End-to-End Car Make and Model Classification using Compound Scaling and Transfer Learning

Omar BOURJA[1], Abdelilah MAACH[2], Zineb ZANNOUTI[3], Hatim DERROUZ[4],
Hamza MEKHZOUM[5], Hamd AIT ABDELALI[6], Rachid OULAD HAJ THAMI[7], François BOURZEIX[8]
RIME Departement, Mohammadia School of Engineers, Mohammed V University in Rabat, 10100, Morocco[1,2]
Department of Electronics and Informatics (ETRO), Vrije Universiteit Brussel, Brussels[5]
Embedded Systems and Artificial Intelligence Departement, MAScIR, 10100, Morocco[1,3,4,6,8]
IRDA team, ADMIR Laboratory, Rabat IT center, ENSIAS, Mohammed V University, Rabat 10100, Morocco[3,4,7]
Corresponding Author: Omar BOURJA

Abstract—**Recently, Morocco has started to invest in IoT systems to transform our cities into smart cities that will promote economic growth and make life easier for citizens. One of the most vital addition is intelligent transportation systems which represent the foundation of a smart city. However, the problem often faced in such systems is the recognition of entities, in our case, car and model makes. This paper proposes an approach that identifies makes and models for cars using transfer learning and a workflow that first enhances image quality and quantity by data augmentation and then feeds the newly generated data into a deep learning model with a scaling feature–that is, compound scaling. In addition, we developed a web interface using the FLASK API to make real-time predictions. The results obtained were 80% accuracy, fine-tuning it to an accuracy rate of 90% on unseen data. Our framework is trained on the commonly used Stanford Cars dataset.**

Keywords*—*Vehicles classification; deep learning; compound scaling; transfer learning; IoT

I. Introduction

Intelligent Transportation Systems (ITS) represent a combination of advanced information and communication technologies. They are used in transportation and traffic management systems to enhance road transportation networks' safety, efficiency, sustainability, reduce congestion, and improve the driving experience. The performance of road networks can be monitored and adjusted in real-time. Video surveillance systems have become so used in ITS. With the massive adoption of high-definition cameras, advanced analytics, and AI, surveillance systems are faced with increased workloads and are no longer just for security purposes. An intelligent transportation system should include the minimum requirements for managing traffic, including vehicle detection [1], vehicle tracking [2], [3] , and vehicle type classification [4], [5]. Because of traffic jams, lack of vehicle parking spots, and pollution, traffic control has always been a problem in the urban areas of Morocco [6]. Due to such events, traffic monitoring is crucial for collecting statistical data to design better and plan transportation infrastructure, other functionalities that can be integrated in an intelligent transport system is inter-vehicle distance estimation [7], [8]. Vehicle classification can solve numerous problems and help for a better traffic organization. Motivated by this fact, we have developed a framework to classify models and car makes in real-time to solve practical use case scenarios in ITS. In general, identifying car make and model has not been an easy process for computers because of its visual complexity and differences between classes. However, Humans can simply identify a car by its logo or hood ornaments. Given the complexity of the problem, various approaches were used, starting from classical machine learning models to intricate deep learning models that achieved state-of-art results. Deep learning has been talked about a lot in recent years. And for a good reason, this subset of machine learning has imposed itself impressively in several research fields of which car classification was a part. Several methods and algorithms of deep learning were used to classify car make and model, primarily, Convolutional Neural networks (CNN), which are powerful programming models allowing in particular image recognition by automatically assigning each image provided as input a label corresponding to its class. Also, deep Neural Networks(DNN) a multilayer neural networks which can include millions of neurons, divided into several dozen layers. Deep learning empowers Artificial Intelligence to learn new rules to be more reliable and efficient. The exponential improvement in computing power and the development of related applications allow artificial intelligence to generate more complex and dense layers of neurons.

The challenge with model and car classification is the fine-grained feature. Compared to basic image recognition, the dataset is more diverse in contrast to the similarities found pixel-wise. The question is whether a model can differentiate between different cars model and make found on the fine-grained dataset. To solve this issue, a deep learning model should be able to adapt and recognize the similarities found in the dataset and enhance the prediction. Hence, a compound scaling model in which width, depth, and resolution are scaled so that the model captures more fine-grained patterns. In this paper, we present a novel approach to classifying models and car makes. Inspired by the recent work on scaling neural networks, we worked with the EfficientNet [9] model pre-trained on the ImageNet dataset. We fine-tuned the model to our needs, thus, adding layers to reduce complexity. We also used transfer learning for prior knowledge of the model weights. With such behavior, the model can adapt to different scenarios, therefore, better prediction accuracy. We conducted experiments on the challenging Stanford Cars dataset [10], which contains 196 different categories of cars taken from different angles. We have used FLASK API to create a web interface so as to achieve real-time prediction.

