CNNs for Fine-Grained Car Model Classification

H. Corrales, D. F. Llorca(✉), I. Parra, S. Vigre, A. Quintanar, J. Lorenzo, and N. Hernández

Computer Engineering Department, Universidad de Alcalá, Alcalá de Henares, Spain
{hector.corrales,david.fernandezl,ignacio.parra,susana.vigre, alvaro.quintanar,javier.lorenzod,noelia.hernandez}@uah.es

Abstract. This paper describes an end-to-end training methodology for CNN-based fine-grained vehicle model classification. The method relies exclusively on images, without using complicated architectures. No extra annotations, pose normalization or part localization are needed. Different full CNN-based models are trained and validated using CompCars [31] dataset, for a total of 431 different car models. We obtained a top-1 validation accuracy of 97.62% which substantially outperforms previous works.

Keywords: Vehicle model · Fine-grained classification · CNNs

1 Introduction

Fine-grained classification of cars, also known as model classification, has a great interest for a considerable number of applications such as traffic regulation, surveillance, tolls automation or parking monitoring. This task can be extremely challenging due to big similarities and subtle differences between related car models, differences that can be easily lost with changes in location, viewpoint or pose. Most of fine-grained classification methods make use of techniques such as pose normalization, part localization [8] and modeling [18] or additional annotations to accomplish this task. As a result complex models are obtained and a large amount of time is spent labelling datasets.

In this paper we propose an end-to-end training methodology for CNN-based fine-grained vehicle model classification (see Fig. 1). Our method relies exclusively on images, without using complicated architectures, extra annotations, pose normalization or part localization. Different full CNN-based models have been trained and validated using CompCars [31] dataset and with our methodology we substantially outperform previous works obtaining a top-1 validation accuracy of 97.62% for a total of 431 different car models.

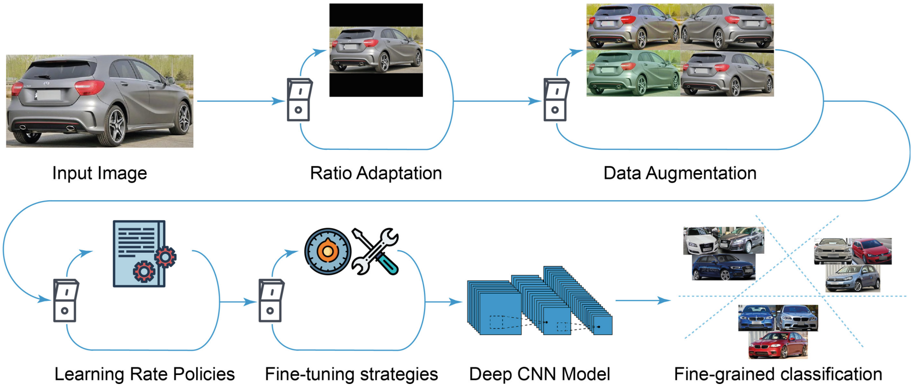


Fig. 1. General overview of the proposed methodology.

2 Related Work

Nowadays, there is a large amount of datasets of fine-grained categories, among which we can find birds [2,27,29], flowers [1,21], dogs [10,16], leaves [14], aircrafts [20,26] and cars [12,31]. Many approaches have been used in order to improve fine-grained classification tasks, like 3D object representations [12], pose normalization [3] or part localization [11,32].

Prior to the popularization of CNNs, classification tasks laid on hand-crafted features such as HOG [5], SIFT [19] or more recent visual word features like [4,28,30] used together with classifiers like SVM. Thus, in December 2013, Krause et al. [12] proposed a method to extract 3D model based features. Jointly, they presented the ultra-fine-grained BMW10 dataset and the cars196 dataset. In [3], Branson et al. proposed an architecture to normalize the pose of birds and extract features using CNNs that will be fed to a SVM classifier. Following this line, Zhang et al. [32] presented a method to semantically localize key parts of objects and extract the features from them. Krause et al. [11] also proposed a method to align the images through segmentation to extract the pose without part annotations, making the training process easier.

In [18], Llorca et al. presented a vehicle model recognition approach by modeling the geometry and appearance of car emblems from rear view images using a linear SVM classifier with HOG features. Classification is performed within the set of models of each car manufacturer, which is previously recognized by means of logo classification [17]. Lin et al. [15], instead of manually defining the parts of the objects from which the features will be extracted, used bilinear networks to automatically extract the features with two twin CNNs and multiplex its outputs to feed them to a SVM. In [8], Fang et al. developed a coarse-to-fine method in which they automatically detect discriminative regions and extract features to feed them to a one-versus-all SVM.

