Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Are we ready for accurate and unbiased fine-grained vehicle classification in realistic environments?

HÉCTOR CORRALES[1], NOELIA HERNÁNDEZ[1], IGNACIO PARRA[1], EDUARDO NEBOT[2], (Fellow Member, IEEE), DAVID FERNÁNDEZ-LLORCA[1,3], (Senior Member, IEEE)

[1]Computer Engineering Department, University of Alcalá, Spain
[2]Australian Centre for Field Robotics, The University of Sydney, Australia
[3]European Commission, Joint Research Center, Seville, Spain

Corresponding author: H. Corrales (e-mail: hector.corrales@uah.es).

This work has been funded by research grant DPI2017-90035-R (Spanish Ministry Science Innovation), and partially funded by research grants S2018/EMT-4362 SEGVAUTO4.0-CM (Community Region of Madrid, Spain) and CLM18-PIC-051 (Community Region of Castilla la Mancha, Spain). Some of the GPUs used to develop this research were donated by the NVIDIA GPU Program.

ABSTRACT Fine-grained vehicle classification from images, also known as Vehicle Make and Model Recognition (VMMR), has become an important research topic in the last years, with a growing number of scientific contributions in multiple application areas, such as autonomous vehicles, surveillance systems, traffic monitoring and management, among others. Recent techniques based on deep learning have proven to be very effective in addressing this problem. So effective that, based on the state-of-the-art results (above 95% accuracy), it would seem that the problem is practically solved. However, our main hypothesis is that the existing datasets to date have limited variability, which precludes good and unbiased generalisation of the models trained with them. In particular, it is observed that the test datasets are very similar in nature to those used for training and validation which makes these benchmarks prone to dataset bias and to overfitting. When these systems are tested with more challenging data or data from different datasets performance degrades considerably. In this paper, on the one hand, we evaluate state-of-the-art deep learning models to perform fine-grained vehicle classification and explore multiple training techniques, such as curriculum learning or weighted losses, to mitigate the bias between different makes and models and to assess the limits of current approaches. On the other hand, we analyse the existing datasets, present an additional dataset from a challenging scenario, and merge all the data into a cross-dataset that includes common samples and classes from the existing datasets. In this way, we can evaluate geographical, make and model biases, and performance and generalisation capabilities from a more realistic perspective. The obtained results suggest that we are still far from accurate and unbiased vehicle make and model recognition in realistic traffic and driving scenarios.

INDEX TERMS Fine-grained classification, vehicle make and model, dataset bias, curriculum learning, weighted loss, cross-datasets.

I. INTRODUCTION

FINE-GRAINED vehicle classification consists in the classification of vehicles according to make and model and even differentiating between different versions of a particular model (ultra-fine-grained classification). This task is especially useful when used in combination with other applications such as license plate recognition systems to detect if a vehicle is driving with a fake number plate, or in a public car park to detect an attempted theft. In conjunction with keypoint detection methods [1], it is also possible to project 3D structures of a known model obtaining distance, size and perspective information in 2D images. Regarding number plate recognition systems, this information can be used to obtain vehicle data and solve the vehicle classification task, but this approach is vulnerable to recognition errors, license plate swap, and license plate information is not always available. For this reason, a robust system that is able to classify make and model efficiently could be extremely useful.

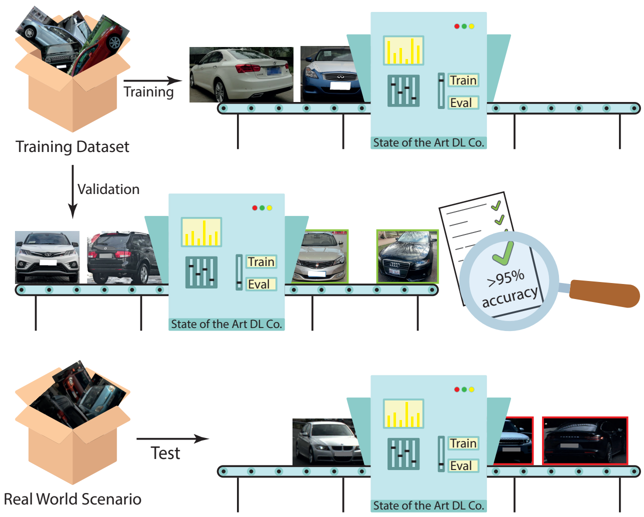

FIGURE 1. Fine-grained vehicle classification from images has achieved above 95% accuracy on validation. Those same models experience a drop in performance when faced with different datasets making them not suitable for real world applications.

There are three main problems when working with fine-grained classification. First, *multiplicity*, i.e., the same model has different shapes and/or appearance depending on the year of manufacture (different versions of the model). Second, *ambiguity*, i.e., two models from different or the same manufacturers have similar appearance. Third, *bias*, i.e., distribution of makes and models is not representative of the actual study population. These issues make the problem of fine-grained vehicle classification a major challenge in which a correctly constructed dataset is of vital importance.

There are a considerable number of existing datasets to deal with the task of fine-grained vehicle classification, which can be divided into two categories. First, specific datasets, created to solve a particular task or limited to a given scenario, such as surveillance [2]–[4]. They tend to be smaller, offering little flexibility and little generalisation potential. Second, general purpose datasets, that aim to advance the state-of-the-art of classification and are intended to be multipurpose, as for example [5]–[8]. The difficulty of constructing a general dataset that accurately represents the reality usually makes them biased, with poor variety of viewpoints, lighting or scenarios, making them less suitable for real world applications.

Most work focuses on solving a specific problem or obtaining raw results, either on previous datasets or on a new general dataset. This leads to the current situation of performance saturation, with datasets such as CompCars [6] saturating at around 98% performance in validation, which suggest that the problem of fine-grained classification is mostly solved. However, when we use these models in more challenging scenarios or analyse performance by each individual class, the results are not entirely satisfactory.

Dataset bias is not usually taken into account, yet it clearly materialises as a class imbalance problem. In datasets with hundreds of classes, one can report 95% average accuracy, even if multiple classes report very low performance. This is because the number of samples for these classes is so low that it barely affects the overall results. We have empirically observed this behaviour when trying to use one of our models, that achieves state-of-the-art results in CompCars [9], in a more realistic scenario (see Fig 1).

In line with the above statements, the aim of this paper is not to present a new method and compare it with previous approaches. The difference in performance between current methods in current datasets is practically negligible. Instead, we focus on the empirical assessment of current limitations and problems, proposing several solutions to address them. In particular, the main contributions of our work can be summarised as follows:

- We study the applicability of curriculum learning techniques to fine-grained vehicle classification problem and evaluate its performance.
- We analyse the effect of bias on the per-class performance of fine-grained models and explore techniques, such as weighted loss, to mitigate its effects and improve performance and generalisation capabilities.
- We propose a test set built from the PREVENTION dataset [10] to externally evaluate performance and generalisation capabilities in realistic scenarios both for makers and models.
- We present a cross-dataset to mitigate biases, and assess the complexity and generalisation capabilities of existing datasets. This cross-dataset is publicly available[1].

The remainder of the paper is organised as follows. Section II briefly summarises the state-of-the-art and the most relevant datasets. The methodology and data augmentation techniques are presented in Section III. An extensive experimental evaluation is provided in Section IV. Conclusions and future work are finally discussed in Section V.

II. RELATED WORK

A. EXISTING DATASETS

Despite the existence of a significant amount of vehicle make and model classification research, most of the existing datasets are small or medium in size, with only a few of large datasets being publicly accessible. This has led researchers to work with their own datasets which, as we have said, are small in size given the high cost of acquiring and labelling thousands of images. Because of this, it is extremely difficult to compare the different approaches as each use a different dataset.

Many previous works perform and extensive analysis of the different datasets [7], [8]. In this paper, we are going to focus exclusively on the most extensive and/or important ones. Among them, we can find Cars-196 [5], CompCars [6], BoxCars [3], [11], VMMR-db [7] or Frontal-103 [8].

Cars-196 dataset is the first large-scale fine-grained vehicle classification dataset. It contains 16,185 images from multi-

[1] https://github.com/ninte/fusion-cross-dataset

ple viewpoints of 196 classes of cars with labels for make, model and year. Being the first one with a relevant size, many later works make use of it, but even so, it still has a limited number of images, these images have professional quality, making them far from real world application and a large number of vehicles are from the same year (2012), which implies poor diversity.

CompCars dataset is probably one of the most relevant vehicle datasets, proposing three different tasks: fine-grained classification, attribute prediction and car verification. It contains data from 2 scenarios, one of web-nature and other of surveillance-nature. The web-nature data was collected from internet forums, websites and search engines with a total of 136,727 images from 163 makers and 1,716 models with different viewpoints. The surveillance-nature data was collected from road surveillance cameras with a total of 44,481 images, all of them from the frontal view. Focusing on the fine-grained classification task, they propose the use of a web-nature subset composed of 52,083 images of 431 different car models. Overall, the CompCars dataset is of reasonable quality. However, there is a considerable geographical bias since most of the vehicle makes and models are specific to the China region, which can be a problem for applications in other regions. The images have professional quality, with even some renders, making them far from a real world situation. The multiplicity problem has been ignored, grouping all versions from a model in one class, even though they have different appearances.

BoxCars is a vehicle dataset focused on surveillance applications. The images were taken from surveillance cameras and, for each vehicle correctly detected, there are 3 images from different viewpoints. The dataset contains 21,250 different vehicles with a total of 63,750 images, 27 makes and 126 models. They also provide make, model, submodel and model year classes annotations and 3D bounding box information. This is a robust dataset, with real-world quality and diversity of views, but low image size and quality. It also suffers from geographical bias as all the images were recorded in the city of Brno in the Czech Republic.

The VMMR-db is probably the most ambitious vehicle classification dataset ever created. It contains a total of 291,752 images of 9,170 different classes. These images were taken by different users and cameras, ensuring a great variety of views, lighting and quality making it realistic. To build the dataset the images were gathered from online vehicle selling web pages and automatically annotated using the title and description provided by the sellers. They provide a subset of 51 classes overlapping with CompCars and a subset of 3,036 classes containing all of those with more than 20 images. Unfortunately, although the dataset does not suffer from the multiplicity problem, having used automatic annotation, for the same model of vehicle there are several classes for different years, even if in those years the specific model was the same. In addition, although with less impact, it also suffers from a certain geographic bias.

Frontal-103 dataset is, to our knowledge, the most recently published vehicle dataset. It is comprised of a total of 65,433 frontal view web-nature images from 103 makers and 1,759 models tackling the multiplicity problem with a different class for each version of a model. Although Frontal-103 is promising, some shortcomings can be found. As in CompCars, the images have professional quality with a non despicable amount of renders. Many of the images are very similar (almost repeated in some cases). A significant number of vehicles have been found to be mislabelled. No training/validation/test split is provided. This is important, as many of the images are very similar or repeated, so many of the images seen in training can also appear in validation and testing. In spite of all this, they face the problem of multiplicity in a competent and effective manner. Finally, as it happens in all other cases, there is a considerable geographical bias. Most of the car manufacturers are from China, which makes a model trained on this data not applicable in other regions in the absence of such vehicles.