The paper is repartitioned as follows. First, we present a literature review on the subject of vehicle classification. Then we discuss the methodology in which we formulate the problem, explain the architecture used and discuss the image preprocessing, data augmentation, and transfer learning phase. Following is the experimentation section, where we discuss the dataset used, the model implementation, and the training loss function. After that, the results section is where we discuss the obtained results. Finally, a conclusion and perspective section.

II. Related Work

Car make and model classification problem has widely been addressed using two research category methods. The first method focuses on handcrafted feature engineering. The second one, instead, focuses on machine learning and deep learning techniques. Since our method is based on a deep learning architecture, we mainly focus on related work with similar approaches. Xingyang Ni et al [11] 2021, compared two methods of car models and makes classification: a straightforward classification and a more flexible metric learning method. They built their model based on the ResNet 50 pre-trained on the ImageNet dataset and retraining it on the VERI-WILD dataset that contains approximately 0.4 million images with 14 types of vehicles, and 149 different car makes. As for the result, upon using a cross-entropy loss, they found a 96.8% accuracy rate on the type classification and 95.6% on the make classification. While for the triplet loss method, they found an accuracy rate of 97.4% on the type classification and 95.3% on the make classification. In addition, they used a lifted structured loss method in which they found 97.7% on the type classification and 96.2% on the model classification. Ye Xiang et al [12] 2019, proposed a four-stage pipeline that consists of part detection, part assembling, topology constraint, and classification for fine-grained vehicle recognitions. They used a backbone model trunked at the middle in the first stage. Afterward, they called for pointwise convolutional layers that put together related parts into the same feature map. Eventually, the topology constraint covers depth wise convolutional layers and approximates the possibility of the topology correlation between associated parts. Finally, they evaluated the model on two public datasets: the Stanford Cars dataset and the CompCars dataset. In both datasets, various car viewpoints can be found. The results obtained were 94.3% accuracy on the CompCars dataset, 94.3% on the Stanford Cars dataset for the model classification and 99.6% for the make classification. Rachmadi et al. [13] 2018, proposed a pseudo-long short-term memory classifier for identifying a single image. The presented technique considers the split pictures to be time-series frameworks. Those images are outlined by cropping input images with a two-level spatial pyramid region configuration given to the P-LSTM classifier in a cycle. And to calculate the prediction of each class, they added a fully connected layer. They used the MIO-TCD dataset, which contains 648,959 vehicle images of 11 types of vehicles. They obtained a 97.98% accuracy rate. Jung et al. [14] 2017, trained ResNet models using actual traffic surveillance recordings. A joint fine-tuning method is employed to fine-tune all parameters and not only the final dense layer. They used DropCNN that arbitrarily drops the probabilities from the aforementioned backbones during training. They used the MOI-TCD dataset. They obtained a 97.9% accuracy rate. Hu et al [15] 2017, presented a spatially weighted convolutional neural network that accommodates a predefined amount of pooling channels. The model then takes out deep convolutional neural network features with the enlightenment of its learned masks. They have achieved an accuracy of 93.1% on the Stanford Cars dataset and 97.6% on the CompCars dataset. Lee and Chung [16] 2017, proposed twelve local expert networks and six global networks. They used three neural network structures: AlexNet, GoogLeNet, and ResNet18 .The local expertise and global networks are trained with the particular subsets and entire training set, respectively. They generated the prediction by combining the predictions of one local expert network and multiple global networks. They used the MIO-TCD dataset to get an accuracy rate of 97.92%. Huttunen et al. [17] 2016, presented a deep learning neural network that employed SVM (Support vector machine) on a dataset that contains over 6500 images. They found an accuracy rate of 97.75% for the deep learning method and 96.19% for the SVM method. Dong et al. [18] 2015 proposed a sparse Laplacian filter learning method to minimize the parameters of convolutional layers with a large number of unlabeled samples. They collected the BIT-Vehicle dataset, which contains 9850 high-resolution vehicle frontal-view images. They achieved an accuracy rate of 96.1%. Yang et al. [19] 2015, proposed a model based on pre-trained weights of the ImageNet dataset and fine-tuned it with the CompCars dataset. The result obtained was 82.9% accuracy in the car make, 76.7% in the car model, and 80.8% in car parts. Girshick et al. [20] 2014 proposed the fastest model, taking approximately 2 seconds for an object to be detected. The approach used similar layers for both the detection and classification tasks. First, the model detects the spatial geometric position of an object using a sliding window method. This allowed the model to classify vehicles accordingly, utilizing the image's extracted objects(features). The final accuracy of the model was a fascinating 73.2% in image classification. Although the classification accuracy wasn't that high, the model was fast enough to compensate for the lack. Wang et al. [21] 2013 used SPM (Spatial Pyramid Matching). The method mainly focuses on detecting the spatial distance that can be found between detected objects of an image. In addition to SPM, SIFT was used to extract features, followed by an LLC (Locality-constrained Linear Coding) to extract locations. The classification task achieved a 59.3% accuracy, improved by an SVM classifier later on. Krizhevsky et al. [22] 2012 proposed a low-level features extraction method using Gabor filters. The model was composed of higher layers that deal with classification tasks and lower layers that deal with extracting features, the classifier used was an SVM. The model performed a 93.3% accuracy on image identification and 83.3% on image classification. Cheung et al. [23] 2008 Used SIFT algorithm (ScaleInvariant Feature Transformation), the model matches interest points in car images. The framework consists of an optimization network that uses the geometry of the image to spot interest points in cars. If the matched points in the training model are similar to the test phase, those two points are called inliers; thus, the classification of the car matches. The only drawback of this framework is that it matches images in the dataset for the same angle only, resulting in a mismatch if the angle is modified. Bay et al. [24] 2008 developed an unsupervised learning model that acts on the behavior of labeled image subcategories. These subcategories were generated using a segmentation. The model allowed only focus on essential image features. Hence, removing the background as it's considered as noise, the extracted features are then used for the classification task using a categorial loss function. This method led to the foundation of segmentation in car and model classification. Many methods have developed a fine-tuned model using segmentation filters. Some models even allowed modification in terms of kernel density. This ability make the model more robust in detecting and filtering important car image features. In recent work on vehicle classification, compound scaling has been used to extract fine-grained features. In our work, we sought to explore more the use of transfer learning with EfficientNet compound scaling coefficients pre-trained on the ImageNet and the MobileNet model architecture to classify model and car make.