Since the appearance of AlexNet [13] in 2012 the use of CNNs has growth considerably. The appearance of other architectures like VGG [22], GoogLeNet/

Inception [25] or ResNet [9] as evolution confirms that CNNs have come to stay. For example, in [31], Yang et al. presented CompCars, a dataset for fine-grained car classification and verification. This is the largest car dataset to date, with a total of $208,826$ images extracted from two scenarios, web-nature and surveillance-nature, from which, $136,727$ images are of entire cars from the web-nature scenario and $44,481$ from the surveillance one. They also made various experiments, finding out that the best results are achieved when the model is fine-tuned using images from all viewpoints, and compared the performance of different deep models.

In [23] and [24] Sochor et al. proposed a system for vehicle recognition on traffic surveillance. This method consisted of using additional data like 3D bounding box of vehicles, with which the vehicles are "unpacked" to obtain an aligned representation of them. Dehghan et al. [6] described the details of Sighthound's vehicle make, model and color recognition system. As this is a private commercial solution they didn't showed the full system, but they tested it in multiple datasets like CompCars, which can be used for performance comparison purposes.

3 System Description and Results

As we have previously introduced we are going to use CompCars dataset. Specifically a subset of 431 different car models with a total of 52 083 images, 36 456 for training and 15 627 for validation.

In order to carry out the different experiments and compare their results, a basic architecture will be used on which modifications will be made. This basic architecture is an Imagenet [7] pretrained ResNet50 model fine-tuned for 50 epochs with a constant learning rate of 0.001 for all layers. The loss function used is cross entropy and stochastic gradient descent with 0.9 momentum as optimizer.

We have tried a variety of modifications over the data (data-augmentation), different models (ResNet50, ResNet101 and InceptionV3) and different fine-tuning approaches and learning rate policies.

The top-1 and top-5 validation accuracy achieved with this base configuration is 88.49% and 97.45%.

3.1 Ratio Adaptation

CompCars images come in a variety of sizes and aspect ratios. One problem of fully connected classification CNNs is that input images have to be of a given size (224×224 pixels for ResNet and 299×299 pixels for Inception). So, to feed these images into the CNN we need to resize them to fit the expected sizes. The problem is that all images that do not have a 1:1 aspect ratio will be deformed in the resizing process.

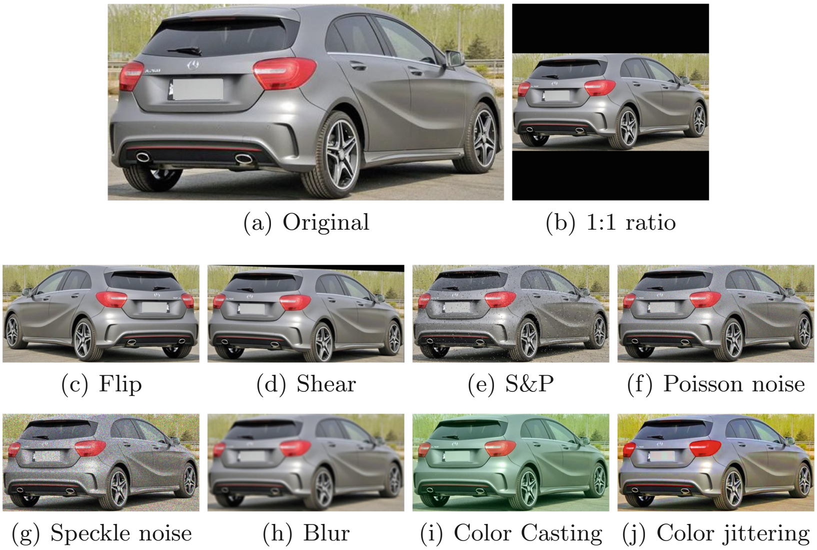
This could be an obstacle to the network learning. To discern if this is the case, we have developed an experiment: in the process of training each image is padded with two vertical or horizontal bands to adapt its ratio and prevent the deformation. An example of this operation can be seen in Fig. 2(b).

We found that the top-1 accuracy drops from 88.49% to 82.12% when the ratio is adapted. This is a 6.37% loss in accuracy. This could be explained by the fact that the net is losing generalization capacity because of the vertical/horizontal bands that are being introduced in the images provoking a reduction in the area with relevant information.

Moreover, even if the images are deformed, the network has the ability to learn and interpret the content. So, the ratio adaptation technique has been discarded.

3.2 Data-Augmentation

Data augmentation its a common tool used in deep learning to artificially increment datasets. Its use is compulsory when available data is limited as it helps to fight overfitting. Although in our case the dataset that we are going to use has a huge amount of images, we can get benefit from data augmentation. To do so, we have implemented the following data augmentation operations:



(a) Original (b) 1:1 ratio

(c) Flip (d) Shear (e) S&P (f) Poisson noise

(g) Speckle noise (h) Blur (i) Color Casting (j) Color jittering

Fig. 2. Data augmentation and ratio adaptation examples

- *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.*
- *Bluring*: 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.

