Transfer Learning for classifying front and rear views of vehicles

Sara Baghdadi and Noureddine Aboutabit
IPIM Laboratory, National School of Applied Sciences, Khouribga, Sultan Moulay Slimane University, Beni Mellal, Morocco
E-mail: sara92.baghdadi@gmail.com

Abstract. Various computer systems have been proposed to classify vehicles according to several criteria (category, brand, model). Unfortunately, there is not much research on the classification of views, especially front and rear views. Several factors make this classification very difficult including similarity in shape, size, and color. This work aims to classify front and rear views of vehicles using the Transfer Learning (TL) approach. Here, we used a pre-trained CNN (AlexNet) that has been trained on more than a million images and can classify images into 1000 object categories. Thus, we transferred its learned knowledge and applied it to our new task (Classifying vehicle views). We conducted then two experiments. The first experiment has two scenarios: the first scenario is devoted to Transfer Learning using the AlexNet model, and the second scenario aims to build a network from scratch inspired from AlexNet. Experimental results reveal that the Transfer Learning approach gives high results. On the other hand, in the second experiment, we decided to use TL-AlexNet to extract features and train them with an SVM classifier instead of fully connected layers. And also, we combined the SVM with the fully connected layers. The accuracy rates have been improved after this experiment.

1. Introduction
Recently, most of the cities of the world started to integrate diverse technologies in the area of transport to increase comfort and safety. These technologies are referred to as ITS (Intelligent Transportation Systems). ITS includes two main parts: vehicle-based systems and infrastructure-based systems. Vehicle-based ITS are vehicle automation, active safety systems, and advanced driver assistance systems (ADAS). Infrastructure-based ITS are tolling, traffic monitoring and surveillance, traffic control, etc [1]. Vehicle classification is an essential part of the ITS. It has recently become an active subject of study for control and monitoring issues [2]. Different systems have been developed to detect and classify vehicles according to several criteria (category, make, model). However, there is not much work on the classification of views [3]. In this paper, we aim to classify the front and rear views of vehicles including buses, cars, motorcycles, and trucks. Many factors make the classification of front and rear views very difficult including similarity in shape, size, and color. This classification can be used for various purposes such as automatic parking, one-way street surveillance, automatic road enforcement, ADAS systems. This work is part of an overall objective of building a traffic monitoring system. The current approaches to object recognition use Machine Learning methods [4]. Deep learning is a sub-category of Machine Learning. With data availability and computing power, Deep Neural Networks have achieved higher results in the object classification field. Object classification is
basically carried out in two steps: feature extraction and classification. In traditional Machine Learning techniques, features are extracted manually however Convolutional Neural Networks extract features automatically. A Convolutional Neural Network is one of the most well-known deep neural networks. It specializes in image recognition. A CNN consists of pairs of convolution and pooling layers followed by a fully connected neural network consisting of at least one layer [5].

Networks can be designed and trained from scratch or from pre-trained models. In the literature, several pre-trained networks have been created for many tasks such as image, text, and audio classification [6]. The reuse of a pre-trained network on a new task through the transfer of knowledge is called Transfer Learning [7]. We will define TL and explain how knowledge can be transferred between tasks, in Section 3.1. Transfer learning has recently become more and more popular [6]. It is generally efficient, faster, and easier than training a network with randomly initialized weights from scratch [4]. In this paper, we conducted two experiments. The first one has two scenarios. The first scenario aimed to apply TL. we used the Alexnet model: a pre-trained CNN on more than a million images (1000 object categories). The second scenario aimed to use training from scratch in order to prove the robustness of Transfer Learning by comparing the two scenarios. In training from scratch, to keep the same evaluation conditions, we tried to design a CNN inspired from the AlexNet structure. Finally, we decided to use TL-AlexNet to extract features and train them with an SVM classifier instead of fully connected layers. And also, we combined these two classifiers.

The remainder of this paper is structured as follows. In Section 2, we highlighted some works that are closely related to vehicle classification. Section 3 describes all the techniques used in this work. Our experimental results and analysis are shown in Section 4. The final section concludes the paper and presents future work.

The next section highlights the related works to vehicle view classification.

2. Related Works

In literature, there is unfortunately no previous work on the subject of classification of vehicle views, except our work [8]. We have constructed a classification system for front and rear views of vehicles using two feature extractors (HOG and LBP) and two classifiers (SVM and kNN). LBP+SVM, LBP+kNN, and HOG+SVM achieved respectively the accuracies 83.54%, 84.81%, and 94.94%. KNN+ HOG achieved the best performance. It reached an accuracy of 97.47%.

We present now some related works that are closely related to the papers subject. In [9], the authors developed a recognition system of cars that combines two methods. The first is based on the shape of the rear-view. The second is based on the rear lighting of the car. The system achieved an accuracy of 89%. [10] proposed a linear SVM-based method to address vehicle make and model classification issues. The authors used SIFT (Scale Invariant Transform Feature) as a feature extractor. It reached an accuracy of 89%.

Recently, Transfer Learning become more and more popular. Pre-trained models have been used to solve many tasks, especially in image classification. [11] trained AlexNet and VGG-16 using Transfer Learning and training from scratch on the task of facial expression recognition. The results have shown that the TL technique outperforms the training from scratch. In [11], the authors used the Convolutional Neural Networks for colonic polyp classification using training from scratch and Transfer Learning evaluated on 8-HD-endoscopic databases. The experiments prove that merging classical features with CNN features is a smart method for better results. [12] presents a comparative study between transfer learning algorithms and traditional machine learning algorithms under the condition of domain class imbalance. The condition of domain class imbalance is characterized by the fact that the source and target domains have different probabilities of class, which can cause marginal distributional differences between the source and target data. From experimental results Graph Co-Regularization Transfer Learning (GTL)
3. Methods
This section defines the Transfer Learning approach with Convolutional Neural Networks in detail.