III. Methodology

We describe our method as displayed in Fig. 1. The framework consists of three phases, the preprocessing and data augmentation phase, the model implementation phase, and the transfer learning phase. The model is trained end-to-end, and we developed a web interface for real-time prediction using the trained model weights. We will discuss each of these phases in the following sub-sections.

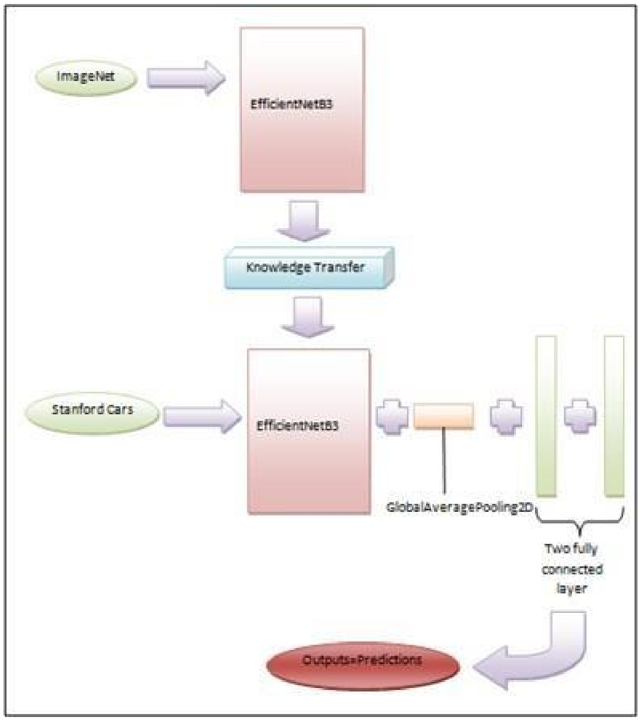


Fig. 1. The Training Workflow of our System Combining Transfer Learning and EfficientNetB3 Pre-Trained Model

A. Problem Formulation

We define the problem as a categorical classification scenario in which we ought to classify model and car make according to the scalability of the convolution neural network in question–That is, the learning behavior. Therefore we can define a Convolution network as:

$$N = \bigodot_{i=1..s} \rho_{i^i}^{L}(X_{<H_i,W_i,C_i>}) \tag{1}$$

where:

N defines a list of composed layers.
ρ denotes layer ρ_i is repeated L_i times in i.
$< H_i, W_i, C_i >$ denotes the shape of tensor X where H_i , W_i are the spatial dimension and C_i is the channel dimension.

The objective is to find the best ρ, yet we want our model to be scalable in order to extract fine-grained features, so instead of focusing on finding the best ρ, we focus on finding the best scaling dimensions. As described in [9], the model fixes ρ and uniformly explores all layers parameters with a constant ratio. As such, we have an optimization problem, which can be formulated as follows:

$$\underset{d,w,r}{max}\ N = \bigodot_{i=1..s} \hat{\rho}^{d\cdot\hat{L}_i}(X_{<r\cdot\hat{H}_i, r\cdot\hat{W}_i, w\cdot\hat{C}_i>}) \tag{2}$$

where:

d, w and r are the coefficients, depth, width and resolution respectively.
$< r \cdot \hat{H}_i, r \cdot \hat{W}_i, w \cdot \hat{C}_i >$ defines the predefined parameters multiplied by the coefficients.

