Severity of Catastrophic Forgetting in Object Detection for Autonomous Driving

Christian Witte[1,2], René Schuster[3], Syed Saqib Bukhari[1], Patrick Trampert[1], Didier Stricker[2,3] and Georg Schneider[1]

[1]*ZF Friedrichshafen AG, Saarbrücken, Germany*

[2]*University of Kaiserlautern - TUK, Kaiserslautern, Germany*

[3]*DFKI - German Research Center for Artificial Intelligence, Kaiserslautern, Germany*

Keywords: Continual Learning, Catastrophic Forgetting, Object Detection, Autonomous Driving.

Abstract: Incorporating unseen data in pre-trained neural networks remains a challenging endeavor, as complete retraining is often impracticable. Yet, training the networks sequentially on data with different distributions can lead to performance degradation for previously learned data, known as *catastrophic forgetting*. The sequential training paradigm and the mitigation of catastrophic forgetting are subject to Continual Learning (CL). The phenomenon of forgetting poses a challenge for applications with changing distributions and prediction objectives, including Autonomous Driving (AD).

Our work aims to illustrate the severity of catastrophic forgetting for object detection for class- and domain-incremental learning. We propose four hypotheses, as we investigate the impact of the ordering of sequential increments and the underlying data distribution of AD datasets. Further, the influence of different object detection architectures is examined. The results of our empirical study highlight the major effects of forgetting for class-incremental learning. Moreover, we show that domain-incremental learning suffers less from forgetting but is highly dependent on the design of the experiments and choice of architecture.

1 INTRODUCTION

Training a neural network sequentially on new data results in a degradation of performance on previously learned knowledge, which is called catastrophic inference or catastrophic forgetting (McCloskey and Cohen, 1989). As the real world is non-stationary, autonomous systems are exposed to ever-changing data distributions. For Autonomous Driving (AD), the distribution change is induced by a variation in the visual domain, e.g., weather, time, country, or the appearance of new, unknown classes, e.g., drones or electric scooters. Incorporating knowledge of new domains and new classes into a Neural Network (NN) is a non-trivial task. Common practice is to extend the previous dataset with data samples containing new classes or from different domains and to retrain the NN from scratch. As storing previous data is not desired and an entire retraining results in a high computational overhead, Continual Learning (CL) aims to alleviate the effects of catastrophic forgetting for sequential training. Comparing current CL approaches is difficult due to various assumptions, different settings, and downstream tasks, which highly influence the results.

We investigate the effects of different realistic CL scenarios with respect to catastrophic forgetting for object detection, which is an essential element for AD. Guided by four proposed hypotheses, we evaluate how the aspects *data distribution*, *order of sequential tasks*, and *architecture choice* have an impact on the performance of object detection. We compare domain- and class-incremental learning and conclude with suggestions for the evaluation of CL methods.

2 CONTINUAL LEARNING SCENARIOS

Continual Learning, also known as Incremental Learning, describes an iterative learning procedure composed of sequential tasks to be learned, which either differ in the data distribution or the prediction objective. As only data for the current task, also referred to as increment, is available for the current training step, weights important to previous tasks are altered. Changing these weights results in the effect of *catastrophic forgetting*. We can formally define

Witte, C., Schuster, R., Bukhari, S., Trampert, P., Stricker, D. and Schneider, G.
Severity of Catastrophic Forgetting in Object Detection for Autonomous Driving.
DOI: 10.5220/0011634500003411
In *Proceedings of the 12th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2023)*, pages 262-269
ISBN: 978-989-758-626-2; ISSN: 2184-4313

the task $T_t = (X_t, Y_t)$ as *t-th* incremental task in the learning procedure with the set of input samples $X_t = \{x_i\}_{1 \leq i \leq N_t}$ and output targets $Y_t = \{y_i\}_{1 \leq i \leq N_t}$ (Hsu et al., 2018). The sequence of tasks $\{T_1, T_2, ..., T_K\}$ describes the entire CL scenario. CL scenarios can be categorized based on the incremental task to be solved (Hsu et al., 2018; Van de Ven and Tolias, 2019). Let X_s, X_t be the input data for two tasks $T_s, T_t, \forall s,t \in K, s \neq t$ and Y_s, Y_t are the sets of annotations. **Class-incremental Learning** describes learning an exclusive subset of classes for each subsequent task. For classification tasks, this implies $Y_s \cap Y_t = \emptyset$ and the setting is class instance-injective, which means each input sample corresponds to only one task. However, the downstream-task object detection demands multiple class label instances per data sample and additionally predicts not only class annotations, but also spatial information of the object. Hence, only the sets of class annotations for different tasks are disjoint. The differing marginal distributions of target annotations, i.e., $P(Y_s) \neq P(Y_t)$, imply consequently a different distribution of the input data $P(X_s) \neq P(X_t)$.

