Karlsruher Schriften
zur Anthropomatik

Band 62

Jürgen Beyerer, Tim Zander (Eds.)

Proceedings of the 2022 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory

KIT Scientific
Publishing

Jürgen Beyerer, Tim Zander (Eds.)

Proceedings of the 2022 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory

Karlsruher Schriften zur Anthropomatik
Band 62
Herausgeber: Prof. Dr.-Ing. habil. Jürgen Beyerer

Eine Übersicht aller bisher in dieser Schriftenreihe erschienenen Bände finden Sie am Ende des Buchs.

Proceedings of the 2022 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory

by
Jürgen Beyerer, Tim Zander (Eds.)

KIT Scientific Publishing

Impressum

Scientific
Publishing

Karlsruher Institut für Technologie (KIT)
KIT Scientific Publishing
Straße am Forum 2
D-76131 Karlsruhe

KIT Scientific Publishing is a registered trademark
of Karlsruhe Institute of Technology.
Reprint using the book cover is not allowed.

www.ksp.kit.edu

Print on Demand 2023 – Gedruckt auf FSC-zertifiziertem Papier

ISSN 1863-6489 (Schriftenreihe)
ISSN 2510-7259 (Tagungsband)
ISBN 978-3-7315-1304-9
DOI 10.5445/KSP/1000158900

Preface

In 2022, the annual joint workshop of the Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB) and the Vision and Fusion Laboratory (IES) of the Institute for Anthropomatics, Karlsruhe Institute of Technology (KIT) was hosted again in a Black Forest house near Triberg.

For a week from the 31st of July to the 5th of August, the PhD students of both institutions delivered extended reports on the status of their research and participated in heated discussions on topics ranging from computer vision and optical metrology to usage control, control theory and neural networks. Most results and ideas presented at the workshop are collected in this book in the form of detailed technical reports. This volume provides a comprehensive and up-to-date overview of some of the research programs of the IES Laboratory and the Fraunhofer IOSB.

The editors thank Arno Appenzeller, Jonas Vogl, Paul Wagner and Zeyun Zhong for their efforts resulting in a pleasant and inspiring atmosphere throughout the week. We would also like to thank the doctoral students for writing and reviewing the technical reports as well as for responding to the comments and suggestions of their colleagues.

Jürgen Beyerer & Tim Zander

Contents

Personalized Explanations

Maximilian Becker

Vision and Fusion Laboratory
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT), Germany
maximilian.becker@kit.edu

Abstract

Machine learning systems are often hard to investigate and intransparent in their decision making . Explainable Artificial Intelligence (XAI) tries to make these systems more transparent. However, most work in the field focuses on technical aspects like maximizing metrics. The human aspects of explainability are often neglected. In this work, we present personalized explanations, which instead focus on the user. Personalized explanations can be adapted to individual users to be as useful and relevant as possible. They can be interacted with to give users the ability to engage in an explanatory dialog with the system. Finally, they should also protect user data to increase the trust in the explanation system.

1 Introduction

Artificial intelligence and machine learning have become extremely popular technologies that are widely used because of their many advantages. However, learned models like neural networks also have some major disadvantages, especially their lack in transparency. During training, the models learn correlations from the training data that enable them to make predictions on unseen data and take decisions. What exactly a model learned, what they pay attention to and how they make decisions is however hard to comprehend. The field of explainable artificial intelligence (XAI) tackles this problem and tries to make learning systems more transparent. The goal is to create explanations for systems that help users or operators to better understand the models and their inner workings[2].

However, much of the research in the field is very technical, neglects the human aspect of explainability and only relies on researchers intuitions. Many works concentrate on technical aspects and try to maximize metrics that are not validated by user studies or grounded in psychology. Papers focused on users are mostly focused on user interfaces and not on the underlying algorithms[12, 10, 1].

In this work we present ideas for XAI methods that are better adapted to users to make them more useful and relevant. The first aspect is that methods should be individualized to the user to make them more helpful. Users should have the ability to customize an explanation in order to adapt it to their needs. Existing methods of interaction and individualization are presented in Section 2. A new approach for individualized and interactive explanations will be discussed in Section 3. Section 3.1 explains the approach and focuses on the individualization. However, users have to be able to interact with the system to get explanations they understand and are relevant for them. Methods of interaction with the new approach are shown in Section 3.2. Individualized explanations do however require personal data of the user in order to personalize the explanations. In order to build trust with the system, the user data has to be protected. Explanations can also help users to understand what kind of data is needed for a system to function properly. Concepts for data protection and data minimization will be shown in Section 4.

2 Background

There are different existing approaches in the literature to interact with XAI systems. The first one is to give the user the option to generate multiple explanations[15]. By generating multiple explanations the user gets different view points and has a better chance of understanding them. This can be done by generating multiple explanations of the same type or explanations of different types. The user can also get the option to change input data[5]. By changing the instances that are explained by the system the user can get a better overview over the feature space and the behavior of the system. Other works evaluate the interaction with graphical representations or user interfaces[6]. They investigate how different visualizations or interfaces help users to understand the explanations. Another way to interact with an explanation system is through an interface using natural language processing[4]. This way the user can use natural language to interact with the system which makes it much more suited for end users with little technical knowledge. All these approaches leave the explanation system itself untouched and only build different user experiences around them. By interacting with an explanation system, the explanations will also be individualized on a basic level. However, explanations can also be individualized explicitly. DiCE[14] can generate counterfactual explanations that are diverse, meaning that different explanation instances are different from each other. The method can also be used to set feature constraints that are used to ensure feasibility of the explanations but can also be used to adapt explanations to individual users.

3 Individualized Explanations

Individualized explanations should be adaptable to the use case as well as the individual user. The explanations can be adapted by an admin or professional user or the end user of the system itself. Different aspects can be considered when adapting explanations. One aspect is general knowledge or world knowledge as well as knowledge about the use case in which a system is deployed. In a medical use case for example, other aspects are relevant compared to a financial use case. Different features are important in different contexts and different applications so the explanations have to reflect that. Explanations should also be adapted to the user group. Different user groups have different abilities and knowledge levels in machine learning and the application domain. Machine learning experts, domain experts and end users have different capabilities and require different explanations. However, the explanations should not only be adapted to the user group but also to the individual user itself. To achieve this, different sources of background knowledge can be used (see Section 3.1) or the user can be given the ability to customize the explanations by herself (see Section 3.2).

3.1 Personalized Counterfactuals

Counterfactual explanations are a form of explanation for machine learning systems. A counterfactual describes an alternative state in which some changes were made that lead to a different outcome. In machine learning, the factual is an instance whose prediction from a model should be explained. The counterfactual is an instance with small changes that lead to a different classification. For example, if a credit application is rejected a counterfactual explanation could tell that the application would have been accepted if the credit amount was lowered by a certain amount[13].

Counterfactuals are a local explanation method, which means that they explain a single data point or decision of a model in contrast to global explanations which explain the behavior of the whole model. They are calculated by searching for the closest instance from the one that should be explained that changes the prediction of the classifier. This can be done by random sampling[3], using a gradient[9], formulating the problem as an optimization problem[16] or with genetic algorithms[14]. Counterfactuals originate from counterfactual thinking which people engage in regularly[12]. Thus people are already used to the concept which makes these explanations especially user friendly and suited for non technical end users.

It is however not obvious how different features with different value ranges and units should be treated when comparing counterfactual explanations. It is for example not possible to objectively compare what change in the credit amount equals to what change in the credit length. The idea behind personalized counterfactuals is to have a weighted distance metric to calculate the distance between a factual and different counterfactuals. The weights can be chosen by the user to represent her preferences. If for example the weight for the credit duration is low and the weight for the credit amount is high, a change in the credit amount will be penalized more and a change in the credit duration will be preferred. This means that by changing the weights of the distance function the users can adapt the counterfactual explanations to their needs or preferences.

Besides the weighting of features, users have some more ways to personalize the explanations. Some features are unchangeable like a persons place of birth. Users can tell the system to ignore such features in the explanations. Other options would be to restrict the value range in which a feature can be changed or make it only changeable in one direction, like an age which can only get larger. If it is a multi-class classification problem the user can also specify the class the counterfactuals should be classified as.

In addition to options for single features, users also have the option to set global metrics to diversify the different counterfactual explanations shown to the user. These global metrics evaluate a set of different counterfactuals. By changing them, the user can get multiple similar explanations or more different ones. Users can also adjust a weight that measures in how many columns changes were made. With these global metrics, the user can configure the overall set of different counterfactual explanations

All these settings can be adjusted by an administrator to represent world knowledge or adapt the explanations to a specific use case. The administrator will set these options once for a specific application. In a second step, the end user can adjust all settings or a smaller subset in the application itself. This is done to personalize the explanations to the specific user.

After all the weights and metrics are set, the search for personalized counterfactuals is done with an evolutionary algorithm. Features of the original instance are randomly changed, excluding the ignored features. The new instances generated in this way are passed to the model to check if they are counterfactuals or if they are classified as the wanted class. Then, the distance from the original instance is measured by a weighted Euclidean distance using the previously specified weights. The set of instances is also evaluated by the global metrics. The best instances are chosen and changed again. This process is iterated until the rating by the metrics does not change anymore. This approach is completely model agnostic because it uses no internal information about the model that should be explained, like gradients. Only predictions of the model are used to check if instances are counterfactuals or if they are classified as the wanted class.

3.2 Interactive Explanations

An important aspect to adapt explanations to users is to enable users to interact with the system. This way, users can influence explanations and customize them to their needs, be it their knowledge level in a certain domain or their technical expertise. Explanations between humans are also often given in a form of dialog. The explainee can ask about things she does not understand or for details on the given explanation. The explainer can then give additional information, reformulate the given explanation or come up with a new one. Interaction and individualization go hand in hand because users are only able to individualize their explanations if they are able to interact with the system and users that can interact with an explanation system will try to get a better understanding by adapting the explanations to their needs. The personalized counterfactual explanations can also be interacted with in several ways. The first way is to change the weights for different features. This influences how much a feature is changed in the generated counterfactuals. Users can also exclude features from the search by marking them as unchangeable. This helps to only generate satisfiable explanations and not ones that are impossible or unrealistic by for example demanding to change a persons race. The next way to interact with the system is to adjust the global metrics that compare the set of generated counterfactuals. With these metrics, the users can get more or less diverse explanations and influence how sparse the explanations are, meaning how many features are changed. The final way is by changing the target class. With this setting, users can tell the system to generate explanations for a specific target class. By looking for counterfactuals with a specific target class, the user can see what changes to an instance are needed to reach the desired class.

4 Protection of Personal Data

The previous Sections showed how explanations can be personalized and interacted with to better meet the user's needs. However, personalization also has a drawback: it requires personal data from the users. Without some form of data about the user of an explanation system it is not possible to adapt explanations to the user. Personal data underlies the European Union's General

Data Protection Regulation (GDPR)[8] as well as the Artificial Intelligence Act[7]. These regulations demand the protection of personal data. One way to protect user data in any application is by the use of trusted computing methods. An approach to this is shown in section 4.1. Explanations can also be used to show users what influence sharing or not sharing some data has on a system. Users can see how system behavior changes with their decision and find a configuration that works for them. With such explanations users can make informed decisions on what data they want to share with a system and what data they want to keep private. This way users are able to minimize the data they have to share with a system. The idea for using XAI to explain the effect of sharing data is shown in Section 4.2.

4.1 Explainable AI and Trusted Computing

Users may be uncomfortable with sharing their data with a system that they do not understand and cannot trust. XAI can explain a systems behavior to a user but to get relevant explanations users often have to share their data first in order to get explanations that are relevant to their situation. In order to keep personal data safe and contribute to increasing the trust in the system trusted computing methods can be used. Two trusted computing technologies that are useful for this are Trusted Platform Modules (TPMs) or Trusted Execution Environments (TEEs). TPMs are trusted hardware modules that can verify the state of a system. TEEs are a separate part of the processor that enables secure data processing and is not accessible even by the operating system. These methods can be used to secure an explanation system and make it more trustworthy either by verifying that the system is in a trustworthy state with TPMs or by executing code on TEEs. The combination of these technologies can create trust in the system through trusted computing methods and trust in the underlying machine learning model through XAI.

4.2 Explainable AI for Data Minimization

Some applications require different kinds of data from users. For example, a health app may be interested in health data, location data and general information about the user. However, users may not want to share all this data with an application. In order to let users make an informed decision about the data they want to share, they have to know how the system behavior changes if they decide to keep some of the data private. Here, we will present a concept on how XAI can be used to generate such explanations.

The idea is to use a combination of SHAP[11] and counterfactual explanations. SHAP is a XAI method building on Shapley values. It calculates a feature importance by omitting features from an instance and replacing them with values from random instances from the training set. The predictions of the model with the random feature values are averaged and compared to the result of the original instance. This way, the influence of the feature value on the original instance can be calculated. This idea can be combined with counterfactual explanations explained in Section 3.1. The combination of the two methods should be able to explain users how not sharing some of their data would influence the system behavior.

5 Summary

In this work, we presented some ideas for making explanations for AI systems more relevant to users. At first, methods for individualizing explanations were shown that make it possible to adapt explanations to individual users. Afterwards, existing principles of interaction with explanation systems were presented and it was shown how users can interact with personalized explanations. Interaction and individualization are interrelated because users have to interact with a system in order to individualize an explanation. Methods for protecting user data in the explanation process through trusted computing were shown. At the end, an idea on how to minimize the data a user has to share by providing explanations was presented.

References

[1] Amina Adadi and Mohammed Berrada. “Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)”. In: *IEEE access* 6 (2018), pp. 52138–52160.

[2] Nadia Burkart and Marco F Huber. “A survey on the explainability of supervised machine learning”. In: *Journal of Artificial Intelligence Research* 70 (2021), pp. 245–317.

[3] Alberto Carlevaro et al. “Counterfactual Building and Evaluation via eXplainable Support Vector Data Description”. In: *IEEE Access* 10 (2022), pp. 60849–60861.

[4] Shuo Chang, F Maxwell Harper, and Loren Gilbert Terveen. “Crowd-based personalized natural language explanations for recommendations”. In: *Proceedings of the 10th ACM conference on recommender systems*. 2016, pp. 175–182.

[5] Michael Chromik. “reshape: A framework for interactive explanations in xai based on shap”. In: *Proceedings of 18th European Conference on Computer-Supported Cooperative Work*. European Society for Socially Embedded Technologies (EUSSET). 2020.

[6] Michael Chromik and Andreas Butz. “Human-XAI interaction: a review and design principles for explanation user interfaces”. In: *IFIP Conference on Human-Computer Interaction*. Springer. 2021, pp. 619–640.

[7] European Commission. *Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL LAYING DOWN HARMONISED RULES ON ARTIFICIAL INTELLIGENCE (ARTIFICIAL INTELLIGENCE ACT) AND AMENDING CERTAIN UNION LEGISLATIVE ACTS*. URL: `https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52021PC0206`.

[8] European Commission. *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General*

Data Protection Regulation). URL: https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[9] Chihcheng Hsieh, Catarina Moreira, and Chun Ouyang. “Dice4el: interpreting process predictions using a milestone-aware counterfactual approach”. In: *2021 3rd International Conference on Process Mining (ICPM)*. IEEE. 2021, pp. 88–95.

[10] Mark T Keane et al. “If only we had better counterfactual explanations: Five key deficits to rectify in the evaluation of counterfactual xai techniques”. In: *arXiv preprint arXiv:2103.01035* (2021).

[11] Scott M Lundberg and Su-In Lee. “A Unified Approach to Interpreting Model Predictions”. In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 4765–4774. URL: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[12] Tim Miller. “Explanation in artificial intelligence: Insights from the social sciences”. In: *Artificial intelligence* 267 (2019), pp. 1–38.

[13] Christoph Molnar. *Interpretable machine learning. A Guide for Making Black Box Models Explainable*. Lulu. com, 2020. URL: https://christophm.github.io/interpretable-ml-book/.

[14] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. “Explaining machine learning classifiers through diverse counterfactual explanations”. In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 2020, pp. 607–617.

[15] Chelsea M Myers et al. “Revealing neural network bias to non-experts through interactive counterfactual examples”. In: *arXiv:2001.02271* (2020).

[16] Sandra Wachter, Brent Mittelstadt, and Chris Russell. “Counterfactual explanations without opening the black box: Automated decisions and the GDPR”. In: *Harv. JL & Tech.* 31 (2017), p. 841.

Structural Inspection Planning for Mobile Robots

Raphael Hagmanns

Vision and Fusion Laboratory
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT), Germany
raphael.hagmanns@kit.edu

Abstract

In this report we present a pipeline for static coverage planning of known objects, which is an important task in the field of mobile robot based inspection. We analyse the main components of the Structural Inspection Planner [1] and embed an improved implementation into a autonomous flight pipeline for UAVs. Triangle mesh models serve as input for an initial viewpoint sampling. Inspection quality and path length are optimized by formulating the viewpoint sampling as constraint QP. We thoroughly evaluate the ROS-based inspection pipeline on synthetic and real models using a Gazebo simulation. Our experimental evaluation shows that while an efficient inspection trajectory could be generated for most of the tested models, the result is very dependent on regular and well formed input models.

1 Introduction

Automated structural inspection tasks have become increasingly important in the last couple of years. As facilities grow larger, it is difficult to guarantee a continuous and smooth operation without generating a high manual workload. The field of automating those operations is called *Non-Destructive Inspection* (NDI) [9]. Utilizing mobile robots to perform NDIs can keep workers out of

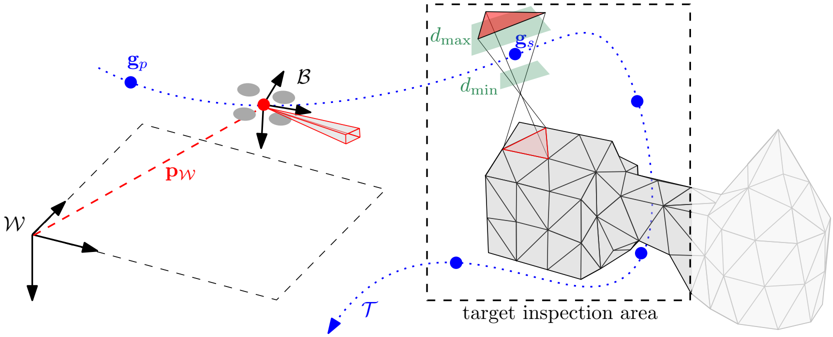


Figure 1.1: Outline of a UAV inspection flight. The drone with a fixed camera is following the inspection trajectory (blue), while tracking its exact position in the world coordinate frame $\mathcal{W}$. Blue points on the trajectory mark viewpoints for specific triangles. The position of the viewpoint g_s is dependent on a set of constraints, for instance the one constructed by the parameters $d_{\min}$ and $d_{\max}$ given in green.

dangerous situations and save a lot of time and resources. Especially with the availability of UAVs (Unmanned Aerial Vehicles), high risk operations such as inspection of buildings, bridges or oversea structures become feasible. Nooralishahi et al. recently provided an in-depth review on the current state of UAVs in NDI [9]. Oftentimes, it is desirable to create an inspection on the basis of an existing model of the respective structure. This allows to quantify the structural damage on the surface compared to the existing model. However, such an approach requires the mobile inspection robot to approach specific viewpoints with a high accuracy in order to inspect the correct target area. This is very difficult to achieve by a manual UAV operator and leads to artifacts in the imagery generated during the inspection flight due to the accumulation of positioning errors. These issues can be dealt with by automatically generating suitable viewpoints and a corresponding inspection trajectory. This way, consistent inspection imagery can be generated in an automated fashion through a number of repetitions.

Therefore, we propose a framework which allows for inspection planning of structures using different kinds of mobile robots, even though we focus on UAVs here. A schematic visualization of the inspection process is given in Figure 1.1. More detailed building blocks of the proposed framework are depicted in Figure 3.1. The inspection planning approach conceptually bases heavily on the Structural Inspection Planner (SIP) [1], which has been developed to calculate inspection trajectories for fixed wing drones as well as UAVs for existing triangle meshes. Our work identifies weaknesses of the SIP and re-implements it with certain modernisations and adoptions. We also abstract the planner component in a way that it can be used inside a larger simulation framework. This allows us to easily perform tests and simulation flights for reconstruction purposes using the framework. Section 2 gives a more detailed overview on the structural inspection planner and other related work, before we present the main structure of our inspection pipeline in Section 3. We apply the planner to different artificial and real models in a Gazebo simulation. This allows us to easily test different UAV, scenario and sensor setups and also account for measurement uncertainties. The qualitative and quantitative evaluation of these experiments is presented in Section 4. Finally, we conclude our work, identify drawbacks and comprehensively describe possible future adoptions in Section 4.

The proposed framework is built within the Robot Operating System (ROS) [11] connecting the different components as visualized in Figure 3.1. It is widespread in the robotic community as it comes with sensor drivers, state of the art simulation frameworks (Gazebo [7]) and lots of prebuilt algorithms for perception and navigation. [1,2] This allows us to abstract the inspection planner into a single node as modular component in a greater UAV stack developed in a previous work [5]. This stack is supported by a Gazebo simulation of a UAV platform running the px4 software stack.[3] Px4 [8] is an open source autopilot running on various drones. It comes with Software-in-the-loop *SiL* and Hardware-in-the-loop *HiL* features which allows ours scenarios to be simulated in a realistic way.

[1] `https://ros.org/`

[2] `https://gazebosim.org/`

[3] `https://px4.io/`

2 Related Work

UAVs are a natural choice for structural inspection as they allow for agile movements in complicated and cluttered environments. This led to extensive research for flight planners in the last couple of years. Oftentimes, the desired goal is to create a reconstruction of a previously unknown environment. In this work, we focus on *model-based inspection*, where we require an accurate mesh of the target. Previous works with these conditions are rare.

The work by Yan et al. uses a multistage approach to generate a coarse reconstruction in a first step and then samples viewpoints for a high quality reconstruction in a second step [15]. Such an approach targets large scale reconstruction as prior knowledge of the target shape is not utilized. Instead, the skeleton is being build with a costly Structure-from-Motion technique. Schmid et al. provide an online informative path planner, where only one inspection flight is required. It uses an RRT*-inspired exploration scheme with object coverage as optimization target. They showed a TSDF-based reconstruction of previously unknown target areas [12, 4].

The Structural Inspection Planner (SIP) [1] is one of the few frameworks which explicitly uses triangle meshes as input to sample a viewpoint trajectory. It samples viewpoints for each triangle in the mesh. Viewpoints have geometrically derived constraints, which are solved as global optimization problem. In a next step, all sampled viewpoints are connected in an efficient way by interpreting the trajectory generation as Traveling Salesman Problem. The steps of viewpoints sampling and trajectory generation via TSP are combined in an iterative fashion until a minimal-length path is found. In practical application however, we found the planner to sample not admissible viewpoints or not converging at all for difficult meshes. The initial viewpoint sampling is highly dependant on the structure of the triangle mesh. Jing et al. [6] also uses an explicit model as input representation. However, they require a voxelized version of the model to first sample a suitable inspection area (via-points) using voxel dilation of the target. Suitable path primitives are then randomly sampled and verified by estimating the target visibility at each point. In a final step, a graph based method is used to generate the final UAV trajectory from the path primitives.

For larger objects, the operation on single voxels can become costly, so that the visibility calculation is not feasible for all admissible via points.

A recent work by Debus and Rodehorst [2] on the inspection of buildings provides evaluation metrics for path planning approaches. The corresponding *Bauhaus Path Planning Challenge* comes with a framework implementing these metrics for a set of models, which will also be targeted in this report. Despite typical evaluation metrics such as *path length* and *runtime*, they also focus on measurable reconstruction quality and surface resolution.

3 Planning Pipeline

In the following, we briefly describe our pipeline architecture before we discuss the inspection planner design in greater detail. The main building blocks of the pipeline are depicted in Figure 3.1. We embed the planner into our UAV framework presented in a previous work [5]. We design the main building blocks “Viewpoint Sampling” and “Trajectory Generation” to be components of a Planner Manager, as this manager also takes care of the sequential control for replanning and avoiding obstacles during the mission. In the original work [1], both blocks of viewpoint sampling and trajectory generation were supposed to run multiple times in an alternating scheme. However, in the experimental evaluation we show that the viewpoint arrangement resulting from the initial sampling iteration is already an intuitive result providing full coverage. Therefore, we mostly apply only one step of sampling and trajectory planning routines. This reduces the overall planning time at the cost of longer trajectories.