To perform an independent evaluation of the different models and assess their generalisation capabilities, we have created a test set based on the PREVENTION dataset [10], which is designed for vehicle intention prediction and contains images from real driving scenarios. The viewpoint and nature of the images is very different from those found in most fine-grained datasets, which is suitable for assessing generalisation capabilities. The PREVENTION dataset has a total of 356 minutes of records for a distance of 540km. Images were obtained from two cameras (front and rear view). We manually selected a total of 2,685 vehicles, 1,452 are front facing images and 1,233 from the back. A total of 33 different makers have been labelled. From these 2,685 vehicles, a total of 1,113 have been labelled at model level, 618 front facing and 515 from the back. A total of 87 different models have been obtained. The reasons why not all vehicles have a model label are the impossibility of obtaining the model reliably or the lack of consensus among annotators.

A summary of all the datasets can be found in Table 1. Fig. 2 depicts four examples of four classes for each dataset (CompCars, VMMR-db, Frontal-103 and PREVENTION).

B. FINE-GRAINED VEHICLE CLASSIFICATION

Fine-grained vehicle classification is a widely explored task. Before CNNs became the standard, classification tasks laid on hand-crafted features. Some of the most remarkable works of the pre-CNN era focused on the inherent characteristics of the vehicles by modelling their geometry and appearance. This approach was used in [12]–[14]. Santos et al. [14] proposed an automatic car recognition system composed of two recognition methods, both relying on the external features of the car. One makes use of the rear view shape, dimension and edges, while the other makes use of features of the back lights. Llorca et al. [13] also used rear view images. They applied a license plate recognition module and a previously developed vehicle make recognition system [15] based on the logo to predict the car make and, after that, learn the geometry and appearance of rear car emblems to predict the

TABLE 1. Summary of the most relevant fine-grained classification datasets.

Dataset	Viewpoint	Nature	# Samples	# Classes	Problems
Cars-196 [5]	Mixed	Web	16,185	196 models	Geographical bias, # images, far from real world, poor diversity
CompCars [6]	Mixed	Web	52,083	431 models	Geographical bias, far from real world, multiplicity problem ignored
BoxCars [3], [11]	Mixed/3D	Surveillance	63,750	126 models	Geographical bias, images size and resolution
VMMR-db [7]	Mixed	Real	291,752	9,170 models	Geographical bias, multiple labels for the same class
Frontal-103 [8]	Front	Web	65,433	1,759 models	Geographical bias, far from real world, labelling issues
Makers Test Set	Front/Back	Driving	2,685	33 Makers	Geographical bias, # images and classes
Models Test Set	Front/Back	Driving	1,113	87 Models	Geographical bias, # images and classes

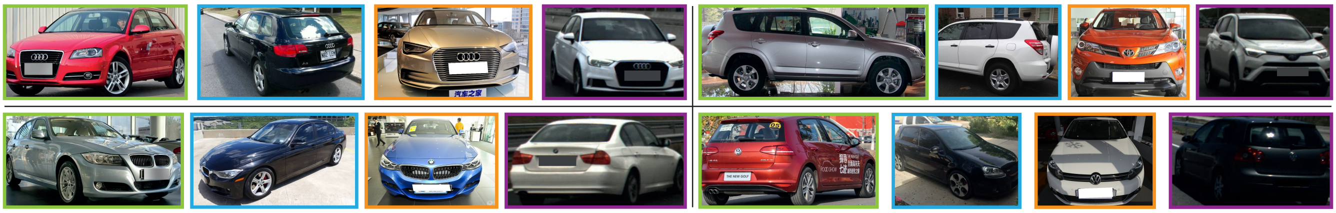

FIGURE 2. Example of images from the three main datasets (CompCars (green). VMMR-db (blue) and Frontal-103 (orange)) and the PREVENTION test set (purple). The images are arranged per-class in groups of four. We have Audi A3 (top left), Toyota RAV4 (Top righ), BMW 3 Series (bottom left) and Volkswagen Golf (bottom right).

model. In [12], Gu et al. proposed a method to deal with severe pose variation. They presented a mirror morphing scheme exploiting the symmetry of cars to normalise any orientation image into a typical view.

Looking deeper into the existing works of the CNNs era, different approaches have been taken to tackle the fine-grained vehicle classification problem, such as focusing on location, appearance and/or parts, working in 3D space or using different networks, modules or training techniques, among others.

In the first group we have those that focus on location, appearance and/or parts [16]–[24]. In [16], Lin et al. proposed a novel end-to-end trained CNN architecture for fine-grained visual recognition called Bilinear CNNs. The idea is to have two networks that extract location and appearance related features and then, combine them as a pooled outer product, obtaining localised feature interactions invariant to translations. They also proved that these bilinear features are highly redundant and that can be reduced an order of magnitude keeping performance practically unaltered. They report results on Cars-196, and although they do not surpass the state-of-the-art, they are close to it. In [17], Krause et al. proposed a method that, instead of using part annotations (like in their previous work [5]), generates parts using co-segmentation and alignment in combination with R-CNN. They show that this approach achieves state-of-the-art results in their dataset (Cars-196), outperforming methods that use part annotations during training. One interesting approach is the taken by Fang et al. [18], in which they tackle the fine-grained vehicle recognition problem by locating discriminative parts where the differences are more evident. To do so, they propose a coarse-to-fine method that makes use of CNNs to extract feature maps and locate these discriminative regions. The feature maps are then used to detect refined regions and extract their features until there are no regions left. Then, all the features (global and local) are used together on a one-versus-all SVM classifier obtaining state-of-the-art results in CompCars surveillance subset. Following the discriminative region approach, Fu et al. [19] presented a novel framework that uses a recurrent attention CNN to recursively learn discriminative region attention and region-based feature representation at multiple scales obtaining similar results to [17] in Cars-196, but without human defined bounding boxes. In [20], Zhao et al. proposed a Diversified Visual Attention Network (DVAN) that is able to gather discriminative information using multiple attention canvases from which it extracts convolutional features. An LSTM recurrent unit is then used to learn the attentiveness and discrimination of these canvases. In [21], Tian et al. followed an approach similar to the one taken by Fang et al. obtaining local and global features too. They proposed an iterative discrimination CNN based on selective multi-convolutional region feature extraction. Two types of features are extracted (local and global), and then used to iteratively localise deep pivotal features and feed them to a fully-connected fusion layer. They report near state-of-the-art results in Cars-196 and in CompCars. In [22], Elkerdawy et al. proposed the use of a co-occurrence layer to discover parts in a unsupervised way, avoiding the use of parts or 3D bounding boxes annotations. They report state of-the-art results in BoxCars and competent results in CompCars. In [23], Du et al. proposed a novel method that adds new layers in each training step exploiting information of the last step and a jigsaw puzzle generator to enhance network input by forming images that contain information from different granularity levels. They report results on several fine-grained classification datasets obtaining state-of-of-the-art results on Cars-196. Recently, Ding et al. [24] outperformed these results using enhanced feature representations and discriminative regions. To do so they presented the Attention Pyramid Convolutional Neural Network (AP-CNN), consisting of two feature and attention pathways used to learn high-level semantic features and low-level detailed features. Following this, they use a ROI-guided strategy that refines features and eliminates background noise.

Among those that work in 3D space we have [3], [5], [25], [26]. One of the limitations of 2D recognition models is that their ability to generalise across different viewpoints is limited. In [5], Krause et al. upgrade two 2D methods to 3D, outperforming its 2D counterparts. To do so they first estimate the 3D geometry of the object and then represent the appearance of local features and their locations in 3D space. In [25], Ramnath et al. proposed a method to recognise make and model from an arbitrary view. They first create a 3D hull from the image an then project 3D space curves and refine them using three-view curve matching. These 3D curves are then matched to 2D image curves using an alignment technique. Lin et al. [26] proposed to optimise 3D model fitting and fine-grained classification jointly. First, they use Deformable Part Models (DPM) to extract initial part locations. Second, they use regression techniques to estimate landmark locations. Then, they fit the 3D model landmarks of a deformable model to the predicted 2D landmarks. With this information they extract part-based features and use them on a SVM classifier. Finally they use the prediction to refine the landmark fitting. In [3], Sochor et al. proposed an enhanced input to a CNN. Instead of using the plain image, they obtain a 3D bounding box used to "unpack" the vehicle image, the shape and orientation, boosting performance both for classification and recognition. The main problems with 3D methods are their high complexity and the need for much denser labeling. If 3D information is not relevant, 2D methods are more efficient and provide, in general, better results.

Finally, there is a plethora of existing work that makes use of different networks, modules or training techniques [9], [27]–[32]. In [27], Anderson et al. used a modular approach combining pretrained networks with new untrained ones. In this way, they get new modules to learn complementary features to those of the pretrained ones. They used Cars-196 to prove their approach. Instead of a new network or training technique, Hu et al. [28] proposed the use of a Spatially Weighted Pooling (SWP) layer to improve the robustness and effectiveness of CNNs feature representations. This novel pooling layer contains a predefined number of spatially weighted masks that are learnt to pool the extracted features in a discriminative way. They obtain state-of-the-art results in both Cars-196 and CompCars. Other approaches focus on the loss function instead of the CNN structure, as in [31], where Li et al. proposed a new regularisation term to cross-entropy loss. The resulting loss function, Dual Cross-Entropy Loss, can help alleviate the vanishing gradient problem and demonstrates good performance with small datasets. They use Cars-196 to prove their approach and obtain state-of-the-art results. In [9], Corrales et al. presented an end-to-end training methodology for fine-grained vehicle classification. By applying diverse techniques like data augmentation, learning rate policies and fine-tuning strategies they achieved state-of-the-art results in CompCars. In [32], Buzzelli et al. *revisited* CompCars, defining a new more challenging and realistic train/test split and propagated the existing type-level annotations to the whole dataset. They also designed and implemented three different methods: one that directly predicts make-model-year, a two-step approach that first predicts vehicle type and then make-model-year and a multilabel approach that predicts both type and make-model-year. They show interesting results, with a new baseline that goes down from ~ 90% to 61% accuracy and achieving 70% accuracy with the two-step method.

As we have seen, there are multiple datasets and approaches to address fine-grained vehicle classification. However, there is a clear tendency to increase the complexity of the models to improve the overall results, neglecting other key aspects such as class imbalance and generalisation capability.

A summary of the CNNs era fine-grained vehicle classification approaches can be found in Table 2.

C. IMBALANCED CLASSES

One of the key problems when working with large classification datasets with a large number of classes is class imbalance. In our experience, we have empirically found that models that perform well on average can have poor generalisation capabilities, reporting very poor results for underrepresented classes. Typically, there are two re-balancing approaches to address this problem, one is re-sampling the data (over-sampling under-represented classes or under-sampling over-represented ones) and the other is to use weights to balance the training. In the case of re-sampling, over-sampling seeks to artificially increase the number of samples of under-represented classes (dataset bias problem). The initial way to solve it was to add repeated samples, at the cost of increasing the risk of overfitting. To prevent it new samples can be either interpolated from existing samples [33], [34] or synthesised [35]–[37]. But, although these new samples prevent overfitting, they could also be noisy, negatively conditioning model performance. The other re-sampling technique, under-sampling, has the risk of leaving behind relevant data, which still seems preferable to over-sampling [38]–[40].

Regarding weight-based methods, a common approach is to use the inverse frequency of each class [41]–[43]. Other approach is to focus on the difficulty measured by the loss of each class [44], use cost-sensitive weighting [45], [46] or use a meta-learning algorithm that learns to assign weights based on the gradients like the one used by Ren et al. [47].