Domain-incremental Learning considers the CL scenario, in which the output targets do not change, while the domain shift induces a change in the input data distribution (Pan and Yang, 2009). The elements of the target space, i.e., all possible annotations to be learned, stay the same for consecutive tasks ($Y_s = Y_t$).

For each task in **Task-incremental Learning**, the set of output labels is different, i.e., output spaces are disjoint (Hsu et al., 2018). Task-incremental learning settings require a task identifier, which is an additional ground truth annotation for the current task. Class- and domain-incremental learning scenarios for AD are visualized in Figure 1.

3 RELATED WORK

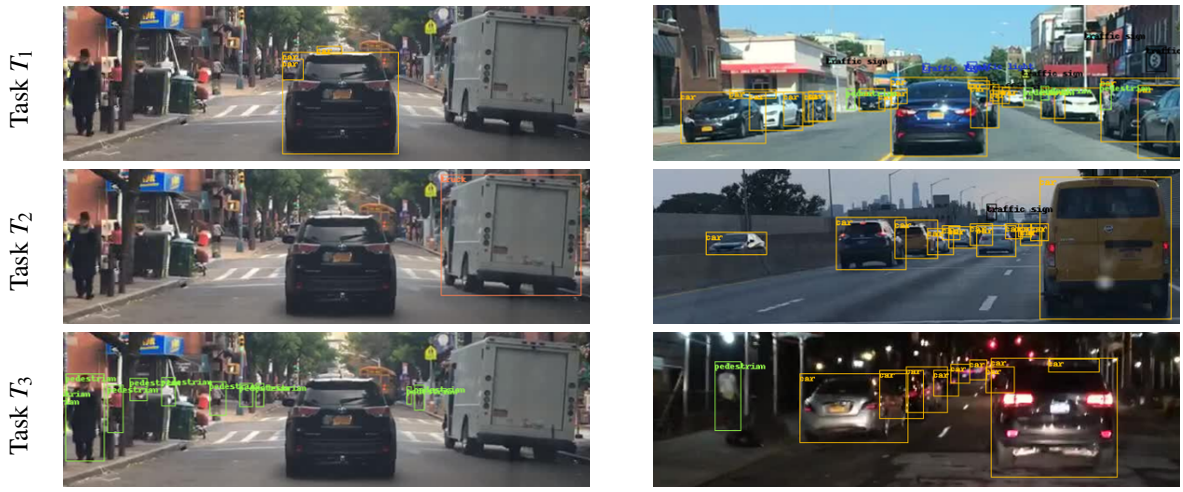
3.1 Continual Learning

Previous work on CL primarily focused on different techniques to alleviate the effects of catastrophic forgetting. Delange et al. (2021) proposed the categorization into replay, regularization-based, and parameter isolation methods. Replay methods aim to mitigate forgetting by revisiting previous knowledge explicitly (Lopez-Paz and Ranzato, 2017; Rebuffi et al., 2017) or inducing self-generated data during training (Shin et al., 2017). Constraining or penalizing updates of model parameters is the focus of regularization-based methods (Kirkpatrick et al., 2017; Zenke et al., 2017). Algorithms that employ a dynamic expansion of NNs (Rusu et al., 2016) or assign different model parameters to each task (Serra et al., 2018) are referred to as parameter isolation methods. While most approaches focus on classification, CL for object detection is a less explored field of research. One type of mitigation strategy is knowledge distillation (Shmelkov et al., 2017; Peng et al., 2020, 2021). Another line of work analyzed the diverse influences on CL in various settings. Farquhar and Gal (2018) assessed the experimental setup for CL. They claimed that the evaluation is biased, so they define fundamental desiderata for the empirical evaluation of CL. Among them is the evaluation without test-time task labels, which are commonly used to indicate the task to be solved. Consequently, we neglect task-incremental learning in this empirical study. Recently, Mirzadeh et al. (2022) performed an in-depth analysis of the significance of the architecture for classification in CL. They show that each architecture has an individual trade-off between stability and plasticity (Mirzadeh et al., 2022; Pham et al., 2022). Ramasesh et al. (2021) show in their empirical study that pre-trained ResNet (He et al., 2016) and Transformer (Dosovitskiy et al., 2021) architectures are more robust in terms of forgetting than randomly initialized models for CL.