3.1 Viewpoint Sampling

We use the same optimization scheme as Bircher et al. in [1], as we iterate through all triangles in the mesh to generate one viewpoint each. A viewpoint needs to fulfill all *intrinsic* and *extrinsic* constraints. The first refers to the visibility of the targeted triangle, while the latter refers to boundary constraints given by the user. Opposing to [1], we increase the flexibility of the optimization problem by allowing an arbitrary number of constraints in the solver. This also

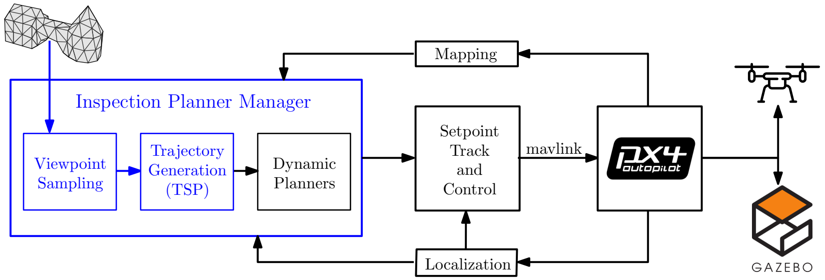

Mapping
Inspection Planner Manager
Viewpoint Sampling
Trajectory Generation (TSP)
Dynamic Planners
Setpoint Track and Control
mavlink
PX4 autopilot
Localization
GAZEBO

Figure 3.1: Structure of the whole simulation pipeline. The relevant blocks for inspection planning are highlighted in blue. The system consists of individual components for localization, mapping and control, which enable to fly the inspection trajectory within a simulated or real environment. The pipeline works with system based on the px4 [8] firmware, which implements the mavlink protocol for communication.

allows to build more generic constraints for polygons instead of triangles. We first calculate the normal a_N of each of the polygons with vertices $\{x_1, \dots, x_n\}$, as well as all edge normals $n_1, \dots, n_n$. In addition, we need to specify the camera parameters horizontal and vertical field of view as well as the pitch. Using these values as input, we build the following set of constraints.

$$\begin{aligned}
\min_{g^k} \quad & (g^k - g_p^{k-1})^T(g^k - g_p^{k-1}) + (g^k - g_s^{k-1})^T(g^k - g_s^{k-1}) \\
& + \alpha_w (g^k - g^{k-1})^T(g^k - g^{k-1}) \\
\text{s.t.} \quad & \underbrace{\begin{bmatrix} \infty \\ \infty \\ \vdots \\ \infty \\ d_{\text{max}} \\ \infty \\ \infty \\ \infty \\ \infty \end{bmatrix}}_{\text{ubA}} \geq \underbrace{\begin{bmatrix} n_1^T \\ n_2^T \\ \vdots \\ n_n^T \\ {a_N}^T \\ n_{\text{right}}^T \\ n_{\text{left}}^T \\ n_{\text{lower}}^T \\ n_{\text{upper}}^T \end{bmatrix}}_{\text{A}} \cdot g^k \geq \underbrace{\begin{bmatrix} n_1^T \cdot x_1 \\ n_2^T \cdot x_2 \\ \vdots \\ n_n^T \cdot x_n \\ d_{\text{min}} + {a_N}^T \cdot m \\ n_{\text{right}}^T \cdot m \\ n_{\text{left}}^T \cdot m \\ n_{\text{lower}}^T \cdot x_{\text{lower}}^{\text{cam}} \\ n_{\text{upper}}^T \cdot x_{\text{upper}}^{\text{cam}} \end{bmatrix}}_{\text{lbA}}
\end{aligned} \tag{3.1}$$

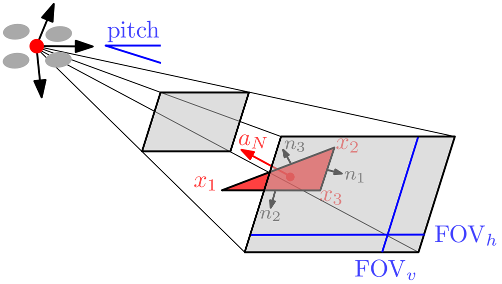


Figure 3.2: Visualization of the camera parameters FOV and pitch as well as the notation for triangle vertices and normals.

The left and right parts of the above equation ubA $= \{x_{min}, y_{min}, z_{min}\}$ and lbA $= \{x_{max}, y_{max}, z_{max}\}$ quantify the admissible viewpoint sampling space. The first n constraints force the viewpoint g^k to be sampled “in front” of the triangle, as the projection of the hyperplane normal $n_i^\top$ of the respective edge onto the vector spanned by the viewpoint $(g^k - x_i)^\top$ is required to be positive. The next constraint forces the distance of the viewpoint to be in $[d_{\text{min}}, d_{\text{max}}]$ by restraining projected distance of the triangle normal a_N. Finally, the last four constraints are exactly the *field of view* constraints from [1]. They guarantee the viewpoint to lie inside the horizontal and vertical FOV of a camera with a specific pitch. To accomplish this, four hyperplane normals n_{upper}, n_{lower}, n_{left}, n_{right} with respective anchor points $x_{<\cdot>}$ are sampled using the camera pitch and FOV. A more detailed derivation of these constrains can be found in the original work of SIP [1].

The optimization objective is to minimize the distance between two consecutive viewpoints g_p and g_s, which optimizes the total path length. In the original formulation from [1], the viewpoint sampling was meant to run multiple iterations, such that the squared distance between g_p^k, g^k and g_s^k and their previous iteration $k - 1$ is minimized. The quadratic problem is then solved using qpOASES [3], resulting in a optimal position g_{opt} for the current iteration k. In a next step, the rotation is sampled by performing an explicit visibility analysis. In a simple UAV scenario, pitch and roll of the rotorcraft are always fixed, while the yaw angle γ is subject to change. For the sampled position and all possible yaw angles in $[0, 2\pi]$ with a step size of 0.2rad we check if (a) all distance constraints are met, (b) all vertices lie withing the FOV of the camera, and (c) there are no collisions on the way from camera to the polygon. The collision check can be invoked by performing a simple raycast which allows it to consider other static obstacles in the scene. The first position and orientation pair which passes all requirements, is selected as viewpoint.

3.2 Trajectory Planning

Given the set of N viewpoints $\{g_1, g_2, \ldots, g_N\}$, one for each triangle, we now connect them into a single shortest path trajectory $\mathcal{T}_{\text{opt}}$. Connecting such as set of "must visit"-points is a typical application for the *Traveling Salesman Problem* (TSP). Each viewpoint must be visited exactly once while optimizing the overall path length. Even though TSP it is a NP-hard problem, efficient solvers exist for the comparatively small number of viewpoints. We use a TSP-solver developed as part of the *Google OR-tools* [10]. It allows to use a custom distance matrix between all viewpoints as input. This allows it to implicitly embed more metrics such as the change of yaw or inspection angle into the optimization. However, in the current version we simply use an Euclidean distance matrix in order to optimize for path length as primary objective. The TSP is then solved using the guided local search heuristic which is considered one of the most efficient sampling heuristics for routing problems.

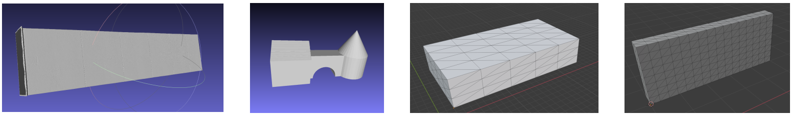
4 Experimental Evaluation

The utilized planning framework allows to use the px4 *Software-In-The-Loop* component to run experiments with Gazebo as simulator. We tested the planning procedure on a number of different models. Figure 4.1 shows some of them, either taken from the Bauhaus Challenge [2], coming with the SIP-implementation [1] or created from real-world objects on our premises. Even if Gazebo is not capable of rendering the environment in a photorealistic way, it allows to test

	Parameter Name	Units	Default	Explanation
1	**Planner**			
	$d_{\text{min}}, d_{\text{max}}$	meter	$[6.0, 10.0]$	distance constraints
	Min. incidence angle	degree	$10°$	$\angle(a_n, n_{<\cdot>})$ c.f. Fig. 3.2
2	**Rotorcraft**			
	Max. velocity	m/s	2	
	Max. angular velocity	rad/s	0.5	
3	**Space boundary**			
	Max. space size	meter	$[200, 200, 50]$	x, y and z size
	Space center	meter	$[0, 0, 0]$	3D coordinate [x, y, z]
6	**Camera**			
	FOV	degree	$[120°, 120°]$	[horizontal, vertical] field of view
	Pitch	degree	$30°$	Pitch angle of the camera

Table 4.1: List of the most important parameters and their respective default values within the inspection framework.

results of the view-point sampling using different sensor setups and environment data.



(a) Bridge Pier

(b) House

(c) Hall

(d) Wall Model

Figure 4.1: Exemplary triangle mesh models used for experimental evaluation. The first two were taken from the Bauhaus Path Planning Challenge [2], while the last two have been created for simple ablation studies.

The framework has various parameters, which heavily influence the experimental results. We tried to use similar defaults as in [1]. Table 4.1 gives an overview over the most important parameters and their default values.

We quantitatively evaluate the planners in different configurations on a set of standard metrics. We leverage the *planning time*, *path length* and *mean yaw*

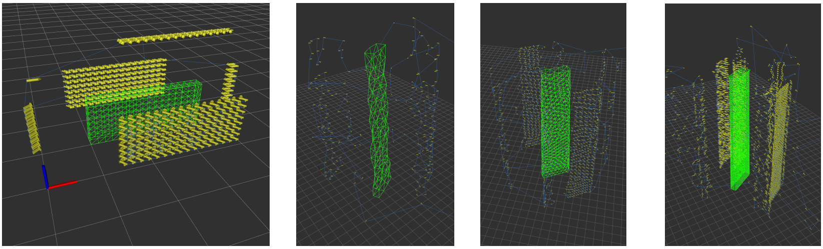
Model	Mesh Size facets	Path Length $[m]$	Mean Yaw Rate $[^{\circ}]$	Planning Duration $[s]$
Hall	148	216	14.36	2.60
Artificial House	154	152	14.23	2.51
Bridge Pier	186	408	21.17	2.63
	965	441	11.09	8.85
	3930	2011	0.28	262.02

Table 4.2: Quantitative evaluation on the three main models hall, house and bridge pier. Mesh size denotes the number of triangles and planning duration the total planning time for all steps.

rate as main metrics. We also verify the number of *rejected* triangles. The path length is specified as $\mathcal{L} = \sum_{i=0}^{N-1} d_{i \to i+1}$ where N is the total number of viewpoints and $d_{i \to i+1}$ is the distance between the i^{th} viewpoint and the subsequent one. The mean yaw rate $\overline{\Delta\psi} = \frac{1}{N} \sum_{i=0}^{N-1} \|\Delta\psi_{i \to i+1}\|$ specifies the mean change in yaw angle over time, with $\Delta\psi_{i \to i+1}$ being the change in yaw between two consecutive viewpoints.

Table 4.2 shows the results for some of the models in different resolutions. All metrics are dependant on the number of triangles in the mesh. The main cause for an increasing path length are outliers in the viewpoint sampling, while the increased planning time results mostly from the exponential increase in TSP solving time. The decrease in yaw rate simply follows from the fact that the many viewpoints are interpolations of viewpoints from the lower resoluted mesh and thus not contributing to any turns of the UAV.

We qualitatively inspect the generated inspection trajectory on some of the models in Figure 4.2. For the simplest wall model in Figure 4.2(a) the viewpoint generation works as expected when running one iteration of viewpoint sampling. The remaining images in Figure 4.2 show the bridge pier in different resolutions. Even though the generated flight plan looks regular in general, more outlier viewpoints are generated for the higher resoluted meshes. The reason for this is typically the result from non optimal QP outputs for some difficultly placed



(a) Wall

(b) Bridge Pier 186 facets

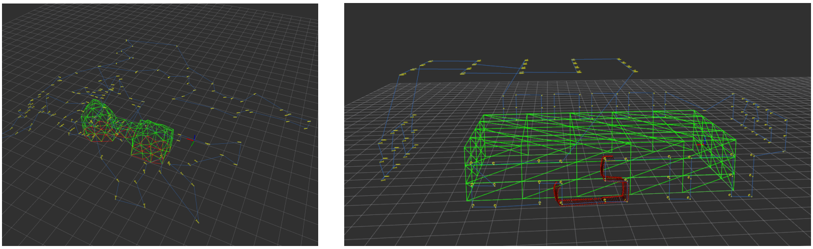
(c) Bridge Pier 965 facets

(d) Bridge Pier 3930 facets

Figure 4.2: Generated paths for the wall model and for different resolutions of the bridge model. The light green lines are the input mesh, blue the trajectory and the yellow arrows denote the viewpoints and their direction.

triangles. The solver is configured in a way, that some constraints may be relaxed, if a global optimum cannot be found. One approach to prevent such outliers from being sampled is to restrict the d_{max} parameter for the admissible sampling space. In addition, it could be considered to not sample one viewpoint per triangle but to combine multiple similar viewpoints in a later step to reduce the overall path length. This could also allow to filter sampled outlier viewpoints, which could lead to a smoother trajectory than in Figure 4.2(d). We observe a similar behaviour for the models in Figure 4.3. We use red to indicate triangles for which no viewpoints could be generated. As the camera pitch is fixed and the UAV cannot fly below the ground, all ground triangles are marked red in Figure 4.3(a) and therefore not participating in the trajectory planning. Figure 4.3(b) shows the inspection path on the simple hall model and additionally marks the UAV odometry from a simulated flight in red.

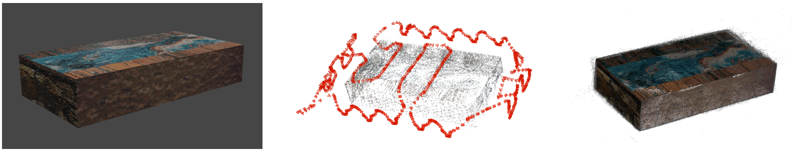
Despite of sampling some outlier viewpoints, we can observe a successful coverage for all tested models. This can also be verified by performing a reconstruction using the recorded images from the simulation. Figure 4.4 shows the result of this procedure on the Hall model. We post-processed the images with colmap [13] to obtain the sparse and dense reconstruction results. Note that the inspection planning itself does not target 3D-reconstruction applications in particular. We do not ensure within the viewpoint generation that a triangle



(a) House (b) Hall

Figure 4.3: Generated paths for the artificial house and the hall. In addition to the trajectory we mark rejected triangles in red.

must be observed from two distinct positions. Nevertheless, the generated paths often allow a dense reconstruction as two consecutive viewpoints oftentimes target neighboring triangles on the mesh, resulting in a stereo baseline for both triangles.

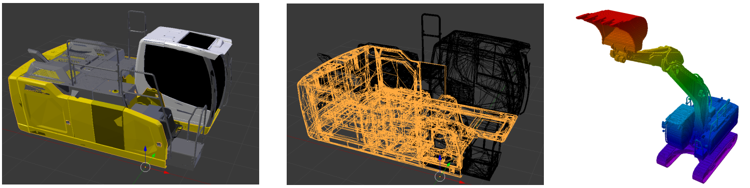


(a) Textured Hall Model (b) Generated Inspection Plan (c) Reconstructed Model

Figure 4.4: Application of the inspection planning for reconstruction purposes. The first image (a) shows a textured version of the model from Figure 4.1(c). We then use the generated inspection plan to fly it in simulation and perform a reconstruction using the saved images (b). Finally, (c) shows the dense reconstruction.

5 Conclusion and Future Work

In this work, we presented and evaluated a structural inspection pipeline for mobile robots using triangle meshes as input. We showed that intuitive inspection trajectories could be generated for a set of different models. In order to thoroughly test the inspection, we embedded it into a UAV autonomy pipeline with simulation capabilities. Using that simulation, we also applied the routine for the purpose of 3D-reconstruction. We identify the availability and variability of input data as main drawback in the presented approach. Given an arbitrary input mesh, it is a very error-prone pre-processing step to transform it into a regular triangle mesh with a desired amount of facelets. On the other side, the number of facelets is the only parameter to control the initial number of sampled viewpoints and thus the runtime required for the initial sampling step. Some approaches, such as ACVD [4] resulting from [14] exist to simplify existing meshes but as soon as the geometries get complex and non-convex, we were not able to produce a regular mesh (see 5.1(b)).



(a) Excavator (b) Upper Carriage Mesh (c) Voxelized Object

Figure 5.1: Example for an excavator model, which is difficult to tackle using a triangle mesh based inspection scheme. Using a dynamically generated voxelized version of the model such as in (c) might improve the viewpoint sampling and also allows resampling for different joint positions.

One way to overcome these limitations in the future is the usage of a different input modality. For instance, one could use a voxelized structure of the mesh such as visualized in 5.1(c), which is comparatively easy to generate even for dynamic joint positions. This approach would require a strategy to divide the voxel structure into different regions as the workload for sampling one viewpoint per voxel would be too high. This also raises the question, if the formulation as QP is even necessary if we only run one iteration of sampling and planning. We can influence the resulting inspection trajectory either by running multiple

[4] `https://github.com/valette/ACVD`

iterations or by adjusting the input modality in a way that less viewpoints are sampled in the first place.

Further extensions to the current approach are conceivable. It could be useful to explicitly encode inspection or reconstruction quality into the optimization. The first would require some dynamically generated distance constraints in order to achieve a user-definable ground sampling distance (GSD). The ladder requires that a triangle can be inspected from at least two viewpoints, so ideally each viewpoint must encode the constraints for multiple triangles.

6 Acknowledgments

A majority of this work is based on the implementation done by Shreyas Manjurnath during his Master Thesis, which has been carried out at Fraunhofer IOSB in 2021 and was supervised by Boitumelo Ruf and Raphael Hagmanns. Therefore, most ideas for the extensions presented in this work have been worked out in collaborative think sessions.

References

[1] A. Bircher et al. “Structural Inspection Path Planning via Iterative Viewpoint Resampling with Application to Aerial Robotics”. In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. May 2015, pp. 6423–6430.

[2] P. Debus and V. Rodehorst. “Evaluation of 3D UAS Flight Path Planning Algorithms”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLIII-B1-2021 (2021), pp. 157–164. DOI: 10.5194/isprs-archives-XLIII-B1-2021-157-2021.

[3] Hans Joachim Ferreau et al. “qpOASES: a parametric active-set algorithm for quadratic programming”. In: *Math. Program. Comput.* 6.4 (2014), pp. 327–363. DOI: 10.1007/s12532-014-0071-1.

[4] Jonathan D. Gammell, Siddhartha S. Srinivasa, and Timothy D. Barfoot. “Informed RRT*: Optimal Incremental Path Planning Focused through an Admissible Ellipsoidal Heuristic”. In: *CoRR* abs/1404.2334 (2014). arXiv: `1404.2334`.

[5] Raphael Hagmanns. “Dynamic Planning Pipeline for Indoor Inspection Flights”. In: *Proceedings of the 2021 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*. 2021, pp. 87–103. ISBN: 978-3-7315-1171-7.

[6] Wei Jing et al. “Coverage Path Planning using Path Primitive Sampling and Primitive Coverage Graph for Visual Inspection”. In: Nov. 2019, pp. 1472–1479. DOI: `10.1109/IROS40897.2019.8967969`.

[7] Nathan Koenig and Andrew Howard. “Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Sendai, Japan, Sept. 2004, pp. 2149–2154.

[8] Lorenz Meier, Dominik Honegger, and Marc Pollefeys. “PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms”. In: *Proceedings - IEEE International Conference on Robotics and Automation* 2015 (June 2015), pp. 6235–6240. DOI: `10.1109/ICRA.2015.7140074`.

[9] Parham Nooralishahi et al. “Drone-Based Non-Destructive Inspection of Industrial Sites: A Review and Case Studies”. In: *Drones* 5.4 (2021). ISSN: 2504-446X. DOI: `10.3390/drones5040106`.

[10] Laurent Perron and Vincent Furnon. *OR-Tools*. Version 7.2. Google, July 2019. URL: `https://developers.google.com/optimization/`.

[11] Morgan Quigley et al. “ROS: an open-source Robot Operating System”. In: vol. 3. Jan. 2009.

[12] L. Schmid et al. “An Efficient Sampling-Based Method for Online Informative Path Planning in Unknown Environments”. In: *IEEE Robotics and Automation Letters* 5.2 (Apr. 2020), pp. 1500–1507. ISSN: 2377-3774. DOI: `10.1109/LRA.2020.2969191`.

[13] Johannes Lutz Schönberger and Jan-Michael Frahm. “Structure-from-Motion Revisited”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[14] Sebastien Valette, Jean-Marc Chassery, and Remy Prost. “Generic Remeshing of 3D Triangular Meshes with Metric-Dependent Discrete Voronoi Diagrams”. In: *IEEE Trans. Vis. Comput. Graph.* 14 (Mar. 2008), pp. 369–381. DOI: 10.1109/TVCG.2007.70430.

[15] Feihu Yan et al. “Sampling-Based Path Planning for High-Quality Aerial 3D Reconstruction of Urban Scenes”. In: *Remote Sensing* 13.5 (2021). ISSN: 2072-4292. DOI: 10.3390/rs13050989.

Measuring the Effects of Catastrophic Forgetting in Neural Networks

Tobias Kalb

Porsche Engineering Services GmbH, Germany
Tobias.Kalb@porsche-engineering.de

Technical Report IES-2022-11

Abstract

When developing versatile machine learning systems, catastrophic forgetting poses a significant challenge, by which models trained on tasks sequentially suffer significant performance drops when put to use on earlier tasks. In spite of the prevalence of catastrophic forgetting, its underlying cause and process are still poorly understood. Commonly, the performance of a continual learning algorithm is only measured using accuracy on the test set of the tasks within a sequence. While test set accuracy is useful for comparing different continual learning algorithms on their respective benchmarks, they cannot provide insights into how and where the model is affected by catastrophic forgetting, as they only provide final accuracy metrics. Therefore, we study how comparing representations, re-training schemes and layer stitching can help to reveal effects and causes of catastrophic forgetting.

1 Introduction

A desired property of many machine learning models exposed to a changing environment is the ability to progressively acquire new knowledge without negatively interfering with previously learned knowledge. A major challenge in achieving this goal is to overcome catastrophic forgetting, where the model forgets knowledge learned from previous tasks while learning a new task [8, 16]. This is especially relevant in constantly changing environments, like automated driving, in which a model for semantic scene parsing has to adapt to new unseen objects, e.g., e-scooters, or different driving situations or new environments e.g. different countries or adverse weather conditions. Continual learning is a rapidly evolving field, aiming to overcome the limitations of catastrophic forgetting. Continual learning algorithms often attempt to overcome specific known causes of catastrophic forgetting like weight drift, activation drift, inter-task confusion and task the recency bias [15, 13]. However, the performance of a continual learning algorithm is mostly measured using accuracy on the test set of the tasks within a sequence. While test set accuracy is useful for comparing different continual learning algorithms on their respective benchmarks, they cannot provide insights into how and where the model is affected by catastrophic forgetting, as they only provide final accuracy metrics. Therefore, the goal of this work is to demonstrate how causes and effects of catastrophic forgetting can be revealed with existing methods that are used to measure the representational similarity, weight distance and the inter-task confusion of a continually trained model. Additionally, we identify the limitations of the approaches that should be taken into account when using them.

2 Related Work

As the previously mentioned underlying effects of catastrophic forgetting cannot be measured exclusively by the accuracy achieved on the test sets, several methods were proposed to gain additional insights of the causes and effects of catastrophic forgetting. Mirzadeh et al. [17] use Linear Mode Connectivity to show that multi-task minima are connected to continual learning minima by a linear path of low error on their respective tasks, while the individual single-task optima are not similarly connected. More recent work uses Linear Probing [5] to investigate representational forgetting, which measures the difference in accuracy a optimal linear classifier achieves before and after introducing a new task. They confirm the notion that the observed test accuracy of continual learning algorithms only allow restrictive insights into the model and that some methods perform better mitigating the effects of forgetting than the test accuracy indicates. Similarly, for semantic segmentation Decoder Re-Training is used to gain comparable insights and to measure the impact of inter-task confusion in class-incremental learning [10]. Central Kernel Alignment (CKA) introduced by Kornblith measures the representational similarity of neural architectures after training. Recently it has also been applied in continual learning to measure the shift of representations for previous tasks after training on a new task [5, 20]. Furthermore, the Dr. Frankenstein toolset proposed Csiszárik et al. [4], which measures the functional similarity of representations, was used to identify the causes of forgetting in class-incremental semantic segmentation [10]. Finally, ongoing research proposes several interpretability methods for deep learning models that help to explain why a model made a particular prediction [2]. These methods can also be utilized in the setting of continual learning, but are not discussed in our work. As there is an apparent lack of work comparing the different methods, in this work we aim to evaluate the given methods on similar tasks to understand how they can complement each other.

3 Preliminaries

3.1 Effects of Catastrophic Forgetting

A machine learning task $T = \{(x_m, y_m)\}_{m=1}^{M}$ consists of a set of M inputs $x \in \mathcal{X}$ and corresponding labels $y \in \mathcal{Y}$. In classical machine learning the parameters of a model f_w are optimized by minimizing the negative gradient of the empirical risk g over T w.r.t to a loss function $\mathcal{L}$. Most commonly g is approximated by calculating the stochastic gradient $\tilde{g}$ on a mini batch $T' \subset T$ with p_i as the sampling probability for a training sample.

$$\mathbb{E}\left[\, \tilde{g} \,\right] = \sum_{(\mathbf{x}_i, y_i) \in T'} p_i \, \nabla \mathcal{L}\left(f_w(\mathbf{x}_i), y_i\right) \tag{3.1}$$

If p_i is uniform for all training samples, the expectation of $\tilde{g}$ is equal to g. However, in continual learning p_i is not uniform, as the model f is sequentially optimized on a sequence of tasks $T_{1..K}$. So that p_i is only uniformly distributed over samples belonging to the current task T_k. However, the goal is to minimize the empirical risk g over the entire sequence $T_{1..K}$. Thus, when optimizing on data of task T_k, optimization disregards other task distributions and the model is optimized without regard of previous data. Therefore, the sample distribution for all samples of task is $\{T_t | t \neq k\}$ is $p_i = 0$, which leads to catastrophic forgetting.

$$\begin{aligned}\mathbb{E}\,[\,\tilde{g}\,] &= \sum_{t} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{T}_t} p_i \, \nabla \mathcal{L}\,(f_w(\mathbf{x}_i), y_i) \\ &= \sum_{(\mathbf{x}_i, y_i) \in \mathcal{T}_c} p_i \, \nabla \mathcal{L}\,(f_w(\mathbf{x}_i), y_i)\end{aligned} \tag{3.2}$$

There are four main effects how catastrophic forgetting manifests itself in incremental learning [15].

- **Weight Drift:** During optimization on T_k, the weights of the model that were relevant to the previous task T_{k-1} are updated without regard to the previous task, resulting in drop of performance on task T_{k-1}.
- **Activation Drift:** A change of the weights of the model directly results in a change of internal activations and to the output of the model. While activation drift is a direct result of the weight drift, activation drift additionally also takes the input data distribution into account.
- **Inter-task confusion:** The objective in class-incremental learning is to correctly discriminate between all the observed classes. However, as the classes are never jointly trained, the learned features are not optimized to discriminate classes from different tasks, as shown in Figure 3.1. Related to inter-task confusion are task-specific spurious features that can also arise in domain incremental learning [13].
- **Task-recency bias:** In the class-incremental setting, the model is optimized to predict new classes without regarding the old classes. This leads to a strong bias for the most recently learned classes, especially in the classification layers of the models.