A technique that has recently gained special attention is *Focal Loss* [44]. They proposed a modification of standard

Authors	Year	Dataset	Approach	Model
Krause et al. [5]	2013	Cars-196	3D space	SVM
Ramnath et al. [25]	2014	Custom	3D space	Three-view curve matching
Lin et al. [26]	2014	FG3DCar	3D space	DPM+SVM
Lin et al. [16]	2015	Cars-196	Location, appearance, parts	Bilinear CNNs
Krause et al. [17]	2015	Cars-196	Location, appearance, parts	R-CNN: co-segmentation and alignment
Anderson et al. [27]	2016	Cars-196	Other	CNN: complementary features
Sochor et al. in [3]	2016	BoxCars	3D space	CNN
Fang et al. [18]	2017	CompCars	Location, appearance, parts	CNN+SVM
Fu et al. [19]	2017	Cars-196	Location, appearance, parts	Recurrent attention CNN
Hu et al. [28]	2017	Cars-196 and CompCars	Other	CNN: SWP layer
Zhao et al. [20]	2017	Cars-196	Location, appearance, parts	DVAN+LSTM
Tian et al. [21]	2018	Cars196 and CompCars	Location, appearance, parts	CNN: fully-connected fusion layer
Elkerdawy et al. [22]	2018	BoxCars and CompCars	Location, appearance, parts	CNN: co-ocurrence layer
Li et al. [31]	2019	CompCars	Other	CNN: dual cross-entropy loss
Corrales et al. [9]	2020	CompCars	Other	CNN: fine tuning strategies
Du et al. [23]	2020	Cars-196	Location, appearance, parts	CNN: incremental # of layers
Ding et al. [24]	2021	Cars-196	Location, appearance, parts	AP-CNN
Buzzelli et al. [32]	2021	CompCars	Other	CNN: hierarchical approaches

TABLE 2. CNNs era fine-grained vehicle classification approaches summary.

cross entropy loss by adding a new term that reduces the relative loss of well-classified data and focuses on the harder misclassified ones.

Recently, Cui et al. [48] presented a novel framework to measure data overlapping and compute the effective number of samples for each class. After that, they use a re-weighting scheme to apply the effective number of samples previously computed and re-balance the loss obtaining significant increases in performance on long tailed datasets.

III. METHODOLOGY

In order to tackle the fine-grained vehicle classification problem and evaluate generalisation capabilities, multiple experiments will be carried out. For this purpose, a variety of strategies and methods have been adopted. This section describes the different architectures used, data augmentation techniques, learning rate policies, curriculum learning methods and different loss weighting strategies.

A. ARCHITECTURES

Many years have passed since AlexNet [49]. During this time, CNNs have evolved and today there are countless different models. From VGG [50], the direct evolution of AlexNet, through Inception [51], [52], ResNet [53] or ResNext [54], to Google's EfficientNets [55]. In [56], Bianco et al. presented an in depth analysis of the main Deep Neural Networks (DNNs) used for image recognition reporting multiple performance indices. In this paper, we propose to use the ResNet50 and InceptionV3 models due to two main reasons. First, these two models have a good balance between performance and complexity ratio, with a very efficient use of their parameters [56]. Second, these two models are perfectly capable of addressing the fine-grained vehicle classification problem allowing us not only to obtain a good overall performance, but also enabling the study of the impact of different learning techniques on per-class performance and generalisation, as well as to analyse the quality of the datasets.

B. DATA AUGMENTATION

It is widely accepted by the community that data augmentation is essential to improve model performance and prevent overfitting [57]. In our previous paper [9] we empirically proved the benefits of using data augmentation and tested various techniques:

- *Horizontal Flip*: an horizontal flip (over y axis) with a probability of 50% is performed over the image.
- *Salt and Pepper*: each pixel of the image is set to 0 or 255 with a probability of 2%.
- *Poisson noise*.
- *Speckle noise*.
- *Blurring*: gaussian blur operation is performed over the image with a random kernel size between 3 and 11 and standard deviation of 6.
- *Color Casting*.
- *Color Jittering*: the image is converted to HSV color space and saturation and value are independently randomly modified.

Our data augmentation strategy is applied in each epoch to the training data and performed as follows. First, we randomly apply the flipping operation to each image. Second, we randomly select one of the other six data augmentation operations and apply it to the resulting image. Finally, we apply ImageNet [58] normalisation.

C. LEARNING RATE POLICIES

As with data augmentation, multiple learning rate policies are extensively used by the community. After several experimental validation we selected the following: to keep the learning rate constant (*constant lr*), and to reduce it by an order of magnitude every n epochs in a stepped pattern (*step-n*). The initial learning rates that we use are 0.01 and 0.001 along with Stochastic Gradient Descent (SGD), with 0.9 momentum and 0.0001 weight-decay.

D. CURRICULUM LEARNING

The fact that learning processes can be much more efficient when information is presented in an organised way, progressively expanding the different concepts and difficulty, rather than presented randomly, is an intuitive and reasonable approach that has not yet been sufficiently applied to the domain of deep learning. This is particularly interesting for the fine-grained vehicle classification problem due to the hierarchical structure of the data (makes → models). This idea was first proposed in 1993 by Elman et al. [59] and subsequently explored in 2009 by Bengio et al. [60], showing solid improvements in performance for multiple tasks. In this paper, we conduct a series of experiments to assess the feasibility and impact on per-class performance of two different curriculum learning techniques. The first consists in training an easier, more general problem and then retraining for the desired task. In our case, it seems reasonable to first train the network to classify vehicle makers (general task) and, after that, refine the network to classify models (desired task). In our experiments we refer to this approach as *incremental-learning*. The second technique is to start training an easier problem and, at each epoch, gradually increase the difficulty. For example, in a multi-class classification problem one starts with the easier classes and gradually adds the most difficult ones. We start with the fully connected layer initialised for all classes and show to the model only a subset of the dataset (the classes with the best performance) to gradually add the rest of the classes. We apply two slightly different versions of this technique by adding 5 and 10 new classes every epoch respectively until all the classes are in use. After this, we continue the training for a few more epochs to ensure that the last classes added to the model are trained for more than 1 epoch. We refer to these techniques as *progressive5* and *progressive10* in our experiments.

E. WEIGHTED LOSS FOR CLASS IMBALANCE

The class imbalance problem occurs when one or more of the classes present in the dataset have a weight (number of samples) several orders of magnitude below the rest of the classes. This often means that, in the training process, these classes are irrelevant during back-propagation, so that, although the overall performance of the model is apparently good, these particular classes perform well below average. When these classes appear in real world conditions, we have a bias problem in the dataset. This effect can be mitigated by using loss weights to favour under-represented classes or penalise over-represented ones.

We evaluate up to three different sets of weights. The first one, which we refer as *standard*, is defined in Eq. 1:

$$W_i^1 = 1 - \Big(num\_samples_i / \sum(num\_samples)\Big) \tag{1}$$

where i represents the specific class. This way, all weights are less than 1. However, when no weights are used (all equal to 1), the sum of the weights is the number of classes. We can therefore maintain the proportions by normalising the weights so that adding them together gives the number of classes. This is how the second set of weights is defined, as a modification of the *standard* technique in which weights are normalised so that they add up to the number of classes. We refer to this set as *standard normalised*. For the third and last of the sets we modify the weights with a non-linear function, as defined in Eq. 2. First, we calculate the percentage of representation in the dataset of each class and, after that, we use the non-linear function $-log(x)$. Additionally, we use the number of classes normalised version as in the second set. We refer to this set as *log*.

$$\begin{gathered} w_i^2 = num\_samples_i / \sum(num\_samples) \\ W_i^2 = -log(w_i^2) \end{gathered} \tag{2}$$

where i represents the class. We also use focal loss and evaluate various values for α and γ. The definition of focal loss is given by Eq. 3:

$$FL(p_t) = -\alpha_t(1-p_t)^\gamma log(p_t) \tag{3}$$

where p_t is given by the following equation:

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise} \end{cases} \tag{4}$$

where $y = 1$ means that the class has been correctly classified, and p is the predicted class probability.

IV. RESULTS

A. CURRICULUM LEARNING

First, we aim to study the applicability of curriculum learning techniques to the fine-grained vehicle classification problem and evaluate its performance. All the experiments have been made using the fine-grained classification subset of CompCars, with 431 classes and 52,083 images. We have chosen CompCars, a widely used and well known dataset, because of its large number of classes and images.

1) Incremental Learning

In Table 3 we compare the performance of a standard trained ResNet50 and InceptionV3 with its counterparts trained using *incremental-learn* (first we train an easier problem -makers- and then retrain it for models). All these models have been trained for 50 epochs using a learning rate of 0.001 and constant policy.

We can observe consistent performance obtaining a slight improvement with the *incremental-learn* technique. This indicates that the *incremental-learn* technique is working and,

TABLE 3. Comparison of the accuracy on validation of the first curriculum learning method (from general to specific) with standard training. The times for the *incremental-learn* method models (marked with *) are roughly double their standard counterparts because we are counting the time required for training the maker and model networks.

Model	Method	top1/5 acc (%)	Train Time
ResNet50	standard	95.28 / 99.26	3h5m
ResNet50	incremental-learn	95.55 / 99.42	6h3m*
InceptionV3	standard	95.02 / 99.10	4h25m
InceptionV3	incremental-learn	95.39 / 99.29	8h53m*

although the training time practically doubles, it can be useful to enhance generalisation. Given that the results obtained for InceptionV3 and ResNet50 are very similar, for the remaining of this section we will only show the results for ResNet50.

These tests alone do not allow us to properly interpret the results. As we have said, we first trained makers and after that models. How has *incremental-learn* method affected the performance? A comparison of ResNet50 models per-class performance can be seen in Fig. 3. In order to visualise the data more clearly, we decided to subtract the original per-class performance, thus obtaining results centred on zero (same performance), above zero (standard model performs better) and below zero (*incremental-learn* performs better). We also applied a color coding with a threshold of 2.5% difference in performance to divide the classes in three groups. The group below -2.5% in green (*incremental-learn*). The group above 2.5% (standard model). And the group in between (similar performance with both models). We can see that most of the values are at 0 or very close with some outlayers going to differences of more than 10%. Analysing the data, both trainings are balanced having practically the same number of improvements and losses in performance so we wondered if these variations could be caused by the number of samples in the classes.

In Fig. 4 we can see the per-class difference in performance between the two models depending on the number of samples in each class. This gives us valuable information. There is a clear tendency to obtain similar results the more samples a class has with the greatest differences concentrated in some of the classes with the least number of samples. We can see a homogeneous distribution between improving and worsening performance so we can say that the variations are related with the number of samples, but, we think that the main reason for this behaviour is the fact that the classes with fewer samples are more exposed to the random variations of each training.

