Continual Cross-Dataset Adaptation in Road Surface Classification

Paolo Cudrano[1], Matteo Bellusci[1], Giuseppe Macino[2] and Matteo Matteucci[1]

Abstract— **Accurate road surface classification is crucial for autonomous vehicles (AVs) to optimize driving conditions, enhance safety, and enable advanced road mapping. However, deep learning models for road surface classification suffer from poor generalization when tested on unseen datasets. To update these models with new information, also the original training dataset must be taken into account, in order to avoid catastrophic forgetting. This is, however, inefficient if not impossible, e.g., when the data is collected in streams or large amounts. To overcome this limitation and enable fast and efficient cross-dataset adaptation, we propose to employ continual learning finetuning methods designed to retain past knowledge while adapting to new data, thus effectively avoiding forgetting. Experimental results demonstrate the superiority of this approach over naive finetuning, achieving performance close to fresh retraining. While solving this known problem, we also provide a general description of how the same technique can be adopted in other AV scenarios. We highlight the potential computational and economic benefits that a continual-based adaptation can bring to the AV industry, while also reducing greenhouse emissions due to unnecessary joint retraining.**

I. INTRODUCTION

Autonomous vehicles (AVs) have gained significant attention in recent years, with tremendous advancements in various aspects of their operation. One crucial aspect is the detection and understanding of the road environment, which includes the ability to classify different types of road surfaces. Road surface classification is critical for AVs as it provides valuable information about the road material, allowing the vehicle to adapt its driving operating conditions accordingly. For instance, when traversing an unpaved rural road, the vehicle can modulate its velocity and adjust its steering to navigate safely through the challenging terrain. Similarly, on a cobblestone road, the presence of rain can increase the risk of skidding, necessitating proactive measures to avoid potential hazards. In addition, road surface classification facilitates the creation of high-definition (HD) maps, enabling AVs to receive information about road conditions in advance. With this information, vehicles can plan their routes more effectively, enhancing the overall driving experience. Moreover, knowledge of the pavement material is crucial when developing realistic simulations for testing and validating AVs and Advanced Driver Assistance Systems (ADAS). Indeed, this knowledge enables the recreation of the driving scenario with higher fidelity.

To tackle the road surface classification problem, current approaches rely on deep learning techniques due to their exceptional ability to extract intricate features from visual data. However, deep learning models heavily rely on the quality and diversity of their training dataset, and several issues can arise when applying these models to real-world scenarios [1]. For instance, biased or unrepresentative training data can lead to models that perform poorly or unfairly [2]. Additionally, the lack of diversity in the training data can limit the ability of models to generalize to unseen situations or novel datasets. It is known that the distribution shift between different datasets can indeed significantly impact model performance, as models trained on one dataset often fail to generalize when tested on other datasets (Fig. 1a–b), i.e., they show poor cross-dataset generalization [3].

In the context of AVs, a large number of datasets have been collected. For the task of road surface classification, in particular, three datasets are publicly available and contain all the necessary information: RTK [4], KITTI [5], and CaRINA [6]. It has been shown in the literature [4] that existing deep learning models trained on one of these three datasets exhibit inadequate generalization when tested on the remaining two. This limitation poses a significant obstacle to the deployment of robust road surface classification systems in autonomous vehicles, as it highlights how these models cannot operate reliably in all plausible real-world scenarios.

Addressing this cross-dataset generalization problem is critical. A straightforward approach would be to perform finetuning on the model using the target dataset. However, finetuning only on novel data leads to catastrophic forgetting [7], i.e., the model's performance on previously learned tasks quickly deteriorates as it becomes optimized only for the new data (Fig. 1c). A consequent idea would be to include the original dataset in the data used for finetuning, forcing the model to remember past data while learning from novel samples. Including the entire original dataset in the training, however, can be computationally inefficient, especially when dealing with large historical datasets. Additionally, access to past datasets may not even be feasible in settings of online training, or it may be forbidden due to privacy restrictions.

To overcome this limitation, an alternative solution is necessary: updating the model solely based on the new data, yet avoiding forgetting past knowledge (Fig. 1d). This problem is precisely the focus of continual learning (CL), a research field studying how to mitigate catastrophic forgetting.

In this work, we exploit one such continual learning strategy to solve the problem of dataset adaptation in road surface classification. In doing so, our contribution is twofold. First, we propose a practical solution to the known problem of poor

[1,2]Department of Electronics Information and Bioengineering, Politecnico di Milano, p.zza Leonardo da Vinci 32, Milan, Italy name.surname @ { polimi.it[1], mail.polimi.it[2]}

This paper is supported by "Sustainable Mobility Center (Centro Nazionale per la Mobilità Sostenibile – CNMS)" project funded by the European Union NextGenerationEU program within the PNRR, Mission 4 Component 2 Investment 1.4.