To the best of our knowledge, an empirical study concerning the influence of catastrophic forgetting on object detection has not been performed, yet.

3.2 Object Detection

Faster-RCNN (Ren et al., 2015) is a widely used two-stage detection architecture, which employs a Convolutional Neural Network (CNN) for feature extraction and for region proposal generation. Recently, the first Vision Transformer (ViT) (Dosovitskiy et al., 2021) has shown competitive performance with manageable computational resources. The ViT was improved upon by computing attention in shifted windows, called Swin-Transformer (Liu et al., 2021). Carion et al. (2020) introduced Detection Transformer (DETR), the first transformer-based end-to-end architecture for detection. As this approach suffers from slow convergence and low performance on small objects due to the limited spatial resolution, it was improved by introducing deformable attention modules. Deformable DETR (DDETR) (Zhu et al., 2021) mitigates these effects by computing attention only on a small set of sampling points. Tian et al. (2019) introduced FCOS (Fully Convolutional One-Stage Object Detector), an anchor-free and proposal-free one-stage detector.



(a) Class-incremental learning (b) Domain-incremental learning

Figure 1: Incremental learning for object detection is visualized for the dataset BDD100K (Yu et al., 2020). Each image corresponds to one task. In the presented class-incremental learning setting (a), the classes *Car*, *Truck*, and *Pedestrian* are learned sequentially. For the domain-incremental scenario (b), the tasks T_1, T_2, T_3 differ in their domain, as each task contains only images at *Daytime*, *Dawn/Dusk*, or *Night*, respectively.

4 EXPERIMENTAL SETUP

As clear guidance for the experiments, first, we define four hypotheses to be investigated to analyze the effects of catastrophic forgetting. Then, we introduce three datasets and the relevant CL scenarios. Lastly, we examine relevant metrics and provide implementation details for our experiments.

4.1 Hypotheses

To analyze the effects and influence of catastrophic forgetting for 2D bounding box detection, we designed and performed various experiments to investigate the following hypotheses:

H1. *The order of tasks influences the forgetting.*

H2. *The data distribution of the input influences the severity of forgetting.*

H3. *The architecture of the detector has an influence on the forgetting.*

H4. *Class-incremental learning leads to more severe forgetting compared to domain-incremental learning.*

4.2 Datasets

This investigation considers object detection for AD datasets. To obtain a high degree in diversity, we consider the datasets **BDD100K** (Yu et al., 2020) and **SODA10M** (Han et al., 2021).

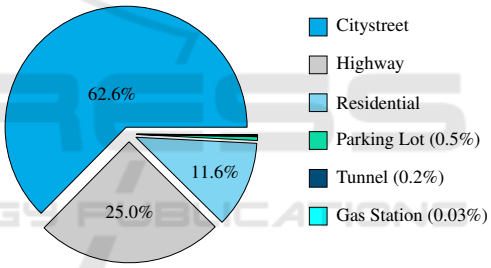


Figure 2: Distribution of the image occurrences for the domain *Scene* for BDD100K.

BDD100K was released in 2020 to embrace real-world complexity by incorporating a high diversity geographically and environmentally. It was recorded in the USA and it includes scene type information for each image, such as *citystreets, highways*, or *residential areas*, as well as the time of day annotation. Further, weather condition information, e.g., *sunny, overcast, rainy, etc.*, is provided.

SODA10M is a large-scale autonomous driving dataset with 10 million unlabeled and 20,000 labeled images. It provides annotations for the domains weather, scene, and time of day. The training dataset contains images from one domain, while the validation and testing datasets include data from all domains. To investigate hypothesis ***H2***, i.e., that the distribution of domains influences the incremental training, the *scene* domain distribution is visualized for

BDD100K in Figure 2. We must acknowledge, that the class and domain distributions are vastly unbalanced.