An example of the previous described data augmentation operations can be seen in Fig. 2.

The process of data augmentation is as follows: in first place an horizontal flip its applied over the image, then, one of the other operations is randomly selected. This data augmentation process is computed online for each batch, therefore, all the images are slightly different in each epoch.

With this configuration we achieved a 95.48% top-1 validation accuracy, which is an improvement of 6.99% over the base model.

3.3 Learning Rate Policies

Until now, the learning rate used has remained constant during the training. A commonly used tool is to implement learning rate policies to modify it throughout the training. Of all those that have been tested, the one that has obtained the best results is the stepped one. This is, reduce the learning rate every n epochs.

In our case, we have added to the previous best model (base model + data augmentation) a 10-step policy rate (divide by 10 the learning rate every 10 epochs).

With this configuration we achieved a 97.03% top-1 validation accuracy, which is 1.55% better.

3.4 Fine-Tuning Process

As we previously said, we have been fine-tuning the model using the pretrained weights on Imagenet. As we have changed the last fully connected layers in order to adapt the network, this weights are randomly initialized, so, they have a difference in training compared with the rest of the network. An interesting approach is to train the fully connected layer alone and after that, the full network as we have been doing. We call this process 2-step fine-tuning.

After having tried multiple combinations of learning rate policies with 2-step fine-tuning the best results have been obtained when using constant learning rate in the fully connected training and 10-step in the full training.

With this configuration we achieved a 97.16% top-1 validation accuracy.

3.5 Other Models

So far Resnet50 has been used as the base model. In orther to achieve best results we have tried deeper models as Resnet101 and InceptionV3. After multiple trainings and configurations the best results for each model have been achieved with the 2-step fine-tuning process for both of them and constant+10-step learning rate policy in the case of Resnet101 and 10-step for InveptionV3.

With this configuration we achieved a 97.59% and 97.62% top-1 validation accuracy for Resnet101 and InceptionV3 respectively.

The best result was obtained with InceptionV3 with a top-1 validation accuracy of 97.62%.

A comparison of the different models can be seen in Table 1. Figure 3 shows some examples of the classification results with validation images.

Table 1. Results of the different configurations.

Model	Ratio/data-augmentation	Fine-tuning	Lr policy	Top-1/5 validation accuracy (%)
ResNet50 (base model)	✗/✗	Full	Constant	88.49/97.45
ResNet50	✓/✗	Full	Constant	82.12/92.21
ResNet50	✗/✓	Full	Constant	95.48/99.26
ResNet50	✗/✓	Full	Step-10	97.03/99.62
ResNet50	✗/✓	2-step	Constant+10-step	97.16/99.60
ResNet101	✗/✓	2-step	Constant+10-step	97.59/99.68
InceptionV3	✗/✓	2-step	Step-10	**97.62/99.64**
Yang et al. (CompCars)	–	–	–	91.20/98.10
Sighthound	–	–	–	95.88/99.53

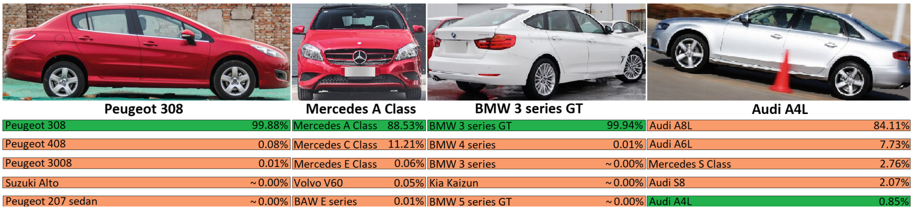


Fig. 3. Fine-grained classification results. Three correct classifications (left) and one error (right).

4 Conclusions and Future Works

In this paper we have described an end-to-end training methodology for CNN-based fine-grained vehicle model classification. Compared to other methods, our proposal relies exclusively on images, without using complicated architectures or high time demanding datasets (pose normalization, part localization, extra info, etc.).

Data augmentation has been found to significantly improve performance, even with a large dataset. The use of 2-step fine-tuning and adaptive learning rate allows the system to reach the best results and by combining it with data augmentation we achieve 97.62% top-1 accuracy which outperform previous models like the one proposed by Yang et al. [31] or the private commercial solution from Sighthound [6].

As future work we have identified two promising lines. The first one is to adapt the system to track and reidentify vehicles in complex traffic scenes, which has a great potential in traffic surveillance. The second one is a classification reinforcement method based on pose and structural modeling in orther to achieve better results with an even wider variety of car models.