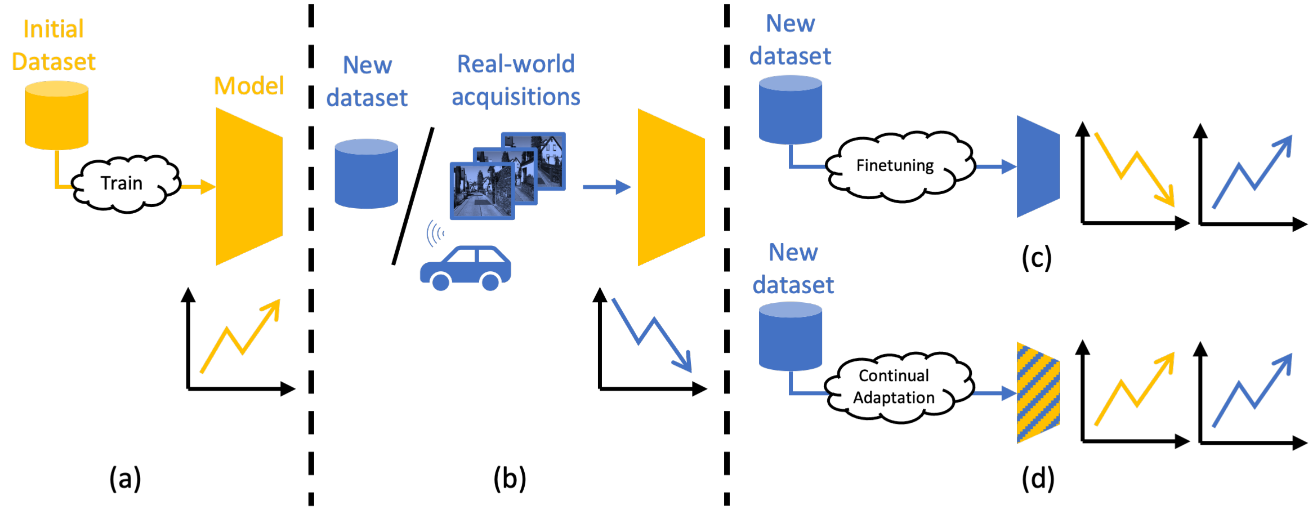

Fig. 1: Models adopted in AVs are typically trained on a single well-known dataset (yellow). These models seem to perform well on the test data from the same dataset (a), but when applied to other datasets or in the real-world (blue), their performance drops as they lack generalization (b). It is crucial to still update such models as new data is available. Finetuning only on new data, however, leads to catastrophic forgetting, where the model learns the new information but loses previous knowledge (c). Instead, we propose to adopt continual learning strategies to allow for model updates while retaining past knowledge (d).

cross-dataset generalization in road surface classification, showing how it is possible to efficiently update a model using only new data. Second, we showcase the applicability of continual learning strategies not only to this specific task, but to any AV task that necessitates updating a model with new data effectively yet efficiently.

II. RELATED WORK

Road surface classification has been addressed by several works in the literature. Pereira et al. [8] distinguish between paved and unpaved road surfaces using a Convolutional Neural Network (CNN) based on VGG [9]. They train their deep model with smartphone-collected images acquired under several weather conditions. Nolte et al. [10], instead, compare CNN architectures at classifying the road surface (asphalt, dirt, grass, wet, cobblestone, snow) under different weather conditions, exploiting various datasets as Oxford RoboCar [11] and KITTI [5]. Using a custom CNN architecture, Rateke *et al.* [4] classify the road material (asphalt, paved, unpaved), and then proceed to analyze road quality and conditions (e.g., presence of potholes) with an ad-hoc system. They train their road material classifier using monocular images from KITTI [5], CaRINA [6], and to properly train on unpaved conditions, they additionally propose a new purposefully-collected dataset, RTK. In the same work, Rateke *et al.* [4] also report cross-dataset generalization issues when training on one single dataset and testing on the other two. They propose to mitigate this problem by training their model jointly on all three datasets. This, however, is not only inefficient, but it implies that, if a criticality is discovered in the future, retraining on all available datasets will be necessary again in order to fix it.

Similar poor cross-generalization performances are reported by Gesnouin *et al.* [12] in the context of pedestrian crossing detection, a completely different task in autonomous vehicle perception. This shows that the problem of poor cross-dataset generalization is widespread and hinders the reliability of potentially any AV perception system. A systematic approach must be found to solve it consistently.