The advantage of scaling optimal d, w, r is that when scalling depth (d), the network tends to capture more complex features. Scalling the width (w) will allow the network to capture more fine-grained features. For the resolution (r), the network will have the ability to capture different patterns due to the enlargement of the resolution, making it easy to extract fine-grained features. With this in mind, the issue persists when maximizing the accuracy in contrast to the scalable parameters. The network will often become challenging as the scaling values depend on each other. Hence, we use EfficientNet with compound scaling parameters. Intuitively, the network scale is according to a compound coefficient determined by a grid search. This allows the network to fine-tune itself according to the optimal need. Fig. 2 shows the scalling behavior of EfficientNet compared with different other methods.

B. Model Architecture

We describe our model as a set of a combination between EfficientNet for compound scaling and MobileNet [25] as the model architecture. EfficientNet is a convolutional neural network that relies on scaling the width, depth, and resolution uniformly. In addition to that, the network has a small number of parameters compared to other models: it has only 12,320,535 parameters, but it has proven to reach better results on the ImageNet dataset compared to other models with a higher number of parameters. Thus, we transfer knowledge of the trained EfficientNet model and use it in our system using transfer learning. Fig. 3 shows the EfficientNet architecture. As for the MobileNet architecture, it uses mobile inverted bottleneck convolution (MBConv), applying depth-separable convolution with residuals.

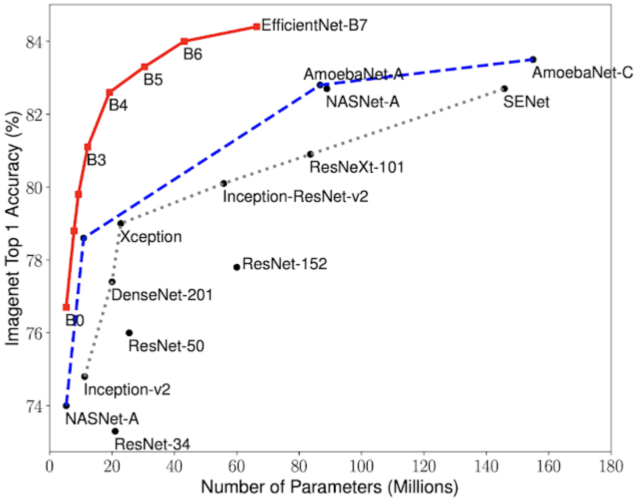


Fig. 2. Comparison between EfficientNets and other Existing CNNs on ImageNet

The main difference between the regular residual block and the invested residual block is that the latter follows a narrow > wide > narrow approach. In contrast, the first follows a wide > narrow > wide path approach. For example, Fig. 4 shows the difference between Residual and Inverted residual blocks.

C. Image Preprocessing and Data Augmentation

Our dataset has a small number of cars in each class: about forty pictures. Training a deep neural network on few images is often challenging: the model having access to only a limited number of observations will tend to Overfit. In this case, the performance is poor on the test set while they are good on the training set. This phenomenon is often solved by increasing the size of the dataset and/or reducing the number of model parameters. The first method is often challenging to set up because the work of collecting/labeling new observations is laborious. The second possibility is conceivable for an image recognition problem. However, even the most miniature complex models can contain hundreds of thousands of parameters, which are tricky to achieve. As data augmentation allows new labeled images to be generated from those already available, it is a relatively more straightforward solution to implement, and the results can be surprising. The most well-known technique of data augmentation is image data augmentation. It combines the methods used to artificially increase the size of a training dataset by creating modified versions of images from the available training images. We can then effectively improve the learning process as it results in more training samples for the neural network model. Augmentation techniques can create image variations that can enhance the ability of training models to generalize what they have learned to new images, which significantly improves model performance. The data augmentation applies only to the training set and not the validation or test set. This differs from data preparation, such as image resizing, which must be performed consistently across the entire dataset interacting with the model. Fig. 5 shows a sample of data augmentation on the Stanford Cars dataset.