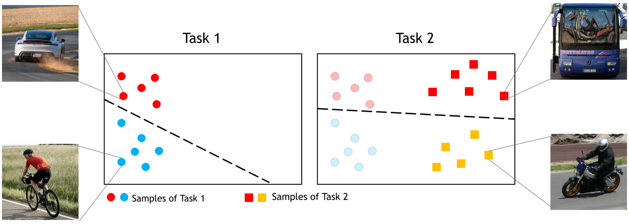


Figure 3.1: Visualization of task confusion in class-incremental learning. As classes of Task 1 (car and bicycle) and classes of Task 2 (bus and motorcycle) are never trained at the same time, the classifier never learns to discriminate between bicycle and motorcycles, which causes inter-task confusion.

3.2 Notation

A training task $T = \{(x_m, y_m)\}_{m=1}^{M}$ consists of a set of M images $x \in \mathcal{X}$ with $\mathcal{X} = \mathbb{R}^{H \times W \times 3}$ and corresponding labels $y \in \mathcal{Y}$. Given the task T an artificial neural network learns a mapping function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that maps the input space to the output space. The neural network consists of N consecutive layers so that $f = g_N \circ \ldots \circ g_1$, where $g_n : \mathcal{A}_{n-1} \rightarrow \mathcal{A}_n$ are mappings between the activation spaces $\mathcal{A}_{n-1}$ and $\mathcal{A}_n$ with $\mathcal{A}_0 = \mathcal{X}$. In continual learning f is not trained on a single task T but on a sequence of tasks T_t. We denote the neural network that was successively trained on t tasks as $f_t = g_{t,N} \circ \ldots \circ g_{t,1}$ with corresponding activations $\mathcal{A}_{t,n}$. Our goal is to evaluate methods that measure the activation drift between $\mathcal{A}_{t,n}$ and $\mathcal{A}_{t-1,n}$ or the weight drift between the parameters of $g_{t,n}$ and $g_{t-1,n}$, which layer n is subjected to during continual learning. Furthermore, we define two incremental settings that define how subsequent task expand the first learned task:

- Class-incremental learning, in which each new task extends the existing set of classes by a set of novel classes.
- Domain-incremental learning, in which the classes stay the same, but the images of each task are obtained from a different distributions and therefore have distinct visual appearance.

4 Experiments

Our study compares different methods to measure catastrophic forgetting in a class- and domain-incremental setup, because the effects of catastrophic forgetting differ vastly between these setups. In the domain-incremental setting, we train the model incrementally on Cityscapes [3] (CS) and then on the ACDC-*Night* [22]subset. ACDC and CS are both large-scale datasets for semantic understanding of urban street scenes for autonomous driving and share a common 19 class labeling policy, so that the increment is purely the change from day (CS) to night images (ACDC). For the class-incremental setting, we use the commonly used PascalVOC [7] dataset with a 15-5 split. The PascalVOC-15-5 split is a two step incremental learning sequence, which consists of learning 15 classes (1–15) in the first step T_0 and the remaining 5 classes (16–20) in the second step T_1. We evaluate all methods on the ERFNet [21] architecture and compare different methods to train the models in an continual manner, namely: Fine-Tuning (FT), the prior-regularization method EWC [11] and Replay.

5 Activation Drift

Methods in this section measure the activation drift between a model f_0 and f_1, where the model f_0 is trained on T_0 andf_1 is initialized with the parameters of f_0 and incrementally trained on T_1. This is done on a layer-wise manner so that the activations $\mathcal{A}_{0,n}$ and $\mathcal{A}_{1,n}$ of layer n of the models f_0 and f_1 are compared. The current key methods to measure activation drift in neural networks are Centered-Kernel Alignment (CKA) [12] and Layer Matching [10, 4]. In this

Method	**Class-Incremental**			**Domain-Incremental**		
	0-15	*15-21*	*Forgetting*	*Cityscapes*	*Night*	*Forgetting*
Fine-Tuning	4.6	23.0	9.0	36.4	39.1	31.9
EWC [11]	28.1	10.1	23.8	40.2	27.8	28.1
Replay	42.2	29.1	39.1	58.2	40.4	10.1

Table 4.1: Comparison of EWC, Replay and Fine-Tuning in the class- and domain-incremental learning scenarios. Evaluation is run after training on the entire task sequence.

section we discuss how these methods measure activation drift in continual learning and what the difference between those methods are.

5.1 Layer Matching with Dr. Frankenstein

The Dr. Frankenstein toolset aims to analyze the similarity of representations in deep neural networks, by matching the activations of two networks on a given layer by joining them with a stitching layer [4]. The goal of the stitching layer is to transform the activations of a specific layer of f_0 to the corresponding activations of a model f_1. We measure the similarity of the learned representations by comparing the initial accuracy of the model f_0 with that of the resulting Frankenstein Network. The higher the resulting relative accuracy is, the closer the learned representations of the models are to each other. Previous work in continual learning omits the stitching layer and directly uses the activations of f_0 in f_1, as the models are closely related, because f_1 is initialized with the parameters of f_0 [10]. The setups are displayed in Figure 5.1. If the accuracy of the resulting Frankenstein network is not adversely affected, this is clear evidence that the internal representations of f_1 were not altered drastically during training on T_1. This analysis will give insights into how much the activation at a specific layer has changed after incremental training.

Results: The layer-wise activation drift measured with layer stitching for the incremental learning scenarios is displayed Figure **??**. It is apparent that in the class-incremental scenario (Pascal-15-5) the encoder layers up until layer 8 are not at all affected by activation drift and only later encoder layers or specific layers in the decoder show significant representation shift, as already pointed out in recent work [10, 5, 20]. However, in the domain-incremental learning setting we see that primarily the first layers are affected by activation drift and later layers only change slightly.

Limitations: A limitation of this approach is that it cannot measure positive backward transfer without the additional stitching layer, in which the model would learn a new improved representation for old data while learning a new

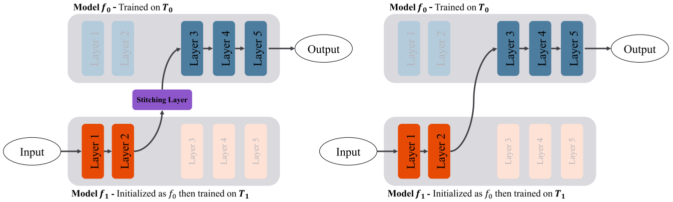


Figure 5.1: Comparing the original Dr. Frankenstein layer matching [4] (left) with the approach without additional stitching layer [10] (right).

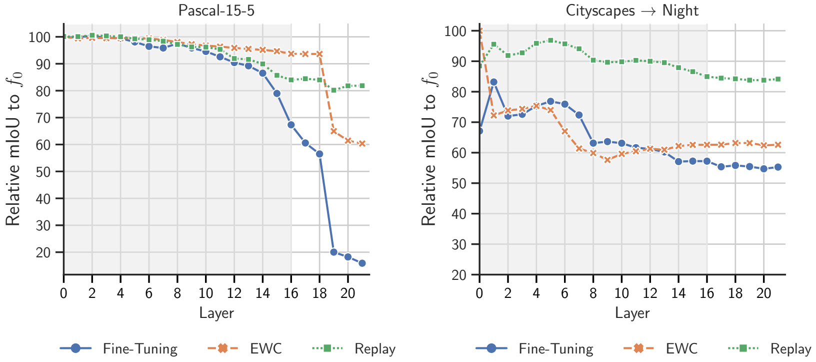


Figure 5.2: Activation drift between f_1 to f_0 measured by relative mIoU on the first task of the Frankenstein Networks stitched together at specific layers (horizontal axis). The layers of the encoder are layer 0–16 (grey area), the decoder layers are 17–20 (white area). In the class-incremental Pascal-15-5 setting the activations the early layers of the encoder stay very stable for all methods, only EWC and Fine-Tuning have a severe drift in activations in the decoder layers of the network. In the domain-incremental learning setting only the first layers (0–8) are affected by activation drift and layers 8–20 layers only change slightly.

task. This could occur when a feature that was discriminative for T_0 is replaced with a feature that is more useful for discriminating all classes. In that case f_0 could no longer extract useful information from the stitched representations of f_1, which would lead to a performance drop although these representations are still useful for f_1 to classify all classes.

5.2 Centered Kernel Alignment (CKA)

CKA [12] is a similarity index that measures the similarity between internal representation of neural networks. Given $|T|$ samples and their corresponding matrices of activations $A_n \in \mathbb{R}^{|T| \times p}$ and $B_n \in \mathbb{R}^{|T| \times p}$ of p neurons at a specific layer n of two neural networks f_a and f_b linear CKA is defined as:

$$CKA(A_n, B_n) = \frac{||A_n^T B_n||_F^2}{||B_n^T B||_F ||A_n^T A_n||_F} \tag{5.1}$$

$|| \cdot ||_F$ denotes the Frobenius norm. Linear CKA has recently been used to compare the intermediate representations of models f_0 and f_1 in continual learning [5, 20], in which a high CKA score equates to lower representational forgetting. Csiszárik et al. [4] investigated the relationship between representational similarity that is measured by CKA and functional similarity measured by Dr. Frankenstein. In this case functional similarity means that the representation lead to the similar output of the model, whereas representational similarity is directly measuring the distance between representations. They demonstrate that a network can retain high functional similarity using Dr. Frankenstein while simultaneously decreasing the similarity index measured by CKA. In other words they can change the representations of a layer while the output of the entire network is not affected.

Results: In Figure 5.3 we compare layer stitching results with the similarity measured by CKA. For the encoder layers we observe very similar layers, where we only see a mild shift in representations. However, in the decoder layers we see vast differences, as EWC retains much higher representational similarity thane Replay or FT. This seemingly contradicts the results in Table 4.1, where EWC shows significantly more forgetting than Replay. In combination with the layer stitching plots this could also indicate that small representational changes in the decoder lead to significant functional changes in the output.

Limitations: Similar to layer stitching CKA is also not able to measure positive backward transfer, so that one needs to be aware that not every drift

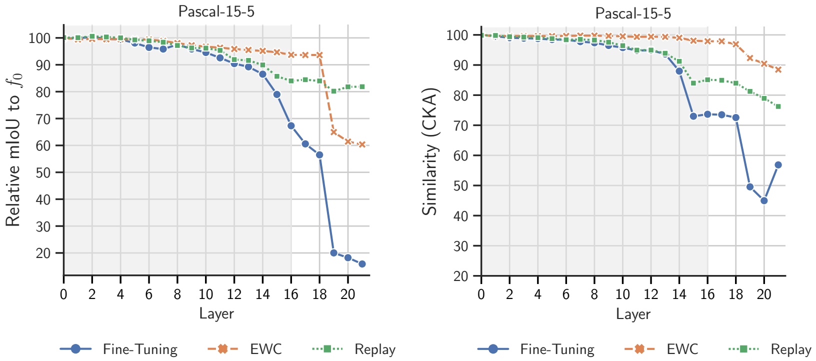


Figure 5.3: Activation drift between f_1 to f_0 measured by CKA [12] (right) and layer stitching [10] (left) at specific layers (horizontal axis).

in activation has a adverse effect on the performance of the previous task. The results show that higher representational similarity does not directly indicate better performance on the previous task.

6 Re-Training and Re-Estimation

Re-Training and Re-Estimation methods try to freeze specific layers of the network and re-train the remaining layers on all datasets, to show how useful the features of the frozen layers are to solve the joint task. Decoder Re-Training and Linear Probing freeze the backbone of the model and re-train the classification layer or the decoder of the network, to estimate how discriminate the features of the backbones are. Partial Retraining Accuracy, on the other hand, measures forgetting for single layers of the network while the remaining are trained from scratch. Finally, Batch Normalization Re-Estimation is used to measure the contribution of the changing Batch Normalization population statistics to catastrophic forgetting.

6.1 Batch Normalization Re-Estimation

A major contributor to the activation drift of a model trained incrementally are the changing population mean and variance of the Batch Normalization (BN) layers, which are collected during training to achieve a deterministic behavior for inference [9]. While this works for i.i.d.[1] data, in the non-i.i.d incremental learning setting the BN estimates of the population mean and variance are heavily biased towards the most recent task, leading to a significant drop in accuracy on old tasks [14]. A straightforward method to measure the impact of changing BN statistics is to re-estimate them on the joint dataset. This can be achieved by simply doing a forward pass over the entire joint dataset, without the backward pass.

Results: Table 6.1 shows the respective re-estimation results for the domain- and class-incremental experiments. By comparing the ΔmIoU$_{BN}$ we see that the changing BN statistics have much more significant impact in domain-incremental learning. Furthermore, in the domain-incremental setting replay alleviates the change of BN statistics completely as re-estimation even slightly decreases the mIoU. Therefore, we conclude that changing BN statistics are a significant contributor to forgetting in the domain-incremental setting and that BN re-estimation can be an important tool to reveal this effect.

Limitations: BN Re-estimation can only give a measure on which BN layers are affected by the changing BN population statistics, but allows no insights into the direct causes of the change. However, it can be vital to understand how a continual learning algorithm is affecting the BN statistics, e.g. Replay stabilizes population statistics using the Replay buffer. Finally, it should be noted that this method is not applicable to the recent Vision Transformer [6] architectures as they use Layer Normalization [1] instead of BN.

[1] independent and identically distributed

Method	**Class-Incremental**		**Domain-Incremental**	
	mIoU$_{BN}$	ΔmIoU$_{BN}$	mIoU$_{BN}$	ΔmIoU$_{BN}$
Fine-Tuning	4.4	0.1	47.0	10.6
EWC [11]	36.7	4.1	57.6	17.4
Replay	46.3	0.8	57.9	-0.3

Table 6.1: Performance in mIoU [%] of the adapted model f_1 after re-estimating the population statistics of all BN layers. By measuring and comparing the increase after re-estimating BN statistics (ΔmIoU$_{BN}$), we see that in class-incremental learning re-estimating BN statistics leads to a less significant increase compared to the domain-incremental setting.

6.2 Partial Retraining Accuracy (PRA)

Murata, Toyota, and Ohara [18] measure representational forgetting of a specific layer $g_{t,i}$ with *Partial Retrain Accuracy* (PRA), which is the accuracy that can be gained after freezing $g_{t,i}$ while the remaining part of the model is re-initialized and re-trained on all data from the previous tasks. After that they re-order the sequence in which the tasks are learned, to prevent the effect the task order has on the learned representations. Using this method they show non-negligible amount of forgetting is already happening at shallow layers.

Limitations: The validity of this method is questionable, because the majority of the network is re-trained on the joint data of the model so that activation drift of intermediate layers can potentially be rectified by following layers as they are trained on the joint task. E.g. when freezing only the very first block of a network in the domain-incremental setting the remaining layers will amend the activation drift of the first layer, although the very first layers are known to be causes of severe forgetting in this setting. So while the aforementioned approaches that directly compare the activations are not able distinguish whether the model has learned a new representation for old data or if the previous representation has been overwritten, PRA can falsely lead to the conclusion that a new representation has been learned due to re-training.

6.3 Decoder Retrain Accuracy and Linear Probing

Decoder Retraining [10] and Linear Probing [5] aim to measure representational forgetting by calculating the difference in accuracy an optimal classifier layer achieves on an old task before and after introducing a new task. Since the methods are similar except that Decoder Retraining is intended for semantic segmentation and Linear Probing for classification, we only consider Decoder Retraining in this section. To measure the Decoder Retraining Accuracy, the encoder of the model is frozen while the decoder is re-trained on all classes with the same training configuration and subsequently evaluated on all tasks. While the gain in mIoU ΔmIoU$_R$ gives a measure on how much the decoder is contributing to forgetting, the mIoU$_R$ shows how useful the learned representations are to discriminate between classes of different tasks. Linear Probing and Decoder Re-Training both have been used to show that continual learning methods that seem not effective in the class-incremental setting such as EWC, are in fact able to stabilize internal representations and that only a few final layers are the main contributor to deteriorating performance on the old task [10, 5].

Limitations: Decoder Retrain Accuracy and Linear Probing are aimed at differentiating between representational forgetting in the encoder and forgetting in the the decoder. They indicate how discriminative the features of the backbone are to distinguish all observed classes. However, they cannot give further insights into which layers are affected. Furthermore, it is not as useful in the domain-incremental setting, because the forgetting is mainly affecting early layers.

7 Weight Drift

Instead of measuring activation drift, it is also possible to measure the changes of the model from f_0 to f_1 simply by calculating the ℓ_2 distance of the models normalized parameters $\frac{\theta_0}{||\theta_0||_2}$ to $\frac{\theta_1}{||\theta_1||_2}$ [19]. Furthermore, to see how individual layers are affected by weight drift it is also possible to measure the distance between specific convolutional layers or BN layers.

Method	Class-Incremental	Domain-Incremental
	$\ell_2(\frac{\theta_1}{\|\theta_1\|_2} - \frac{\theta_0}{\|\theta_0\|_2})$	$\ell_2(\frac{\theta_1}{\|\theta_1\|_2} - \frac{\theta_0}{\|\theta_0\|_2})$
Fine-Tuning	0.1	0.2
EWC [11]	0.02	0.11
Replay	0.14	0.25

Table 7.1: Weight distance calculated as ℓ_2 distance of the models parameters θ_0 to θ_1. The distance between the models' parameters is lowest for EWC as it explicitly constraints updates on existing parameters. Weight distance is largest for Replay which is least affected by drop an mIoU, indicating that weight distance does not always correlate with a drop in performance.

Results: In Table 7.1 we display the ℓ_2 distance for the models parameters for the class- and domain-incremental setting. We notice that the models trained with EWC stay the closest to θ_0, as it explicitly constraints updates on existing parameters. Although Replay is least affected by catastrophic forgetting, we observe the largest ℓ_2-distance between the model's weights. This indicates that less weight drift does not directly indicate more forgetting. We additionally compare the layer-wise distances of the convolutional layers in Figure 7.1. Interestingly, the model trained with EWC has only very minor changes in the weights of the model up until later layers in the decoder of the network, which coincides with the layers, in which we also observe a significant drop in similarity for layer stitching. Finally, we also compare the ℓ_2-distance of the BN layers in Figure 7.1 for the domain-incremental setting, in which we see that the very first BN layer undergoes the most drastic changes.

Limitations: The major difference between measuring weight drift instead of activation drift is that weight drift does not take the training data into account. However, we observe that the distance of parameters of the model is not indicative of the performance drop on the previous task. Therefore, we conclude that it can be used to interpret how the weights have changed, but it should not be understood as a direct measure for catastrophic forgetting.

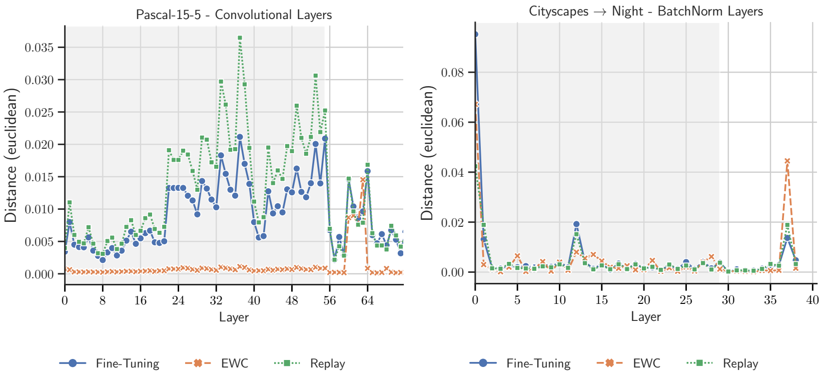


Figure 7.1: The distance of f_1 and f_0 measured by ℓ_2 distance of the weights of the convolutional layer in the class-incremental setting (left) and the running mean of the BN Layers in the domain-incremental setting.

Method	Class-Incremental						Domain-Incremental				
	Forgetting	CKA	Layer Stitching	Weight Distance	ΔmIoU$_R$	ΔmIoU$_{BN}$	*Forgetting*	CKA	Layer Stitching	Weight Distance	ΔmIoU$_{BN}$
Fine-Tuning	9.0	45.0	15.9	0.1	18.9	**0.1**	31.9	78.4	54.7	0.2	10.6
EWC	23.8	**88.5**	60.4	**0.02**	11.3	4.1	28.1	78.3	57.6	**0.11**	17.4
Replay	**39.1**	76.2	**80.2**	0.14	**3.8**	0.8	**10.1**	**94.8**	**83.8**	0.25	**-0.3**

Table 7.2: Comparison of the discussed methods to measure catastrophic forgetting. As CKA and layer matching measure similarity of acitvations for every layer, we report only the minimum value. Values in bold indicate the method that supposedly is least affected by forgetting according to the measure used. The results clearly indicate that catastrophic forgetting is nuanced as many effects contribute to forgetting in a different manner, so that there is no single measure that can show the full picture.

8 Conclusion

In this report we evaluated and discussed tools to assess the effects of catastrophic forgetting. In a series of experiments, we demonstrate the strengths and weaknesses of these tools. We find that these approaches work best in combination since they complement each other and capture different effects. For example, measuring activation drift with CKA or layer stitching is helpful to locate forgetting, but BN reestimation and Decoder Re-Training are required to identify the causes. Furthermore, we found that evaluating weight distances does not correlate with the drop in performance of previous tasks and should not be interpreted as a measure of catastrophic forgetting. Finally, we note that measures of activation drift such as layer matching and CKA are useful in both domain- and class-incremental settings, whereas BN Re-Estimation is more insightful in domain-incremental learning and Decoder Re-Training in the class-incremental learning. A summary the results is also displayed in Table 7.2. Our report reveals that catastrophic forgetting is nuanced as many effects contribute to catastrophic forgetting in a different manner, so that there is no single measure that can show the full picture.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. *Layer Normalization*. 2016. DOI: 10.48550/ARXIV.1607.06450. URL: https://arxiv.org/abs/1607.06450.

[2] Vanessa Buhrmester, David Münch, and Michael Arens. “Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: A Survey”. In: *Machine Learning and Knowledge Extraction* 3.4 (2021). DOI: 10.3390/make3040048.

[3] Marius Cordts et al. “The Cityscapes Dataset for Semantic Urban Scene Understanding”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[4] Adrián Csiszárik et al. "Similarity and Matching of Neural Network Representations". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 5656–5668. URL: `https://proceedings.neurips.cc/paper/2021/file/2cb274e6ce940f47beb8011d8ecb1462-Paper.pdf`.

[5] MohammadReza Davari et al. "Probing Representation Forgetting in Supervised and Unsupervised Continual Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 16712–16721.

[6] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *ICLR* (2021).

[7] M. Everingham et al. "The Pascal Visual Object Classes (VOC) Challenge". In: *International Journal of Computer Vision* 88.2 (June 2010), pp. 303–338.

[8] R. French. "Catastrophic Forgetting in Connectionist Networks". In: *Trends in Cognitive Sciences* 3.4 (1999), pp. 128–135. DOI: `10.1016/s1364-6613(99)01294-2`.

[9] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 448–456. URL: `https://proceedings.mlr.press/v37/ioffe15.html`.

[10] Tobias Kalb and Jürgen Beyerer. "Causes of Catastrophic Forgetting in Class-Incremental Semantic Segmentation". In: *Proceedings of the Asian Conference on Computer Vision (ACCV)*. Dec. 2022, pp. 56–73.

[11] James Kirkpatrick et al. "Overcoming catastrophic forgetting in neural networks". In: *Proceedings of the National Academy of Sciences* 114.13 (2017), pp. 3521–3526. DOI: `10.1073/pnas.1611835114`. eprint: `https://www.pnas.org/doi/pdf/10.1073/pnas.1611835114`. URL: `https://www.pnas.org/doi/abs/10.1073/pnas.1611835114`.

[12] Simon Kornblith et al. "Similarity of neural network representations revisited". In: *36th International Conference on Machine Learning, ICML 2019*. Vol. 2019-June. May 2019, pp. 6156–6175. ISBN: 9781510886988. arXiv: 1905.00414. URL: https://arxiv.org/abs/1905.00414.

[13] Timothée Lesort. "Continual Feature Selection: Spurious Features in Continual Learning". In: *arXiv preprint arXiv:2203.01012* (2022).

[14] Vincenzo Lomonaco, Davide Maltoni, and Lorenzo Pellegrini. "Rehearsal-Free Continual Learning Over Small Non-I.I.D. Batches". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2020.

[15] Marc Masana et al. "Class-incremental learning: survey and performance evaluation". In: *arXiv preprint arXiv:2010.15277* (2020).

[16] Michael McCloskey and Neal J. Cohen. "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem". English (US). In: *Psychology of Learning and Motivation - Advances in Research and Theory* 24.C (Jan. 1989), pp. 109–165. ISSN: 0079-7421. DOI: 10.1016/S0079-7421(08)60536-8.

[17] Seyed Iman Mirzadeh et al. "Linear Mode Connectivity in Multitask and Continual Learning". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=Fmg_fQYUejf.

[18] Kengo Murata, Tetsuya Toyota, and Kouzou Ohara. "What is happening inside a continual learning model? - A representation-based evaluation of representational forgetting - A R". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. Vol. 2020-June. 2020, pp. 952–956. ISBN: 9781728193601. DOI: 10.1109/CVPRW50498.2020.00125.

[19] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. "What is being transferred in transfer learning?" In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 512–523. URL: https://proceedings.neurips.cc/paper/2020/file/0607f4c705595b911a4f3e7a127b44e0-Paper.pdf.

[20] Vinay Venkatesh Ramasesh, Ethan Dyer, and Maithra Raghu. “Anatomy of Catastrophic Forgetting: Hidden Representations and Task Semantics”. In: *International Conference on Learning Representations*. 2021. URL: `https://openreview.net/forum?id=LhY8QdUGSuw`.

[21] Eduardo Romera et al. “ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation”. In: *IEEE Transactions on Intelligent Transportation Systems* 19.1 (2018), pp. 263–272. DOI: `10.1109/TITS.2017.2750080`.

[22] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. “ACDC: The Adverse Conditions Dataset with Correspondences for Semantic Driving Scene Understanding”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021.