As continual learning gains attention in the deep learning community, applications in industrial and real-world setups are emerging, including works on other autonomous driving problems. Shaheen *et al.* [13] propose a review of the literature and discuss emerging applications and related challenges for autonomous vehicles, including insights on scene recognition, lane keeping, and Reinforcement Learning (RL)-based vehicle control. Verwimp *et al.* [14], instead, recently proposed a new benchmark for continual learning, specifically designed for autonomous vehicle tasks. Zhang and Mueller [15] assess the feasibility of online continual learning for on-board vehicle devices, proposing a lightweight object detection architecture. Kalb *et al.* [16] adapt the formulation of semantic segmentation to a continual learning setup, validating their work on autonomous driving benchmarks. In a successive work, Kalb *et al.* [17] further investigate the performance of replay techniques when dealing with segmentation tasks, field previously almost unexplored. In segmentation, Termöhlen *et al.* [18] use frequency-based style transfer to perform continual domain adaptation and contrast catastrophic forgetting. Also traffic monitoring and forecasting has seen interest in early continual learning works. Among these, Tsai *et al.* [19] perform traffic forecasting using an online version of EWC [20], with application to traffic pattern discovery during COVID-19 road regulation changes. Sun *et al.* [21], instead, apply continual techniques to a camera-based traffic advisory system.

III. CONTINUAL LEARNING

Before framing the problem of road surface classification in the context of continual learning, we briefly present the field and its current developments.

Continual Learning (CL) aims at making learning systems able to acquire knowledge through time from a possibly-infinite stream of data, without forgetting past learned information. The main problem faced by CL is catastrophic forgetting, i.e., the inability of deep learning models to consistently retain past knowledge when trained on new data.

This characteristic is rooted in the optimization procedure adopted to train these models. Let us consider, without loss of generality, the most simple scenario of a deep learning model (or network), whose parameters (or weights) are optimized (or trained) in a fully supervised setting on real-world high-dimensional data, e.g., camera images. Let us also assume we have a mechanism to sample input data and associated target outputs from their real-world statistical distribution.

To find the best weights for our model, we start acquiring a random data sample and its associated target output, and use our model with its current weights to predict an output. We can then compare this prediction to its associated target by means of a loss function (e.g., a distance measure), which assesses the model's performance. As model and loss function are smooth, by chain rule we can compute the gradient of this loss with respect to each model weight, and perform a step of gradient descent (or variations) in such direction. Doing so will slightly pull the weights of our model towards a configuration that performs better on the current data sample.

Repeating this procedure over different samples, the weight updates give rise to a tug-of-war in the weight space, pulling weights in the optimal direction for each data instance. We can expect that, after a large number of samples, the weights converge to a region of compromise that performs adequately over all data samples. It can be shown that this indeed happens, but only if our sampling procedure generates *i.i.d.* (independent and identically distributed) samples over the *entire data distribution*. Intuitively, this means that the samples must span over the entire distribution and that regions of such distribution are eventually revisited.

Clearly, this condition is strict and unrealistic for real-world data, but to approach it, two steps are always taken. First, a very large amount of samples is used in the optimization, in order to best approximate the entire data distribution; and second, these samples are shuffled, in order to break any inherent dependency. To carry on these two operations practically, these samples are preemptively collected and stored in the form of a dataset. It is for this reason that, trying to run this procedure on an already-trained model and sampling only from a new dataset, inevitably leads to catastrophic forgetting, as we are sampling only from a portion of the entire data distribution of interest, thus violating the above condition. Intuitively, this is because the tug-of-war mechanism internal to the optimization is not balanced, and will end up favoring the performance of the model only on newly-seen data, at the expense of older ones.

Although solutions for catastrophic forgetting are still currently under research, several training strategies have been proposed in the literature to mitigate it, finding a trade-off for the stability-plasticity dilemma [22]. Such strategies are traditionally categorized into three classes: replay-based, architecture-based, and regularization-based [22]. In recent times, hybrid versions [23] are also emerging.

Replay-based strategies, such as iCaRL [24], DGR [25], and GEM [26], represent the most straightforward approaches. They focus on storing in a memory buffer a few of the seen samples during training, in order to mix them with the new data when learning future experiences. In this way, they recreate a simile-i.i.d. setup and thus maintain good performance on past experiences. The main focus in this direction is devoted, of course, to the selection of as few most representative samples as possible, in order to maintain low memory consumption. Generative models and latent replay are also commonly adopted as rehearsal strategies [27].

Architecture-based strategies, instead, physically modify the network architecture to account for the incoming new information. A simple baseline consists in adding a new head to the network every time a new experience is encountered. This expedient untangles stability and plasticity, as it learns custom weights for each experience, at the cost of rapidly-increasing memory consumption. Among others, examples are progressive neural networks (PNN) [28] and HAT [29].