4.3 Continual Learning Scenarios

As described in Section 2, there are three different CL settings. Since task-incremental learning represents an artificial scenario due to the required task-identifier, the focus of this study lies on class- and domain-incremental learning.

We investigate the influence of several class-incremental learning scenarios for object detection and we perform all experiments with the assumption that instances of previous or future classes may also appear at any step and that their labels are only available at their respective task increment. This means that only annotations for classes in the current task increment, e.g., vehicles, are provided. Previous or future class instances that appear also in the current increment, e.g., pedestrians co-occurring with vehicles, are not annotated.

As commonly done in class-incremental learning (Shmelkov et al., 2017), we investigate setups with various classes in the first increment and different numbers of classes added per increment. These setups can be denoted by a tuple, e.g., 5-5 or 9-1, where the first integer of the tuple represents the number of *initial* classes trained in the first increment T_1 and the second integer represents the number of classes added per increment.

To evaluate domain-incremental learning, we use the available domain information of the datasets BDD100K and SODA10M. For each CL scenario, we perform sequential training on the incremental datasets, also referred to as naïve fine-tuning. After training on a specific increment, the performance of the network is evaluated for the current task. The evaluation takes into account only the previous and current tasks for class-incremental learning and all tasks for domain-incremental learning.

4.4 Metrics

The mean Average Precision (mAP) is a common performance metric for object detection. It is computed for all known classes for a fixed validation set. The performance itself does not yield information about the forgetting in CL. To obtain a notion of the influence one increment has on the previous or the following increment, the metrics *Backward Transfer (BWT)* and *Forward Transfer (FWT)* were introduced (Lopez-Paz and Ranzato, 2017).

With K tasks to be learned, the model is evaluated on all K tasks after the training of each task T_i. Then, a matrix $P \in \mathcal{R}^{K\times K}$ is obtained with $p_{i,j}$ as mAP for task T_j after being trained on task T_i and all previous tasks $T_1, ..., T_{i-1}$. A baseline vector $\bar{b}$ is computed previously to the training by evaluating the performance of the initial model on all tasks. The above mentioned metrics are then computed as follows:

$$BWT = \frac{1}{K-1}\sum_{i=1}^{K-1} p_{K,i} - p_{i,i} \tag{1}$$

$$FWT = \frac{1}{K-1}\sum_{i=2}^{K} p_{i-1,i} - \bar{b}_i \tag{2}$$

The BWT denotes a measure for the average difference between the initial performance after training on T_i and the final performance after training on all tasks, i.e., after T_K. For positive values of BWT, the model is gaining performance compared to the initial task, while negative BWT values imply that the network has forgotten previous knowledge. The impact that the learning of task T_i has on the subsequent task T_{i+1} is described by FWT. Positive values of FWT indicate that the acquired knowledge on task T_i transfers well to the subsequent task. If the FWT is negative, the training of task T_i worsens the performance of the model for task T_{i+1} compared to the initial baseline $\bar{b}_i$. As the initial performance is evaluated after the initialization of the network for object detection, significant values for $\bar{b}$ are not to be expected. Even for pre-trained backbone networks, we employ models with randomly initialized heads for object detection. Consequently, negative values seldom occur. We introduce BWT(%) as the BWT normalized by $\frac{1}{K-1}\sum_{i=1}^{K-1} P_{K,i}$ to obtain a relative notion of the forgetting. Similar to the average accuracy (Lopez-Paz and Ranzato, 2017), we introduce the performance metric avg. mAP, which denotes an average over the mAPs after training on all K tasks, i.e., $avg.\ mAP = \frac{1}{K}\sum_{i=1}^{K} P_{K,i}$. The gap between avg. mAP for incremental learning and mAP for joint, i.e., non-incremental, training describes the performance difference due to sequential training. We refer to this difference as overall forgetting.

4.5 Implementation Details

To quantify the influence of the architecture, the introduced state-of-the-art detection networks from Section 3.2 are evaluated. Faster-RCNN (Ren et al., 2015) with a ResNet50 (He et al., 2016) backbone is used as the default architecture for all experiments, if not otherwise stated. For our CL scenarios, the

Table 1: Influence of the order of tasks for domain-incremental learning on BDD100K and for the detector architecture Faster-RCNN. For the available domains, the obtained performance (avg. mAP), the relative forgetting (BWT(%)) and the ability to generalize to the next domain (FWT) are presented. Each row corresponds to a different sequence of tasks.