Constraint-based Causal Discovery by using Path Constraints gained from Signal Injection and Recovery

Josephine Rehak

Vision and Fusion Laboratory
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT), Germany
Josephine.Rehak@kit.edu
ORCiD: 0000-0001-6139-9703

Abstract

The discovery of causal relations via interventions has proven to be simple when only one observed variable is affected or unaffected. However, in a multivariate setting, it is likely that more than one variable is affected by the intervention. Thus drawing conclusions about the true causal graph becomes far more difficult as we can not retrieve information of any obvious causal relationship or causal order. We demonstrate, that causal discovery with multiple affected variables is possible by introducing a novel definition of path constraints for constraint-based causal discovery. We exercise our novel technique on a combustion engine simulation, were we inject wavelets of our choice in a variable of investigation and try to rediscover this wavelet in the other, observed variables to gain such path constraints and thus to restraint the causal graph search.

1 Introduction

Causal discovery deals with finding cause and effect relationships in data. Its subdomain of interventional causal discovery tries to achieve this knowledge gain by performing experiments [5].

Current methods for interventional causal discovery either inspect the interventional effect in upmost one variable and thus recover few causal information or cannot handle interventional effects in multiple variables. We want to investigate a specific approach that is low-intrusive and injects a signal into a running process and gathers information about its occurrence in the causal graph to deduce information about causal relations. We will demonstrate an example experiment on a combustion engine simulation. The paper is structured as follows: In Section **??**, we shed some light on existing approaches for interventional structure learning. In Section 3, we introduce our fundamentals for the novel low-invasive technique. In Section 4, the signal injections are demonstrated on a combustion engine example step by step. In Section 4, we draw the conclusion.

2 Causal Graphs

Causal graphs consist of a set of nodes representing variables V and a set of edges E representing causal relations. If a directed edge points from $A \in V$ to $B \in V$, then variable B is caused by A. A path from A to B is a chain of consistently directed edges $C(A, B) = \{A \rightarrow X_1, \ldots, X_i \rightarrow B\}$, $X_i \in V$, $i \in \mathbb{N}$ directed from A to B with a number of edges being equal or greater than one $|C(A, B)| \geq 1$. A direct causal relation between variables indicates $|C(A, B)| = 1$, but an indirect causal relation indicates a path $|C(A, B)| > 1$.

The goal of causal discovery is to gain knowledge over the true causal graph for the inspected environment. According to the theory of constraint-based learning [6], all the potential graphs form an equivalence class in respect to our knowledge about the edges and variables of the graph itself. If the number of graphs in the equivalence class equals one, we assume to have found the true causal graph. But if no knowledge of the causal graph is given and the larger the amount of inspected variables, the more edges have to be inspected. Table 2.1

shows, how the number of graphs in an equivalence class grows exponentially with the number of variables and edges. One can gain these constrains, by either inspecting data, or as in our case performing experiments.

2.0.1 Interventional Causal Discovery

Using interventions for causal discovery is one of the oldest and most popular approaches in science. Even despite their costliness, their potential of being unethical or by being simply not feasible. It is assumed that a variable is a cause of another variable B if an intervention on A also affects the associated variable B [8]. [2] distributed the existing approaches in two major categories called structural interventions and parametric interventions.

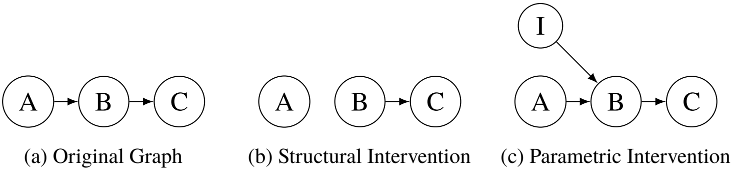
2.0.2 Structural Interventions

As shown in Figure 2.1, structural interventions (also called hard interventions) cut off all causal influences to the variable under intervention and determines its probability distribution completely. For example in randomized controlled drug trails [3], the treatment drug a patient receives is determined randomly but always one of several options. Here, as stated by the potential outcomes framework, the causal effect is identified by structurally intervening on the one variable while observing the effect in the other.

Parametric Interventions Parametric interventions (also called soft intervention) intervene on the probability distribution of a variable by adding another cause to it or its causes. They do not disturb the original causal structure, but

Count of Variables	1	2	3	4	5	6
Count of Potential Edges	0	1	3	6	10	15
Count of Potential Graphs	0	4	64	4,096	1,048,576	1,073,741,824

Table 2.1: Overview of the exponential increase in graphs with growing number of edges and variables



(a) Original Graph (b) Structural Intervention (c) Parametric Intervention

Figure 2.1: Causal graphs after a structural and a parametric intervention on variable B

no foreign influences of other variables can be prevented with certainty. In comparison, parametric interventions are a rather new technique.

3 Essentials of Wavelet Injections

The new intervention method can be classified according to Section **??** as soft intervention-based, since the original causal network is not perturbed but an additional variable is added as a cause to the inspected variable. The particularity of these interventions lies in the fact that a wavelet in the form of a wavelet is added to the variable and is tried to be rediscovered to gain causal information.

When injecting a wavelet into a variable A, the injected wavelet and the timeseries coming from the causing variable are added up. We assume the wavelet to spread in the graph in direction of the causal relations. If we find the wavelet in one variable, we assume a direct causal relation to be present, but in case of a discovery in several other variables, including B, we may not. Instead, we gain knowledge about an existing path between the variables with $|C(A, B)| \geq 1$, since the wavelet must have traveled somehow from A to B. We will refer to such path information as path constraints from here on. With each gained path constraint, we can remove all graphs from the equivalence class of potential graphs that have no path between the investigated variables present and hence do not support it and thus the number of potential graphs is decreased.

For signal recovery, we normalized the measured values. Otherwise, the different scales of the variables would make a comparison difficult. Then we applied on each measured variable the fast pattern matching algorithm called Mueens ultra-fast Algorithm for Similarity Search (MASS) [9]. It stepwise matches a desired pattern to a subsequence of the inspected timeseries and calculates the z-normalized distance. The aggregation of these distances results in an overall distance profile. If its minimal distance is below a chosen threshold, we assume the position to be our signal. Otherwise, we assume the signal to be absent in the observed variable and thus we gain no path constraint.

Note, that in general, we do not consider information about variables in which the injected signal could not be found, as the wavelet may be lost due to various reasons. For example, the signal may be of unfortunate form and hence be canceled out by the causal graph itself, it may be heavily deformed and thus be not recoverable or simply be too weak to be noticeable in other variables.

4 Applying Signal Injections

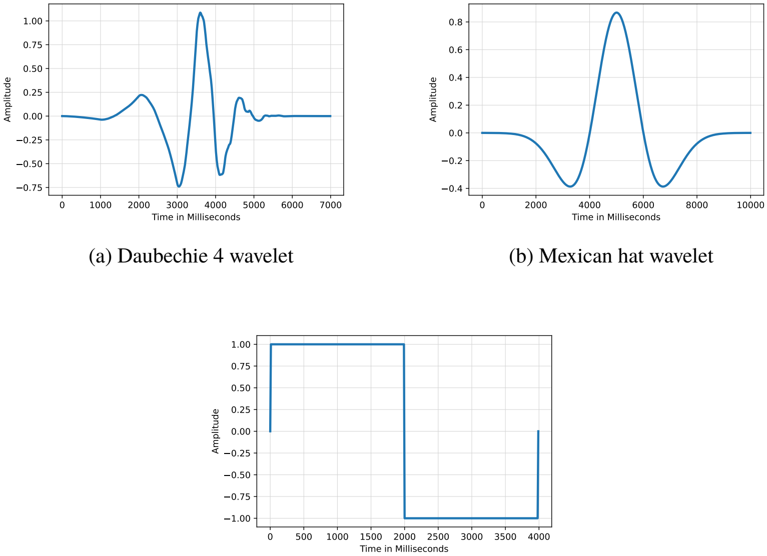
Here we demonstrate how we applied the signal injections on a combustion engine dataset and explain the experiment step by step.

Step 1: Wavelets for Injection

We decided to use three very distinct and well-defined wavelets for our signal injection in the combustion engine. These are a Daubechie 4 wavelet, a Mexican Hat wavelet and a Haar wavelet. They are depicted in Figure 4.1. We have chosen these wavelets because they contain amplitudes in the positive and negative value range and have a unique shape.

Step 2: Simulation Setup

As a testing environment, we used a running combustion engine simulation [7, 4, 1]. To evaluate the performance of the novel discovery approach, we inferred the simulation's true causal graph as is shown in Figure 4.2. Here, we give a brief explanation of the causal relations: The *angle* of the throttle plate influences



(a) Daubechie 4 wavelet

(b) Mexican hat wavelet

(c) Haar wavelet

Figure 4.1: The wavelets we used for signal injection

how much *air intake* in the motor cylinder is possible. The *air intake* over time adds up to the *aircharge* in the cylinder before combustion. After combustion, depending on the *aircharge*, the *torque* increases and finally the overall *engine speed* rises. The increase in *engine speed* also depends on the *load* carried by the engine.

We added multiple sensors to the simulation to retrieve the timeseries required for signal recovery. In total we took measurements of six variables including the actuated variable. We injected the signals in the actuation of the angle only and inspected all the other measured timeseries for traces of the injected signal. According to Figure 4.2, we expect to find the wavelets in the air intake, aircharge, torque and engine speed variable, but not in the load variable, as it is

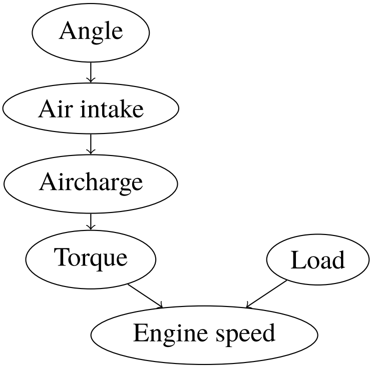


Figure 4.2: The true causal graph of the combustion engine simulation

independent of the angle variable. The wavelet rediscovery method are required to come to the same conclusion.

Step 3: Signal Discovery

As an implementation of the pattern matching algorithm, we used the python package stumpy [1]. With it, we found several wavelets in all variables depending on the influenced angle variable. Table 4.1 gives an overview of the wavelets we rediscovered. All in all, the Daubechie 4 wavelet and the Mexican hat wavelet performed best, as they were found in all variables depending on the angle variable in their actual positions. For evaluation purposes, we determined the actual position in advance, by comparing the measurements with wavelet injections with measurements without injections. Any divergence between those measurements must be caused by the wavelet. We decided to use this information only for evaluation, as we want to gain causal information with minimum number of measurements.

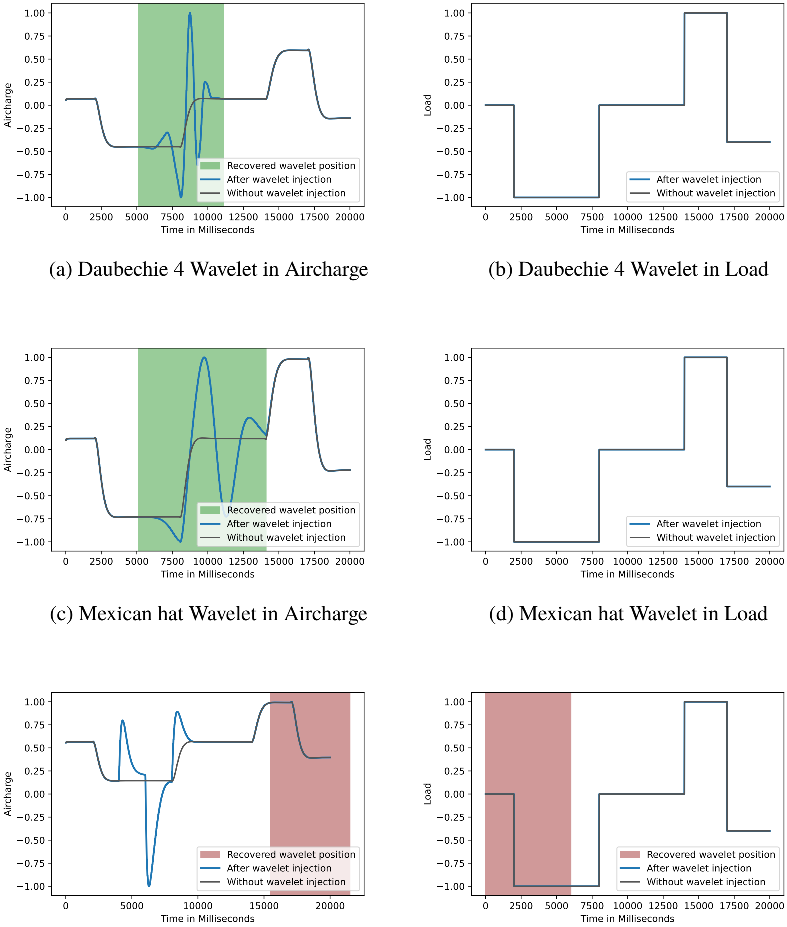
[1] `https://stumpy.readthedocs.io/`

	Air Intake	Aircharge	Torque	Engine Speed	Load
Daubechie 4 wavelet	✓	✓	✓	✓	×
Mexican hat wavelet	✓	✓	✓	✓	×
Haar wavelet	✓	×	×	×	×
Ground Truth	✓	✓	✓	✓	×

Table 4.1: An overview of the wavelets we injected into the angle variable and if they could be recovered in the other variables in their actual position.

Figure 4.3 presents an excerpt from our results for the aircharge and the load variable for each of the three wavelets. The colored area is where the signal was rediscovered by the pattern matching algorithm. It is colored green, when the wavelet is found in its actual position, but if it is red, it was found in a wrong position or in a variable, where no wavelet influence is present. In the aircharge variable, both the Daubechie 4 wavelet and the Mexican hat wavelet were rediscovered in their actual position. Only the Haar wavelet was found in the wrong position (Subfigure 4.3(e)).

In the Load variable, no signal should be found at all, as the variable is not influenced by the angle variable where we injected the wavelet. But the discovery algorithm wrongly found the Haar wavelet in the Load variable (Subfigure 4.3(f)). We assume the mistaken discoveries of the Haar wavelet to be because of the simple wavelet form, it fits in several places of a time series, even if it is not present at all. Both Mexican hat wavelet and Daubechie 4 wavelet proved complex enough to allow a safe rediscovery.



(a) Daubechie 4 Wavelet in Aircharge

(b) Daubechie 4 Wavelet in Load

(c) Mexican hat Wavelet in Aircharge

(d) Mexican hat Wavelet in Load

(e) Haar Wavelet in Aircharge

(f) Haar Wavelet in Load

Figure 4.3: The wavelets as they were discovered in the exemplary aircharge and load variable. The area where the lines diverge indicates the presence of a signal and is highlighted green as a mark for successful recovery. Red highlights indicate a wrong recovery. If the lines do not diverge, no wavelet is present and nothing should be discovered.

Step 4: Causal Inference and Results

From the previous step, we were able to retrieve from the Mexican hat and the Daubechie 4 wavelet injection independently four path constraints $\{C(\text{Angle}, \text{Air intake}), C(\text{Angle}, \text{Aircharge}), C(\text{Angle}, \text{Torque}),$ $C(\text{Angle}, \text{Engine Speed})\}$. Next, we implemented a brute force algorithm, that generated all viable graphs for given variables and eliminated all graphs from the set not supporting the constraints. In total, we were able to reduce the number of graphs from 1,073,741,824 to 23,855,104 and eliminated with this method approximately 97.8% of potential causal graphs. The number of graphs may be reduced further by performing additional wavelet injections.

5 Conclusion

We explained the idea of discovering causal knowledge by injecting and retrieving wavelets in causal variables. For injection, we simply added a chosen wavelet to the incoming timeseries of a variable and tried to rediscover it in the depending variables via fast pattern matching. We gained causal information by defining path constraints to restrict the equivalence class for the true causal graph, as a path is assumed to be present between injected variable and the variable of rediscovery. We demonstrated the procedure on a running combustion engine simulation by adding three different wavelets (Haar, Daubechie 4 and Mexican hat) to an actuated variable. The procedure performed well for the Mexican hat wavelet and the Daubechie 4 wavelet. By using either of them, we were able to receive four path constraints and to reduce with them the number of graphs from 1,073,741,824 to 23,855,104.

References

[1] PR Crossley and JA Cook. “A nonlinear engine model for drivetrain system development”. In: *International Conference on Control 1991. Control’91*. IET. 1991, pp. 921–925.

[2] Frederick Eberhardt and Richard Scheines. “Interventions and causal inference”. In: *Philosophy of science* 74.5 (2007), pp. 981–995.

[3] Ronald Aylmer Fisher. “Design of experiments”. In: *British Medical Journal* 1.3923 (1936), p. 554.

[4] John J Moskwa and J Karl Hedrick. “Automotive engine modeling for real time control application”. In: *1987 American Control Conference*. IEEE. 1987, pp. 341–346.

[5] Josephine Rehak. “A Proposal on Discovering Causal Structures in Technical Systems by Means of Interventions”. In: *Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*. 2021, p. 91.

[6] Josephine Rehak. “A Review on Approaches for Causal Structure Identification”. In: *Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*. 2022, p. 105.

[7] Robert W Weeks and John J Moskwa. “Automotive engine modeling for real-time control using matlab/simulink”. In: *SAE transactions* (1995), pp. 295–309. URL: `https://www.mathworks.com/help/simulink/slref/modeling-engine-timing-using-triggered-subsystems.html`.

[8] James Woodward. *Making things happen: A theory of causal explanation*. Oxford university press, 2005.

[9] Chin-Chia Michael Yeh et al. “Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets”. In: *2016 IEEE 16th international conference on data mining (ICDM)*. Ieee. 2016, pp. 1317–1322.

A Detailed Study of the Association Task in Tracking-by-Detection-based Multi-Person Tracking

Daniel Stadler

Vision and Fusion Laboratory
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT), Germany
daniel.stadler@kit.edu

Abstract

Many multi-person trackers follow the tracking-by-detection paradigm applying a person detector in each frame and linking detections of the same target to form tracks in the association task. While the basic concept is the same among these methods, various motion models, distance metrics to measure the similarity of targets, and matching strategies are used. This makes it difficult to compare different methods and also to assess the influence of single tracking components on the final performance. For these reasons, all parts of the association task are thoroughly investigated in this study. Starting with a simple baseline which is consequently improved with the help of experimental results, a strong tracking-by-detection-based framework is developed that achieves state-of-the-art performance on two multi-person tracking benchmarks.

1 Introduction

The objective of multi-person tracking (MPT) is to detect and identify all persons in every frame of a given video. Applications range from crowd monitoring to autonomous driving and surveillance related tasks.

To solve the MPT problem, most methods pursue the tracking-by-detection (TBD) paradigm. A detector is applied on each image independently and the obtained detection sets are matched such that detections of the same target form a track with a unique ID. This problem of assigning the correct detections to the corresponding tracks is called the association task. While some approaches try to integrate detection and association more tightly [2, 9, 31, 33, 43], the strict separation of the two sub-tasks in TBD can still achieve state-of-the-art results. Currently, the top performing entries of the standard MPT benchmarks MOT17 [22] and MOT20 [6] follow the TBD paradigm [1, 7, 23, 28, 46] leveraging an of-the-shelf detection model and focusing on the association task.

Different strategies to improve the association can be observed in the literature. Motion models based on Kalman filter [16] are used to make the estimated target positions more accurate [4, 5, 10, 40]. In addition, camera motion compensation techniques are integrated to deal with motion of non-static cameras [2, 10, 15, 28, 29]. The core of the association is the distance measure which determines how likely a detection belongs to a so-far tracked target. On the one hand, motion-based metrics such as Intersection over Union (IoU) are utilized and on the other hand, the appearance of targets is leveraged. For example, in DeepSORT [40] and its further development StrongSORT [10], a person re-identification model is applied to extract appearance features from the image patches of the detections and cosine distance between the high-dimensional features is taken as association metric. While DeepSORT uses motion distance only for gating, i.e., prohibiting unlikely assignments, StrongSORT combines it with appearance distance as also done in [1, 7]. Besides the distance metric, the association strategy has a large influence on the performance. While most methods make all assignments at once with the Hungarian algorithm [17], DeepSORT proposes a matching cascade that prefers previously observed targets and ByteTrack [46] performs a second matching step in which low-confident detections are utilized.

In this study, all aforementioned components of the association task in MPT are analyzed in detail. Starting with a baseline TBD approach with strong motion models, a large number of experiments with different distance measures, both motion- and appearance-based, and their combinations are conducted. In addition, matching strategies with multiple stages are investigated. With the help of the experimental results, a strong TBD method is developed which achieves state-of-the-art results on the two MPT datsets MOT17 [22] and MOT20 [6]. Furthermore, ablative experiments of the proposed framework are performed showing the influence of the single tracking components as well as the sensitivity of the tracking parameters on the final performance.

2 Baseline and Motion Models

A baseline tracker using only IoU as matching metric is built before more advanced matching measures and strategies are investigated. In addition, various motion models for target and camera motion are compared in this section.

Let $\mathcal{T}^{t-1} = \{T_1^{t-1}, \dots, T_k^{t-1}\}$ be the tracks found until frame I^{t-1} and $\mathcal{D}^t = \{D_1^t, \dots, T_l^t\}$ the detections generated on the frame I^t of a video $V = [I^1, \dots, I^n]$ of length n. The association task is to assign the detections $\mathcal{D}^t$ to its corresponding targets $\mathcal{T}^{t-1}$. For this, distances between all confident detections and tracks are calculated and used as cost values. Afterwards, the overall costs of assignments are minimized, e.g., with the Hungarian algorithm [17]. More precisely, given a detection $D = (B_D, s) \in \mathcal{D}_t$ with box B_D and confidence s and the box B_T of a track $T \in \mathcal{T}^{t-1}$, a distance measure d can be calculated using the IoU between detection box B_D and track box B_T:

$$d_{\text{IoU}} = 1 - \text{IoU}(B_{\text{D}}, B_{\text{T}}) \tag{2.1}$$

Before calculation of the distance matrix of detections and tracks, the detections are filtered w.r.t. confidence, i.e., detections with a score s smaller than the threshold s_{track} are removed and not used in the association. In addition, a maximum distance $d_{\max}$ is enforced to prohibit unlikely assignments. Unmatched tracks that are not assigned a detection become *inactive* and are kept for $i_{\max}$ frames in the set of tracks before deletion. Thus, they can be re-activated for a short time period to bridge occlusions, for instance. Unmatched detections with high confidence $s \geq s_{\text{init}}$ start new tracks. Note that some trackers [10, 28, 40, 46] follow an initialization strategy, in which detections first start tentative tracks that have to be confirmed in subsequent frames in order to become active. While this strategy suppresses frame-wise false positive detections, it introduces false negatives since the tentative tracks do not contribute to the tracking output.

If the quality of detections is high and a large threshold s_{init} is set, such an initialization technique can reduce the overall performance, so it is not used in this study unless otherwise stated.

Most MPT approaches have in common, that a Kalman filter (KF) [16] is used to model the motion of targets. However, various formulations of the state vector $\mathbf{x}$ and different implementation details can be found in the MPT literature. The most used variants are originally from the SORT [4] and DeepSORT [40] frameworks. The state vectors of the two KF types are as follows:

$$\mathbf{x}_{\text{SORT}} = (u, v, a, r, \dot{u}, \dot{v}, \dot{a})^{\mathsf{T}} \tag{2.2}$$

$$\mathbf{x}_{\text{DeepSORT}} = (u, v, r, h, \dot{u}, \dot{v}, \dot{r}, \dot{h})^{\mathsf{T}} \tag{2.3}$$

The box center position is (u, v) and the aspect ratio is $r = w/h$ with w and h denoting box width and height, respectively. A derivative of a variable x with respect to time is indicated by $\dot{x}$. Whereas SORT explicitly models the box area $a = w \cdot h$ and its derivative $\dot{a}$ but keeps the aspect ratio r fixed, DeepSORT instead models the box height h and its derivative $\dot{h}$. Thus, the process and measurement noise covariance matrices also differ next to other implementation details, which can be found in the papers [4, 40] or the public source code.

Recently, further developments have been proposed for the DeepSORT variant – the Noise Scale Adaptive (NSA) KF [8] and the height preservation (HP) adaptation [30]. In the update step of the NSA KF, the measurement noise covariance matrix $\mathbf{R}$ is weighted with the confidence of the measurement, i.e., the detection confidence score s, as follows:

$$\mathbf{R}_{\text{NSA}} = (1 - s) \cdot \mathbf{R} \tag{2.4}$$

The higher the detection confidence, the smaller the adapted measurement noise covariance $\mathbf{R}_{\text{NSA}}$ and the more influence has the detection on the track state update. The other adaptation is related to the state vector $\mathbf{x}$. It is empirically found in [30], that predicting inactive tracks for multiple frames without state update, the track box size can change dramatically which hinders re-activation after occlusion. To prevent this, HP can be applied simply setting the derivative $\dot{h}$ to zero before the KF prediciton step, which is also done in [1] and [46].

Besides target motion, modelling camera motion is also important. For camera motion compensation (CMC), again two different methods from literature are

Table 2.1: Motion Model Results.

KF Type	NSA	CMC	HOTA	KF Type	NSA	CMC	HP	HOTA
SORT	✗	✗	67.61	DeepSORT	✗	✗	✗	67.40
SORT	✓	✗	67.67	DeepSORT	✓	✗	✗	67.82
SORT	✗	ECC	67.77	DeepSORT	✗	ECC	✗	68.03
SORT	✗	ORB	**68.36**	DeepSORT	✗	ORB	✗	68.13
SORT	✓	ECC	68.03	DeepSORT	✓	ORB	✗	68.62
SORT	✓	ORB	68.35	DeepSORT	✓	ORB	✓	**68.67**

investigated – the Enhanced Correlation Coefficient (ECC) Maximization [12] and a model from [28] that is based on the ORB [26] feature detector and the RANSAC [13] algorithm. The ORB method is a sparse image registration technique in that foreground objects like moving persons can be neglected, in contrast to the global ECC method. A similar approach is found in [1].