Regularization-based strategies, at last, deal with catastrophic forgetting by constraining the weight updates through regularized losses, in order to prevent large changes of the weights more responsible for modeling past data distributions. In a sense, these techniques deal with the root of catastrophic forgetting, controlling the tug-of-war mechanism and dealing with the problem of credit assignment [30]. Based on theoretical neuroscience considerations on synaptic consolidation [30], [31], these approaches provide the advantage of having low extra-memory requirements. Among these methods, Elastic Weight Consolidation (EWC) [20] limits weights' gradients proportionally to their importance for previous experiences. These importances are estimated by computing their Fisher information matrix at the end of each experience. Synaptic Intelligence (SI) [32], instead, estimates similar weight importances in an online manner, allowing for a smooth online transition between consecutive experiences. These strategies, however, must be applied from the beginning of the training, and thus cannot be adopted with pretrained models. Conversely, Less-Forgetful Learning (LFL) [33] directly uses the new data as a proxy for the old ones. In this way, it is able to regularize the embedding space generated by the last layer of the classifier.

IV. CONTINUAL ADAPTATION TO NEW DATA

With a background in CL, we can now frame the problem of dataset adaptation in the context of continual learning. Specifically, we focus our attention on the task of road surface classification, although the following formulation is general to other AV applications.

In road surface classification, we consider a deep network F_θ of parameters θ that has the task of classifying the road surface between three categories: asphalt, unpaved (e.g., gravel roads) and paved (other than asphalt, e.g., cobblestone). Let us assume that F is first trained on dataset D_0. We can denote this trained network with $F_\theta = F_{\theta_0^*}$, where θ_0^* represents the optimal parameters obtained training on D_0. Notice that D_0 could be a large pretraining dataset and might be unavailable at a later time. Our objective is to adapt the model weights θ to a new dataset D_i such that, while new knowledge is acquired, the network retains high performance also on all previously seen datasets $(D_0, \ldots, D_{i-1})$. Here, we use i to index experiences, which correspond with a new dataset being presented to the network.

In a general continual learning setup, at every new experience, changes can occur in the distribution of the input data, the distribution of the targets, or the set of target labels. In dataset adaptation, however, we consider only changes in the input data distribution. This setup is also known in the literature as domain-incremental learning [27]. Nevertheless, our formulation can be adapted for task-incremental, class-incremental, and data-incremental setups [27].

Among all possible strategies proposed in the literature, for the task of dataset adaptation we rely on Less-Forgetful Learning (LFL), proposed by Jung *et al.* [33]. In the following, we summarize its working mechanisms and highlight its main advantages over the alternatives.

A. Less-Forgetful Learning (LFL)

Less-Forgetful Learning (LFL) [33] is a regularization-based strategy perfect for the problem of continual domain adaptation, i.e., avoiding catastrophic forgetting due to input domain shifts. LFL can be applied on pretrained models as-is, as it does not require any additional computation during training on the first dataset. This is a crucial feature in the context of this work, as we aim to rapidly adapt already existing models as new information becomes available.

Considering a deep feed-forward neural network with a final softmax classification layer, the authors of LFL consider the top layer F^L as a linear classifier, while the rest of the network F^{L-1} is considered as a feature extractor, mapping the input data to a linearly separable embedding space. Building on this consideration, the authors then highlight two key desiderata for their continual strategy:

A. The decision boundaries outlined by the top layer F^L must remain unchanged between different experiences.
B. The embedding of data from past experiences should not move significantly when updating the network.

Clearly, B. cannot be guaranteed in general, as it requires constant access to past datasets. Nevertheless, the authors find that using data from new experiences as a proxy for the old ones still provides good performances. As a consequence, given a model $F_\theta = F_{\theta_0^*}$, pretrained on datasets D_0, their strategy for updating F_θ using dataset D_1 prescribes to:

1) Copy the current optimal network weights $\theta_1 := \theta_0^*$.
2) Freeze the top softmax layer F^L (to achieve A.).

