. In: *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, vol 2. IEEE, pp 1491–1498

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits any noncommercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if you modified the licensed material. You do not have permission under this license to share adapted material derived from this chapter or parts of it.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