To compare the different motion models, several experiments are run on the validation split (Val) of MOT17, which is created by dividing the train sequences into two halves and using the second ones [35, 46, 48]. As detection model, a publicly available YOLOX [14] model from [46] is utilized, which has been trained on a combined dataset consisting of CrowdHuman [27], CityPersons [45], ETH [11], and the first half of MOT17 train split. Note that this YOLOX model can be regarded as the current standard in MPT on the MOT datasets, since many state-of-the-art methods are using it [1, 10, 5, 28, 23, 37, 36, 46]. If not otherwise stated, the parameters of the tracker are set to $s_{\text{init}} = 0.7$, $s_{\text{track}} = 0.6$, $d_{\max} = 0.8$, $i_{\max} = 30$ and the resolution of the input images is 1440×1080 pixels. To measure the overall tracking accuracy, HOTA [20] is evaluated.

The results with different KF types, KF adaptations and CMC models are summarized in Table 2.1. Without any extensions, the SORT KF performs slightly better than the DeepSORT KF. However, the results of the DeepSORT KF can be largely improved with the NSA adaptation, while NSA in combination with SORT does not enhance the results in all configurations. This is not surprising, as NSA is developed as extension for the DeepSORT KF and the measurement noise covariance matrices $\mathbf{R}$ differ among the KF types. As expected, ORB outperforms ECC in all experiments. W.r.t. the baselines, ORB improves the overall tracking performance by 0.75 HOTA and 0.73 HOTA for SORT KF and DeepSORT KF, respectively. Additionally adding the height preservation (HP) in the DeepSORT KF variant, a HOTA of 68.67 is achieved which is a gain of 1.27 HOTA in comparison to the DeepSORT KF baseline. Therefore, the DeepSORT KF with NSA and HP extensions is used in all subsequent experiments, together with the CMC model based on ORB features.

3 Distance Measures

As mentioned previously, the distance measure is the core of each TBD algorithm. In the baseline experiments of the last section, the IoU has been leveraged which is the most used motion-based distance metric in MPT. In this section, further distance measures for the association are explored. First, motion-based matching is analyzed in Section 3.1. Then, appearance-based matching is studied in Section 3.2. Both types of infomation are combined in Section 3.3, before further techniques like incorporating the detection confidence and applying gating mechanisms are treated in Sections 3.4 and 3.5, respectively.

3.1 Motion-based Matching

The authors of SimpleTrack [18] experiment with the Generalized IoU (GIoU) [24] as similarity measure in combination with appearance information, which enhances the performance of their tracker. This raises the question, whether other IoU related measures also can improve the matching accuracy. Therefore, different adaptations of the original IoU are investigated in the following. Given two boxes $A = (x_A, y_A, w_A, h_A)$ and $B = (x_B, y_B, w_B, h_B)$, the IoU is the relation of the intersection $A \cap B$ to the union $A \cup B$:

$$\mathrm{IoU} = \frac{A \cap B}{A \cup B} \tag{3.1}$$

The IoU has the drawback that non-overlapping boxes always yield an IoU of 0, independent from how far away the boxes are from each other. To solve this issue, the GIoU is proposed as

$$\mathrm{GIoU} = \mathrm{IoU} - \frac{C \setminus (A \cup B)}{C} \tag{3.2}$$

where C denotes the smallest enclosing box of A and B. While the spatial distance of the boxes A and B has influence on the box C, it is not modelled explicitly. In contrast, the euclidean distance $d_{\mathrm{L2}}(A, B) = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$ is directly used in the Distance IoU (DIoU) [47]:

$$\mathrm{DIoU} = \mathrm{IoU} - \frac{d^2_{\mathrm{L2}}(A, B)}{c^2} \tag{3.3}$$

Here, c denotes the diagonal of the smallest enclosing box C. The same paper further introduces the Complete IoU (CIoU) [47], which not only explicitly models spatial distance but also aspect ratio consistency:

$$\mathrm{CIoU} = \mathrm{DIoU} - \alpha v \tag{3.4}$$

$$v = \frac{4}{\pi^2} \left(\arctan\left(\frac{w_A}{h_A}\right) - \arctan\left(\frac{w_B}{h_B}\right)\right)^2 \tag{3.5}$$

$$\alpha = \frac{v}{(1 - \mathrm{IoU}) + v} \tag{3.6}$$

Note that the IoU and its variants are similarity measures with a maximum similarity of 1. Thus, a distance measure can be created by subtracting the value from 1 as in Equation 2.1.

If a Kalman filter is used as motion model, it is possible to integrate the uncertainty of the motion estimation into the distance measure. For this, DeepSORT [40] and StrongSORT [10] calculate the squared Mahalanobis distance between a detection D and a track T, given the state formulation of the detection box $\mathbf{d} = (u, v, r, h)^{\mathsf{T}}$ and the projection of the track state (mean and covariance) into measurement space $(\mathbf{y}, \mathbf{S})$:

$$d_{\mathrm{mahal}} = (\mathbf{d} - \mathbf{y})^{\mathsf{T}} \mathbf{S}^{-1} (\mathbf{d} - \mathbf{y}) \tag{3.7}$$

Whereas DeepSORT uses only the Mahalanobis distance for gating, i.e., preventing unlikely assignments by enforcing a maximum distance, here, d_{mahal} is directly used as matching distance. Additionally, the euclidean distance d_{L2} between detection and track center is considered.

Table 3.1: Motion-based Matching Results.

d	IoU	GIoU	DIoU	CIoU	L2	Mahal
HOTA	68.67	68.47	**68.74**	**68.74**	64.70	62.94

To compare the performance of the aforementioned motion-based distance measures d in the association, several experiments are conducted tuning the maximum distance threshold d_{max} for each metric separately. The highest achieved HOTA values are reported in Table 3.1. One can see that the IoU-based distance measures work much better than taking the L2 distance or the Mahalanobis distance. While L2 distance does not consider the important information of box dimensions, the Mahalanobis distance is only a rough estimation of the object location if the state uncertainty is high [40]. In the experimental setup, DIoU and CIoU achieve the highest HOTA value of 68.74, closely followed by IoU and GIoU. Note that DIoU and CIoU yield the *exactly* same tracking results. Since the aspect ratio of targets does not vary significantly in MPT, v in Equation 3.4 becomes a very small value, thus CIoU $\approx$ DIoU holds. For this reason, the DIoU is used in the rest of this study.

3.2 Appearance-based Matching

Similar to adopting an of-the-shelf detector, many MPT approaches take over a model from the re-identification community for extracting appearance features of targets [1, 10, 19, 34, 40, 41]. Such a network takes a small image patch of a detected person as input and computes a high-dimensional feature vector that represents the appearance of the person. In appearance-based matching, several design choices have to be made when comparing the features of detections and tracks. Which distance measure should be used? How many time steps shall be considered to describe the appearance of a track? What is the best way to combine features from different time steps? In this section, a large amount of experiments is conducted to answer these questions empirically. Given two m-dimensional feature vectors f_D and f_T from a detection and a track, respectively, one can calculate either the cosine distance d_{cos} or the euclidean distance d_{L2} to measure their appearance similarity:

$$d_{\cos} = 1 - \frac{f_D \cdot f_T^{\intercal}}{\|f_D\| \cdot \|f_T\|} \tag{3.8}$$

$$d_{\mathrm{L2}} = \sqrt{(f_{D,1} - f_{T,1})^2 + (f_{D,2} - f_{T,2})^2 + \cdots + (f_{D,m} - f_{T,m})^2} \tag{3.9}$$

Note that $d_{\cos} \in [0, 2]$ and $d_{\mathrm{L2}} \in [0, \infty]$ holds and $\| \cdot \|$ represents the Euclidean norm. Studying the source code of a few appearance-based MPT methods, it is observed that some methods apply a mask to the cosine distance matrix before solving the assignment problem with the Hungarian method. More precisely, all entries above the maximum distance threshold $d_{\max}$ are set to $d_{\max} + \epsilon$ with ϵ being a very small value, e.g., $1\mathrm{e}^{-5}$. This causes unlikely assignments with a distance above the matching threshold $d_{\max}$ to have the same contribution to the overall cost that is minimized by the Hungarian algorithm.

While the detection feature f_D is simply the output of the re-identification model, there are multiple possibilities to build the track feature f_T. In the simplest case, the feature f_D^{t-1} from the last assigned detection D^{t-1} of the track $T^t = [D^{t_{\text{init}}}, \ldots, D^{t-2}, D^{t-1}]$ is used as track feature: $f_T = f_D^{t-1}$. To benefit from temporal information, DeepSORT [40] builds a feature bank $F_T = [f_D^{t-N}, \ldots, f_D^{t-2}, f_D^{t-1}]$ with the features of the past N time steps. The distance to a current detection feature f_D^t is calculated for each feature of the bank. The appearance distance $d(D, T)$ between a detection D and a track T is then chosen to be the minimum of all distances derived from the feature bank:

$$d_{\min}(D, T) = d_{\min}(D, F_T) = \min_{i \in [1, \ldots, N]} d(f_D^t, f_D^{t-i}) \tag{3.10}$$

If the target is clearly visible both in one of the last N frames and the current frame, the extracted features are of high quality and taking the minimum appearance distance is a good choice. However, this is not always the case in MPT, especially when facing severe occlusions. In such situations, the mean distance might be a better choice:

$$d_{\text{mean}}(D, T) = d_{\text{mean}}(D, F_T) = \frac{1}{N} \sum_{i}^{N} d(f_D^t, f_D^{t-i}) \tag{3.11}$$

Moreover, it is possible to average the two measures, which results in a third strategy to calculate the appearance distance between a detection and a track:

$$d_{\text{mean+min}}(D, T) = \frac{1}{2}(d_{\text{mean}}(D, F_T) + d_{\text{min}}(D, F_T)) \tag{3.12}$$

The last investigated strategy for computing the appearance distance is adopted from [38]. Instead of using a feature bank, the track feature f_T is updated in an exponential moving average (EMA) fashion with the newly assigned detection feature f_D^t and a weighting factor α in each time step:

$$f_T^t = \alpha f_T^{t-1} + (1 - \alpha) f_D^t \tag{3.13}$$

The re-identification model from [1] is leveraged for feature extraction in the experimental evaluation. It is a BoT (SBS) [21] model with ResNeSt50 [44] as backbone, trained on the first half of MOT17 [22] train split. The performance of the aforementioned appearance-based distance measures and strategies is again compared on the MOT17 Val split, whereby the maximum distance threshold d_{max} is optimized for each configuration separately. For experiments using the EMA technique, the corresponding parameter α is also tuned.

The resulting HOTA values are reported in Table 3.2. One can see that masking the distance matrix is beneficial for cosine distance but not euclidean (L2) distance. With masking, cosine distance outperforms L2 distance by 0.36 HOTA. Taking $N = 10$ past time steps in a feature bank into account, the results improve significantly by 1.40 to 2.25 points, depending on the strategy of the distance calculation. This shows the importance of temporal information in appearance-based matching. The best results are achieved by averaging the mean and minimum distance of the features (mean+min). Increasing the number of features yields improvements up to $N = 10$, while HOTA values decrease again using 20 or even 100 features. The EMA strategy achieves competitive results but HOTA is 0.25 points worse than the best configuration – the mean+min strategy with $N = 10$ past features and masked cosine distance – which achieves 68.72 HOTA. Note that the overall performance of the appearance-based matching is on par with the motion-based matching from the previous section (Table 3.1). However, on indiviual sequenves of the dataset, differences in HOTA up to 4 points are observed. This motivates the combination of motion- and appearance-based matching which is investigated in the next section.

Table 3.2: Appearance-based Matching Results.

d	Masking	N	Strategy	EMA	HOTA
Cosine	✗	1	✗	✗	66.19
Cosine	✓	1	✗	✗	66.47
L2	✗	1	✗	✗	66.11
L2	✓	1	✗	✗	66.03
Cosine	✓	10	min	✗	67.87
Cosine	✓	10	mean	✗	68.20
Cosine	✓	10	mean+min	✗	**68.72**
Cosine	✓	1	mean+min	✗	66.47
Cosine	✓	2	mean+min	✗	67.25
Cosine	✓	5	mean+min	✗	68.26
Cosine	✓	10	mean+min	✗	**68.72**
Cosine	✓	20	mean+min	✗	68.60
Cosine	✓	100	mean+min	✗	68.05
Cosine	✓	1	✗	✓	68.47

3.3 Combined Matching

Motion- and appearance-based distance measures provide different types of information. Thus, combining both kinds to an advanced distance measure is a promising approach which is also followed in other works [1, 10, 18]. Given two distance measures d_1, d_2 and corresponding weights w_1, w_2, a combined distance d_{comb} can simply be built by a weighted sum:

$$d_{\text{comb}} = w_1 d_1 + w_2 d_2 \tag{3.14}$$

For motion information, the IoU-based distance measures d_{IoU}, d_{GIoU} and d_{DIoU} are considered, while the feature cosine distance d_{cos} is used for appearance information. Experiments with different configurations are conducted on MOT17 Val. Note that the maximum distance threshold $d_{\max}$ is adjusted when changing distance measures or one of the weights w_1 or w_2. The resulting HOTA values are listed in Table 3.3. The previously achieved results using either motion-

Table 3.3: Combined Matching Results.

d_1	d_2	w_1	w_2	HOTA	d_1	d_2	w_1	w_2	HOTA
IoU	✗	✗	✗	68.67	IoU	Cosine	1	2	69.16
GIoU	✗	✗	✗	68.39	IoU	Cosine	1	3	69.13
DIoU	✗	✗	✗	68.74	IoU	Cosine	1	4	69.22
Cosine	✗	✗	✗	68.72	IoU	Cosine	1	5	69.04
IoU	Cosine	1	1	68.91	GIoU	Cosine	1	4	69.37
IoU	Cosine	2	1	68.62	DIoU	Cosine	1	4	**69.41**

or appearance-based information are also given for reference. The best results are very similar with HOTA $= 68.74$ for DIoU and HOTA $= 68.72$ for cosine distance, which justifies the usage of both cues. Combining IoU distance and cosine distance with equal contribution ($w_1 = w_2 = 1$), HOTA improves to 68.91. Giving more weight to the motion-based measure ($w_1 = 2$, $w_2 = 1$), the performance decreases. However, if the appearance information is taken more into account ($w_1 = 1$, $w_2 > 1$), HOTA can be further enhanced up to 69.22 for $w_2 = 4$. The same holds true for combining GIoU or DIoU distance with appearance cosine distance. The largest HOTA value of 69.41 is obtained by combining DIoU distance and cosine distance while setting $w_1 = 1$ and $w_2 = 4$, i.e., giving four times the weight to the appearance information. This is a gain of 0.69 points in HOTA compared to using only one distance measure.

Note that experiments have also been conducted with the Mahalanobis distance d_{mahal} (Equation 3.7) in combination with the appearance cosine distance as it is done in StrongSORT [10]. The highest achieved HOTA in the experimental setup is 69.13. While this is also an improvement w.r.t. using only appearance information, the performance is worse than combining DIoU distance with the appearance cosine distance. Therefore, the combination of DIoU distance and cosine distance is utilized in the remainder of this study.

Table 3.4: Use Detection Confidence Results.

d	d_{score}	HOTA	d	d_{score}	HOTA	d	d_{score}	HOTA
IoU	✗	68.67	DIoU	✗	68.74	DIoU+Cosine	✗	**69.41**
IoU	✓	**68.78**	DIoU	✓	**68.79**	DIoU+Cosine	✓	69.19

3.4 Use of Detection Confidence

Some IoU-based MPT methods incorporate the detection confidence s into the distance calculation by simple multiplication [1, 28, 46]:

$$d_{\text{IoU,score}}(D, T) = 1 - (\text{IoU}(B_D, B_T) \cdot s) \tag{3.15}$$

The motivation behind it is that more confident detections should be favored in the association. Note that this strategy can also be applied together with other IoU-based metrics and its influence is investigated empirically. Because the multiplication of $s \in [s_{\text{track}}, 1]$ changes the scale of the distance measure d, the maximum distance threshold d_{max} has again been tuned. The results are depicted in Table 3.4. Integrating the detection score into the distance matrix slightly improves HOTA by 0.11 and 0.05 points for IoU and DIoU distance, respectively. However, in combination with the appearance cosine distance, which yields the overall best results, using the detection score degrades the performance. Thus, the detection score is not leveraged in the distance calculation in the remainder of the study.

3.5 Gating

As mentioned before, DeepSORT [40] utilizes the Mahalanobis distance to prevent unlikely assignments which is referred to as gating. The distance measure is only used to prohibit assignments with a distance value above a threshold but is not integrated into the matching distance. In this section, the influence of such a gating mechanism on the tracking performance is analyzed. Besides Mahalanobis distance, IoU, DIoU and appearance cosine distance are tested as gating measures. The combination of DIoU and cosine distance from

Table 3.5: Gating Results.

Gating	✗	IoU	DIoU	Cosine	Mahal
HOTA	69.41	69.45	**69.47**	69.41	69.42

Section 3.3 is taken as distance for matching. Tracking results with additional gating are depicted in Table 3.5. In the experiments, only small HOTA gains up to 0.06 points are achieved, although the gating thresholds have been tuned carefully. For this reason and because a too small gating threshold can degrade the tracking performance, gating is not used in the rest of this work.

4 Multiple Matching Stages

It is the common practice in MPT to solve the assignment problem for all tracks and detections at once as also done in this study so far. However, a few works split the set of tracks or detections into subsets which are processed one after another [1, 28, 30, 40, 46]. Two strategies are revisited – a matching cascade from the famous DeepSORT [40] tracker (Section 4.1) and the BYTE [46] association method which recently lead to notable improvements (Section 4.2).

4.1 DeepSORT Matching Cascade

Given an example track $T^t = [D^{t_{\text{init}}}, \ldots, D^{t-k}]$ at time step t, its age a is defined as the time since the track has been observed for the last time. For this example track, $a = k$ holds. Note that in this definition, *active* tracks have an age of 1, whereas *inactive* tracks have an age greater than 1. In DeepSORT [40], tracks with an age of 1 are matched with all available detections. Then, all tracks with an age of 2 are matched with the remaining unmatched detections and so forth. The motivation behind this strategy is to favor tracks that have been observed recently, since the accuracy of propagated track locations decreases over time. However, in StrongSORT [10] – a further development of DeepSORT – it is found that this matching cascade harms the tracking performance when the

Table 4.1: DeepSORT (DS) Matching Cascade Results.

DS Matching Cascade	HOTA	DS Matching Cascade	HOTA
✗	**69.41**	✓	67.86

tracker gets stronger because the additional prior constraints limit the matching accuracy [10]. To investigate the influence of the DeepSORT matching cascade on the so-far best tracker of this study (Section 3.3), it is utilized in an additional experiment. The result is shown in Table 4.1. Integrating the matching cascade significantly decreases HOTA by 1.55 points which confirms the results from [10]. Obviously, this matching cascade is not used in further experiments.

4.2 BYTE Association

Usually, only high-confident detections are used in the association as low-confident ones include many false positives that harm the tracking performance. In contrast, an association technique named BYTE is proposed in [46], which allows to make use of low-confident detections in a second matching stage. Detections with confidence score below s_{track} are not removed but compared to unmatched tracks that have not been assigned a high-confident detection in the first association. Since the low-confident detections are not utilized to start new tracks but only for assignment to already tracked targets, the overall performance can be largely increased. The authors of [46] show this by applying the BYTE association to different trackers which leads to consistent improvements. Among the trackers, the varying distance measures are kept in the first matching stage. However, in the newly introduced second matching stage, only the IoU distance is leveraged as the authors argue that most tracks in this stage suffer from occlusion or motion blur, where appearance features are not reliable [46].

Since the tracking pipeline of this study differs quite a lot from other approaches with the improved motion modelling from Section 3.1 and the combined distance measure from Section 3.3, it is also experimented with appearance-based cosine distance next to other distance measures in the second association stage. Although it is not mentioned in the paper [46], the publicly available source

Table 4.2: Second Matching Results. The best result using only one matching stage is achieved with a combination of DIoU and cosine distance: HOTA = 69.41 (Table 3.3).

Use Inactive	Distance	HOTA	Use Inactive	Distance	HOTA
✗	IoU	69.66	✓	IoU	**70.29**
✗	DIoU	69.70	✓	DIoU	70.22
✗	Cosine	69.68	✓	Cosine	70.22
✗	DIoU+Cosine	69.73	✓	DIoU+Cosine	70.14

code reveals that only *active* tracks are considered in the second matching stage. In this study, it is also tested whether the inclusion of *inactive* tracks in this stage can be beneficial. Resulting HOTA values of the conducted experiments related to the second matching stage can be found in Table 4.2.

In contrast to [46], appearance-based distances like the cosine distance and the combination with DIoU also achieve good results. Compared to the baseline, where only one matching stage is used ($\text{HOTA} = 69.41$), gains up to 0.32 HOTA are obtained. Note that the applied distance threshold of the second stage $d_{\max,2}$ influences the performance, so it is tuned carefully for each configuration.

When additionally inactive tracks are used, IoU-based matching results in 70.29 HOTA which is a huge improvement compared to using only active tracks in the second matching stage. It is observed that the optimized distance threshold $d_{\max,2}$ is much lower than in the implementation of [46] (0.19 vs. 0.5). Setting such a low threshold ensures that only inactive tracks with accurately predicted locations can be matched. With the usage of inactive tracks, no prior constraints are applied that could limit the matching accuracy, similar as the matching cascade of DeepSORT (see Section 4.1). Since the IoU-based matching in the second stage yields an improvement of 0.88 HOTA in comparison to the one-stage baseline, it is leveraged in all further experiments.

Table 5.1: Parameter Tuning Results.

	$i_{\max}$	N	Strategy	s_{init}	s_{track}	$d_{\max,1}$	$d_{\max,2}$	HOTA
Before Tuning	30	10	mean+min	0.7	0.6	3.18	0.19	70.29
After Tuning	**28**	**16**	**mean**	0.7	0.6	**3.13**	0.19	**70.77**

5 Parameter Tuning and Sensitivity

Before evaluating different motion models to develop a baseline tracker for this study, some parameters had to be set initially: the number of frames an inactive track is kept ($i_{\max}$), confidence thresholds for detections to be considered in the association and to start new tracks (s_{track} and s_{init}) and the maximum distance threshold to prevent unlikely assignments ($d_{\max,1}$). Extending the tracking framework with additional components, further parameters are introduced. Integrating appearance features (Section 3.2), the number of past time steps in the feature bank (N) and the strategy how to calculate the cosine distance (min, mean, mean+min) have to be chosen. With the utilization of a second matching stage (Section 4.2), another maximum distance threshold has to be set ($d_{\max,2}$). Since the number of parameters has increased during this study, some might not be set optimal anymore. For this reason, an extensive grid search has been performed to find the best parameter configuration of the tracker. The results are summarized in Table 5.1, whereby parameters that have changed are bold. Optimizing the set of parameters gives a notable plus of 0.48 HOTA.

To get a better understanding of the importance and the sensitivity of the tracking parameters, hundreds of experiments have been conducted in that each parameter has been varied within a decent interval around the best value (specified by grid search), while all the other parameters were fixed at their optimum. The resulting HOTA curves are shown in Figure 5.1.

The confidence threshold s_{init} of a detection to initialize a new track obviously has a large influence on the tracking performance. With a too low threshold, many false positives are introduced, whereas with a too large threshold, many targets are missed. The track threshold s_{track} decides, whether a detection is considered

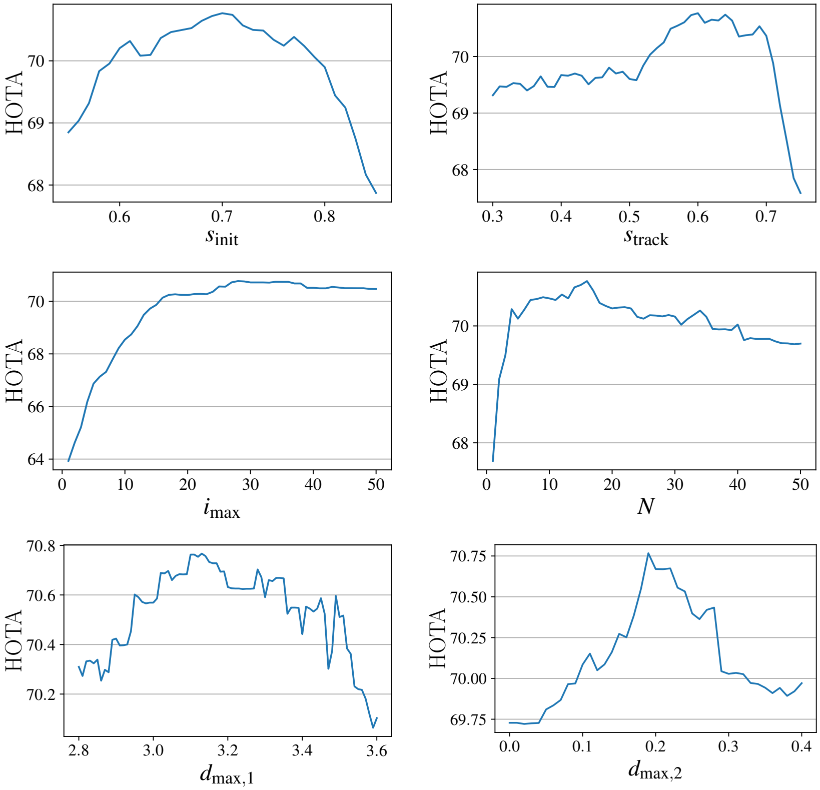


Figure 5.1: Sensitivity of Tracking Parameters.

in the first association stage or the second. Priority is given to detections with confidence above s_{track} and in the second stage, a stricter maximum distance is enforced for the lower-confident detections. In the experiments, $s_{\text{track}} = 0.6$ achieved the best results. This value is 0.1 smaller than s_{init}, which equals the relation in [46].