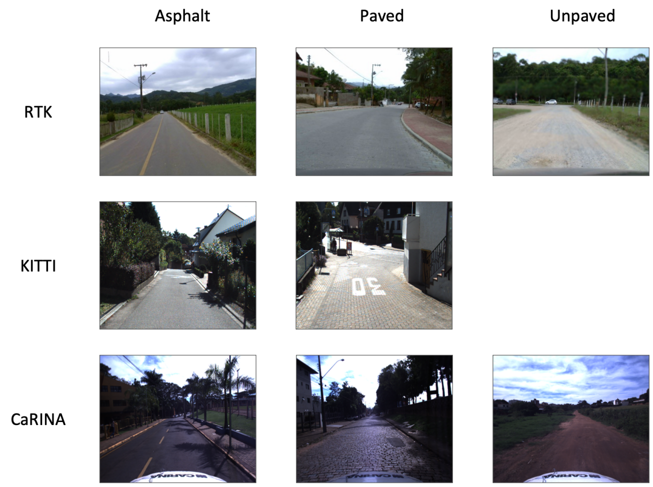

Fig. 2: The three mainly known public datasets for road surface classification: RTK, KITTI, and CaRINA. They provide three label categories: asphalt, paved (e.g., cobblestone), and unpaved. Empty spaces are left where no data is present.

3) Train F_θ on dataset D_1 with loss:

$$\mathcal{L}(x; \theta_1, \theta_0^*) = \mathcal{L}_c(x; \theta_1) + \lambda_e \mathcal{L}_e(x; \theta_1, \theta_0^*) + \mathcal{R}(\theta_1), \quad (1)$$

where $x \in D_1$, $\mathcal{R}$ is a regularization term (such as an L_2 penalty), and $\mathcal{L}_e$ (enforcing B.) is:

$$\mathcal{L}_e(x; \theta_1, \theta_0^*) = \frac{1}{2} \left\| F^{L-1}_{\theta_0^*}(x) - F^{L-1}_{\theta_1}(x) \right\|_2 . \quad (2)$$

This procedure can be repeated for any further new dataset D_i becoming available over time. In the end, F_θ will have converged to a configuration that has acquired as much knowledge as possible from the last dataset, without forgetting what was learned from past datasets.

V. EXPERIMENTS

A. Experimental setup

To validate the efficacy of our methodology in achieving efficient dataset adaptation, we follow the same experimental setup proposed by Rateke *et al.* [4]. In particular, we replicate their findings that the task of road surface classification presents poor cross-dataset generalization, and we show how the continual adaptation strategy presented in Sec. IV significantly mitigates the problem without needing to store and process past data.

To maintain replicability, we employ the CNN architecture described in their work [4]. Moreover, we employ the same three datasets: RTK [4], KITTI [5], and CaRINA [6]. Each dataset is divided into train and test splits. Fig. 2 displays a few samples of the images present in each dataset, highlighting the three possible output labels: asphalt, unpaved, and paved. In their work, [4] consider RTK as the first training source (D_0 in our notation) because of its larger size, more suitable for the initial training; KITTI and CaRINA instead, containing fewer samples, are considered as secondary datasets (D_1, D_2). When training only on RTK and testing on KITTI and CaRINA, Rateke *et al.* report

TABLE I
NAIVE STRATEGY

	Testing dataset					
	RTK		KITTI		CaRINA	
Training dataset	AUROC	F1	AUROC	F1	AUROC	F1
RTK	0.9932	0.9614	0.3778	0.4462	0.7206	0.5660
RTK → KITTI	0.7526	0.5083	0.6521	0.9462	0.4564	0.3868
RTK → KITTI → CaRINA	0.5972	0.6924	0.6189	0.8385	0.9995	0.9717

TABLE II
LFL STRATEGY

	Testing dataset					
	RTK		KITTI		CaRINA	
Training dataset	AUROC	F1	AUROC	F1	AUROC	F1
RTK	0.9932	0.9614	0.3778	0.4462	0.7206	0.5660
RTK → KITTI	0.9742	0.9186	0.6278	0.8692	0.6606	0.5660
RTK → KITTI → CaRINA	0.9774	0.9338	0.6115	0.8077	0.9983	0.9528

TABLE III
JOINT STRATEGY

	Testing dataset					
	RTK		KITTI		CaRINA	
Training dataset	AUROC	F1	AUROC	F1	AUROC	F1
RTK ∪ KITTI ∪ CaRINA	0.9852	0.9648	0.6353	0.8769	0.9848	0.9340

TABLE IV
COMPARISON OF THE FINAL MODEL AGAINST THE JOINT STRATEGY

	Testing dataset					
	RTK		KITTI		CaRINA	
Strategy	ΔAUROC	ΔF1	ΔAUROC	ΔF1	ΔAUROC	ΔF1
Naive	0.3880	0.2724	0.0164	0.0384	−0.0147	−0.0377
LFL	0.0078	0.0310	0.0238	0.0692	−0.0135	−0.0188

significantly lower metrics on the latter two, highlighting poor cross-dataset generalization. For this reason, they then repeat the experiment training jointly on all three datasets, achieving satisfactory performances on all three test sets, at the cost of a longer and more expensive training. We replicate their results and extend their experimental setup to evaluate the performance of the reference model when trained using two baselines (naive and joint training [27]) and the LFL continual learning strategy. In particular:

Naive finetuning. After our model has been trained on RTK, we continue finetuning it, first on KITTI and then on CaRINA. This replicates the findings of [4] and represents a lower bound for our expected performance, as it is the best setup for catastrophic forgetting to occur.