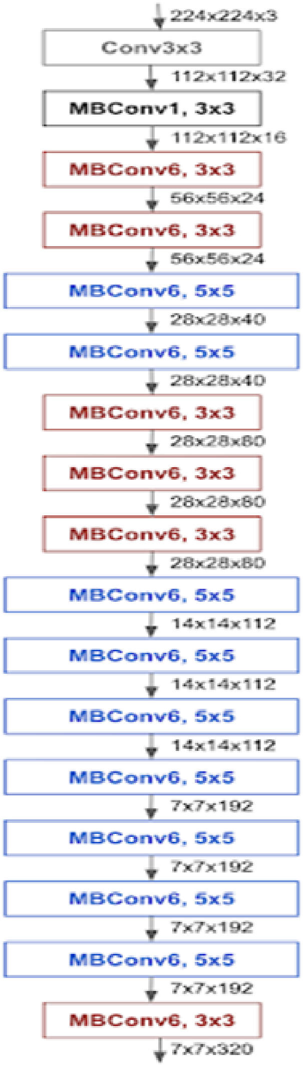


Fig. 3. EfficientNet Architecture

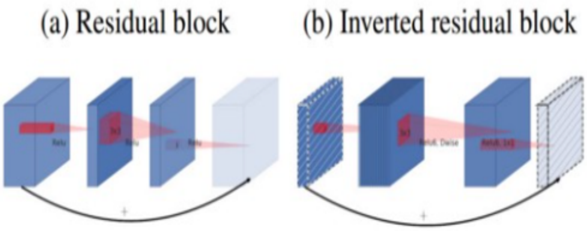


Fig. 4. Comparison between Residual Block and Inverted Residual Block

D. Transfer Learning

Transfer learning [26] has become common in the past few years because it has proven to achieve better results even with the use of a small amount of data. In our work, we have used transfer learning, a supervised learning technique that consists of taking a pre-trained model and reusing it on another dataset. Fig. 6 shows the workflow of the transfer learning technique.

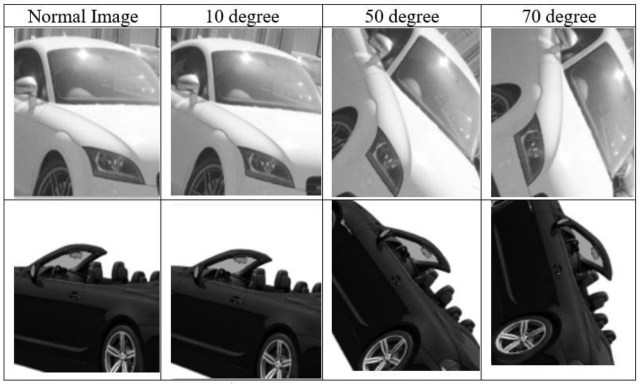


Fig. 5. Augmentation Technique Implemented on the Stanford Cars Dataset

Transfer Learning consists of the transfer of knowledge from one task to another. This behavior allows the network to solve similar problems with the same pre-trained model. This will eventually improve the quality of learning and reduce the computation time. However, deep learning requires always having a large dataset to operate the neural networks at their foremost. Therefore, we can adjust using Transfer Learning to get better predictions even with a small dataset.

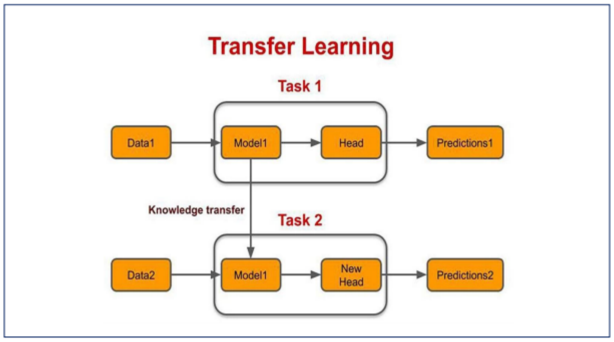


Fig. 6. Transfer Learning Workflow

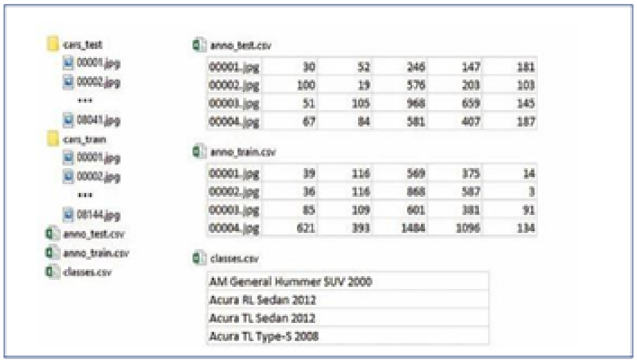
IV. Experimentation

A. Understanding the Stanford Dataset

Stanford Cars dataset has been widely used for model and car classification. The dataset was collected in 2013 and contained 196 different categories of cars (model, make, and year). The dataset is split into 8,144 pictures for the train set and 8,041 pictures for the test set. Fig. 7 shows an example of some quantitative car images in the Stanford dataset.

The files anno-test.csv and anno-train.csv are a table of six columns. The first column represents the name of the picture. From the second to the fifth column, we have four values of pixels that show the exact emplacement of the vehicle in the image. The last column contains information that classifies the car. The list of all classes is in the ”classes.csv” file. Fig. 8 shows the dataset files structure.