Order	avg. mAP	BWT(%)	FWT	mAP (joint)
Time of Day (Day, Night, Dawn/Dusk)				
Day → Dawn → Night	0.263	-10.5%	0.253	0.304
Day → Night → Dawn †	0.228	-15.0%	0.209	
Dawn → Day → Night	0.241	-5.7%	0.21	
Dawn → Night → Day ‡	0.264	**11.5%**	0.189	
Night → Day → Dawn	0.263	-10.2%	**0.263**	
Night → Dawn → Day	**0.268**	7.2%	0.223	
Scene (Citystreet, Highway, Residential, ...)				
Descending by occurrences	0.244	-18.3%	**0.267**	0.304
Ascending by occurrences	**0.286**	**176%**	0.085	
Weather (Clear, Overcast, Snowy, Rainy, ...)				
Descending by occurrences	0.229	-7.9%	**0.238**	0.304
Ascending by occurrences	**0.250**	**35.8%**	0.166	

†: Descending by occurrences ‡: Ascending by occurrences

dataset of each task is constructed by filtering the original dataset according to the domain or class information. For class-incremental training, the original validation dataset is preserved, if available. The domain-incremental datasets are each split randomly into training and validation by the ratio of 80:20. For SODA10M, both training and validation datasets are merged into one and then split randomly into training and validation datasets. We employ the CL framework Avalanche (Lomonaco et al., 2021) and utilize MMDetection (Chen et al., 2019) for the object detection architectures. For all experiments, we only use random flipping as augmentation and train on the number of epochs with learning rate scheduling as proposed by the framework for each network.

5 RESULTS & DISCUSSION

For both datasets, experiments are conducted with different orderings of domains (***H1***). The data distributions are taken into account for the evaluation and possible sources of diverse effects are outlined (***H2***). The network architectures are subject to the analysis by comparing their performance for identical scenarios (***H3***). The first three hypotheses are analyzed for domain-incremental learning. Lastly, we examine the severity of forgetting for class-incremental learning and compare the results to domain-incremental learning (***H4***).

5.1 Hypothesis *H1* – The Order of Tasks Influences the Forgetting

Table 1 depicts the results for domain-incremental learning. The overall forgetting is evident for all domains, as the mAP is evidently below the performance of non-incremental training. Major performance differences are apparent for the same domain if the order of increments is altered. For the *Time of Day* domain, setups with presumably different strengths of domain gaps were analyzed, e.g., the difference in data distribution of subsequent tasks is assumed to be lower for the sequence Day → Dawn than it is for Day → Night. The results show that the ordering with respect to the number of occurrences has a higher impact on the backward transfer than the strength of the domain gap as the descending/ascending case has the lowest/highest forgetting. However, consecutive tasks with low domain gap strength, e.g., Day → Dawn, and simultaneously descending occurrences have high FWT values, thus, generalize well to new data. Due to the intractability of all permutations of domains, we determined the order of the increments for the domains *Scene* and *Weather* only by the image occurrences in descending and ascending order. For both datasets, the relative BWT and, thus, the forgetting is lowest in the ascending case, while the descending order achieves the highest FWT.

Table 2: Performance overview of different architectures for SODA10M for domain-incremental training. The obtained performance (avg. mAP), the forgetting (BWT(%)) and the ability to generalize to the next domain (FWT) are presented. The joint training serves as upper baseline as all data is available at once without sequential training.

Architecture	avg. mAP	BWT(%)	FWT	mAP (joint)
Time of Day (Day → Night)				
Faster-RCNN (ResNet)	0.434	-19.6%	0.255	0.446
Faster-RCNN (Swin)	**0.452**	**-14.8%**	**0.304**	0.475
FCOS (ResNet)	0.376	-27.1%	0.205	0.397
DDETR (ResNet)	0.416	-25.4%	0.286	0.479
Scene (Citystreet → Highway → Countryroad)				
Faster-RCNN (ResNet)	**0.428**	-10.9%	**0.366**	0.446
Faster-RCNN (Swin)	0.426	**-9.8%**	0.331	0.475
FCOS (ResNet)	0.333	-20.2%	0.283	0.397
DDETR (ResNet)	0.331	-20.9%	0.271	0.479
Weather (Clear → Overcast → Rainy)				
Faster-RCNN (ResNet)	0.431	-15.2%	**0.417**	0.446
Faster-RCNN (Swin)	**0.448**	**-8.5%**	0.415	0.475
FCOS (ResNet)	0.365	-20.9%	0.353	0.397
DDETR (ResNet)	0.384	-27.0%	0.411	0.479