LFL (continual adaptation). After our model has been trained on RTK, we use the continual learning strategy LFL (Sec. IV-A) to finetune it first on KITTI and then on CaRINA. Notice that LFL does not need to be applied when training on RTK.

Joint training. We discard our model trained only on RTK, and retrain it from scratch over the combination of all three datasets, as proposed in [4]. This represents an upper bound to our expected performance as, merging and shuffling all available data, no forgetting can occur.

Our entire experimental setup is implemented in PyTorch [34], using the Avalanche framework [35]. We train each experience for 30 epochs, using an SGD optimizer with learning rate of 0.002 and $\lambda_e = 1$. These values have been finetuned to yield the best results and thus propose a fair evaluation. All experiments are executed on an Nvidia Quadro RTX 6000 GPU.

B. Experimental results

We evaluate the performance of the reference model in terms of AUROC and F1-score, with the former being the most indicative as it balances precision and recall and is not affected by class imbalance.

To evaluate the impact of forgetting, we compute these metrics after training on the initial dataset (RTK), and after adding each additional dataset (KITTI, CaRINA). These metrics are computed individually on the test splits of each dataset. In this way, we can also evaluate the performances yielded on KITTI and CaRINA even before they are seen by the model, showing the model's cross-dataset performance.

Tables I, II and III report the results of this analysis for Naive, LFL, and Joint strategies respectively. Notice that the Naive strategy can be considered as an approximate lower bound for performances, while the Joint strategy as an approximate upper bound.

From Tables I and II, we notice that both Naive and LFL strategies report the same performance when training on the first dataset, RTK. This is because LFL does not intervene during the first training, thus both strategies perform the same exact computation at this stage. We notice that while the model has great performance on RTK, it generalizes poorly to both unseen datasets KITTI and CaRINA, despite the classification task being the same. This replicates the poor cross-dataset generalization reported by Rateke *et al.* [4].

When finetuning on the second dataset (KITTI), Naive and LFL present different behaviors. In both cases, the performance on KITTI rises, as the model is adapting to its data distribution. However, Naive leads to a significant drop in performance on the RTK test set, going from an AUROC of 0.9932 to 0.7526. This is a clear manifestation of catastrophic forgetting: the model has adapted to the new data distribution but has lost performance on the previous distribution. LFL, on the contrary, displays only a minor perturbation, and its AUROC performance on RTK remains above 0.97 even when learning from a new dataset. The same trend is noticeable when we move to the third dataset (CaRINA), as the AUROC on RTK drops further to 0.5972 under the Naive strategy. This confirms that this strategy leads to significant catastrophic forgetting, while LFL maintains a high performance throughout the model's lifetime.

When comparing to the upper bound performance given by the Joint strategy in Table IV, we notice that the final model obtained through sequential updates achieves comparable performance when adopting the LFL strategy. Indeed, the model obtained with LFL reports a difference of at most 2.38% AUROC on any dataset. The Naive strategy, instead,

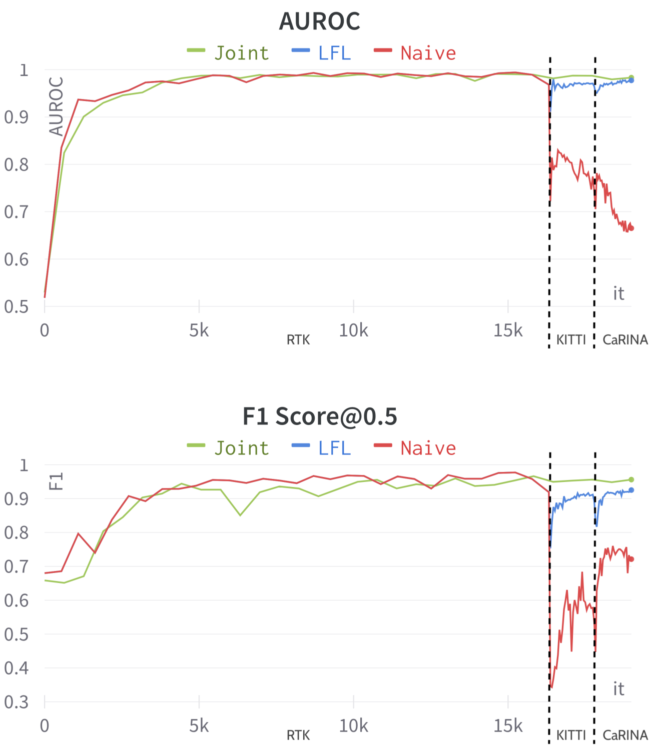

Fig. 3: Trend of AUROC and F1 metrics while training on successive datasets (RTK, followed by KITTI, followed by CaRINA). The metrics are computed over the union of the test sets of these three datasets. It is evident how the Naive strategy sustains a substantial drop in performance when a new dataset is introduced, and never recovers from this occurrence. The continual strategy, LFL, instead, although slightly affected by the change of distribution, quickly stabilizes and maintains almost perfect performances (with respect to a Joint full retraining).