Another important parameter is i_{max}. The higher the value, the longer the occlusions that can be bridged. If this so-called inactive patience, however, is too high, wrong assignments to inactive tracks can occur, since the location accuracy decreases over time. For the number of appearance features in the feature bank, the empirically found best value is $N = 16$. If only a few features are considered, the full potential of the temporal information is not leveraged, whereas features from too far in the past might not be representative anymore due to changes in appearance.

The best values for the matching thresholds $d_{\max,1}$ and $d_{\max,2}$ are 3.13 and 0.19, respectively, on MOT17 Val. Too small values prevent correct assignments while too large values allow wrong assignments. The fluctuations in the corresponding HOTA curves are caused by the small depicted HOTA ranges and in addition – like for all parameters – are attributable to the finite dataset size.

6 Post-processing

The so-far developed tracking framework works fully online which means that the tracking results are final after processing each frame of the input video. Some applications without real-time requirements allow to refine the tracking results with post-processing techniques to improve the performance. Besides simple linear interpolation of fragmented tracks, two more sophisticated post-processing methods introduced in StrongSORT [10] are investigated – the Appearance Free Link (AFLink) model and Gaussian Smoothed Interpolation (GSI).

AFLink is a small convolutional neural network that takes the center positions and corresponding frames of two tracks as input and computes a connectivity score solely based on spatio-temporal information. If this connectivity score is higher than a threshold and some spatio-temporal constraints are fulfilled, the two tracks are linked hypothesizing that they belong to the same target. Implementation details can be found in the StrongSORT paper [10].

Since the maximum gap in a fragmented track is $i_{\max} = 28$ (see Table 5.1), which corresponds to roughly one second on MOT17, many of those gaps can be successfully filled with linear interpolation (LI). However, in some cases the linear approximation is not accurate enough. Therefore, GSI employs Gaussian process regression [39] to model non-linear motion of targets. Another

Table 6.1: Post Processing Results.

AFLink	Interpolation	HOTA	AFLink	Interpolation	HOTA
✗	✗	70.77	✓	LI	72.52
✓	✗	70.84	✓	GSI	**72.81**

advantage compared to the linear interpolation is that the noisy trajectories are smoothed. It is referred to [10] for details of the GSI algorithm.

Table 6.1 depicts the post-processing results after application of AFLink as well as linear and Gaussian smoothed interpolation. AFLink slightly improves HOTA by 0.07 points. Since the model does not integrate appearance information, strict spatio-temporal constraints have to be enforced to prevent wrong connections. For potentially larger improvements, more sophisticated approaches like ReMOT [42] could be applied which is left for future work. Based on appearance features enhanced by self-supervised learning, tracks are not only merged in [42], but erroneous tracks consisting of different targets are additionally cut apart. Looking at the results of the two interpolation techniques, it is observed that both significantly improve the overall performance with gains of 1.68 and 1.97 points in HOTA for LI and GSI, respectively. As expected, the non-linear GSI outperforms the simple linear interpolation.

7 Ablation Study

In this work, several components related to the association task in MPT have been investigated and a strong tracking framework based on the TBD paradigm has been developed. Starting from a simple baseline with standard Kalman filter (KF) for track propagation and IoU distance as association metric, extensions of the KF and a camera motion compensation (CMC) module were introduced. Then, motion-based matching was combined with appearance-based matching leading to a sophisticated distance measure. Afterwards, low-confident detections were integrated into the association within a second matching stage. Finally, parameter tuning and post-processing were performed. All these steps lead to consistent

Table 7.1: Ablation Study. Abbreviations: CMC = camera motion compensation, NSA+HP = Noise Scale Adaptive Kalman filter + height preservation, DIoU+Cosine = Distance IoU + cosine distance, PT = parameter tuning, PP = post-processing (AFLink + Gaussian smoothed interpolation).

CMC	NSA+HP	DIoU+Cosine	2$^{\text{nd}}$ Matching	PT	PP	HOTA
✗	✗	✗	✗	✗	✗	67.40 (±0.00)
✓	✗	✗	✗	✗	✗	68.13 (+0.73)
✓	✓	✗	✗	✗	✗	68.67 (+0.54)
✓	✓	✓	✗	✗	✗	69.41 (+0.74)
✓	✓	✓	✓	✗	✗	70.29 (+0.88)
✓	✓	✓	✓	✓	✗	70.77 (+0.48)
✓	✓	✓	✓	✓	✓	**72.81** (+2.16)

improvements of the overall tracking performance measured in HOTA that are summarized in Table 7.1. Besides the offline post-processing, the largest gains in the online tracker come from the second matching stage (+0.88 HOTA), the combined distance measure (+0.74 HOTA), and the CMC model (+0.73 HOTA). All components together boost HOTA significantly from 67.40 to 72.81.

8 Comparison with the State-of-the-Art

The final tracker of this study is named *StrongTBD* because of the large improvements w.r.t. the TBD baseline from Section 2. StrongTBD is compared to the state-of-the-art on MOT17 [22] and MOT20 [6] test splits in this section. Before delving into the results, it should be noted that annotations of the test splits are not publicly available and evaluation is done by submitting the tracking results to the official server (*motchallenge.net*). Besides HOTA, other performance measures such as MOTA [3] and IDF1 [25] are also computed. To prevent parameter tuning on the test data, one is restricted to four submissions. However, the tracking performance is highly dependent on the setting of some parameters, especially on the detection thresholds s_{init} and s_{track} (see Section 5). For example, changing s_{init} and s_{track} from 0.7 to 0.4 and 0.6 to 0.3, respectively, MOTA increases by approximately 10 points on the MOT20-08 sequence in the

Table 8.1: State-of-the-Art Methods on MOT17.

Method	MOTA	IDF1	HOTA	FP	FN	IDSW
MAATrack [30]	79.4	75.9	62.0	37320	77661	1452
RTU++ [36]	79.5	79.1	63.9	29508	84618	1302
StrongSORT [10]	79.6	79.5	64.4	27876	86205	1194
SAT [37]	80.0	79.8	64.4	25125	86505	1356
ByteTrack [46]	80.3	77.3	63.1	25491	83721	2196
QuoVadis [7]	80.3	77.7	63.1	25491	83721	2103
FOR_Tracking [23]	80.4	77.7	63.6	28674	79452	2298
BoT-SORT [1]	80.5	80.2	65.0	**22521**	86037	1212
ByteTrackV2 [28]	80.6	78.9	63.6	35208	**73224**	1239
StrongTBD	**81.6**	**80.8**	**65.6**	24171	78759	**954**

Table 8.2: Values of s_{init} on MOT17 and MOT20 test sets.

MOT17	01	03	06	07	08	12	14	MOT20	04	06	07	08
s_{init}	0.8	0.75	0.75	0.7	0.7	0.8	0.65	s_{init}	0.7	0.4	0.7	0.4

submissions of StrongTBD. This and the fact that some works do not report their applied thresholds makes a fair comparison among methods difficult. The trend of using various thresholds for different sequences of the datasets [1, 28, 46] further complicates the comparison.

Nevertheless, Table 8.1 lists the 10 best performing trackers on MOT17 with ascending MOTA values. StrongTBD achieves the highest values in MOTA, IDF1, and HOTA. Furthermore, it has the least number of identity switches (IDSW). Despite the aforementioned comparability issues, the results show that the developed tracker can compete with the state-of-the-art. To make these results reproducible, the s_{init} values of the submission for the sequences of MOT17 are reported in Table 8.2. Note that for the tracking thresholds $s_{\text{track}} = s_{\text{init}} - 0.1$ holds, just as in [1, 28, 46].

Table 8.3: State-of-the-Art Methods on MOT20.

Method	MOTA	IDF1	HOTA	FP	FN	IDSW
SAT [37]	75.0	76.6	62.6	**15549**	113136	**816**
OC-SORT [5]	75.7	76.3	62.4	19067	105894	942
RTU++ [36]	76.5	76.8	62.8	19247	101290	971
FOR_Tracking [23]	76.8	76.4	61.4	27112	91254	1443
ByteTrackV2 [28]	77.3	75.6	61.4	22867	93409	1082
ReMOT [42]	77.4	73.1	61.2	28351	**86659**	1789
ByteTrack [46]	77.8	75.2	61.3	26249	87594	1223
QuoVadis [7]	77.8	75.7	61.5	26249	87594	1187
BoT-SORT [1]	77.8	**77.5**	63.3	24638	88863	1313
StrongTBD	**78.0**	77.0	**63.6**	25473	87330	1101

Table 8.2 also shows the values of s_{init} on the final submission on the MOT20 dataset. The results on this benchmark of the 10 best performing trackers are given in Table 8.3. StrongTBD obtains the highest MOTA and HOTA as well as the second highest IDF1, which confirms the competitiveness of the developed tracking framework. Note that the parameter configuration of StrongTBD has been adapted on the MOT20 dataset in order to be more comparable to the second best entry BoT-SORT [1]. More precisely, the input resolution of the MOT20-04 and MOT20-07 sequences are set to 1600×896 pixels, while a resolution of 1920x736 pixels is used in MOT20-06 and MOT20-08. In addition, an IoU distance threshold of 0.7 is integrated, which helps to prevent IDSW in crowded scenes. Furthermore, the same initialization strategy as in [1, 28, 46] is followed, in that new tracks are tentative until they get confirmed with an assigned detection in the subsequent frame. As already discussed in Section 2, such a strategy is beneficial if the threshold s_{init} is quite low which is the case for MOT20-06 and MOT20-08 (see Table 8.2). The target density on MOT20 with 127 persons per image is much higher than on MOT17 with only 21.1 persons per image [32]. As StrongTBD has been developed on MOT17 Val, some design choices are not optimal for very crowded scenes as in MOT20. In the future, more focus should be put on tracking in such challenging scenarios.

9 Conclusion

In this study, all components of the association task in MPT have been analyzed in detail. Two of the most important findings are that the combination of motion- and appearance-based distance measures outperforms the sole usage of one information type and that leveraging low-confident detections in a second association stage yields significant improvements. The influence of various tracking components from motion models to post-processing techniques has been investigated as well as the sensitivity of the results to the setting of tracking parameters. The empirical results were used to develop a sophisticated tracking-by-detection method that achieves state-of-the-art performance on the two challenging MPT benchmarks MOT17 and MOT20. Further potential lies in enhancing the association accuracy in very crowded scenes as in the MOT20 dataset, which sould be investigated more thoroughly in the future.

References

[1] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky. “BoT-SORT: Robust Associations Multi-Pedestrian Tracking”. In: *CoRR* abs/arXiv:2206.14651 (2022).

[2] P. Bergmann, T. Meinhardt, and L. Leal-Taixé. “Tracking Without Bells and Whistles”. In: *ICCV*. 2019, pp. 941–951.

[3] K. Bernardin, A. Elbs, and R. Stiefelhagen. “Multiple Object Tracking Performance Metrics and Evaluation in a Smart Room Environment”. In: *ECCV Workshops*. 2006.

[4] A. Bewley et al. “Simple online and realtime tracking”. In: *ICIP*. 2016, pp. 3464–3468.

[5] J. Cao et al. “Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking”. In: *CoRR* abs/arXiv:2203.14360 (2022).

[6] P. Dendorfer et al. “MOT20: A benchmark for multi object tracking in crowded scenes”. In: *CoRR* abs/arXiv:2003.09003 (2020).

[7] P. Dendorfer et al. “Quo Vadis: Is Trajectory Forecasting the Key Towards Long-Term Multi-Object Tracking?” In: *CoRR* abs/arXiv:2210.07681 (2022).

[8] Y. Du et al. “GIAOTracker: A comprehensive framework for MCMOT with global information and optimizing strategies in VisDrone 2021”. In: *ICCV Workshops*. 2021, pp. 2809–2819.

[9] Y. Du et al. “Looking Beyond Two Frames: End-to-End Multi-Object Tracking Using Spatial and Temporal Transformers”. In: *CoRR* abs/arXiv:2103.14829 (2021).

[10] Yunhao Du et al. “StrongSORT: Make DeepSORT Great Again”. In: *CoRR* abs/arXiv:2202.13514 (2022).

[11] A. Ess et al. “A mobile vision system for robust multi-person tracking”. In: *CVPR*. 2008.

[12] G. D. Evangelidis and E. Z. Psarakis. “Parametric Image Alignment Using Enhanced Correlation Coefficient Maximization”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 30.10 (2008), pp. 1858–1865.

[13] M. A. Fischler and R. C. Bolles. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. In: *Commun. ACM* 24.6 (1981), pp. 381–395.

[14] Z. Ge et al. “YOLOX: Exceeding YOLO Series in 2021”. In: *CoRR* abs/arXiv:2107.08430 (2021).

[15] S. Han et al. “MAT: Motion-aware multi-object tracking”. In: *Neurocomputing* 476 (2022), pp. 75–86.

[16] R. E. Kalman. “A New Approach to Linear Filtering and Prediction Problems”. In: *J. basic Eng.* 82.1 (1960), pp. 35–45.

[17] H. W. Kuhn and B. Yaw. “The Hungarian Method for the Assignment Problem”. In: *Naval Research Logist. Quart.* 2.1–2 (1955), pp. 83–97.

[18] J. Li et al. “SimpleTrack: Rethinking and Improving the JDE Approach for Multi-Object Tracking”. In: *Sensors* 22.15 (2022).

[19] Q. Liu et al. “GSM: Graph Similarity Model for Multi-Object Tracking”. In: *IJCAI*. 2020, pp. 530–536.

[20] J. Luiten et al. “HOTA: A Higher Order Metric for Evaluating Multi-object Tracking”. In: *Int. J. Comput. Vis.* 129.2 (2021), pp. 548–578.

[21] H. Luo et al. “Bag of Tricks and a Strong Baseline for Deep Person Re-Identification”. In: *CVPR Workshops*. 2019, pp. 1487–1495.

[22] A. Milan et al. “MOT16: A Benchmark for Multi-Object Tracking”. In: *CoRR* abs/arXiv:1603.00831 (2016).

[23] M. H. Nasseri et al. “Fast Online and Relational Tracking”. In: *CoRR* abs/arXiv:2208.03659 (2022).

[24] H. Rezatofighi et al. “Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression”. In: *CVPR*. 2019, pp. 658–666.

[25] E. Ristani et al. “Performance Measures and a Data Set for Multi-target, Multi-camera Tracking”. In: *ECCV Workshops*. 2016, pp. 17–35.

[26] E. Rublee et al. “ORB: An efficient alternative to SIFT or SURF”. In: *ICCV*. 2011, pp. 2564–2571.

[27] S. Shao et al. “CrowdHuman: A Benchmark for Detecting Human in a Crowd”. In: *CoRR* abs/arXiv:1805.00123 (2018).

[28] D. Stadler and J. Beyerer. “BYTEv2: Associating More Detection Boxes Under Occlusion for Improved Multi-Person Tracking”. In: *ICPR Workshops*. 2022.

[29] D. Stadler and J. Beyerer. “Improving Multiple Pedestrian Tracking by Track Management and Occlusion Handling”. In: *CVPR*. 2021, pp. 10958–10967.

[30] D. Stadler and J. Beyerer. “Modelling Ambiguous Assignments for Multi-Person Tracking in Crowds”. In: *WACV Workshops*. 2022, pp. 133–142.

[31] D. Stadler and J. Beyerer. “Multi-Pedestrian Tracking with Clusters”. In: *AVSS*. 2021.

[32] D. Stadler and J. Beyerer. “On the Performance of Crowd-Specific Detectors in Multi-Pedestrian Tracking”. In: *AVSS*. 2021.

[33] P. Sun et al. “TransTrack: Multiple Object Tracking with Transformer”. In: *CoRR* abs/arXiv:2012.15460 (2021).

[34] S. Tang et al. “Multiple People Tracking by Lifted Multicut and Person Re-identification”. In: *CVPR*. 2017, pp. 3701–3710.

[35] Q. Wang et al. “Multiple Object Tracking With Correlation Learning”. In: *CVPR*. 2021, pp. 3876–3886.

[36] S. Wang et al. “Extendable Multiple Nodes Recurrent Tracking Framework With RTU++”. In: *IEEE Trans. Image Process.* 31 (2022), pp. 5257–5271.

[37] S. Wang et al. “Tracking Game: Self-adaptative Agent based Multi-object Tracking”. In: *ACM Multimedia*. 2022, pp. 1964–1972.

[38] Z. Wang et al. “Towards Real-Time Multi-Object Tracking”. In: *ECCV*. 2020, pp. 107–122.

[39] C. K. I. Williams and C. E. Rasmussen. “Gaussian Processes for Regression”. In: *NIPS*. 1995, pp. 514–520.

[40] N. Wojke, A. Bewley, and D. Paulus. “Simple online and realtime tracking with a deep association metric”. In: *ICIP*. 2017, pp. 3645–3649.

[41] J. Xu et al. “Spatial-Temporal Relation Networks for Multi-Object Tracking”. In: *ICCV*. 2019, pp. 3987–3997.

[42] F. Yang et al. “ReMOT: A model-agnostic refinement for multiple object tracking”. In: *Image Vis. Comp.* 106 (2021).

[43] F. Zeng et al. “MOTR: End-to-End Multiple-Object Tracking with Transformer”. In: *ECCV*. 2022, pp. 659–675.

[44] H. Zhang et al. “ResNeSt: Split-Attention Networks”. In: *CVPR Workshops*. 2022, pp. 2735–2745.

[45] S. Zhang, R. Benenson, and B. Schiele. “CityPersons: A Diverse Dataset for Pedestrian Detection”. In: *CVPR*. 2017, pp. 4457–4465.

[46] Y. Zhang et al. “ByteTrack: Multi-Object Tracking by Associating Every Detection Box”. In: *ECCV*. 2022.

[47] Z. Zheng et al. “Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression”. In: *AAAI*. 2020, pp. 12993–13000.

[48] X. Zhou, V. Koltun, and P. Krähenbühl. “Tracking Objects as Points”. In: *ECCV*. 2020, pp. 474–490.

A Baseline for Cross-Domain Fine-Grained Vehicle Classification in a Supervised Partially Zero-Shot Setting

Stefan Wolf

Vision and Fusion Laboratory
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT), Germany
stefan.wolf@kit.edu

Abstract

Fine-grained vehicle classification is an important task particularly for security applications like searching for cars of suspects who abuse stolen license plates. However, data privacy and the large number of existing car models render it highly difficult to create a large up-to-date dataset for fine-grained vehicle classification with surveillance images. While a large number of images of vehicles are available in the web due to car selling sites, they have a perspective which is vastly different to surveillance images. Domain adaptation is the field of research that uses domain-wise inappropriate images for training of classification models with the target of running accurate inference on images of a different domain. Since the widely considered unsupervised and semi-supervised domain adaptation settings are unrealistic for fine-grained vehicle classification, we establish a baseline for cross-domain fine-grained vehicle classification in a supervised partially zero-shot setting. Our results indicate that existing domain adaptation methods like domain adversarial training and triplet loss are still advantageous for this setting and we show the benefit of distance-based classification for this task.

1 Introduction

Fine-grained classification tasks like vehicle make and model recognition are relying on large datasets for training. These are needed since the small inter-class variance compared to the large intra-class variance are required to be properly approximated by the learned model. While in the web, a large amount of images for different cars are provided by e.g. car selling sites, fine-grained classification is often applied in different domains. For example, vehicle make and model recognition is useful for security applications like manhunt when applied to cameras on highways which provide a surveillance perspective. However, for these perspectives, the availability of data is scarce. The situation is worsened by the high rate of car manufacturers proposing new vehicle models .

To approach the lack of data, domain adaptation methods can enable the use of the large-scale availability of data of different domains like web-nature images to perform tasks like classification in domains which have a limited availability of data like surveillance. While domain adaptation has been widely approached [33, 45, 48, 22, 4, 25, 11] and also specifically for fine-grained classification [10, 31, 34, 35, 44] applications, an unsupervised or semi-supervised domain adaptation setting is commonly assumed. In these settings, a large number of images is present in the target domain for all classes but the labels aren't present for any or only a part of the images. However, for real-world use-cases, the assumption to have data for all classes is hard to fulfill since it can only be assured if labels would be present. Thus, we focus on a different domain adaptation setting: a supervised partially zero-shot setting [38]. This setting assumes that for a large number of base classes, images and labels are available for both domains while for a small number of novel classes, images and labels are only available for the source domain. For these novel classes, no images are available at all for the target domain during training. However, the evaluation on these novel classes with images from the target domain is the main focus of the setting.

Since the research for such a setting is rather small [32], we provide an extensive evaluation of existing domain adaptation methods to find a good baseline for further research. Besides the widely applied domain adversarial learning [8], we explore the use of metric learning with a triplet loss which also has shown advantages for classification across domains [20, 32].

Based on these experiments, we found that a typical softmax classifier only achieves a low classification accuracy for the novel classes. However, a domain adversarial loss heavily increases the accuracy. A distance-based classifier with a combination of a cross entropy loss and a triplet loss showed promising results which can further be improved by the use of a domain adversarial loss resulting in the overall best model.

In Section 2, existing works in the fields of fine-grained classification, cross-domain classification and cross-domain fine-grained classification are introduced. In Section 3.1, the evaluated methods are described and the evaluation results are shown in Section 4. A conclusion of this work is given in Section 4.

2 Related work

In this chapter, an overview of the literature in the fields of fine-grained classification and cross-domain classification as well as works which employ cross-domain classification for fine-grained classification tasks is given.

2.1 Fine-grained classification

Various approaches have been used to improve the accuracy for fine-grained classification. While all recent approaches share their basis of deep neural networks, there are several different extensions and they can be structured into the following categories. **Part-based models** first detect relevant regions like specific parts of a vehicle before the crops of these parts are fed into a convolutional neural network (CNN) [7, 15, 28, 41]. This reduces the feature space to significant parts and thus, reduces the risk of overfitting. **Bilinear CNNs** employ two networks to separate the localization and the extraction of important features. The networks are combined by calculating the outer product of both resulting feature vectors [23]. Several extensions have been proposed to improve the accuracy and efficiency of bilinear CNNs [9, 19, 43]. Multiple authors employ **multi-task learning** by learning an auxiliary task like predicting the viewpoint of the image that provides support for the main task of fine-grained classification. The auxiliary task is performed only during training to improve the learned features [3] or also during inference to provide the network with additional information [24, 29]. **Hierarchical classification** exploits that fine-grained categories are usually defined on multiple layers, e.g. make, model and year of a car. This technique was explored by training multiple layers of the hierarchy in a round-robin manner [16] and by training cascaded classifiers [2]. **Metric learning** has also been applied to improve the features by minimizing intra-class variance and maximizing inter-class variance [17, 30, 42]. **Temporal classification** uses videos as input modality for fine-grained object recognition [1, 46, 18] instead of single images as done by most works. **Webly-supervised classification** gathers additional data from the web with image databases like Flickr providing images with additional meta information that can be used for defining labels [6, 39].

2.2 Cross-domain classification

Domain adaptation is usually employed if classification has to be done in a domain for which a lack of data exists. The lack of data can be in the form of missing images or missing annotations. Mostly an unsupervised scenario is considered which contains abundant but unlabaled data for the target domain. To approach a cross-domain setting, multiple methods have been proposed. We follow the taxonomy of Wang and Deng [33] for the categorization of the approaches. **Discrepancy-based domain adaptation** methods are based on a criterion during fine-tuning to increase the accuracy for the target domain. Proposed criteria are class-based [31, 45], statistic-based [48], architecture-based [22] or geometry-based [4]. **Adversarial-based domain adaptation** methods target a domain confusion of the trained network which disables the possibility of exploiting the domain of an image for the classification decision. This can be done by generative approaches which transform the appearance of a source sample such that it can not be distinguished from the distribution of target samples [25]. Non-generative approaches have also been explored by using domain adversarial training with a domain classifier that is preceded by a gradient reversal layer during training. This leads to features which are invariant in regard to distinguishing the domains. **Reconstruction-based domain adaptation** methods reconstruct samples from either domain to the other domain to create a domain-invariant representation. This has been explored by using a combination of an encoder and a decoder [11] as well as using a Cycle-GAN [47] that keeps semantic information intact by using a cycle-consistency constraint [14].

2.3 Cross-domain fine-grained classification

Some researchers have already addressed fine-grained classification in a cross-domain setting. Gebru et al. [10] exploit the hierarchical nature of fine-grained classification by adding an attribute consistency loss that enforces a matching of coarse-grained attributes like vehicle types to the fine-grained category. With the coarse-grained attribute prediction being a significantly easier task, it is more domain invariant and thus, supports stabilizing the fine-grained prediction due to the new consistency loss. Tzeng et al. [31] and Wang et al. [34] also exploit the attribute and coarse-grained labels inherent to fine-grained classification tasks to improve the domain adaptation. Wang et al. [35] extends adversarial domain-level adaptation by a category-level domain alignment for semi-supervised domain adaptation. Additionally, a part-wise classification to optimize the fine-grained classification accuracy is introduced. Yu et al. [44] achieve a class confusion by training separate class labels for each domain in a pre-training phase and swapping the class labels in a fine-tuning phase with the target of achieving domain confusion while compared to domain adversarial training, keeping the class-separability of the features intact during the adaption process.

3 Methods

In this section, the evaluated methods are described. They can be mainly divided by the type of classification. We evaluate a softmax classifier and a distance-based classifier. As feature extracting backbone, we use ResNet-50 [12] for both variants. On top of both variants, we evaluate the usage of domain adversarial training [8] to improve the domain invariance. While only common for distance-based classification, we evaluate a triplet loss [36] for both variants due to the reported advantages in regards to cross-domain classification [21].

3.1 Softmax classifier

The softmax classifier employs a fully-connected layer to predict as many logits as number of classes and afterwards applies a softmax activation layer to normalize the scores. On top of this output, a cross entropy loss is used to calculate an error measurement.

Additionally, we evaluate the use of a domain adversarial head and an auxiliary triplet loss to improve the domain invariance of the features. Both additions are applied directly on the features of the backbone.