5.2 Hypothesis *H2* – The Data Distribution of the Input Influences the Severity of Forgetting

As previously shown, the order of tasks influences the final performance of the NN on the validation data. To obtain a fair comparison, the data from underrepresented domains was not artificially up-sampled. An intuition for the higher performance for the ascending order is that more data becomes available in subsequent increments. More data increases the performance of the network also on the previous domains, as the domain gap becomes negligible compared to the performance gain due to the higher amount of data. It can be observed that networks almost exclusively obtain the highest mAPs for the current increment and, that the final increment can have a high impact on the overall performance. For the ascending case, the most common domain is trained last, therefore, the NN is fine-tuned on this data and we obtain better overall performance for ascending order. Vice versa, for scenarios in descending order of occurrences, the last increments contain fewer samples. Consequently, it results in inferior overall performance and increased forgetting.

One further influence is that the domain splits in SODA10M are not as granular as they are in BDD100K. Therefore, the domain gap within each increment for SODA10M is assumed to be more extreme. Another aspect is the different sizes of the datasets. BDD100K consists of 100,000 annotated images, while SODA10M provides 10,000 data samples. Hence, we assume that overfitting to one domain is less likely and the vulnerability to forgetting is decreased.

5.3 Hypothesis *H3* – The Architecture of the Detector Has an Influence on the Forgetting

To evaluate the influence of the NN architecture on incremental learning, all introduced architectures were trained and evaluated on the presented datasets. For domain-incremental learning, the performance of each architecture is shown in Table 2 for SODA10M.

The architecture choice influences the severity of forgetting and the magnitude of the FWT. Faster-RCNN shows consistently the best performance after the incremental training with the least amount of forgetting. Further, the usage of the Swin-Transformer as backbone shows a higher avg. mAPs and lessens forgetting compared to the ResNet backbone. The improved performance can be due to presumably better representations. Also, it can be explained by the higher number of network parameters. FCOS shows low overall performance with high forgetting, as depicted in Table 2. Moreover, it has low FWT values for all domains and datasets, which can indicate a poor generalization to other domains. With some exceptions, DDETR is most affected by incremental learning with respect to the introduced metrics as the forgetting appears to be most severe for this architecture.

Table 3: Class-incremental learning with 5-5 and 9-1 increments for BDD100K and Faster-RCNN. Each row corresponds to one task-increment, while each column represents the performance for the given class.

Inc.	**AP** Car	Truck	Bus	Mot.-cycle	Train	Rider	Bi-cycle	Tr. Sign	Tr. Light	Pedes-trian	**mAP**
5-5											
#1	**0.499**	**0.422**	**0.44**	**0.173**	0.00						0.307
#2	0.00	0.00	0.00	0.00	0.00	**0.331**	**0.234**	**0.224**	**0.372**	**0.233**	0.14
9-1											
#1	**0.497**	**0.436**	**0.445**	**0.195**	0.00	**0.196**	**0.22**	**0.368**	**0.223**		0.287
#2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.333**	0.033

5.4 Hypothesis *H4* – Class-Incremental Learning Leads to More Severe Forgetting Compared to Domain-Incremental Learning

Catastrophic forgetting is evident for class-incremental learning, as depicted in Table 3. The performance is provided as the Average Precision (AP) per class. For all analyzed settings, knowledge of previous classes is directly forgotten once the NN is trained on the subsequent task, as the network fails to detect instances of previous classes.

For class-incremental learning, in each increment only annotations for the current classes are present, and hence no labels for other classes. The NN is trained to detect previous classes, thus, it will output predictions of the previous classes while training on the new data. Due to the missing label information of previous classes, predictions for those classes are treated as *false positives* in the loss computation. This might lead to the intuition that the network is forced to actively forget previous classes. We assessed this assumption by utilizing the ground-truth bounding box information of the unknown instances (of classes from a previous/future increment) of the training data to disregard the corresponding losses for the *false positives* during training. Catastrophic forgetting is neither mitigated nor alleviated by incorporating previous predictions during training. Thus, punishing false positive predictions of the network appears to have only a subordinate influence on forgetting.