leads to drops in AUROC of 38.80% on RTK, significantly impacting the reliability of the system on past data.

As current CL strategies are not perfectly mature, we realize that the performance of LFL remains slightly inferior to a Joint training (Table IV). However, if datasets related to past experiences are not available, this is the best we can achieve, and it improves significantly over the Naive approach. Moreover, even if we do have access to all past data, we might still want to consider whether to use it entirely. Indeed, we must also take into account the speedup induced by continual adaptation. In our experiments, finetuning only on the last dataset was found to be 5 times faster than training using all past data. In this work, the dataset dimensions are modest, and thus the computation required could still be handled. However, with large-scale datasets, the saved time is magnified significantly. As a result, when dealing with similar problems of dataset adaptation, we advise practitioners to evaluate the trade-off between the slight performance loss experienced by continual adaptation and the efficiency gained in terms of computation resources.

Incidentally, we notice from Table IV that, on the last dataset (CaRINA), Naive and LFL strategies achieve slightly higher performance than Joint. Intuitively, this is due to how each strategy deals with the stability-plasticity tradeoff. Naive and LFL can focus their optimization only on new data, thus fitting it slightly better than Joint, which is instead fed random samples from all three datasets.

For additional reference, we report the trend during the entire training of the AUROC and F1 metrics computed on the union of the test sets (Fig. 3). As mentioned above, the Joint strategy serves as an upper bound on the performance. We notice that, while the Naive strategy accumulates a significant gap as new datasets are introduced, LFL maintains almost perfect performance throughout the entire training.

VI. Conclusion

In this work, we addressed the problem of cross-dataset adaptation in road surface classification. We showed how continual learning strategies can be adopted to update a given model when new data becomes available, without the need to store and use past datasets, yet avoiding catastrophic forgetting. The efficacy of continual learning for cross-dataset adaptation is confirmed by our experiments, which also highlighted the occurrence of strong forgetting when adopting a naive finetuning approach. While contributing to tackling this open problem in the field of autonomous vehicles, we believe our formulation provides also the basic tools necessary to apply our approach to other AV tasks. Indeed, the need for updating a model with new data is particularly important for improving the robustness of perception systems when deployed in the real world. With this in mind, we encourage the adoption of continual learning techniques not only in future research, but also in industrial settings, in order to save the computational, economic, and logistic resources due to unnecessary large-scale retraining.

References

[1] T. Tommasi, N. Patricia, B. Caputo, and T. Tuytelaars, “A Deeper Look at Dataset Bias,” in *Domain Adaptation in Computer Vision Applications*. Springer International Publishing, 2017, pp. 37–55. 1

[2] D. Pessach and E. Shmueli, “A review on fairness in machine learning,” *ACM Comput. Surv.*, vol. 55, no. 3, Feb 2022. 1

[3] A. Torralba and A. A. Efros, “Unbiased look at dataset bias,” in *CVPR 2011*. IEEE, 2011, pp. 1521–1528. 1

[4] T. Rateke, K. A. Justen, and A. v. Wangenheim, “Road Surface Classification with Images Captured From Low-cost Camera - Road Traversing Knowledge (RTK) Dataset,” *Revista de Informática Teórica e Aplicada*, vol. 26, no. 3, pp. 50–64, Nov. 2019, number: 3. 1, 2, 4, 5

[5] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1, 2, 4

[6] P. Y. Shinzato *et al.*, “CaRINA dataset: An emerging-country urban scenario benchmark for road detection systems,” in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016. 1, 2, 4

[7] R. M. French, “Catastrophic forgetting in connectionist networks,” *Trends in cognitive sciences*, vol. 3, no. 4, pp. 128–135, 1999. 1

[8] V. Pereira, S. Tamura, S. Hayamizu, and H. Fukai, “Classification of paved and unpaved road image using convolutional neural network for road condition inspection system,” in *2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA)*, 2018, pp. 165–169. 2

[9] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA*, 2015. 2

[10] M. Nolte, N. Kister, and M. Maurer, “Assessment of deep convolutional neural networks for road surface classification,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 381–386. 2