2) Progressive Learning

Continuing with the curriculum learning experiments, in Table 4 we compare the performance of two standard trained ResNet50 (one with constant 0.001 learning rate and the other with step-10 policy and 0.01 as initial learning rate) with a set of ResNet50 models trained using both *progressive* variants. All these models have been trained for 50/80

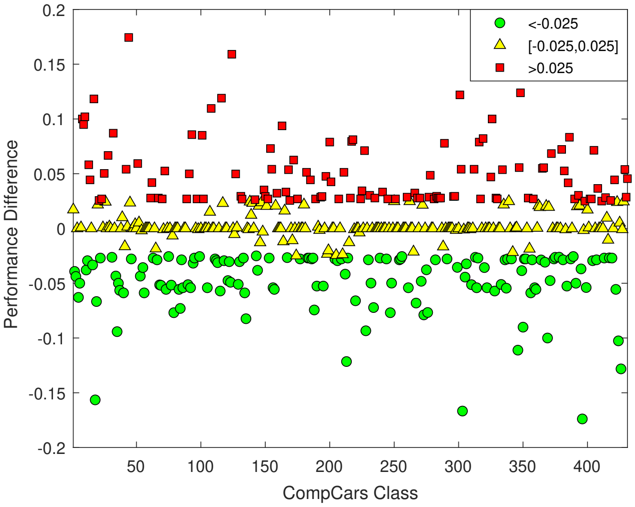

FIGURE 3. Per-class performance differences for the ResNet50 standard and *incremental-learn* models. Difference threshold of 0.025 (2.5%). Differences below -0.025 (green circles) mean better performance for the *incremental-learn* method. Differences above 0.025 (red squares) mean better performance for the standard model. Values in between (yellow triangles) mean similar performance in both models.

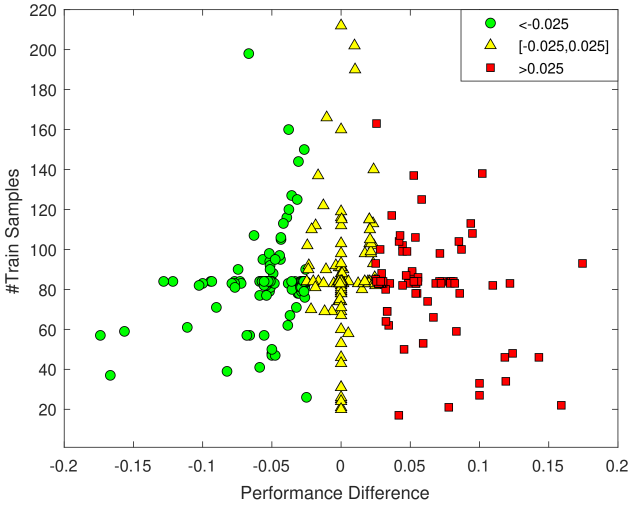

FIGURE 4. Per-class performance differences for the ResNet50 standard and *incremental-learn* models depending on the number of training samples. Difference threshold of 0.025 (2.5%). Differences below -0.025 (green circles) mean better performance for the *incremental-learn* method. Differences above 0.025 (red squares) mean better performance for the standard model. Values in between (yellow triangles) mean similar performance in both models.

epochs (*progressive-10* or *progressive-5*) using learning rates of 0.01/0.001, constant policy and none of theme use the 2-step fine-tuning technique.

If we take a look at the results we can see several things. First, we have consistent results with better performance for all the *progressive-10* models when compared with the *progressive-5* ones. When comparing the different runs of ResNet50 we can see that the 0.01 learning rate seems to work better. The *progressive* ResNet50 models match the performance of the standard one with a slight improvement for the 0.001 learning rate *progressive-10* model. Talking

TABLE 4. Comparison of the accuracy of ResNet50 models trained with progressive learning technique with standard training.

Model	Method	lr	top1/5 acc (%)	Train Time
ResNet50	standard	0.001	95.28 / 99.26	3h5m
ResNet50	standard	step-10 0.01	97.00 / 99.62	3h5m
ResNet50	prog10	0.001	**95.43 / 99.34**	2h44m
ResNet50	prog5	0.001	95.27 / 99.33	4h6m
ResNet50	prog10	0.01	**97.02 / 99.53**	2h44m
ResNet50	prog5	0.01	96.61 / 99.51	4h6m

about times, the *progressive-10* ResNet50 models take less time than the standard ones achieving similar results. With these results, the progressive technique looks like a good option, as it obtains equivalent performance in less time and could be an useful resource to add new classes to an already trained model instead of training it again.

As we have been adding classes from best to worst performance is interesting to analyse the per-class performance. In Fig. 5 we can see a comparison of per-class performance and per-class performance depending on the number of training samples of the ResNet50 *progressive10* trainings and their standard counterparts. Once again, we have applied a threshold of 2.5% difference in performance to divide the classes into three groups. On top, we have the comparison of the models with 0.001 learning rate and on the bottom those with 0.01 learning rate. On the left side we have the per-class performance and on the right side per-class performance by number of training samples. Focusing on per-class performance we can see that the 0.01 learning rate models are more compact (fewer differences), which is consistent with the results (the 0.01 lr models obtain practically the same results while the 0.001 have a greater gap). Focusing on the per-class performance by number of samples we have the same behaviour and the expected pyramidal pattern, with fewer differences the more samples a class has.

Seeing these results, with virtually identical performances using the *progressive* methodology and the standard, we wondered whether gradually increasing the difficulty of the classes is really helping or not. To test this we have trained 2 additional ResNet50 models using 0.01 learning rate, *progressive-10* method, one with random class order and the other with inverse (decreasing difficult) class order.

In Table 5 we can compare the *progressive* models trained with alternative class order with the standard and the *progressive* with increasing difficult ones. As can be seen, all the performances are practically the same which refutes the theory that progressively increasing the difficulty improves performance. These results are somewhat counter-intuitive, as the data structure of the classification problem suggested a potential for improvement. Even so, although no significant performance gain is obtained, it has been shown that the models can be trained progressively, allowing new classes to be added to already trained networks, and achieving equivalent performance with less computational resources, i.e. less time and energy spent on training processes.

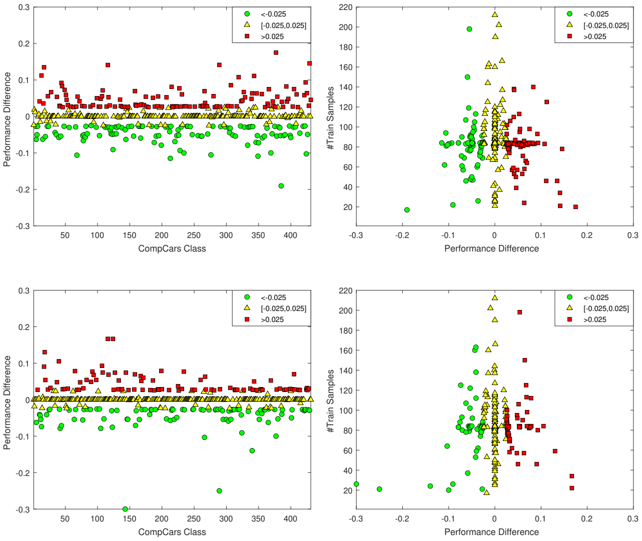

FIGURE 5. Left images are the per-class performance differences of 0.001 and 0.01 learning rate standard ResNet50 and *progressive-10* ResNet50 respectively. Right images are the per-class performance differences for the 0.001 and 0.01 learning rate trains depending on the number of training samples. Difference threshold of 0.025 (2.5%). Differences below -0.025 (green circles) mean better performance for the *progressive* method. Differences above 0.025 (red squares) mean better performance fort the standard model. Values in between (yellow triangles) mean similar performance in both models.

TABLE 5. Comparison of the accuracy of ResNet50 models trained with progressive learning technique and alternative class order (random and inverse) with standard training.

Model	Method	lr	top1/5 acc (%)	T. Time
ResNet50	standard	step-10 0.01	97.00 / 99.62	3h5m
ResNet50	prog10	0.01	97.02 / 99.53	2h44m
ResNet50	prog10-random	0.01	97.03 / 99.51	2h44m
ResNet50	prog10-inv	0.01	97.01 / 99.50	2h44m

B. FINE-GRAINED MODELS

In this section, we analyse the performance of fine-grained classification models comparing them with the baseline results reported by their creators. It could be interesting to compare them with other state-of-the-art methods, but since our aim is not to obtain the best model, we have considered that it does not provide relevant information. We will focus on the results obtained with CompCars, VMMR-db and Frontal-103 and their subsets.

For CompCars we have evaluated 2 subsets. One of makers and other of models, with 73 and 431 classes. For VMMR-db we have evaluated 3 subsets. One of makers and other of models built with the data provided and other called 3,040 built in the same way as the authors built its 3,036 (all the classes with more than 20 images). The number of classes is 43, 472 and 3,040 respectively. For Frontal-103 we have evaluated 3 subsets. One of makers, one of models built with the data provided and one of ultra-fine-grained models. The number of classes is 103, 1,050 and 1,759 respectively.

For clarity, the Table 6 shows the different subsets with the number of classes and the total number of images.

TABLE 6. Information of number of classes and images of each of the subsets.

Subset	# Classes	# Images
CompCars Makers	73	52,083
CompCars Models	431	52,083
VMMR-db Makers	43	246,290
VMMR-db Models	472	246,290
VMMR-db 3,040	3,040	246,290
Frontal-103 Makers	103	65,433
Frontal-103 Models	1,050	65,433
Frontal-103 Ultra	1,759	65,433

All experiments were performed with a 70/30 train/val split. We trained both ResNet50 and InceptionV3 models with step-10 policy and 0.01 learning rate for 50 epochs. As InceptionV3 was the best performing option we will only report its results.

TABLE 7. Comparison of accuracy of the different datasets and subsets with its baseline results.

Model	Subset	top1/5 acc(%)
InceptionV3	CompCars Makers	98.84 / 99.68
InceptionV3	CompCars Models	97.29 / 99.57
CompCars [6]	CompCars Models	91.20 / 98.10
InceptionV3	VMMR-db Makers	97.34 / 99.64
InceptionV3	VMMR-db Models	94.46 / 99.15
InceptionV3	VMMR-db 3,040	42.16 / 91.58
VMMR-db [7]	VMMR-db 3,036	51.76 / 92.90
InceptionV3	Frontal-103 Makers	99.30 / 99.87
InceptionV3	Frontal-103 Models	96.88 / 99.65
InceptionV3	Frontal-103 Ultra	95.62 / 99.48
Frontal-103 [8]	Frontal-103 Ultra	91.28 / -

Table 7 shows the results of the InceptionV3 models for each of the subsets and compares them with the ones reported by their creators. As expected, the best results are obtained in the simplest task, classifying makers, followed by fine-grained models and finally ultra fine-grained models. Analysing the makers results we can see that the best performance is achieved with Frontal-103 as is the easiest one having only images from the front of the vehicles, followed by CompCars and finally VMMR-db as its the most complicated and extensive of the datasets. Focusing on the performance of fine-grained classification, it can be seen that this time the best performance is achieved by CompCars, as it has the least amount of classes, followed by Frontal-103, which, although it has more classes than VMMR-db, is, as we have said, easier having a single view-point. Finally, in the case of ultra fine-grained classification, we can see a big difference between the results obtained by VMMR-db and Frontal-103. While Frontal-103 still achieves a good performance with 95.62% of top1 accuracy, VMMR-db drops to 42.16%. As we have previously said, one of the key problems of VMMR-db dataset is that the labelling contains a class for each year of the same model. Therefore, the actual number of classes is much lower. If we take a look to the top5 accuracy we can see an important leap to 91.58%. In [7] the authors explain this drop in performance by the increased difficulty of going deeper in the hierarchy. However, this statement does not sufficiently hold. As we have seen with Frontal-103, although the ultra classification does indeed have a higher level of difficulty it still has a good performance. This shows that the year-based labelling for models in VMMR-db is not the most appropriate.