Fig. 7. An Overview of the Diversity in the Stanford Dataset



cars_test
00001.jpg
00002.jpg
...
08041.jpg
cars_train
00001.jpg
00002.jpg
...
08144.jpg
anno_test.csv
anno_train.csv
classes.csv

anno_test.csv

00001.jpg	30	52	246	147	181
00002.jpg	100	19	576	203	103
00003.jpg	51	105	968	659	145
00004.jpg	67	84	581	407	187

anno_train.csv

00001.jpg	39	116	569	375	14
00002.jpg	36	116	868	587	3
00003.jpg	85	109	601	381	91
00004.jpg	621	393	1484	1096	134

classes.csv

AM General Hummer SUV 2000
Acura RL Sedan 2012
Acura TL Sedan 2012
Acura TL Type-S 2008

Fig. 8. Data and csv Files Structure

B. Model Implementation

Car classification is a challenging task in machine learning due to the variety of details in each car. Therefore, we used transfer learning on the EfficientNet model pre-trained on the ImageNet dataset and then fine-tuned the model to get better results. First, we started by adding layers to our base model, mainly the globalAveragePooling2D layer, to reduce the variance and complexity of calculations, two fully-connected layers with the activation function ”relu,” and the integration of dropout to reduce overfitting. Next, we trained our model on the Stanford cars database combined with MoVITS Dataset [27]. Finally, to get a higher accuracy rate, we fine-tuned the model by unfreezing our entire model and retraining it.

C. Training on the Stanford Dataset

We used adam as an optimizer and categorial cross-entropy as a loss function to train our model. The use of adam, in this case, is due to the quick convergence that the optimizer allows. Since the dataset is quite complex in term of diversity, adam will help reduce computation time and converges in a significantly lower period. Due to the different classes in the dataset, we chose categorical cross-entropy as our loss function. To evaluate our model, we minimize the loss and compute the accuracy of the training and validation data. Since this is a classification task, we also demonstrate a confusion matrix to help visualize the behavior of the network. We define the categorical cross-entropy loss function as follows:

$$\mathcal{L}(y, y') = \sum_{j=0}^{M} \sum_{i=0}^{N} (y_{i_j} * \phi) \tag{3}$$

where:

$\phi = \; log\; (y'_{ij})$.
y' is the predicted value, y is the ground-truth value.

To compute the accuracy, we use the following principle:

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

where:

A represents the Accuracy.
TP + TN represents the number of correct predictions.
TP + TN + FP + FN represents the total number of predictions.

In summary, anytime the prediction is incorrect, the forecast is False. Otherwise, it is True. Therefore, the final objective is to maximize the prediction as True (True Positive and True Negative) and minimize the prediction as False (False Positive and False Negative).

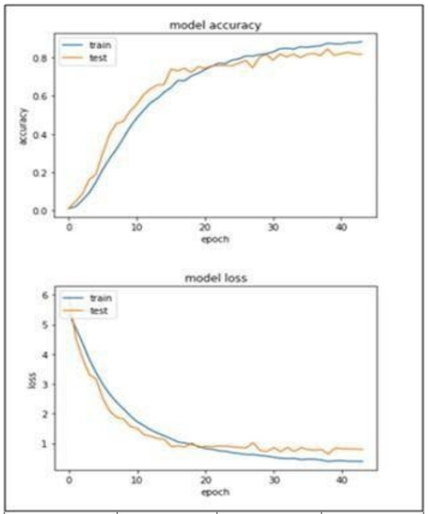
V. Results and Discussion

Compared with ResNet implementation in [11], we show that using EfficientNet has significantly enhanced the network's ability to extract more detailed features of the dataset. Adding data augmentation, the model learns to adapt to different image perspectives making it more robust on unseen data. Furthermore, in contrast to [12], using pointwise convolutional layers to extract fine-grained features, we incorporated scaling coefficients. This allowed the network to scale parameters for an optimal state. Adding MobileNet as a model architecture with mobile inverted bottleneck convolution reduced memory requirement compared to classical residual block.

We trained our model first using 44 epochs. Comparing the training and test loss, as shown in Fig. 9, the model started to find the optimal state using a grid search for the model's scaling coefficient at around 20 epochs. After 20 epochs, the model fluctuated, considering the complexity and deviation of the dataset combined with the augmented images. Finally, After 40 epochs, the model stagnates. Thus, we deduct that the model achieved an optimal state for the given parameters. We found an accuracy of 88% on the train set and 82% on the test set. Fig. 9 shows the results.

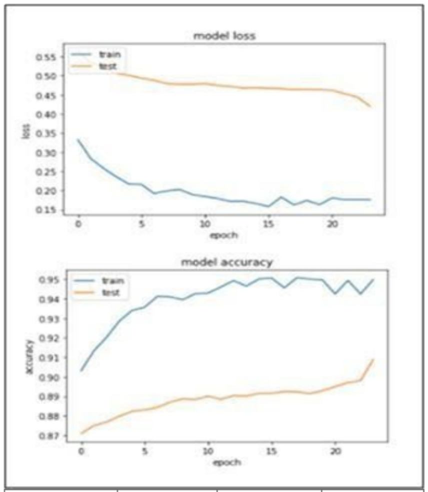
After the first experiment, we understood the behavior of our model, especially after 20 epochs where more fine-grained features are extracted due to the compound scaling of the optimal d, w, and r in the network. To enhance our model's accuracy, we fine-tuned it by retraining it to only 24 epochs where the network understands mostly essential image features. Compared to the previous experiment. We achieved an accuracy of 95% on the train set and 90% on the test. Fig. 10 shows the results.