[11] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, “1 Year, 1000km: The Oxford RobotCar Dataset,” *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017. 2

[12] J. Gesnouin, S. Pechberti, B. Stanciulescu, and F. Moutarde, “Assessing Cross-dataset Generalization of Pedestrian Crossing Predictors,” in *2022 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2022, pp. 419–426. 2

[13] K. Shaheen, M. A. Hanif, O. Hasan, and M. Shafique, “Continual Learning for Real-World Autonomous Systems: Algorithms, Challenges and Frameworks,” *Journal of Intelligent & Robotic Systems*, vol. 105, no. 1, p. 9, May 2022. 2

[14] E. Verwimp *et al.*, “CLAD: A realistic Continual Learning benchmark for Autonomous Driving,” *Neural Networks*, vol. 161, pp. 659–669, 2023. 2

[15] H. Zhang and F. Mueller, “CLAIRE: Enabling Continual Learning for Real-time Autonomous Driving with a Dual-head Architecture,” in *2022 IEEE 25th International Symposium On Real-Time Distributed Computing (ISORC)*, May 2022, pp. 1–10. 2

[16] T. Kalb, M. Roschani, M. Ruf, and J. Beyerer, “Continual Learning for Class- and Domain-Incremental Semantic Segmentation,” in *2021 IEEE Intelligent Vehicles Symposium (IV)*, Jul. 2021, pp. 1345–1351. 2

[17] T. Kalb, B. Mauthe, and J. Beyerer, “Improving Replay-Based Continual Semantic Segmentation with Smart Data Selection,” in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, Oct. 2022, pp. 1114–1121. 2

[18] J.-A. Termöhlen, M. Klingner, L. J. Brettin, N. M. Schmidt, and T. Fingscheidt, “Continual Unsupervised Domain Adaptation for Semantic Segmentation by Online Frequency Domain Style Transfer,” in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, Sep. 2021, pp. 2881–2888. 2

[19] M.-J. Tsai, Z. Cui, C. Liu, H. Yang, and Y. Wang, “An Incremental Learning-based Framework for Non-stationary Traffic Representations Clustering and Forecasting,” in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, Oct. 2022, pp. 3237–3242. 2

[20] J. Kirkpatrick *et al.*, “Overcoming catastrophic forgetting in neural networks,” *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017. 2, 3

[21] J. Sun, S. Kousik, D. Fridovich-Keil, and M. Schwager, “Self-Supervised Traffic Advisors: Distributed, Multi-view Traffic Prediction for Smart Cities,” in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, Oct. 2022, pp. 917–922. 2

[22] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez, “Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges,” *Information fusion*, vol. 58, pp. 52–68, 2020. 3

[23] L. Pellegrini, G. Graffieti, V. Lomonaco, and D. Maltoni, “Latent replay for real-time continual learning,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 203–10 209. 3

[24] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, “iCaRL: Incremental Classifier and Representation Learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[25] H. Shin, J. K. Lee, J. Kim, and J. Kim, “Continual learning with deep generative replay,” in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017. 3

[26] D. Lopez-Paz and M. A. Ranzato, “Gradient episodic memory for continual learning,” in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017. 3

[27] M. De Lange *et al.*, “A continual learning survey: Defying forgetting in classification tasks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3366–3385, 2022. 3, 4, 5

[28] A. A. Rusu *et al.*, “Progressive neural networks,” *arXiv preprint arXiv:1606.04671*, 2016. 3

[29] J. Serra, D. Suris, M. Miron, and A. Karatzoglou, “Overcoming catastrophic forgetting with hard attention to the task,” in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 10–15 Jul 2018, pp. 4548–4557. 3

[30] R. Hadsell, D. Rao, A. A. Rusu, and R. Pascanu, “Embracing change: Continual learning in deep neural networks,” *Trends in cognitive sciences*, vol. 24, no. 12, pp. 1028–1040, 2020. 3

[31] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, “Continual lifelong learning with neural networks: A review,” *Neural networks*, vol. 113, pp. 54–71, 2019. 3

[32] F. Zenke, B. Poole, and S. Ganguli, “Continual learning through synaptic intelligence,” in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 06–11 Aug 2017, pp. 3987–3995. 3

[33] H. Jung, J. Ju, M. Jung, and J. Kim, “Less-forgetful learning for domain expansion in deep neural networks,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018. 3, 4

[34] A. Paszke *et al.*, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019. 5

[35] V. Lomonaco *et al.*, “Avalanche: an end-to-end library for continual learning,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, ser. 2nd Continual Learning in Computer Vision Workshop, 2021. 5