3.2 Distance-based classifier

For the distance-based classifier, during inference, we feed each preprocessed image into the backbone network and calculate the distance between the feature vector of the sample and a prototype feature vector for each class. We choose the class as final prediction for which the distance has the lowest value. The prototype is calculated as the mean of all training samples of a class from the source domain. We also evaluated the use of a medoid instead of a mean but the results indicated an advantage for the mean. Regarding the distance measure, we evaluated the euclidean norm and the negative cosine similarity with the results showing a clear advantage for the negative cosine similarity while the euclidean norm usually prevented the network from converging properly. Since the cosine similarity is originally a similarity instead of a distance measure, we use the negative of the cosine similarity as distance measure. The classification can be described by the following formulas:

$$\mathbf{p_c} = \frac{1}{|X_c|} \sum_{x \in X_c} f(x) \tag{3.1}$$

$$c(x) = \underset{c \in C}{\operatorname{argmin}} - \frac{f(x) \cdot \mathbf{p_c}}{||f(x)|| ||\mathbf{p_c}||} \tag{3.2}$$

where $\mathbf{p_c}$ is the feature prototype for the class c, X_c is the set of training images of a class c from the source domain, f is the backbone feature extractor, $c(x)$ is the predicted class for an image x and C is the set of known classes.

During training, we apply a cross entropy loss with a softmax activation on top of a fully-connected layer. Since the cross entropy loss tends to learn features which are highly dependent on the domain, we use a triplet loss as additional loss function that regularizes the network in regards to the domains. Additionally, the triplet loss ensures that the chosen distance measure is appropriate for the features during inference. After training, the fully-connected layer is dropped and the extracted features are directly used as described above.

3.3 Domain adversarial training

Ganin et al. [8] proposed a domain adversarial training method. It applies a simple domain classifier on top of the features extracted by the backbone and inserts a gradient reversal layer between the network and the domain classifier. The gradient reversal layer leads to learning features which are most inappropriate for a classification of the domain and thus, the features are expected to be invariant in regards to the domain. Therefore, the classification loss which is applied in parallel will focus on learning features which are inherent to the class instead of exploiting the domain.

For the domain classification head, we employ two hidden fully-connected layers with 1024 channels with each being followed by a ReLU activation and a batch normalization layer. A final fully-connected layer with a single output channel which is followed by a sigmoid activation predicts the domain. A binary cross entropy is applied as training loss for the domain classification.

The gradient reversal layer includes a gating that controls the influence of the reversed gradient of the domain classification loss onto the main network. We call this parameter λ. A λ of 1 means an unhindered influence while a λ of 0 means that the domain classification has no influence on the main network at all. A good choice of λ might depend on the current state of training and a pre-set value is probably not appropriate. Our results showed that the loss coupling of λ proposed by Wiedemer et al. [37] was superior to a pre-set value and an increasing schedule of λ as it was originally proposed for the domain adversarial training [8]. The loss coupling sets λ for each iteration based on the domain classification loss value of the previous iteration. The exact formula is $\lambda_i = \exp(-L_{d,i-1})$ with λ_i being the set λ for the iteration i and $L_{d,i-1}$ being the domain classification loss for iteration $i - 1$. This ensures that the domain classification only has a strong influence on the main network if the loss is low meaning that the domain classifier is able to classify the domain adequately. In case of a high domain loss, the domain classifier is not able to classify the domains properly and will not provide a good domain adversarial loss.

3.4 Triplet loss

A triplet loss [36] explicitly minimizes the distance of features of the same class while maximizing the distance of features of different classes with respect to a chosen distance measure. While the cross entropy loss also tends to show a similar behavior, it only enforces a linear separability of classes which can result in features of a single class still being spread in feature space. This can be particularly dramatic for cross-domain scenarios for which the distribution of images is different between training and inference. Thus, we apply a triplet loss as additional loss that directly minimizes the distance of features of the same class.

4 Experiments

We execute quantitative evaluations to find a good baseline for cross-domain classification under a supervised partially zero-shot setting. First, the settings of the comparisons are described. Afterwards, the results are discussed. The comparisons include ablation studies for a softmax classifier, ablation studies for a distance-based classifier and a comparison between both approaches.

4.1 Settings

The datasets used for the experiments are described first. Afterwards, the evaluation metrics and training details are reported.

4.1.1 Dataset

As dataset, we choose CompCars [40] which is one of the largest fine-grained vehicle classification datasets available and consists of a web-nature part (CompCars Web) and a surveillance-nature part (CompCars SV). The CompCars Web has a predefined split of 16.016 training images and 14.939 test images. The predefined split of the CompCars SV contains 31.148 training images and 13.333 test images.

While the CompCars Web is labeled according to the make, model and year of a specific car, the CompCars SV is only labeled up to the model of a car and lacks the year as annotation. Thus, we also only consider the model for all cars in CompCars Web. This results in a total of 431 classes for CompCars Web and a total of 281 classes for CompCars SV. We identify the intersection of both sets of classes and use only these for our experiments. Thus, we consider a total of 181 classes. Based on this set of classes, we create three different random splits of base and novel classes with the base classes containing 90% and the novel classes containing 10% of the classes. While during training, for the base classes abundant labeled images are available in both domains, we restrict the availability of data for the novel classes to the source domain of CompCars Web and no images from CompCars SV are available for the novel classes. For each experiment, a model is trained and evaluated on each split and the results are averaged.

4.1.2 Evaluation metric

We use the F1 score on the CompCars SV as main metric for our experiments. We report the class-wise F1 score averaged over the base and the novel classes separately. Since our focus is on adding new classes to the classification, we focus mainly on the F1 score of the novel classes. Due to images of all classes being included in the test set, base classes still influence the score of the novel classes and vice versa. This is sensible since a network only focused on the prediction of novel classes should still be able to distinguish them from the total of all base classes even when distinguishing the base classes might be of minor importance.

4.1.3 Training details

We choose SGD as optimizer with an initial learning rate of 0.04 and a learning rate reduction by 10× is applied after 2500 iterations. We apply a momentum of 0.9 and a weight decay of 10^{-4}. The training is running for 12000 iterations in total. A batch size of 512 per GPU with two GPUs is used. Each batch contains 256 Web and 256 SV images. We evaluate after every 1000 iterations and apply early-stopping by choosing the checkpoint with the highest F1 score for novel classes on the CompCars SV images. The weights are initialized from a model pre-trained on ImageNet. During training, for each image, a crop spanning an area between 8% and 100% of the original image is taken randomly and is resized to a size of 224×224 pixels afterwards. Additionally, a random horizontal flip is applied with 50% probability. Afterwards, the image is normalized using the mean and the standard deviation values of the pre-training on ImageNet. For experiments with a triplet loss, we employ hard negative mining [13] and a margin of 0.3 since preliminary experiments have shown good results for this value.

4.2 Inference details

During evaluation, the images are resized such that the shorter side has 256 pixels while keeping the aspect ratio. Afterwards, a crop of size 224×224 pixels is taken from the center of the resized image. The normalization is applied similar to the training configuration.

4.3 Softmax classification

We evaluate a softmax classifier as the most common architecture for deep-learning-based classification. Since softmax classifiers tend to heavily exploit domains in the classification, we explore the use of domain adversarial training and an auxiliary triplet loss to improve the domain invariance of the network.

Adversarial training	λ-schedule	λ base value	Base F1	Novel F1
No	-	-	95.4	43.0
Yes	Constant	0.1	96.4	66.3
Yes	Increasing	0.1	96.4	66.3
Yes	Increasing	1.0	96.4	65.9
Yes	Coupled	0.1	**96.5**	**67.7**
Yes	Coupled	1.0	96.0	66.4

Table 4.1: Evaluation of different schedules for the λ parameter of the domain adversarial training. The results indicate a clear advantage for the coupled schedule when focusing on the important novel classes.

4.3.1 Domain adversarial training

Adversarial domain adaptation [8] is a widely applied approach for domain adaptation. In order to find a strong baseline, we evaluate different schedules of the λ parameter that controls the influence of the domain adversarial head onto the main network. Besides a constant value and a widely applied monotonically increasing schedule [8], the coupled schedule by Wiedemer et al. [37] is also evaluated. The set λ base value describes the constant value for the constant schedule, the maximum value for the increasing schedule and the highest possible value (in case of zero domain classification loss) for the coupled schedule. The results are shown in Table 4.1.

The adversarial training leads to a large improvement of the base F1 score but particularly of the novel F1 score with all evaluated schedules for λ. While the impact of the schedule for λ is negligible for the base F1 score, for the important novel F1 score, the best results are achieved with the coupled schedule and a λ base value of 0.1. Based on these results, we continue to use these settings for all further experiments involving adversarial domain adaptation. The adversarial training reduces the impact of the domain onto the features and thus, leads to features of novel classes in the target domain being closer to features of the same class in the source domain. Therefore, the samples of novel classes in the target

Triplet loss	Base F1	Novel F1
No	95.4	43.0
Yes	**95.9**	**54.5**

Table 4.2: Evaluation of a triplet loss as auxiliary loss for a softmax classifier. The results indicate that an auxiliary triplet loss can improve the domain invariance of a softmax classifier.

domain are classified more accurately which in turn leads to less confusion with base classes. Thus, also the base class accuracy is improved.

4.3.2 Auxiliary triplet loss

The triplet loss has shown to be more domain invariant than a pure cross entropy loss. Thus, we evaluate the impact of an auxiliary triplet loss in Table 4.2. The triplet loss uses the negative cosine similarity as distance measure. A training with euclidean norm as distance measure did not converge properly since the euclidean norm enforces a feature space that is not well suited for the cross entropy loss. Thus, results for the euclidean norm are not reported.

The results show a clear advantage of the triplet loss for the accuracy of the base as well as the novel classes. The increase is probably a result of the triplet loss forcing a distance of close to zero in feature space for all samples of a class and thus, reducing the possibility of a spread due to different domains. While this only applies for the base classes in training, it probably also reduces the distance of samples of the novel classes between both domains leading to the improvement in the novel class accuracy. This improvement then leads to an improvement in base class accuracy due to less confusion with novel classes occurring.

Distance measure	Base F1	Novel F1
Euclidean norm	8.4	6.2
Negative cosine similarity	**96.0**	**62.9**

Table 4.3: Comparing distance measures for a distance-based classifier. The negative cosine similarity shows a strong advantage with the euclidean norm showing poor results due to the cross entropy loss not converging properly.

4.4 Distance-based classification

While CNNs are mostly combined with a logit-based classification head, distance-based classification and metric learning provide a higher flexibility due to not limiting the model to a specific set of classes during training.

4.4.1 Distance measure

For the distance-based classification, the choice of the distance measure is a crucial parameter. Thus, we compare the use of an euclidean norm as well as negative cosine similarity. The respective distance measure is applied for the triplet loss as well as for the classification. The results of the comparison are shown in Table 4.3. They indicate a strong advantage of the negative cosine similarity while the training with the euclidean norm does not properly converge. Particularly, the training of the triplet loss with an euclidean norm leads to a non-decreasing cross entropy loss. The embedding induced by a triplet loss with an euclidean norm seems to be incompatible with a logit-based softmax classification and a cross entropy loss. Seemingly, the optimizer can not converge to a proper embedding which suits both losses.

4.4.2 Prototype aggregation

For the classification, we aggregate all training samples from the source domain to estimate a prototype for each class and choose the class whose prototype is the closest to the input samples in terms of feature distance. For the aggregation

Aggregation	Base F1	Novel F1
Mean	**96.0**	**62.9**
Medoid	**96.0**	61.8

Table 4.4: Comparison of the estimation methods for the class prototype. Using the mean of the train samples shows a significant advantage over using the medoid.

Domain adversarial training	Base F1	Novel F1
No	96.0	62.9
Yes	**96.3**	**69.8**

Table 4.5: Evaluation of applying domain adversarial training with distance-based classification. The results show that adversarial training can provide an advantage in combination with a distance-based classifier.

of the samples, we evaluate a mean of the features and a medoid of the features. The medoid is defined as the sample which has the smallest total distance to all other samples. The results are shown in Table 4.4. While the difference on the base classes is negligible, the mean aggregation shows a clear advantage over the medoid for the novel classes.

4.4.3 Domain adversarial training

While the triplet loss already provides a strong improvement in terms of domain invariance for distance-based classification, we evaluate if domain adversarial training can still lead to an improved accuracy. Therefore, we apply domain adversarial training with the best setting as in the previous ablation studies additional to the cross entropy loss and the triplet loss we commonly use for the distance-based classifier. The results are shown in Table 4.5 and indicate a slight increase in terms of base class accuracy and a high increase in terms of novel class accuracy.

Method	Base F1	Novel F1
Softmax classifier	95.4	43.0
Softmax classifier with adversarial training	**96.5**	67.7
Softmax classifier with triplet loss	95.9	54.5
Distance-based classifier	96.0	62.9
Distance-based classifier with adversarial training	96.3	**69.8**

Table 4.6: Comparison of softmax classifiers with and without domain regularization methods and distance-based classification. The results show the advantage of distance-based classification for the accuracy of the novel classes while the softmax classifier with domain adversarial training shows a slight advantage for the base class accuracy.

4.5 Comparison of softmax classification and distance-based classification

We compare softmax-based classification methods with and without domain adaptation extensions to a distance-based classification method in Table 4.6. For the softmax-based classification, a domain adversarial training as well as an auxiliary triplet loss is evaluated to improve cross-domain classification accuracy.

While the softmax classifier with the adversarial training shows the highest accuracy for the base classes, the distance-based classifier combined with a domain adversarial training follows closely behind and has a significant advantage in terms of novel class accuracy compared to all evaluated distance-based classifiers. Without adversarial training, the softmax classifier shows a heavy drop in accuracy particularly of the novel classes. The triplet loss also provides a large benefit for the softmax classifier. However, it still shows a large accuracy gap when compared to the adversarial loss.

5 Conclusion

In this work, different domain adaptation approaches were evaluated in a supervised partially zero-shot setting for fine-grained vehicle classification to employ web images as training data for classification on surveillance images. The results show the importance of domain adversarial training to achieve acceptable results with a softmax-based classifier. However, a distance-based classifier employing a combination of a cross entropy loss and a triplet loss still show competitive results which can still be improved by domain adversarial training. This combination showed the overall best results for the classification of the novel classes in our evaluation.

Evaluation of better backbones as modern vision transformers [26, 5] or state-of-the-art convolutional network architectures [27] is up to future work. Other areas of future research are improvements directly targeting the supervised partially zero-shot setting which have not yet been evaluated for other settings.

References

[1] Yousef Alsahafi et al. “CarVideos: A Novel Dataset for Fine-Grained Car Classification in Videos”. In: *16th International Conference on Information Technology-New Generations (ITNG 2019)*. 2019.

[2] Marco Buzzelli and Luca Segantin. “Revisiting the CompCars Dataset for Hierarchical Car Classification: New Annotations, Experiments, and Results”. In: *Sensors* 21.2 (2021).

[3] Qianqiu Chen, Wei Liu, and Xiaoxia Yu. “A Viewpoint Aware Multi-Task Learning Framework for Fine-Grained Vehicle Recognition”. In: *IEEE Access* 8 (2020), pp. 171912–171923.

[4] Sumit Chopra, Suhrid Balakrishnan, and Raghuraman Gopalan. “Dlid: Deep learning for domain adaptation by interpolating between domains”. In: *ICML workshop on challenges in representation learning*. Vol. 2. 6. 2013.

[5] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *International Conference on Learning Representations*. 2021.

[6] Haodong Duan et al. “Omni-Sourced Webly-Supervised Learning for Video Recognition”. In: *Computer Vision – ECCV 2020*. 2020.

[7] Jie Fang et al. “Fine-Grained Vehicle Model Recognition Using A Coarse-to-Fine Convolutional Neural Network Architecture”. In: *IEEE Transactions on Intelligent Transportation Systems* 18.7 (2017), pp. 1782–1792.

[8] Yaroslav Ganin et al. “Domain-Adversarial Training of Neural Networks”. In: *Journal of Machine Learning Research* 17.59 (2016), pp. 1–35.

[9] Yang Gao et al. “Compact Bilinear Pooling”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[10] Timnit Gebru, Judy Hoffman, and Li Fei-Fei. “Fine-Grained Recognition in the Wild: A Multi-Task Domain Adaptation Approach”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2017.

[11] Muhammad Ghifary et al. “Domain Generalization for Object Recognition With Multi-Task Autoencoders”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2015.

[12] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[13] Alexander Hermans, Lucas Beyer, and Bastian Leibe. “In defense of the triplet loss for person re-identification”. In: *arXiv preprint arXiv:1703.07737* (2017).

[14] Judy Hoffman et al. “CyCADA: Cycle-Consistent Adversarial Domain Adaptation”. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 1989–1998.

[15] Shaoli Huang et al. “Part-Stacked CNN for Fine-Grained Visual Categorization”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[16] Yuqi Huo et al. “Coarse-to-Fine Grained Classification”. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR’19. 2019.

[17] Alper Kayabasi, Kaan Karaman, and Ibrahim Batuhan Akkaya. “Comparison of distance metric learning methods against label noise for fine-grained recognition”. In: *Automatic Target Recognition XXXI*. Vol. 11729. 2021.

[18] Jannik Koch, Stefan Wolf, and Jürgen Beyerer. “A Transformer-Based Late-Fusion Mechanism for Fine-Grained Object Recognition in Videos”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*. 2023.

[19] Shu Kong and Charless Fowlkes. “Low-Rank Bilinear Pooling for Fine-Grained Classification”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[20] Pablo Laiz, Jordi Vitria, and Santi Segui. “Using the Triplet Loss for Domain Adaptation in WCE”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. 2019.

[21] Pablo Laiz, Jordi Vitria, and Santi Segui. “Using the Triplet Loss for Domain Adaptation in WCE”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. 2019.

[22] Yanghao Li et al. “Adaptive Batch Normalization for practical domain adaptation”. In: *Pattern Recognition* 80 (2018), pp. 109–117.

[23] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. “Bilinear CNN Models for Fine-Grained Visual Recognition”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2015.

[24] Yen-Liang Lin et al. “Jointly Optimizing 3D Model Fitting and Fine-Grained Classification”. In: *Computer Vision – ECCV 2014*. 2014.

[25] Ming-Yu Liu and Oncel Tuzel. “Coupled Generative Adversarial Networks”. In: *Advances in Neural Information Processing Systems*. Vol. 29. 2016.

[26] Ze Liu et al. “Swin Transformer V2: Scaling Up Capacity and Resolution”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 12009–12019.

[27] Zhuang Liu et al. “A ConvNet for the 2020s”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 11976–11986.

[28] Marcel Simon and Erik Rodner. “Neural Activation Constellations: Unsupervised Part Model Discovery With Convolutional Networks”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2015.

[29] Jakub Sochor, Adam Herout, and Jiri Havel. “BoxCars: 3D Boxes as CNN Input for Improved Fine-Grained Vehicle Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[30] Kihyuk Sohn. “Improved Deep Metric Learning with Multi-class N-pair Loss Objective”. In: *Advances in Neural Information Processing Systems*. Vol. 29. 2016.

[31] Eric Tzeng et al. “Simultaneous Deep Transfer Across Domains and Tasks”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2015.

[32] Naoto Usuyama et al. “ePillID Dataset: A Low-Shot Fine-Grained Benchmark for Pill Identification”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2020.

[33] Mei Wang and Weihong Deng. “Deep visual domain adaptation: A survey”. In: *Neurocomputing* 312 (2018), pp. 135–153.

[34] Sinan Wang et al. “Progressive Adversarial Networks for Fine-Grained Domain Adaptation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[35] Yimu Wang et al. “An Adversarial Domain Adaptation Network for Cross-Domain Fine-Grained Recognition”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2020.

[36] Kilian Q. Weinberger and Lawrence K. Saul. “Distance Metric Learning for Large Margin Nearest Neighbor Classification”. In: *Journal of Machine Learning Research* 10.9 (2009), pp. 207–244.

[37] Thaddäus Wiedemer et al. “Few-Shot Supervised Prototype Alignment for Pedestrian Detection on Fisheye Images”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2022, pp. 4142–4153.

[38] Stefan Wolf. “Cross-Domain Fine-Grained Classification: A Review”. In: *Proceedings of the 2021 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*. Vol. 54. Karlsruher Institut für Technologie (KIT), 2022, pp. 189–205. ISBN: 978-3-7315-1171-7.

[39] Zhe Xu et al. “Webly-Supervised Fine-Grained Visual Categorization via Deep Domain Adaptation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.5 (2018), pp. 1100–1113.

[40] Linjie Yang et al. “A Large-Scale Car Dataset for Fine-Grained Categorization and Verification”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[41] Hantao Yao et al. “Coarse-to-Fine Description for Fine-Grained Visual Categorization”. In: *IEEE Transactions on Image Processing* 25.10 (2016), pp. 4858–4872.

[42] Baosheng Yu et al. “Correcting the Triplet Selection Bias for Triplet Loss”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

[43] Chaojian Yu et al. “Hierarchical Bilinear Pooling for Fine-Grained Visual Recognition”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Sept. 2018.

[44] Han Yu, Rong Jiang, and Aiping Li. “Striking a Balance in Unsupervised Fine-Grained Domain Adaptation Using Adversarial Learning”. In: *Knowledge Science, Engineering and Management*. 2020.

[45] Xu Zhang et al. *Deep Transfer Network: Unsupervised Domain Adaptation*. arXiv:1503.00591 [cs.CV]. 2015.

[46] Chen Zhu et al. “Fine-grained Video Categorization with Redundancy Reduction Attention”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

[47] Jun-Yan Zhu et al. “Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2017.

[48] Fuzhen Zhuang et al. “Supervised Representation Learning: Transfer Learning with Deep Autoencoders”. In: *Proceedings of the 24th International Conference on Artificial Intelligence*. IJCAI’15. 2015.

Long-term Action Anticipation: A Quick Survey

Zeyun Zhong

Vision and Fusion Laboratory
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT), Germany
zeyun.zhong@kit.edu

Abstract

The ability to anticipate possible human actions in the distant future is of fundamental interest for a wide range of applications, including autonomous driving, surveillance, and human-robot interaction. Consequently, various methods have been presented for action anticipation in recent years, with deep learning-based approaches being particularly popular. In this work, we give a short overview of the recent advances of long-term action anticipation algorithms.

1 Introduction

In the last years, we have seen a tremendous progress in the capabilities of computer systems to classify and segment activities in videos. These systems, however, analyze the past or in the case of real-time systems the present with a delay of a few milliseconds. For applications, where a moving system has to react or interact with humans, this is insufficient. For instance, to be able to offer a hand at the right time or to generate proactive dialog to provide more natural interactions, collaborative robots that work closely with humans have to anticipate the activities of a human in the future. Compared to human action recognition and early action recognition, where entire or part of action segments are observable, action anticipation aims to predict future action without observing any part of it, as displayed in Figure 1.1.

As the anticipation results are just assumptions, this tends to be significantly more challenging than traditional action recognition, which performs well with todays well-honed discriminative models [7, 17]. Consistent with action recognition, anticipation approaches start with prediction on only one single video frame [28] and tend to use longer temporal context [24, 29] in recent years. Apart from using a long action history, many approaches attempt to leverage several modalities other than just the raw video frames, such as the motion information and objects contained in the scene, to further improve the predictive ability.

While many recent works anticipate activities only for a very short time horizon of a few seconds [9, 8], there is a parallel line of work [6] which addresses the problem of anticipating all activities that will be happening within a time horizon of up to several minutes, which is particularly interesting for robot systems that require certain time to react and plan the future tasks.

In spite of the enormous amount of research conducted in this area, the problem is still challenging due to the fundamental challenges inherent to the task such as the multi-modal distribution of future action candidates, especially for the scenario where we are going to predict far into the future (long-term anticipation). As action recognition is usually a fundamental sub-component of an anticipation system, the challenges of action recognition [14] are also included, such as the tremendous intra-class variance among the activities, huge spatio-temporal scale variation, target motion variations, etc. Moreover, low image resolution, object occlusion, illumination change and viewpoint change further aggravate these challenges.

Although classical learning approaches, such as Conditional Random Fields (CRFs) [15], Markov models [23], and other statistical methods [19, 22], have been widely used in the literature, we put our focus on deep learning techniques and how they have been extended or applied to daily-living action anticipation, leaving the classical approaches outside the scope of the present review. In this context, the terms action anticipation, action prediction, and action forecasting are used interchangeably.

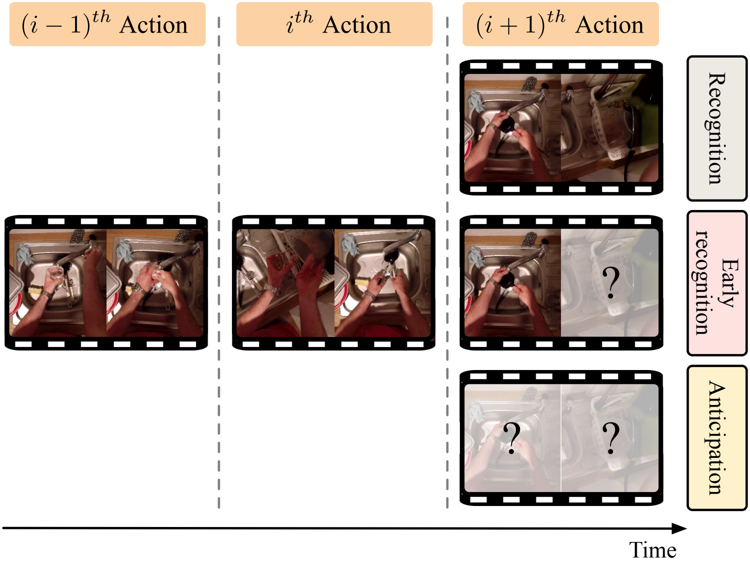
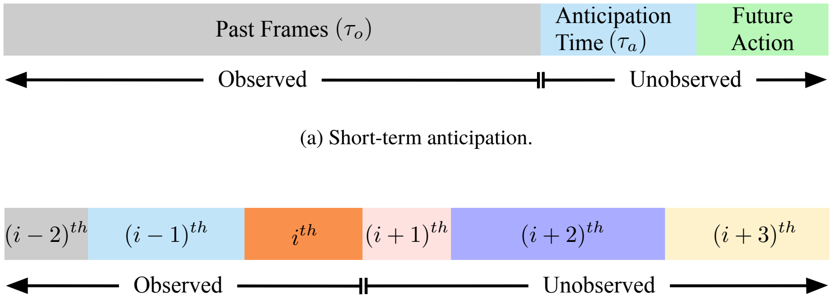


Figure 1.1: The action anticipation task aims to anticipate future action(s) before it happens, whereas action recognition and early action recognition require the observation of complete and partial actions, respectively.