C. WEIGHTED LOSSES

As we have said in the introduction, most articles focus on reporting global results, trying to improve accuracy, without analyzing per-class performance. It is of little use to have spectacular accuracy if a non-negligible number of classes have been somewhat ignored. In this section, we analyse the per-class performance of maker and model classification and explore techniques such as weighted loss to improve its performance and generalisation capabilities. We are going to use VMMR-db Makers and VMMR-db Models for the training and the PREVENTION dataset to externally evaluate performance and generalisation capabilities in realistic scenarios.

1) Why raw precision is not enough?

Fig. 6 shows an histogram of the per-class precision of VMMR-db Maker subset. We can see that even though the top1 accuracy is 97.34% we still have one class performing below 10%. If we look at the results of VMMR-db Models in Fig. 7, we can see that this problem is considerably greater and, even though the top1 accuracy is 94.46%, there is a considerable number of classes with poor performance.

To address the performance problem in particular classes, we first checked the relationship of per-class performance to the number of samples of each class and found that the classes with this problem are among those with the fewest samples. Having verified that there is indeed a problem with under-represented classes, we have employed various weighted losses techniques and focal loss to try to mitigate this problem. To evaluate generalisation capability of the different solutions, in addition to the training performance in VMMR-db Makers and Models, we will use the two test sets (Makers and Models) created from the PREVENTION dataset with rear and front view images in real traffic situations. Of the 33 makers present in the Makers test set, 25 are present in VMMR-db with a total of 1,523 images. And from the 87 models present in the Models test set, 50 are present in VMMR-db with a total of 780 images.

As defined in section III-E, we are going to test three different weighting schemes and the focal loss.

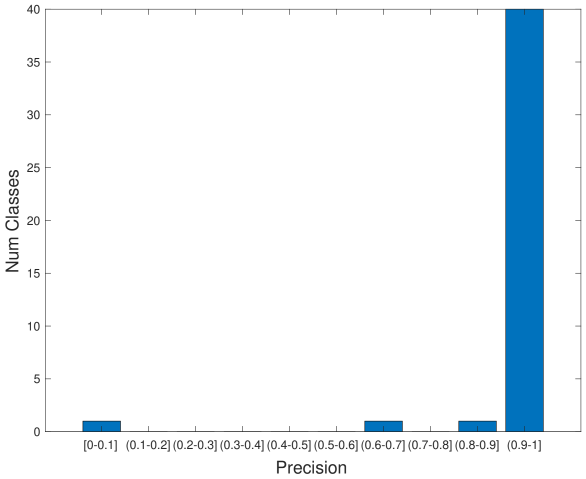

FIGURE 6. Number of classes with a given precision (VALIDATION) for InceptionV3 model trained with VMMR-db Makers subset.

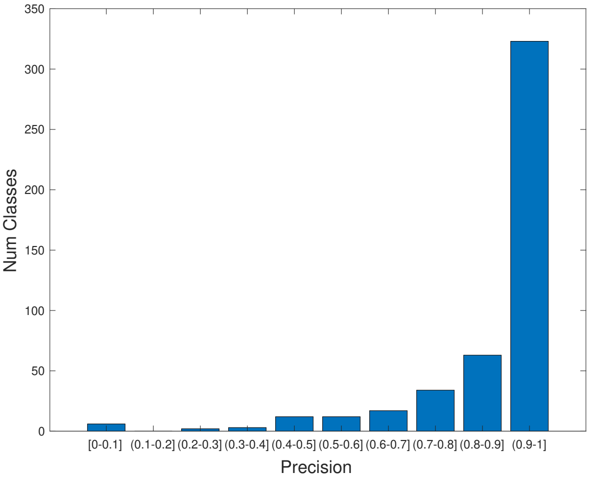

FIGURE 7. Number of classes with a given precision (VALIDATION) for InceptionV3 model trained with VMMR-db Models subset.

2) Weighted losses for maker classification

Table 8 shows the accuracy for the non-weighted, the weighted and the focal loss models trained with VMMR-db Makers and tested on the PREVENTION Makers dataset.

TABLE 8. Comparison of the accuracy of InceptionV3 models trained with VMMR-db Makers with and without weights and test on the PREVENTION Makers dataset. Standard 43 and Log 43 are the normalised version of the weights. The results of focal loss are the ones obtained using $\alpha = 1, \gamma = 2$.

Method	**top1/5 acc (%)**	**test top1 acc(%)** **front / rear / all**
Without weights	**97.34** / 99.64	81.45 / 69.84 / 76.17
Standard weights	97.22 / 99.64	79,16 / 70,85 / 75,38
Standard 43 weights	97.31 / 99.64	**83.37 / 73.02 / 78.66**
Log 43 weights	97.23 / 99.66	81.81 / 69.55 / 76.23
Focal Loss	96.94 / 99.66	82.65 / 70.71 / 77.22

Looking at these results, we can see that, in terms of accuracy in the training phase, the best performance is obtained by the model without weights, but closely followed by the other approaches, with the *standard normalised* weights (standard 43) being the best performing of the weighted models. If we look at the test results, we can see that the normalised weights and focal loss outperform the weightless model with the standard 43 being the best again followed by focal loss.

3) Weighted losses for model classification

Focusing on VMMR-db Models, we can see the accuracy comparison between using and not using weights and focal loss in Table 9. Looking at these results, we have a similar behaviour as with VMMR-db Maker. The best performance in the training phase is again achieved by the weightless model and the weighted models follow closely behind. However, this time the best performing model is the one trained with *focal loss*, even though is the worst performing in validation. In the test results we can see that all the weighted models outperform the weightless one. It is worth noting the large drop in test performance compared to makers. This is most likely due, on the one hand, to the increased difficulty and, on the other hand, to the smaller number of samples in the Models test set, which makes it more biased.

TABLE 9. Comparison of the accuracy of InceptionV3 models trained with VMMR-db Models with and without weights and test on the PREVENTION Models dataset. Standard 472 and Log 472 are the normalised version of the weights. The results of focal loss are the ones obtained using $\alpha = 1, \gamma = 2$.

Method	**top1/5 acc (%)**	**test top1 acc(%)** **front / rear / all**
Without weights	**94.46** / 99.15	51.27 / 57.93 / 54.23
Standard weights	94.30 / 99.16	51.96 / 58.21 / 54.74
Standard 472 weights	94.41 / 99.22	53.12 / 59.37 / 55.90
Log 472 weights	94.40 / 99.22	**53.81** / 58.50 / 55.90
Focal Loss	93.92 / 99.25	53.12 / **59.65 / 56.03**

4) Per-class performance analysis (makers)

But again, we are focusing only on raw performance. It is particularly interesting to look at per-class performance. Fig. 8 shows a comparison of per-class performance for each of the previous models trained with VMMR-db Makers. We can appreciate the effect of the weighted models, with all of them having solved the poor performing class problem of the weightless model. Apart from that, the results are pretty much the same, with standard 43 being the best of the weighted models. Fig. 9 shows a comparison of per-class test performance for each of the previous models trained with VMMR-db Makers and tested on the PREVENTION Makers test set. It can be seen that none of the models have classes below 10% precision and a fairly homogeneous performance, with standard 43 being the best one with a solid performance when compared with the rest of the models, and, even though it has one more class with performance

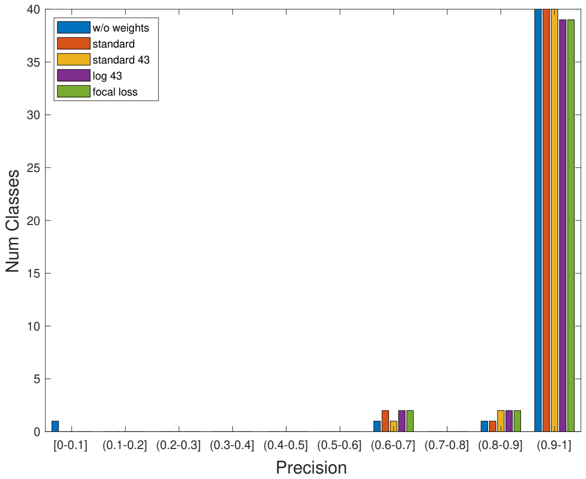

FIGURE 8. Comparison of per-class performance (VALIDATION) in VMMR-db Makers for the different weights sets and focal loss.

below 20%, it also has a noticeable improvement in the range 20-60% when compared with the weightless model. For all of this, it achieves an improvement of almost 2% for front images, 3.18% for rear images and almost 2.5% on all images.

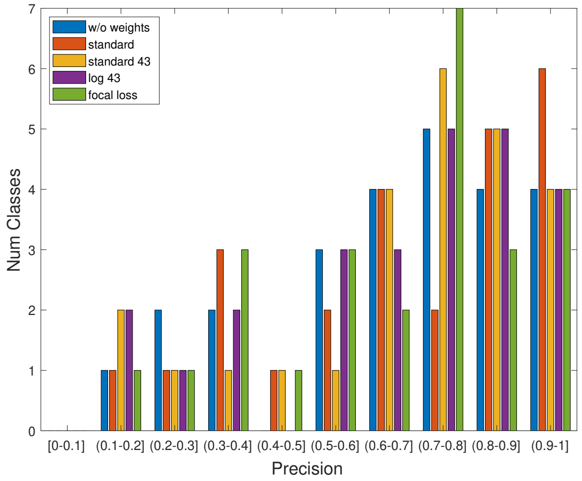

FIGURE 9. Comparison of per-class test performance (TEST) in the PREVENTION Makers dataset for the different weights sets and focal loss trained with VMMR-db Makers.

5) Per-class performance analysis (models)

Continuing with the fine-grained results, Fig. 10 shows a comparison of per-class performance for each of the previous models trained with VMMR-db Models. At first glance we can see that the results are very similar, which makes sense as the performance is almost identical. We can see that the per-class precision distribution is pretty balanced, with all models compensating better performance in one section with worse performance in another.

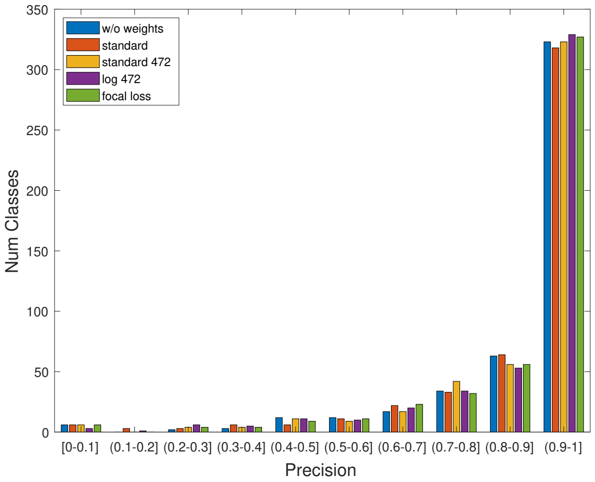

FIGURE 10. Comparison of per-class performance (VALIDATION) in VMMR-db Models for the different weights sets and focal loss.

With these results, it may seem that the use of weights is not justified, as almost identical results have been achieved and there is no clear benefit in terms of the number of poorly performing classes. However, if we look at the test results we can see a considerable improvement, with an increase of more than 2% for front images, almost 2% for rear images and 1.8% on all images.