To showcase the confusion of the network with respect to the predicted values, we generate a confusion matrix which is a summary of the results of predictions about a classification problem to visualize our prediction better. Fig. 11 shows the confusion matrix obtained. Correct and incorrect predictions are highlighted and broken down by class. The results are thus compared with the actual values. This matrix helps understand how the classification model is confused when making predictions.



Train Set Loss	Test Set Loss	Train Set Accuracy	Test Set Accuracy
0 .40	0.80	88%	82%

Fig. 9. Model Accuracy and Loss before Fine-Tuning



Train Set Loss	Test Set Loss	Train Set Accuracy	Test Set Accuracy
0 .18	0.45	95%	90%

Fig. 10. Model Accuracy and Loss after Fine-Tuning

The model accuracy on unseen data is significantly interesting, especially considering the use of augmented data. This allowed the model to understand different image perspective views and find similarities in fine-grained features to be then able to label the correct and incorrect predictions correctly. At this stage, our model's weight is ready to be used for real-time prediction.

Now that we have our model trained, we developed a friendly interface for real-time prediction using the FLASK API and the weighted model trained. The workflow of the API is described in Fig. 12.

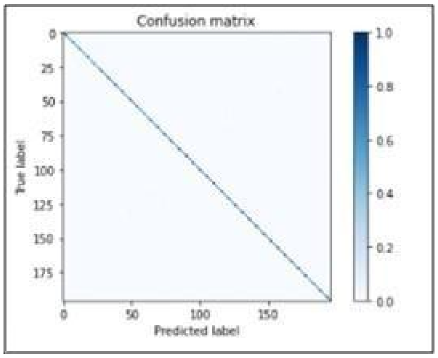


Fig. 11. Confusion Matrix of the Classification Task

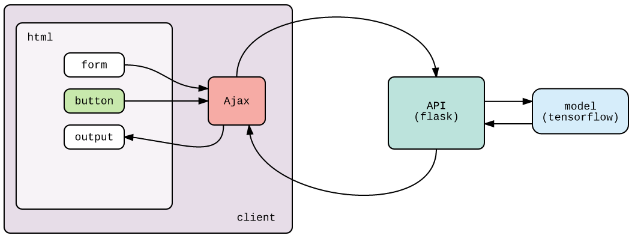
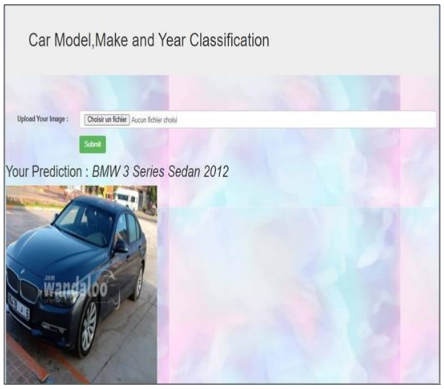


Fig. 12. FLASK API General Workflow

Our final interface is composed of an upload section where the user can freely upload a car image. Upon clicking on submit, the network uses the trained model to predict based on the model weights and outputs the predicted label of the given car image, describing the model and car make. Fig. 13 shows an example of the prediction mechanisms.



Car Model,Make and Year Classification

Upload Your Image : Choisir un fichier Aucun fichier choisi

Submit

Your Prediction : *BMW 3 Series Sedan 2012*

Fig. 13. An Example of Real Time Prediction in our Web Interface using Flask API

VI. Conclusion and Perspectives

In this paper, we explored the use of transfer learning with EfficientNet compound scaling coefficients pre-trained on the ImageNet and the MobileNet model architecture to classify model and car make. The use of such a combination proved to be efficient in this task. We used the Stanford Cars dataset and fine-tuned the model to find an accuracy rate of 90%. Additionally, we have implemented a web interface to predict car images in real-time using Flask API. Extracting fine-grained features is indeed a complex task, yet, we show that combining and fine-tuning the model can significantly enhance accuracy. Furthermore, we can improve the project prediction by building a system that identifies a car's plate number using Optical Character Recognition (OCR), which converts digital images to electronic text. The OCR output is an ASCII code that contains the text of the license plate and which can be compared to existing databases containing additional information on the car owner, such as his issuance badge, serial number, etc. These extracted features will then be used to improve the accuracy of the overall model.

Acknowledgment

This work was supported in part by the CNRST and in part by the MESRSFC, through the Development of an Integrated System for Traffic Management and Detection of Road Traffic Infractions Project.