6 CONCLUSION

Analyzing the four hypotheses demonstrates that the evaluation of CL highly depends on the chosen scenario. Firstly, domain-incremental scenarios show signs of forgetting, yet the effect of catastrophic forgetting is severe for class-incremental learning. We hypothesize that active forgetting due to *false positives* has a subsidiary role. As the AD data is considerably unbalanced, the order must be accounted for when performing CL due to different extents of domain gaps and due to rare domains with limited data. The detection network Faster-RCNN with its task-agnostic RPN is least affected by catastrophic forgetting. Using a transformer as a backbone further increases the robustness.

With the findings of this work, we expect the results for the evaluation of future CL methods to become more trustworthy, reliable, and interpretable. We encourage that future work on CL should acknowledge these results by incorporating random orderings and more realistic scenarios to achieve better comparability between methods. Reducing the effects of forgetting is subject to another line of work (Delange et al., 2021). Our study emphasizes the importance of mitigation strategies for CL for object detection and motivates future research in this direction.

ACKNOWLEDGEMENTS

The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Climate Action" within the project "KI Delta Learning" (Förderkennzeichen 19A19013H). The authors would like to thank the consortium for the successful cooperation.

REFERENCES

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European Conference on Computer Vision*.

Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X.,

Sun, S., et al. (2019). Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*.

Delange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. (2021). A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Farquhar, S. and Gal, Y. (2018). Towards robust evaluations of continual learning. *arXiv preprint arXiv:1805.09733*.

Han, J., Liang, X., Xu, H., Chen, K., Hong, L., Mao, J., Ye, C., Zhang, W., Li, Z., Liang, X., and Xu, C. (2021). Soda10m: A large-scale 2d self/semi-supervised object detection dataset for autonomous driving.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Hsu, Y.-C., Liu, Y.-C., Ramasamy, A., and Kira, Z. (2018). Re-evaluating continual learning scenarios: A categorization and case for strong baselines. *arXiv preprint arXiv:1810.12488*.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A., Milan, K., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Lomonaco, V., Pellegrini, L., Cossu, A., Carta, A., Graffieti, G., Hayes, T. L., Lange, M. D., et al. (2021). Avalanche: an end-to-end library for continual learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

Lopez-Paz, D. and Ranzato, M. (2017). Gradient episodic memory for continual learning. *Advances in Neural Information Processing Systems*.

McCloskey, M. and Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*.

Mirzadeh, S. I., Chaudhry, A., Yin, D., Nguyen, T., Pascanu, R., Gorur, D., and Farajtabar, M. (2022). Architecture matters in continual learning. *arXiv preprint arXiv:2202.00275*.

Pan, S. J. and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*.

Peng, C., Zhao, K., and Lovell, B. C. (2020). Faster ilod: Incremental learning for object detectors based on faster rcnn. *Pattern Recognition Letters*.

Peng, C., Zhao, K., Maksoud, S., Li, M., and Lovell, B. C. (2021). Sid: Incremental learning for anchor-free object detection via selective and inter-related distillation. *Computer Vision and Image Understanding*.

Pham, Q., Liu, C., and Hoi, S. (2022). Continual normalization: Rethinking batch normalization for online continual learning. *arXiv preprint arXiv:2203.16102*.

Ramasesh, V. V., Lewkowycz, A., and Dyer, E. (2021). Effect of scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations*.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. (2017). icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. (2016). Progressive neural networks. *arXiv preprint arXiv:1606.04671*.

Serra, J., Suris, D., Miron, M., and Karatzoglou, A. (2018). Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*. PMLR.

Shin, H., Lee, J. K., Kim, J., and Kim, J. (2017). Continual learning with deep generative replay. *Advances in Neural Information Processing Systems*.

Shmelkov, K., Schmid, C., and Alahari, K. (2017). Incremental learning of object detectors without catastrophic forgetting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Tian, Z., Shen, C., Chen, H., and He, T. (2019). Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Van de Ven, G. M. and Tolias, A. S. (2019). Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*.

Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., and Darrell, T. (2020). Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Zenke, F., Poole, B., and Ganguli, S. (2017). Continual learning through synaptic intelligence. In *International Conference on Machine Learning*.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. (2021). Deformable detr: Deformable transformers for end-to-end object detection. *International Conference on Learning Representations*.