Acknowledgments. This work was supported in part by the Spanish Ministry of Science, Innovation and Universities under Research Grant DPI2017-90035-R, in part by the Community Region of Madrid (Spain) under Research Grant S2018/EMT-4362 and in part by the Electronic Component Systems for European Leadership Joint Undertaking through the European Union's H2020 Research and Innovation Program and Germany, Austria, Spain, Italy, Latvia, Belgium, The Netherlands, Sweden, Finland, Lithuania, Czech Republic, Romania and Norway, under Grant 73746.

References

1. Angelova, A., Zhu, S., Lin, Y.: Image segmentation for large-scale subcategory flower recognition. In: 2013 IEEE Workshop on Applications of Computer Vision (WACV), pp. 39–45, January 2013
2. Berg, T., Liu, J., Woo Lee, S., Alexander, M.L., Jacobs, D.W., Belhumeur, P.N.: Birdsnap: large-scale fine-grained visual categorization of birds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2011–2018 (2014)
3. Branson, S., Van Horn, G., Belongie, S., Perona, P.: Bird species categorization using pose normalized deep convolutional nets. BMVC 2014 - Proceedings of the British Machine Vision Conference 2014 (2014)
4. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision. ECCV, pp. 1–22 (2004)
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 886–893 (2005)
6. Dehghan, A., Masood, S.Z., Shu, G., Ortiz, E.G.: View independent vehicle make, model and color recognition using convolutional neural network. CoRR abs/1702.01721 (2017)
7. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 248–255 (2009)
8. Fang, J., Zhou, Y., Yu, Y., Du, S.: Fine-grained vehicle model recognition using a coarse-to-fine convolutional neural network architecture. IEEE Trans. Intell. Transp. Syst. **18**(7), 1782–1792 (2017)

9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016)
10. Khosla, A., Jayadevaprakash, N., Yao, B., Fei-Fei, L.: Novel dataset for fine-grained image categorization. In: First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition (2011)
11. Krause, J., Jin, H., Yang, J., Fei-Fei, L.: Fine-grained recognition without part annotations. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5546–5555 (2015)
12. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3D object representations for fine-grained categorization. In: The IEEE International Conference on Computer Vision (ICCV) Workshops, pp. 554–561 (2013)
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012)
14. Kumar, N., et al.: Leafsnap: a computer vision system for automatic plant species identification. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7573, pp. 502–516. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33709-3_36
15. Lin, T.Y., RoyChowdhury, A., Maji, S.: Bilinear CNN models for fine-grained visual recognition. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1449–1457 (2015)
16. Liu, J., Kanazawa, A., Jacobs, D., Belhumeur, P.: Dog breed classification using part localization. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7572, pp. 172–185. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33718-5_13
17. Llorca, D.F., Arroyo, R., Sotelo, M.: Vehicle logo recognition in traffic images using HOG and SVM. In: 16th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 2229–2234 (2013)
18. Llorca, D.F., Colás, D., Daza, I.G., Parra, I., Sotelo, M.: Vehicle model recognition using geometry and appearance of car emblems from rear view images. In: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 3094–3099 (2014)
19. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
20. Maji, S., Rahtu, E., Kannala, J., Blaschko, M.B., Vedaldi, A.: Fine-grained visual classification of aircraft. CoRR abs/1306.5151 (2013)
21. Nilsback, M.E., Zisserman, A.: A visual vocabulary for flower classification. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 1447–1454. IEEE (2006)
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556 (2015)
23. Sochor, J., Herout, A., Havel, J.: Boxcars: 3D boxes as CNN input for improved fine-grained vehicle recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3006–3015 (2016)
24. Sochor, J., Spanhel, J., Herout, A.: Boxcars: improving fine-grained recognition of vehicles using 3-D bounding boxes in traffic surveillance. IEEE Trans. Intell. Transp. Syst. **20**, 97–108 (2019)
25. Szegedy, C., et al.: Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9 (2015)

26. Vedaldi, A., et al.: Understanding objects in detail with fine-grained attributes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3622–3629 (2014)
27. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-UCSD birds-200-2011 dataset (2011)
28. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3360–3367 (2010)
29. Welinder, P., et al.: Caltech-UCSD birds 200 (2010)
30. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1794–1801 (2009)
31. Yang, L., Luo, P., Change Loy, C., Tang, X.: A large-scale car dataset for fine-grained categorization and verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3973–3981 (2015)
32. Zhang, N., Donahue, J., Girshick, R., Darrell, T.: Part-based R-CNNs for fine-grained category detection. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 834–849. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_54