This survey is structured as follows. In Section 2, we describe both short-term and long-term anticipation tasks which are commonly used in the literature, so that the reader can better distinguish between them. In Section 3, we introduce the current approaches that address the long-term anticipation task and discuss their limitations. Finally, we conclude this survey in Section 4.

2 Problem Statement

Based on the prediction time horizon, action anticipation approaches can be grouped into two categories: short-term anticipation approaches and long-term anticipation approaches. While short-term approaches predict a single action a



(a) Short-term anticipation.

(b) Long-term anticipation.

Figure 2.1: Category of the action anticipation task. While the short-term anticipation aims at predicting a single future action, long-term task aims to predict a sequence of the following actions.

few seconds into the future, long-term approaches predict a sequence of future actions with their durations up to several minutes into the future. In the following sections, we show the detailed task definition of both categories usually used in the literature.

2.1 Short-term anticipation

Most short-term anticipation approaches follow the setup defined in [28, 4, 5]. As illustrated in Figure 2.1(a), the task aims to predict a future action by observing a video segment of length τ_o. The observation segment is τ_a seconds preceding the action, i.e., from time $\tau_s - (\tau_a + \tau_o)$ to $\tau_s - \tau_a$, where τ_a denotes the "anticipation time", i.e., how many seconds in advance actions are to be anticipated. The anticipation time τ_a is usually fixed for each dataset, whereas the length of the observation segment is typically dependent on the individual method. Methods in this category typically use synchronous data to perform the anticipation task, meaning that the input to the model is a sequence of frames that have the same temporal spacing before the action [9, 8].

Some work [18, 21] attempts to predict the starting time of the next action as well. As this task involves the duration of each action, these approaches usually use asynchronous data as input to the model, containing a sequence of action categories and inter-arrival times. The inter-arrival time is defined as the difference between the starting time of last and the current action. With the predicted inter-arrival time, the starting time of the next action can be easily deduced.

2.2 Long-term anticipation

There is a parallel line of research addressing the long-tern anticipation task, which is proposed in [6]. The goal is to anticipate the category and the duration of future actions for a given time horizon, which can take up to several minutes, as illustrated in Figure 2.1(b). Long-term approaches typically take a sequence of observed action categories and their durations to predict another sequence of actions and durations [6, 1, 31].

3 Long-term Anticipation Approaches

3.1 Methods

Farha et al. [6] first introduced the long-term action anticipation task and proposed two models to tackle the task. One is based on an RNN model, which outputs the remaining length of the current action, the next action class and its length, as shown in Figure 3.1. The long-term prediction is conducted recursively, i.e., observations are combined with the current prediction to produce the next prediction. Another method is based on a CNN model, which outputs a sequence of future actions in a form of a matrix in one single step. Considering the limitations of these two methods, i.e., the RNN model is time-consuming and suffers from error accumulation and the CNN model introduces many parameters when predicting long sequences, Ke et al. [12] proposed a method to explicitly address these issues. They chose to condition on a time variable representing the prediction horizon. Specifically, they transformed the prediction time horizon

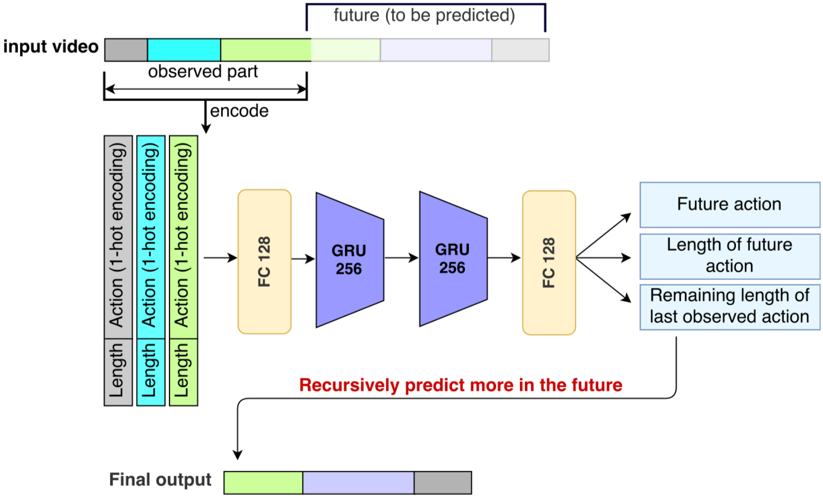


Figure 3.1: Architecture of the RNN system [6]. The input is a sequence of (length, 1-hot class encoding)-tuples. The network predicts the remaining length of the last observed action and the label and length of the next action. Appending the predicted result to the original input, the next action segment can be predicted. Figure is taken from [6].

to a time representation, and concatenated it with the original inputs forming time-conditioned observations. Their model is therefore capable of anticipating a future action at arbitrary and variable time horizons in a one-shot fashion. Additionally, they introduced a time-conditioned skip connection between the last observed action and the initial anticipation based on the intuition that the last action of the observation is generally relevant to the future actions.

Inspired by [12], Gong et al. [10] proposed an encoder-decoder structure based on transformer architecture [27, 2], which effectively captures long-term relations over the whole sequence of actions. The encoder learns to capture fine-grained long-range temporal relations between the observed frames from the past, while the decoder learns a sequence of future action queries, capturing global relations between upcoming actions in the future along with the observed features from the encoder. Because of the proposed parallel decoding, the model is able to make more accurate and faster inference without potential error accumulations caused by autoregressive decoding. However, the number of predictable future actions is also limited to the number of action queries used in the training process, which might need to be adapted, if the model is applied for other datasets.

Predicting future is inherently multi-modal. Given an observed video segment containing an ongoing action, multiple actions could be possible to be the next action following the observed one. This uncertainty becomes even larger if we are going to predict far into the future. Therefore, it may be beneficial to model the underlying uncertainty, allowing to capture different possible future actions. However, in most approaches, action prediction is taken as a classification problem and optimized under cross-entropy loss, suffering from overly high resemblance to dominant ground truth, while suppressing other reasonable possibilities [3]. Moreover, approaches that are optimized with mean square error tend to produce the mean of the modes [28, 20]. To this end, some approaches are proposed to tackle the uncertainty in the future predictions, which are described below.

Farha and Gall [1] introduced a framework that predicts all subsequent actions and corresponding durations in a stochastic manner. In their framework, an action model similar to the one proposed in [6] (shown in Figure 3.1) and a time model are trained to predict the probability distribution of the future action label and duration, respectively. While action labels are taken as classifications and optimized under cross-entropy (CE) loss, durations are taken as real-valued variables which are modeled with a Gaussian distribution and optimized with the negative log likelihood (NLL). At test time, future action label and its duration are sampled from the learned distributions. Long-term predictions are achieved by feeding the predicted action segment to the model recursively.

Zhao and Wildes [31] proposed Conditional Adversarial Generative Networks to address the underlying uncertainty when predicting future action sequence. More specifically, different from many works that operate with continuous time variable [6, 1, 12, 10], they treated both action labels and time as discrete data which are formated as one-hot vectors. These vectors are first projected to higher dimension continuous spaces and concatenated, and then fed to a *seq2seq* generator [26] to compute logits of future action labels and their corresponding time. To obtain differentiable sampling to generate future sequences with both quality and diversity during training, the Gumbel-Softmax relaxation technique [11] that mimics one-hot vectors from categorical distributions and a normalized distance regularizer [30] that encourages diversity are adopted. A ConvNet classifier is used as the discriminator to allow to train the generator adversarially.

Mehrasa et al. [21] proposed using a recurrent variational auto-encoder (VAE [13]) to capture the distribution over the times and categories of action sequences. To overcome the problem that a fixed prior distribution of the latent variable (usually $\mathcal{N}(0, I)$ in VAE models) may ignore temporal dependencies present between actions, authors learned a prior that varies across time. At test time, a latent code is sampled from the learned prior distribution, based on which the probability distributions of the action class and the corresponding time are inferred.

3.2 Limitations

Despite the impressive performance on the standard benchmarks [25, 16], current approaches have several limitations, which are described below.

Limited representativity of the evaluation datasets. The commonly used benchmark datasets for long-term anticipation, i.e., Breakfast [16] and 50Salads [25], contain only videos of a specific kitchen activity, which usually last several minutes. Since there is only one activity per video, i.e., either *preparing a breakfast* or *preparing a salad*, it is easier to predict the following actions than in the real-world scenarios, where a completely different action might occur next. Furthermore, since these videos are typically only several minutes long, the current setting may not be directly applicable for longer videos, especially for real-world applications. Moreover, these datasets do not contain any concurrent actions. However, actions in the real-world scenarios, such as *making a phone call* and *taking notes* may be performed simultaneously.

Difficult deployment of methods that incorporate uncertainty. Methods that incorporate uncertainty typically learn a joint distribution of all data samples. For evaluation, authors usually draw many samples from the learned distribution, and compute the average metric value of all drawn samples [1, 31], or select the most frequent sample as the final result [21]. However, such an evaluation protocol requires multiple runs of the model, which is time-consuming and therefore difficult to deploy for real-time systems.

4 Conclusion

In this survey, we gave a short overview of the current approaches that are proposed to tackle the long-term action anticipation task. We analyzed different methods from two perspectives: research question each individual method addresses and method description. In the end, we also described the limitations of the current approaches. In conclusion, long-term action anticipation is an interesting and relatively new research topic, which attracts increasing attention in the community, and benefits many intelligent decision-making systems. While great strides have been made, there is still large room for improvement in action anticipation using deep learning techniques.

References

[1] Yazan Abu Farha and Juergen Gall. "Uncertainty-Aware Anticipation of Activities". In: *ICCV Workshop*. 2019.

[2] Nicolas Carion et al. "End-to-end object detection with transformers". In: *ECCV*. 2020, pp. 213–229.

[3] Bo Dai et al. "Towards diverse and natural image descriptions via a conditional gan". In: *ICCV*. 2017, pp. 2970–2979.

[4] Dima Damen et al. "Scaling egocentric vision: The epic-kitchens dataset". In: *ECCV*. 2018, pp. 720–736.

[5] Dima Damen et al. "The epic-kitchens dataset: Collection, challenges and baselines". In: *TPAMI* 43.11 (2020), pp. 4125–4141.

[6] Yazan Abu Farha, Alexander Richard, and Juergen Gall. "When Will You Do What? - Anticipating Temporal Occurrences of Activities". In: *CVPR*. 2018. arXiv: `1804.00892`.

[7] Christoph Feichtenhofer et al. “Slowfast networks for video recognition”. In: *ICCV*. 2019, pp. 6202–6211.

[8] Antonino Furnari and Giovanni Farinella. “What Would You Expect? Anticipating Egocentric Actions With Rolling-Unrolling LSTMs and Modality Attention”. In: *ICCV*. 2019.

[9] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. “RED: Reinforced Encoder-Decoder Networks for Action Anticipation”. In: *BMVC*. 2017.

[10] Dayoung Gong et al. “Future Transformer for Long-term Action Anticipation”. In: *CVPR*. 2022. arXiv: `2205.14022 [cs]`.

[11] Eric Jang, Shixiang Gu, and Ben Poole. “Categorical reparameterization with gumbel-softmax”. In: 2017.

[12] Qiuhong Ke, Mario Fritz, and Bernt Schiele. “Time-Conditioned Action Anticipation in One Shot”. In: *CVPR*. June 2019.

[13] Diederik P Kingma and Max Welling. “Auto-encoding variational bayes”. In: *ICLR*. 2014.

[14] Yu Kong and Yun Fu. “Human action recognition and prediction: A survey”. In: *IJCV* 130.5 (2022), pp. 1366–1401.

[15] Hema S. Koppula and Ashutosh Saxena. “Anticipating Human Activities Using Object Affordances for Reactive Robotic Response”. In: *TPAMI* 1 (2016).

[16] Hilde Kuehne, Ali Arslan, and Thomas Serre. “The Language of Actions: Recovering the Syntax and Semantics of Goal-Directed Human Activities”. In: *CVPR*. 2014.

[17] Ze Liu et al. “Video swin transformer”. In: *CVPR*. 2022, pp. 3202–3211.

[18] Tahmida Mahmud, Mahmudul Hasan, and Amit K Roy-Chowdhury. “Joint prediction of activity labels and starting times in untrimmed videos”. In: *ICCV*. 2017, pp. 5773–5782.

[19] Tahmida Mahmud et al. “A poisson process model for activity forecasting”. In: *ICIP*. IEEE. 2016, pp. 3339–3343.

[20] Michael Mathieu, Camille Couprie, and Yann LeCun. “Deep multi-scale video prediction beyond mean square error”. In: *ICLR*. 2016.

[21] Nazanin Mehrasa et al. "A Variational Auto-Encoder Model for Stochastic Point Processes". In: *CVPR*. 2019.

[22] Siyuan Qi et al. "Predicting Human Activities Using Stochastic Grammar". In: *ICCV*. 2017.

[23] Nicholas Rhinehart and Kris M Kitani. "First-person activity forecasting with online inverse reinforcement learning". In: *ICCV*. 2017, pp. 3696–3705.

[24] Fadime Sener, Dipika Singhania, and Angela Yao. "Temporal Aggregate Representations for Long-Range Video Understanding". In: *ECCV*. 2020. arXiv: `2006.00830`.

[25] Sebastian Stein and Stephen J McKenna. "Combining embedded accelerometers with computer vision for recognizing food preparation activities". In: *UbiComp*. 2013, pp. 729–738.

[26] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. "Sequence to sequence learning with neural networks". In: *NeurIPS*. 2014.

[27] Ashish Vaswani et al. "Attention Is All You Need". In: *NeurIPS*. 2017.

[28] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. "Anticipating visual representations from unlabeled video". In: *CVPR*. 2016, pp. 98–106.

[29] Chao-Yuan Wu et al. "MeMViT: Memory-Augmented Multiscale Vision Transformer for Efficient Long-Term Video Recognition". In: *CVPR*. 2022. arXiv: `2201.08383`.

[30] Dingdong Yang et al. "Diversity-sensitive conditional generative adversarial networks". In: *ICLR*. 2019.

[31] He Zhao and Richard P Wildes. "On diverse asynchronous activity anticipation". In: *ECCV*. Springer. 2020, pp. 781–799.

Karlsruher Schriftenreihe zur Anthropomatik (ISSN 1863-6489)

Band 1 Jürgen Geisler
Leistung des Menschen am Bildschirmarbeitsplatz.
ISBN 3-86644-070-7

Band 2 Elisabeth Peinsipp-Byma
Leistungserhöhung durch Assistenz in interaktiven Systemen zur Szenenanalyse. 2007
ISBN 978-3-86644-149-1

Band 3 Jürgen Geisler, Jürgen Beyerer (Hrsg.)
Mensch-Maschine-Systeme.
ISBN 978-3-86644-457-7

Band 4 Jürgen Beyerer, Marco Huber (Hrsg.)
Proceedings of the 2009 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-86644-469-0

Band 5 Thomas Usländer
Service-oriented design of environmental information systems.
ISBN 978-3-86644-499-7

Giulio Milighetti
Band 6 **Multisensorielle diskret-kontinuierliche Überwachung und Regelung humanoider Roboter.**
ISBN 978-3-86644-568-0

Jürgen Beyerer, Marco Huber (Hrsg.)
Band 7 **Proceedings of the 2010 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.**
ISBN 978-3-86644-609-0

Eduardo Monari
Band 8 **Dynamische Sensorselektion zur auftragsorientierten Objektverfolgung in Kameranetzwerken.**
ISBN 978-3-86644-729-5

Die Bände sind unter www.ksp.kit.edu als PDF frei verfügbar oder als Druckausgabe bestellbar.

Band 9 Thomas Bader
Multimodale Interaktion in Multi-Display-Umgebungen.
ISBN 3-86644-760-8

Band 10 Christian Frese
Planung kooperativer Fahrmanöver für kognitive Automobile.
ISBN 978-3-86644-798-1

Band 11 Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2011 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-86644-855-1

Band 12 Miriam Schleipen
Adaptivität und Interoperabilität von Manufacturing Execution Systemen (MES).
ISBN 978-3-86644-955-8

Band 13 Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2012 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-86644-988-6

Band 14 Hauke-Hendrik Vagts
Privatheit und Datenschutz in der intelligenten Überwachung: Ein datenschutzgewährendes System, entworfen nach dem „Privacy by Design“ Prinzip.
ISBN 978-3-7315-0041-4

Band 15 Christian Kühnert
Data-driven Methods for Fault Localization in Process Technology. 2013
ISBN 978-3-7315-0098-8

Band 16 Alexander Bauer
Probabilistische Szenenmodelle für die Luftbildauswertung.
ISBN 978-3-7315-0167-1

Band 17 Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2013 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-0212-8

Die Bände sind unter www.ksp.kit.edu als PDF frei verfügbar oder als Druckausgabe bestellbar.

Band 18 Michael Teutsch
Moving Object Detection and Segmentation for Remote Aerial Video Surveillance.
ISBN 978-3-7315-0320-0

Band 19 Marco Huber
Nonlinear Gaussian Filtering:
Theory, Algorithms, and Applications.
ISBN 978-3-7315-0338-5

Band 20 Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2014 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-0401-6

Band 21 Todor Dimitrov
Permanente Optimierung dynamischer Probleme der Fertigungssteuerung unter Einbeziehung von Benutzerinteraktionen.
ISBN 978-3-7315-0426-9

Band 22 Benjamin Kühn
Interessengetriebene audiovisuelle Szenenexploration.
ISBN 978-3-7315-0457-3

Band 23 Yvonne Fischer
Wissensbasierte probabilistische Modellierung für die Situationsanalyse am Beispiel der maritimen Überwachung.
ISBN 978-3-7315-0460-3

Band 24 Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2015 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-0519-8

Band 25 Pascal Birnstill
Privacy-Respecting Smart Video Surveillance Based on Usage Control Enforcement.
ISBN 978-3-7315-0538-9

Band 26 Philipp Woock
Umgebungskartenschätzung aus Sidescan-Sonardaten für ein autonomes Unterwasserfahrzeug.
ISBN 978-3-7315-0541-9

Die Bände sind unter www.ksp.kit.edu als PDF frei verfügbar oder als Druckausgabe bestellbar.

Band 27 Janko Petereit
Adaptive State × Time Lattices: A Contribution to Mobile Robot Motion Planning in Unstructured Dynamic Environments.
ISBN 978-3-7315-0580-8

Band 28 Erik Ludwig Krempel
Steigerung der Akzeptanz von intelligenter Videoüberwachung in öffentlichen Räumen.
ISBN 978-3-7315-0598-3

Band 29 Jürgen Moßgraber
Ein Rahmenwerk für die Architektur von Frühwarnsystemen. 2017
ISBN 978-3-7315-0638-6

Band 30 Andrey Belkin
World Modeling for Intelligent Autonomous Systems.
ISBN 978-3-7315-0641-6

Band 31 Chettapong Janya-Anurak
Framework for Analysis and Identification of Nonlinear Distributed Parameter Systems using Bayesian Uncertainty Quantification based on Generalized Polynomial Chaos.
ISBN 978-3-7315-0642-3

Band 32 David Münch
Begriffliche Situationsanalyse aus Videodaten bei unvollständiger und fehlerhafter Information.
ISBN 978-3-7315-0644-7

Band 33 Jürgen Beyerer, Alexey Pak (Eds.)
Proceedings of the 2016 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-0678-2

Band 34 Jürgen Beyerer, Alexey Pak and Miro Taphanel (Eds.)
Proceedings of the 2017 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-0779-6

Band 35 Michael Grinberg
Feature-Based Probabilistic Data Association for Video-Based Multi-Object Tracking.
ISBN 978-3-7315-0781-9

Die Bände sind unter www.ksp.kit.edu als PDF frei verfügbar oder als Druckausgabe bestellbar.

Band 36 Christian Herrmann
**Video-to-Video Face Recognition for
Low-Quality Surveillance Data.**
ISBN 978-3-7315-0799-4

Band 37 Chengchao Qu
**Facial Texture Super-Resolution
by Fitting 3D Face Models.**
ISBN 978-3-7315-0828-1

Band 38 Miriam Ruf
**Geometrie und Topologie von Trajektorienoptimierung
für vollautomatisches Fahren.**
ISBN 978-3-7315-0832-8

Band 39 Angelika Zube
**Bewegungsregelung mobiler Manipulatoren für die
Mensch-Roboter-Interaktion mittels kartesischer
modellprädiktiver Regelung.**
ISBN 978-3-7315-0855-7

Band 40 Jürgen Beyerer and Miro Taphanel (Eds.)
**Proceedings of the 2018 Joint Workshop of
Fraunhofer IOSB and Institute for Anthropomatics,
Vision and Fusion Laboratory.**
ISBN 978-3-7315-0936-3

Band 41 Marco Thomas Gewohn
**Ein methodischer Beitrag zur hybriden Regelung
der Produktionsqualität in der Fahrzeugmontage.**
ISBN 978-3-7315-0893-9

Band 42 Tianyi Guan
**Predictive energy-efficient motion trajectory
optimization of electric vehicles.**
ISBN 978-3-7315-0978-3

Band 43 Jürgen Metzler
**Robuste Detektion, Verfolgung und Wiedererkennung von
Personen in Videodaten mit niedriger Auflösung.**
ISBN 978-3-7315-0968-4

Band 44 Sebastian Bullinger
**Image-Based 3D Reconstruction of Dynamic Objects Using
Instance-Aware Multibody Structure from Motion.**
ISBN 978-3-7315-1012-3

Die Bände sind unter www.ksp.kit.edu als PDF frei verfügbar oder als Druckausgabe bestellbar.

Band 45 Jürgen Beyerer, Tim Zander (Eds.)
Proceedings of the 2019 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-1028-4

Band 46 Stefan Becker
Dynamic Switching State Systems for Visual Tracking.
ISBN 978-3-7315-1038-3

Band 47 Jennifer Sander
Ansätze zur lokalen Bayes'schen Fusion von Informationsbeiträgen heterogener Quellen.
ISBN 978-3-7315-1062-8

Band 48 Philipp Christoph Sebastian Bier
Umsetzung des datenschutzrechtlichen Auskunftsanspruchs auf Grundlage von Usage-Control und Data-Provenance-Technologien.
ISBN 978-3-7315-1082-6

Band 49 Thomas Emter
Integrierte Multi-Sensor-Fusion für die simultane Lokalisierung und Kartenerstellung für mobile Robotersysteme.
ISBN 978-3-7315-1074-1

Band 50 Patrick Dunau
Tracking von Menschen und menschlichen Zuständen.
ISBN 978-3-7315-1086-4

Band 51 Jürgen Beyerer, Tim Zander (Eds.)
Proceedings of the 2020 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-1091-8

Band 52 Lars Wilko Sommer
Deep Learning based Vehicle Detection in Aerial Imagery.
ISBN 978-3-7315-1113-7

Band 53 Jan Hendrik Hammer
Interaktionstechniken für mobile Augmented-Reality-Anwendungen basierend auf Blick- und Handbewegungen.
ISBN 978-3-7315-1169-4

Die Bände sind unter www.ksp.kit.edu als PDF frei verfügbar oder als Druckausgabe bestellbar.

Band 54 Jürgen Beyerer, Tim Zander (Eds.)
Proceedings of the 2021 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-1171-7

Band 55 Ronny Hug
Probabilistic Parametric Curves for Sequence Modeling.
ISBN 978-3-7315-1198-4

Band 56 Florian Patzer
Automatisierte, minimalinvasive Sicherheitsanalyse und Vorfallreaktion für industrielle Systeme.
ISBN 978-3-7315-1207-3

Band 57 Achim Christian Kuwertz
Adaptive Umweltmodellierung für kognitive Systeme in offener Welt durch dynamische Konzepte und quantitative Modellbewertung.
ISBN 978-3-7315-1219-6

Band 58 Julius Pfrommer
Distributed Planning for Self-Organizing Production Systems.
ISBN 978-3-7315-1253-0

Band 59 Ankush Meshram
Self-learning Anomaly Detection in Industrial Production.
ISBN 978-3-7315-1257-8

Band 60 Patrick Philipp
Über die Formalisierung und Analyse medizinischer Prozesse im Kontext von Expertenwissen und künstlicher Intelligenz.
ISBN 978-3-7315-1289-9

Band 61 Mathias Anneken
Anomaliedetektion in räumlich-zeitlichen Datensätzen.
ISBN 978-3-7315-1300-1

Band 62 Jürgen Beyerer, Tim Zander (Eds.)
Proceedings of the 2022 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-1304-9

Die Bände sind unter www.ksp.kit.edu als PDF frei verfügbar oder als Druckausgabe bestellbar.

Lehrstuhl für Interaktive Echtzeitsysteme
Karlsruher Institut für Technologie

Fraunhofer-Institut für Optronik, Systemtechnik
und Bildauswertung IOSB Karlsruhe

In 2022, the annual joint workshop of the Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB) and the Vision and Fusion Laboratory (IES) of the Institute for Anthropomatics, Karlsruhe Institute of Technology (KIT) was hosted at a Schwarzwaldhaus near Triberg. For a week from the 31st of July to the 5th of August the doctoral students of both institutions presented extensive reports on the status of their research and discussed topics ranging from computer vision and optical metrology to network security, usage control and machine learning.

The results and ideas presented at the workshop are collected in this book in the form of detailed technical reports. This volume provides a comprehensive and up-to-date overview of the research program of the IES laboratory and the Fraunhofer IOSB.

ISSN 1863-6489 (Schriftenreihe)
ISSN 2510-7259 (Tagungsband)
ISBN 978-3-7315-1304-9

ISBN 978-3-7315-1304-9
9 783731 513049

Gedruckt auf FSC-zertifiziertem Papier