References

[1] A. Saif and Z. R. Mahayuddin, "Robust drowsiness detection for vehicle driver using deep convolutional neural network," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 10, 2020.

[2] H. Ait Abdelali, O. Bourja, R. Haouari, H. Derrouz, Y. Zennayi, F. Bourzex, and R. Oulad Haj Thami, "Visual vehicle tracking via deep learning and particle filter," in *Advances on Smart and Soft Computing*. Springer, 2021, pp. 517–526.

[3] H. A. Abdelali, H. Derrouz, Y. Zennayi, R. O. H. Thami, and F. Bourzeix, "Multiple hypothesis detection and tracking using deep learning for video traffic surveillance," *IEEE Access*, vol. 9, pp. 164 282–164 291, 2021.

[4] H. Derrouz, A. Elbouziady, H. Ait Abdelali, R. Oulad Haj Thami, S. El Fkihi, and F. Bourzeix, "Moroccan video intelligent transport system: Vehicle type classification based on three-dimensional and two-dimensional features," *IEEE Access*, vol. 7, pp. 72 528–72 537, 2019.

[5] H. Derrouz, A. Cabri, H. Ait Abdelali, R. Oulad Haj Thami, F. Bourzeix, S. Rovetta, and F. Masulli, "End-to-end quantum-inspired method for vehicle classification based on video stream," *Neural Computing and Applications*, pp. 1–16, 2022.

[6] O. Bourja, K. Kabbaj, H. Derrouz, A. El Bouziady, R. O. H. Thami, Y. Zennayi, and F. Bourzeix, "Movits: Moroccan video intelligent transport system," in *2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*, 2018, pp. 502–507.

[7] O. BOURJA, H. DERROUZ, H. A. ABDELALI, A. MAACH, R. O. H. THAMI, and F. BOURZEIX, "Real time vehicle detection, tracking, and inter-vehicle distance estimation based on stereovision and deep learning using yolov3," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 8, 2021. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2021.01208101

[8] O. Bourja, A. Maach, Y. Zennayi, F. Bourzeix, and T. Guerin, "Speed estimation using simple line," *Procedia Computer Science*, vol. 127, pp. 209–217, 2018.

[9] Q. V. L. Mingxing Tan, "Efficientnet: Rethinking model scaling for convolutional neural networks," *ICML*, 2019.

[10] 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.

[11] X. Ni and H. Huttunen, "Vehicle attribute recognition by appearance: Computer vision methods for vehicle type, make and model classification," *Journal of Signal Processing Systems*, vol. 93, no. 4, pp. 357–368, 2021.

[12] Y. Xiang, Y. Fu, and H. Huang, "Global topology constraint network for fine-grained vehicle recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 7, pp. 2918–2929, 2019.

[13] R. F. Rachmadi, K. Uchimura, G. Koutaki, and K. Ogata, "Single image vehicle classification using pseudo long short-term memory classifier," *Journal of Visual Communication and Image Representation*, vol. 56, pp. 265–274, 2018.

[14] H. Jung, M.-K. Choi, J. Jung, J.-H. Lee, S. Kwon, and W. Young Jung, "Resnet-based vehicle classification and localization in traffic surveillance systems," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 61–67.

[15] 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.

[16] J. Taek Lee and Y. Chung, "Deep learning-based vehicle classification using an ensemble of local expert and global networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 47–52.

[17] H. Huttunen, F. S. Yancheshmeh, and K. Chen, "Car type recognition with deep neural networks," in *2016 IEEE intelligent vehicles symposium (IV)*. IEEE, 2016, pp. 1115–1120.

[18] Z. Dong, Y. Wu, M. Pei, and Y. Jia, "Vehicle type classification using a semisupervised convolutional neural network," *IEEE transactions on intelligent transportation systems*, vol. 16, no. 4, pp. 2247–2256, 2015.

[19] 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.

[20] T. D. Ross Girshick, Jeff Donahue and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CVPR*, 2014.

[21] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," *Onternational Conference on Human Computer Interactions (ICHCI)*, 2013.

[22] G. E. H. Alex Krizhevsky, Ilya Sutskever, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems 25 (NIPS)*, 2012.

[23] S. Cheung and A. Chu, "Make and model recognition of cars," *Projects in Vision and Learning*, 2008.

[24] H. Bay, T. Tuytelaars, and L. Gool, "Surf: Speeded up robust features," *Computer Vision and Image Understanding*, pp. 346–359, 2008.

[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," *Computer Vision and Pattern Recognition (cs.CV)*, 2018.

[26] N. Donges, "What is transfer learning? exploring the popular deep learning approach," *Built In*, 2019.

[27] H. Derrouz et al., "The moroccan video intelligent transport system dataset for vehicle detection," 2021. [Online]. Available: https://data.mendeley.com/datasets/5jcg5vfx58/3