Fig. 11 shows a comparison of per-class test performance for each of the previous models trained with VMMR-db Models and tested on the PREVENTION Models test set. We can see equivalent or better performance for all the weighted models in terms of number of classes with precision greater than 0.8. In the range of 0.2 to 0.8 the results are pretty balanced, with the weightless model having more classes above 0.7 but the *focal loss* one less in 0.2 to 0.4.

Regarding the poor performing classes, the number of classes below 0.1 is worrying but is practically the same regardless of the model. It is important to remember that the number of images of the Models test set is half that of the Makers test set, making the results more susceptible to variability. However, the results are promising, with a clear improvement in overall test performance, and results that point to an improvement in the generalisation ability of weighted models.

With these results the use of weights is justified, at least in part. The weighted models achieve comparable results to the weightless ones both for makers and models while improving test performance. The claim that weighted models help to reduce the amount of poor performing classes is eclipsed by the worrying amount of them when testing for models. However, results point to an improvement in the generalisation capabilities of the models, as test performance improves by 2.49% and 1.8%. As previously stated, we believe that models test results has a lot to do with the test dataset. It is necessary to conduct further experimentation and build a more extensive and adequate test dataset to properly evaluate

TABLE 10. Fusion sets number of classes, images and distribution of the images between the three source datasets.

Dataset	# Classes	# Images	# CompCars (%)	# VMMR-db (%)	# Frontal-103 (%)
Fusion-Makers	27	265,833	28,960 (10.89%)	198,644 (74.73%)	38,229 (14.38%)
Fusion-Models	75	101,335	13,211 (13.04%)	72,142 (71.19%)	15,982 (15.77%)

TABLE 11. Cross test performance comparison of Fusion-Makers dataset and its source makers datasets. The Test Set column corresponds to the results testing with the PREVENTION Makers subset. From the 27 classes there are 23 common with the PREVENTION Makers test set.

	Test top1/3 acc (%)				**Test Set top1 acc (%)**
Train	Fusion-Makers	CompCars-Makers	VMMR-db-Makers	Frontal-103-Makers	(front/rear/all)
InceptionV3 Fusion-Makers	**98.47 / 99.60**	**99.33 / 99.77**	**98.09 / 99.59**	**99.81 / 99.95**	**89.81 / 81.82 / 86.19**
InceptionV3 CompCars-Makers	47.39 / 58.43	99.17 / 99.77	30.92 / 44.88	93.73 / 97.52	75.26 / 65.07 / 70.64
InceptionV3 VMMR-db-Makers	94.96 / 97.75	79.70 / 88.92	98.04 / 99.47	90.49 / 95.54	82.80 / 73.52 / 78.60
InceptionV3 Frontal-103-Makers	33.32 / 46.18	41.34 / 58.13	19.38 / 34.11	99.78 / 99.92	78.44 / 27.43 / 55.31

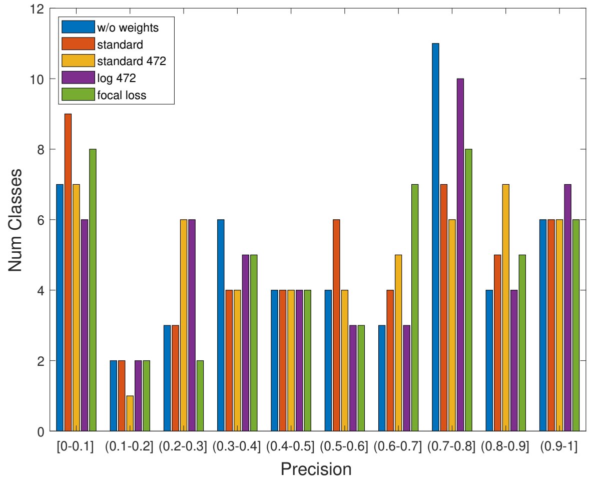

FIGURE 11. Comparison of per-class test performance (TEST) in the PREVENTION Models dataset for the different weights sets and focal loss trained with VMMR-db Models.

fine-grained performance.

D. COMPLEXITY AND GENERALISATION CAPABILITIES

To consider a dataset as a quality dataset, it must capture the real world as reliably as possible, with as few deviations and bias as possible. Therefore, a good dataset will result in models with better generalisation capabilities. As previously mentioned, most datasets are either designed to solve a specific problem, so they are biased, or they are intended for general use. A general-purpose dataset, should capture the world in a reliable way, but when it comes down to it, most of them tend to be conditioned.

Thus, we have decided to build a cross-dataset composed of the common classes between CompCars, VMMR-db and Frontal-103. In this way, we will be able to evaluate the complexity and generalisation capabilities of the models trained with each dataset by performing cross tests. Additionally, we will also test the models with the test set extracted from the PREVENTION dataset.

We have built two sets, one of makers and other of models. The Fusion-Makers set has 27 different manufacturers and a total of 265,833 images, 28,960 from CompCars, 198,644 from VMMR-db and 38,229 from Frontal-103. The Fusion-Models set has 75 different vehicle models and a total of 101,335 images, 13,211 from CompCars, 72,142 from VMMR-db and 15,982 from Frontal-103. It may seem curious, or even a mistake, that the number of images in the set of models is much lower than in makers. The reason is that the makers set is much less strict, allowing models from the same manufacturer that are not present in the three source datasets to be included. In contrast, in the case of models, the requirement that a particular model has to be present in all three datasets brings the total number of classes and images down considerably. Additionally, we have had to group some source classes into a single target class, e.g. different equipment levels that were considered as different classes like BMW 320 vs BMW 325 as BMW 3 Series. As mentioned in the introduction, the correspondence between classes will be publicly available[2]. A summary of the different Fusion sets can be seen in Table 10.

To perform these experiments we used InceptionV3 architecture. First, we are going to analyse makers performance. Table 11 shows the results of the cross-tests with the different makers sets. We can see that the best performing model for all the sets is the one trained with the full Fusion dataset followed by the model trained on the tested set. This demonstrates that the joint use of the datasets brings more variety, resulting in better generalisation capabilities and mitigating the impact of dataset bias. The only model, other than the Fusion one, that is capable of obtaining reasonable results on the other datasets is the one trained with VMMR-db. It is important to notice that VMMR-db almost represents 75% of the Fusion dataset, which could partly justify the good results when testing with Fusion, but not its good performance in the rest of the subsets. When we look at

[2] https://github.com/ninte/fusion-cross-dataset

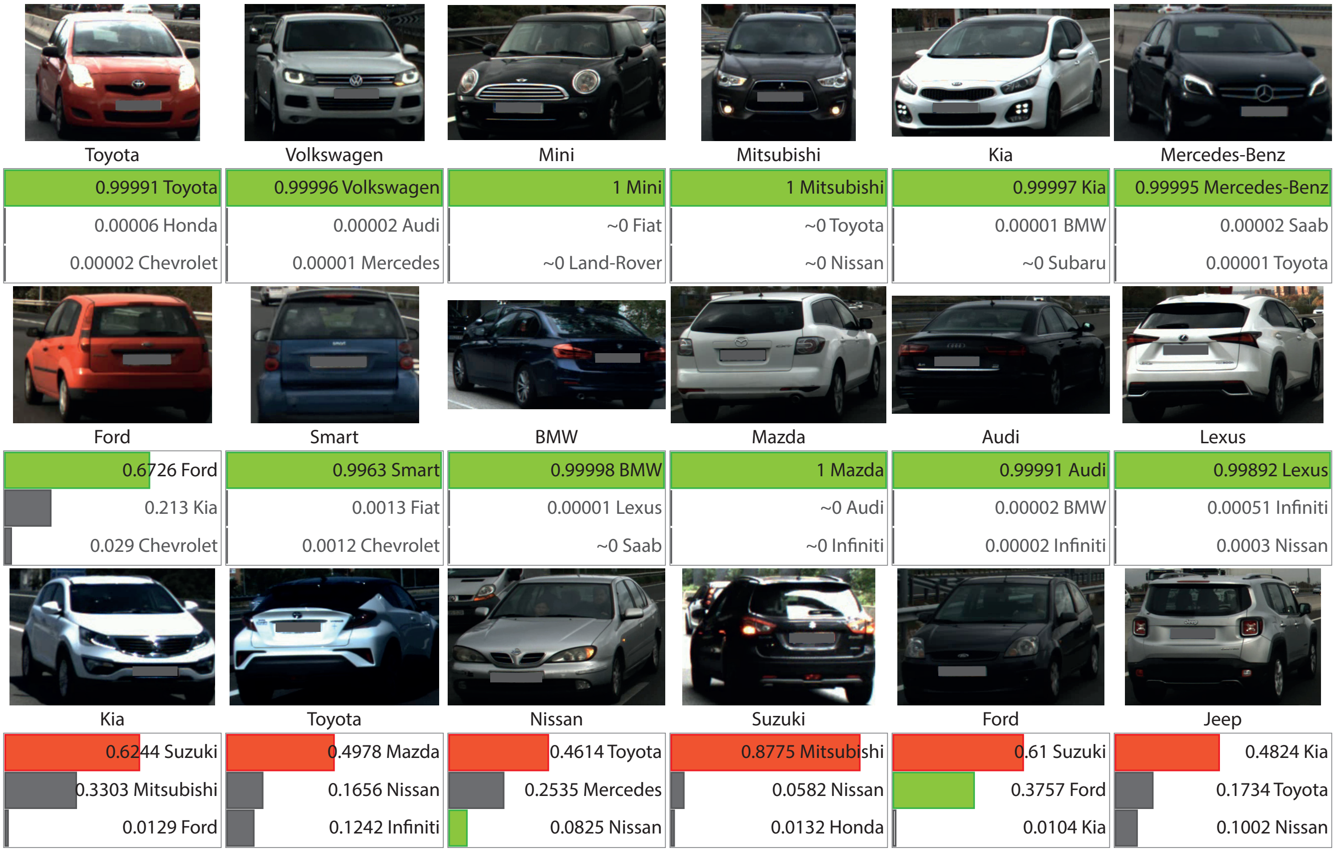

FIGURE 12. Top3 predicted classes of InceptionV3 Fusion-Makers model for sample images from the PREVENTION Makers test set. The first row shows correctly classified front view images. The middle row shows correctly classified rear view images. The bottom row shows misclassified images from both views.

TABLE 12. Cross test performance comparison of Fusion-Models dataset and its source models datasets. The Test Set column corresponds to the results testing with the PREVENTION Models subset. From the 75 classes there are 34 present in the PREVENTION Models test set.

	Test top1/3 acc (%)				**Test Set top1 acc (%)**
Train	Fusion-Models	CompCars-Models	VMMR-db-Models	Frontal-103-Models	(front/rear/all)
InceptionV3 Fusion-Models	**98.51 / 99.58**	**99.32 / 99.75**	**98.11 / 99.47**	**99.62 / 99.64**	**80.23 / 75.67 / 78.47**
InceptionV3 CompCars-Models	43.85 / 56.20	98.41 / 99.57	25.17 / 39.98	83.25 / 93.69	53.35 / 49.43 / 51.56
InceptionV3 VMMR-db-Models	86.28 / 93.19	63.30 / 80.15	97.90 / 99.32	52.67 / 76.24	62.62 / 68.44 / 65.28
InceptionV3 Frontal-103-Models	25.76 / 31.69	34.93 / 44.04	7.90 / 14.43	99.26 / 99.81	53.99 / 4.94 / 31.60

the other two, both CompCars and Frontal-103 obtain poor performance when tested with Fusion. CompCars seems to work well when tested with Frontal-103, outperforming the VMMR-db model, which tells us that they are very similar. Probably, Frontal-103 could get good results in CompCars as well, but it is strongly conditioned by having only frontal images, hence its poor performance. If we take a look to the PREVENTION test results, we can see the same behaviour, with Fusion being the best with 86.19% accuracy for all test images followed by VMMR-db with 78.60%, CompCars with 70.64% and Frontal-103 with 55.31%.

Fig. 12 shows some examples of top3 predicted classes of InceptionV3 Fusion-Makers model in the PREVENTION Makers test set. The first row shows correctly predicted front view images. The middle row shows correctly predicted rear view images. The bottom row shows misclassified images from both views. We can see that the model has practically total confidence in the correctly predicted makers (the mean confidence for the correct predictions is 97.78%). This is not the case for the misclassified ones, which have confidences much lower with the exception of the Suzuki predicted as Mitsubishi (the mean confidence for the wrong predictions is 68.05%).

Table 12 shows the results of the cross-tests with the Fusion-Models dataset. Once again, the best performing model is the one trained with Fusion. We have the same differences for the rest of the tests but this time with performances much lower than when using makers. This may be due to the increase in the number of classes making the problem more complex. Taking a look to the PREVENTION

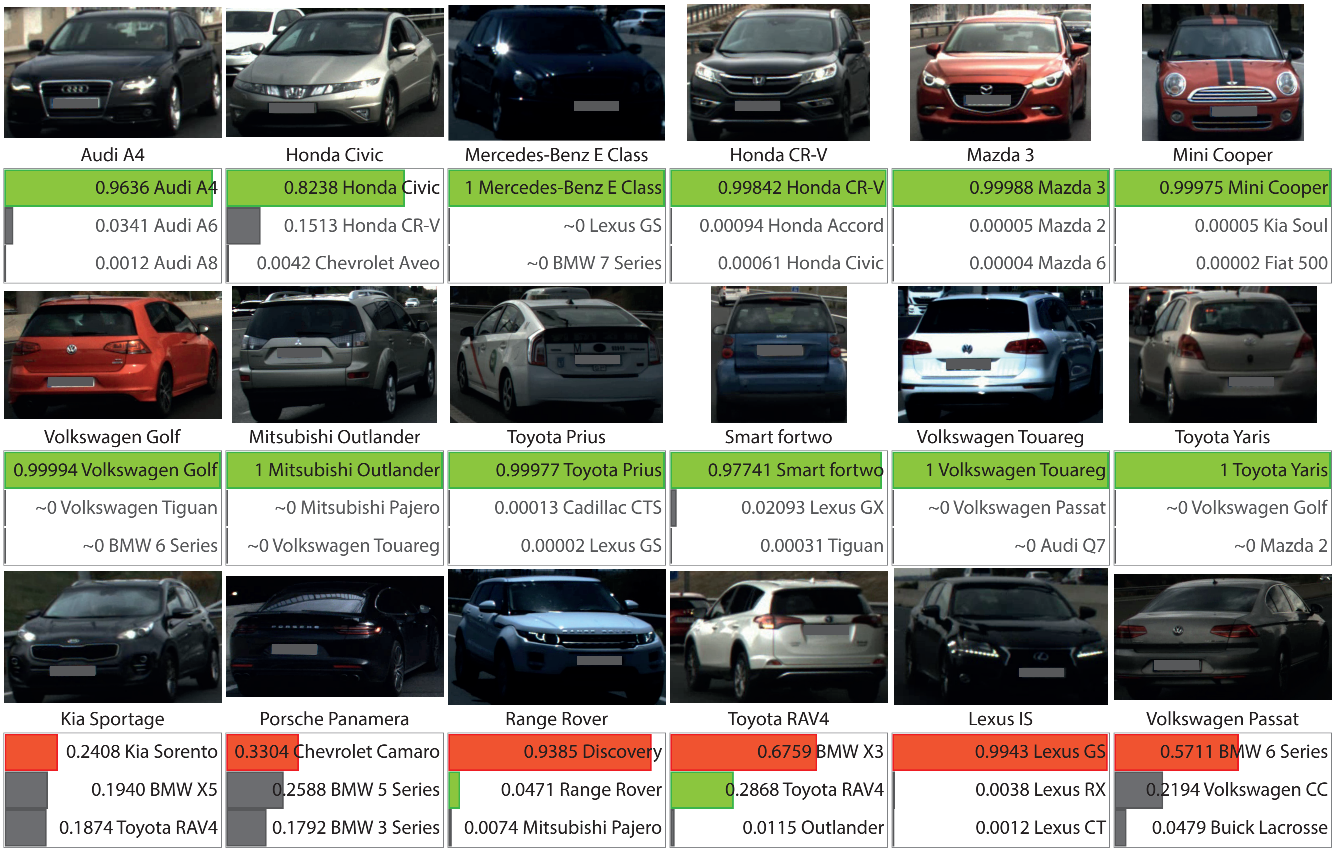

FIGURE 13. Top3 predicted classes of InceptionV3 Fusion-Models model for sample images from the PREVENTION Models test set. The first row shows correctly classified front view images. The middle row shows correctly classified rear view images. The bottom row shows misclassified images from both views.

Models test results, we have the same order in performance (Fusion, VMMR-db, CompCars and Frontal) with 78.47% accuracy for the Fusion model. As expected, the performances are lower than with makers, as it is a more complex problem. However, it should be noted that in this case the Fusion model has a larger performance gap compared to the other models, which supports the importance of having a good dataset that allows better generalisation. Lastly, it is worth mentioning the very poor rear performance of Frontal-103 model (4.94%).

Fig. 13 shows some examples of top3 predicted classes of InceptionV3 Fusion-Models model in the PREVENTION Models test set. The first row shows correctly predicted front view images. The middle row shows correctly predicted rear view images. The bottom row shows misclassified images from both views. We can see that the model has practically total confidence in the correctly predicted models (the mean confidence for the correct predictions is 95.68%). This is not the case for the misclassified ones, which have confidences much lower (the mean confidence for the wrong predictions is 69.91%). Compared to the makers results, the average confidence has gone down for correct predictions (-2.1%) and up a for incorrect ones (+1.86%). This is perfectly normal as it is a more complex problem. In any case, the differences are minimal.

With these results it is clear that of the three datasets, the most complex and the one with the greatest generalisation capabilities is VMMR-db. CompCars is the next, with a significant step down in cross-performance, finally followed by Frontal-103, which is the simplest of all, and is strongly conditioned by having exclusively frontal images. It is also clear that the joint use of the three datasets improves the generalisation capabilities, obtaining reasonably good results in the external test performed with the PREVENTION test set both for makers and models.

The results obtained by making use of the existing datasets suggest that the fine-grained classification can be addressed. However, in cross-testing it is clear that with the exception of VMMR-db, the rest of the datasets are highly biased, and in the case of VMMR-db, although it performs better, it is also biased. It is only when performing these cross-tests and with an external test set that we realise that the problem is not completely solved, and not only for a complex problem such as fine-grained classification, but in a simpler problem such as maker classification. It is necessary to create a sufficiently large and varied dataset, with images of multiple origins, qualities and viewpoints, to be able to tackle the classification problem satisfactorily.

V. CONCLUSIONS AND FUTURE WORK

This paper presents an empirical evaluation of different training methods and approaches for fine-grained vehicle classification as well as an analysis and comparison of the most relevant datasets.

We have analysed the strengths and shortcomings of datasets like CompCars [6], VMMR-db [7] and Frontal-103 [8] and used them in a series of experiments.

In the first place, we have explored different curriculum learning techniques such as *incremental-learn* (training first an easier problem (makers) and after that retrain for a harder one (models)) or *progressive-learn* (start with the easiest best performing classes and gradually add the hardest worst performing) with CompCars dataset. The results show an slight improvement in overall performance for *incremental-learn* with similar gain/losses in per-class performances and a clear relation between per-class performance and number of samples. For *progressive-learn* we have a very similar behaviour, with virtually the same performance and the same per-class differences. The *progressive-learn* results made us consider whether the technique was working as expected, so we performed additional tests, one with decreasing difficulty and other with random order, obtaining identical results. With these results, curriculum learning techniques show a lack of improvement in performance, making it difficult to justify their use as a mechanism for improving learning. However, *progressive-learn* has proven useful as a tool for adding classes to already trained models without having to train from scratch again.

After this, we evaluated the results obtained with different subsets of CompCars (makers and models), VMMR-db (makers, models and 3,040) and Frontal-103 (makers, models and ultra-fine-grained). As expected, the best results are obtained in the easiest task (makers) with Frontal-103 in the first place (as it is the easiest only having frontal images), followed by CompCars and finally VMMR-db (as is the most difficult/extensive of the datasets). For the fine-grained problem (models) the best performance is achieved by CompCars (less classes), followed by Frontal-103 which, although it has more classes, is easier, and finally VMMR-db. Finally, the ultra-fine-grained problem (models and generation) showed a huge difference between Frontal-103 and VMMR-db, with the first one still having a stunning performance and the second one falling bellow 45% top1 accuracy while top5 is still above 90%. This shows the poor class construction of VMMR-db and confirms that even though ultra-fine-grained classification is more challenging, it can still be tackled if the dataset is properly constructed.

Continuing with the experiments we have evaluated the impact of using weighted losses. To do so we have used various weights and focal loss showing that the best results are obtained with the *normalised standard* weights, with practically identical results to those obtained without weights, but with a significant improvement when testing on a new database. Our aim in this part of the article was to analyse the results beyond the raw performance. For this purpose, we have analysed per-class performance showing a clear improvement over the weightless model when working with makers. In the case of models the improvement was not so evident, with more classes over 80% accuracy and similar results in the range 20-80% but no conclusive results for the poor performing ones. While the use of weights have proven to improve generalisation capabilities, we cannot claim the same for reducing the number of poor performing classes. Further experiments and a more extensive adequate test set are needed to properly evaluate fine-grained performance.

Finally, we wanted to analyse the complexity and generalisation capabilities of the existing datasets. To evaluate these characteristics we have built a cross-dataset (Fusion) composed of the common classes between CompCars, VMMR-db and Frontal-103 and performed a series of cross tests. The results show that the best performing model is the one trained with Fusion, both in makers and models, outperforming all the other models in the cross-tests. Regarding the PREVENTION external test set, the Fusion models achieve pretty good results showing really good generalisation capabilities both for makers and models. From the three datasets, VMMR-db is the most complex, with CompCars and Frontal-103 being very similar but Frontal-103 heavily penalised for having exclusively frontal images. These results show that when using the existing datasets by their own, one can think that the fine-grained classification problem is solved. However, cross-testing shows the shortages of the existing datasets, showing a different reality. The problem does not seem to be solved, not only for a complex task like fine-grained classification, but for an easier one like maker classification. It is necessary to create a sufficiently large and varied dataset, with images of multiple origins, qualities and viewpoints, to be able to tackle the classification and fine-grained classification problem satisfactorily.

As future work, we plan to create an extensive dataset, with images of diverse nature, makes and models from different geographical regions, different resolutions, image qualities and viewpoints, with an adequate class hierarchy, enabling the development of more general and unbiased systems capable of performing fine-grained vehicle recognition in multiple, realistic environments.

REFERENCES

[1] H. Corrales, A. Hernández, R. Izquierdo, N. Hernández, I. Parra, and D. Fernández-Llorca, “Simple baseline for vehicle pose estimation: Experimental validation,” *IEEE Access*, vol. 8, pp. 132 539–132 550, 2020.

[2] H. Liu, Y. Tian, Y. Wang, L. Pang, and T. Huang, “Deep relative distance learning: Tell the difference between similar vehicles,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2167–2175.

[3] J. Sochor, A. Herout, and J. Havel, “Boxcars: 3d boxes as cnn input for improved fine-grained vehicle recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3006–3015.

[4] X. Liu, W. Liu, H. Ma, and H. Fu, “Large-scale vehicle re-identification in urban surveillance videos,” in *2016 IEEE International Conference on Multimedia and Expo (ICME)*, 2016, pp. 1–6.

[5] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, “3d object representations for fine-grained categorization,” in *Proceedings of the IEEE international conference on computer vision workshops*, 2013, pp. 554–561.

[6] L. Yang, P. Luo, C. Change Loy, and X. Tang, "A large-scale car dataset for fine-grained categorization and verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3973–3981.

[7] F. Tafazzoli, H. Frigui, and K. Nishiyama, "A large and diverse dataset for improved vehicle make and model recognition," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 874–881.

[8] L. Lu, P. Wang, and H. Huang, "A large-scale frontal vehicle image dataset for fine-grained vehicle categorization," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2020.

[9] H. Corrales, D. F. Llorca, I. Parra, S. Vigre, A. Quintanar, J. Lorenzo, and N. Hernández, "Cnns for fine-grained car model classification," *Computer Aided Systems Theory – EUROCAST 2019. Lecture Notes in Computer Science*, vol. 12014, pp. 104–112, 2020.

[10] R. Izquierdo, A. Quintanar, I. Parra, D. Fernández-Llorca, and M. A. Sotelo, "The PREVENTION dataset: a novel benchmark for PREdiction of VEhicles iNTentIONs," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Oct 2019, pp. 3114–3121.

[11] J. Sochor, J. Spanhel, and A. Herout, "Boxcars: Improving fine-grained recognition of vehicles using 3-d bounding boxes in traffic surveillance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, pp. 97–108, 2019.

[12] H.-Z. Gu and S.-Y. Lee, "Car model recognition by utilizing symmetric property to overcome severe pose variation," *Machine vision and applications*, vol. 24, no. 2, pp. 255–274, 2013.

[13] D. F. Llorca, D. Colás, I. G. Daza, I. Parra, and M. Sotelo, "Vehicle model recognition using geometry and appearance of car emblems from rear view images," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2014, pp. 3094–3099.

[14] D. Santos and P. L. Correia, "Car recognition based on back lights and rear view features," in *2009 10th Workshop on Image Analysis for Multimedia Interactive Services*. IEEE, 2009, pp. 137–140.

[15] D. F. Llorca, R. Arroyo, and M. A. Sotelo, "Vehicle logo recognition in traffic images using hog features and svm," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 2013, pp. 2229–2234.

[16] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear cnn models for fine-grained visual recognition," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1449–1457.

[17] J. Krause, H. Jin, J. Yang, and L. Fei-Fei, "Fine-grained recognition without part annotations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5546–5555.

[18] J. Fang, Y. Zhou, Y. Yu, and S. Du, "Fine-grained vehicle model recognition using a coarse-to-fine convolutional neural network architecture," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1782–1792, 2017.

[19] J. Fu, H. Zheng, and T. Mei, "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4438–4446.

[20] B. Zhao, X. Wu, J. Feng, Q. Peng, and S. Yan, "Diversified visual attention networks for fine-grained object classification," *IEEE Transactions on Multimedia*, vol. 19, no. 6, pp. 1245–1256, 2017.

[21] Y. Tian, W. Zhang, Q. Zhang, G. Lu, and X. Wu, "Selective multi-convolutional region feature extraction based iterative discrimination cnn for fine-grained vehicle model recognition," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 3279–3284.

[22] S. Elkerdawy, N. Ray, and H. Zhang, "Fine-grained vehicle classification with unsupervised parts co-occurrence learning," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018.

[23] R. Du, D. Chang, A. K. Bhunia, J. Xie, Z. Ma, Y.-Z. Song, and J. Guo, "Fine-grained visual classification via progressive multi-granularity training of jigsaw patches," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 153–168.

[24] Y. Ding, Z. Ma, S. Wen, J. Xie, D. Chang, Z. Si, M. Wu, and H. Ling, "Ap-cnn: Weakly supervised attention pyramid convolutional neural network for fine-grained visual classification," *IEEE Transactions on Image Processing*, vol. 30, pp. 2826–2836, 2021.

[25] K. Ramnath, S. N. Sinha, R. Szeliski, and E. Hsiao, "Car make and model recognition using 3d curve alignment," in *IEEE winter conference on applications of computer vision*. IEEE, 2014, pp. 285–292.

[26] Y.-L. Lin, V. I. Morariu, W. Hsu, and L. S. Davis, "Jointly optimizing 3d model fitting and fine-grained classification," in *European conference on computer vision*. Springer, 2014, pp. 466–480.

[27] A. Anderson, K. Shaffer, A. Yankov, C. D. Corley, and N. O. Hodas, "Beyond fine tuning: A modular approach to learning on small data," *arXiv preprint arXiv:1611.01714*, 2016.

[28] Q. Hu, H. Wang, T. Li, and C. Shen, "Deep cnns with spatially weighted pooling for fine-grained car recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 3147–3156, 2017.

[29] D. Liu and Y. Wang, "Monza: image classification of vehicle make and model using convolutional neural networks and transfer learning," 2017.

[30] K. Valev, A. Schumann, L. Sommer, and J. Beyerer, "A systematic evaluation of recent deep learning architectures for fine-grained vehicle classification," in *Pattern Recognition and Tracking XXIX*, vol. 10649. International Society for Optics and Photonics, 2018, p. 1064902.

[31] X. Li, L. Yu, D. Chang, Z. Ma, and J. Cao, "Dual cross-entropy loss for small-sample fine-grained vehicle classification," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4204–4212, 2019.

[32] M. Buzzelli and L. Segantin, "Revisiting the compcars dataset for hierarchical car classification: New annotations, experiments, and results," *Sensors*, vol. 21, no. 2, p. 596, 2021.

[33] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[34] S. Hu, Y. Liang, L. Ma, and Y. He, "Msmote: Improving classification performance when training data is imbalanced," in *2009 second international workshop on computer science and engineering*, vol. 2. IEEE, 2009, pp. 13–17.

[35] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. IEEE, 2008, pp. 1322–1328.

[36] B. Tang and H. He, "Kerneladasyn: Kernel based adaptive synthetic data generation for imbalanced learning," in *2015 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2015, pp. 664–671.

[37] S. Barua, M. M. Islam, and K. Murase, "A novel synthetic minority oversampling technique for imbalanced data set learning," in *International Conference on Neural Information Processing*. Springer, 2011, pp. 735–744.

[38] C. Drummond, R. C. Holte *et al.*, "C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling," in *Workshop on learning from imbalanced datasets II*, vol. 11. Citeseer, 2003, pp. 1–8.

[39] S.-J. Yen and Y.-S. Lee, "Cluster-based under-sampling approaches for imbalanced data distributions," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5718–5727, 2009.

[40] M. Galar, A. Fernández, E. Barrenechea, and F. Herrera, "Eusboost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling," *Pattern recognition*, vol. 46, no. 12, pp. 3460–3471, 2013.

[41] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5375–5384.

[42] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Deep imbalanced learning for face recognition and attribute prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 11, pp. 2781–2794, 2020.

[43] Y.-X. Wang, D. Ramanan, and M. Hebert, "Learning to model the tail," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 7032–7042.

[44] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[45] K. M. Ting, "A comparative study of cost-sensitive boosting algorithms," in *In Proceedings of the 17th International Conference on Machine Learning*. Citeseer, 2000.

[46] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 8, pp. 3573–3587, 2017.

[47] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4334–4343.

[48] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9268–9277.

[49] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012, pp. 1097–1105.

[50] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2015.

[51] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[52] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[53] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[54] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[55] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.

[56] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, “Benchmark analysis of representative deep neural network architectures,” *IEEE Access*, vol. 6, pp. 64 270–64 277, 2018.

[57] J. Muñoz Bulnes, C. Fernández, I. Parra, D. Fernández-Llorca, and M. A. Sotelo, “Deep fully convolutional networks with random data augmentation for enhanced generalization in road detection,” in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 366–371.

[58] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[59] J. L. Elman, “Learning and development in neural networks: The importance of starting small,” *Cognition*, vol. 48, no. 1, pp. 71–99, 1993.

[60] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.

HÉCTOR CORRALES SÁNCHEZ received the B.S. degree in telecommunications engineering (telematics specialty) from University of Alcalá (UAH), Alcalá de Henares, Spain, in 2016 and the M.S. degree in telecommunications engineering (intelligent transportation systems specialty) from UAH in 2018. He is currently pursuing the Ph.D. degree in information and communications technologies at UAH. His current research interests include machine learning, computer vision, deep learning and autonomous driving.

NOELIA HERNÁNDEZ PARRA received the M.S. and Ph.D. in Advanced Electronics Systems (Intelligent Systems) from the University of Alcalá (UAH) in 2009 and 2014, respectively. Her thesis presented a new approach for estimating the global position of a mobile device in indoor environments by using WiFi devices which received the Best PhD Award by the UAH in 2014. She currently works as an Associate Professor at the Computer Engineering Department of the UAH. Her research interest mainly focuses on indoor and outdoor localization, artificial intelligence and Intelligent Transportation Systems.

IGNACIO PARRA ALONSO received the M.S. and Ph.D. degrees in telecommunications engineering from the University of Alcalá (UAH) in 2005 and 2010, respectively. He currently works an Associate Professor at the Computer Engineering Department, UAH. His research interests include intelligent transportation systems and computer vision. He received the Master Thesis Award in eSafety from the ADA Lectureship at the Technical University of Madrid, Spain, in 2006.

EDUARDO NEBOT (Fellow Member, IEEE) received the B.S. degree in electrical engineering from the Universidad Nacional del Sur, Argentina, M.S. and Ph.D. degrees from Colorado State University, Colorado, USA. He is currently a Professor at the University of Sydney, Sydney, Australia, and the Director of the Australian Centre for Field Robotics. His main research interests are in field robotics automation and intelligent transport systems. The major impact of his fundamental research is in autonomous systems, navigation, and safety.

DAVID FERNÁNDEZ-LLORCA (Senior Member, IEEE) received the Ph.D degree in telecommunication engineering from the University of Alcalá (UAH) in 2008. He is currently Scientific Officer at the European Commission - Joint Research Center. He is also Full Professor with UAH. He has authored over 130 publications and more than 10 patents. He received the IEEE ITSS Young Research Award in 2018 and the IEEE ITSS Outstanding Application Award in 2013. He is currently Editor-in-Chief of the IET Intelligent Transport Systems. His current research interest includes trustworthy AI for transportation, predictive perception for autonomous vehicles, human-vehicle interaction, end-user oriented autonomous vehicles and assistive intelligent transportation systems.