3.1. Transfer Learning
Transfer Learning is a machine learning technique that applies and transfers knowledge learned in previous tasks to solve a new task [13] as shown in Figure.1. In fact, there are many pre-trained networks that have been trained on large datasets, we cite: LeNet, Alexnet, VGG, GoogleNet, ResNet, and more. Thus, we can train the new model with a small dataset using the knowledge of one of these pre-trained networks. So, how knowledge can be transferred from one task to another? First, as we know, a pre-trained model has already been trained on millions of images. It has, therefore, become adapted to natural images [14]. Thus, it can efficiently extract features. It already learned how to detect edges, colors, squares, circles, patterns. So, these learned features are basically represented by the values of convolution filters which are the weights of the network. Thus, to transfer this knowledge we can just freeze the weights of the feature extraction network.

![Figure 1. Transfer Learning process.](image)

Training a Deep Learning model from scratch requires a huge amount of data to give higher results. Unfortunately, collecting and training a large dataset takes a long time, and requires high computational resources. Transfer Learning is used to solve this problem; we can achieve then best results with a small dataset by taking the knowledge (weights, parameters) of a pre-trained model and apply it on a new task [15]. In this work, we used the AlexNet model to recognize vehicle views. AlexNet is a Convolutional Neural Network that has been trained by Alex Krizhevsky and his team [4]. More details in Section.4.

3.2. CNN Architecture
Deep Learning is a subfield of Machine Learning. There are many Deep Learning architectures like Deep Neural Networks (DNN), Recurrent Neural Networks (RNN), and Convolutional Neural Networks (CNN). CNNs are extensively applied in computer vision tasks [16]. A CNN is composed of a feature extractor network and a classification network. Before CNN, feature extractors were independent of Machine Learning. However, CNNs integrate the feature extraction phase in the training process. It has transformed the manual process of feature extraction into an automated process thanks to the weight adaptation technique [17]. As shown in Figure.2, the input image passes into the feature extraction network. The extracted features are transformed into one-dimensional vectors and then enter into the classification network which produces the output.
Figure 2. CNN architecture

The feature extractor network contains multiple layers of convolution and pooling. As shown in Figure 3, in a convolution layer, the inputs are convolved with a collection of filters in order to detect features [7]. If we apply a convolution filter to an image, we get combinations of pixels as outputs. For the second convolution layer, the inputs are edges (for example). If we reapply the convolution, the output is combinations of edges and so on. So, the first layers extract the primitive features like edges, brightness, the middle layers regroup the primitive features to form shapes, squares, rectangles, and finally, these shapes are regrouped to form objects [18].

Figure 3. Convolution operation

The feature map is treated by the activation function. ReLU is the most recently used activation function [7]. It is represented by the following equation:

$$ReLU(F) = \begin{cases} \max(0, F), & F \geq 0 \\ 0, & F < 0 \end{cases}$$

Pooling layer groups neighboring pixels into one pixel by computing their average or maximum. The pooling layer aims to reduce the size of images [17].
4. Experiments & Results

To evaluate the performance of the algorithms we used, we built a vehicle views database (3240 images); we collected the images from the internet. And then, we classified the images into two categories: Front and Rear-view images. We divided the dataset into a training set and testing set. The training set aims to create the model, while the test set is used to classify the new images and calculate the evaluation metrics.

In all experiments, we set the maximum number of epochs at 10. This evaluation was done based on different metrics: overall accuracy, misclassification, and true classification rates for each view, and runtime per image.

The characteristics of the machine used to run all of the algorithms are detailed as follows: Lenovo ThinkPad with a processor Intel® Core™ i5 7th Generation CPU @ 2.50GHz 2.71GHz, RAM 8Go. Besides, the algorithms are implemented in MATLAB R2017b.

4.1. Experiment 1

As discussed above, in this work, there are three scenarios: Transfer Learning using AlexNet, CNN-training from scratch, and TL-AlexNet+SVM. In the first experiment, we started by applying Transfer Learning using AlexNet. Then, we designed another CNN architecture from scratch inspired from AlexNet in order to compare the two scenarios. In the second experiment, we used TL-AlexNet features to train an SVM classifier.

4.1.1. Transfer Learning

As said, in this work, we reused the AlexNet model. It has been trained on the famous dataset ImageNet with 1000 object categories. As shown in Figure 4, AlexNet contains 25 layers: five convolutional layers, three pooling layers, seven Rectified Linear Units (ReLU), two normalization layers, two dropout layers, three fully connected layers, one softmax layer [19].

![AlexNet Model](image)

Figure 4. AlexNet Model [20]

Figure 5 presents how we fine-tuned the AlexNet to perform classification on our dataset. As input, we have our own dataset. So, we froze the first part of the network (the weights are fixed) to use the knowledge of extracting features. However, we reset the final layers responsible for
classification (the yellow layers). We configured as output just two categories with labels (Front & Rear) instead of 1000 labels. And then we trained our new model.

![Figure 5](image.png)

**Figure 5. Fine-tuning the AlexNet model**

As shown in Table 1 and Table 2, the results are better when using the activation function ReLU and miniBatch=32. The model achieved an accuracy of 99.75%.

| Table 1. Results when using the activation functions (ReLU and leakyReLU) |
|------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                  | Rear        | Front       | Rear        | Front       | Rear        | Front       | Rear        | Front       | Runtime/Image |
| TL.AlexNet       | Overall accuracy | True Classif rate | Misclassif rate | True Classif rate | Misclassif rate | Runtime/Image |
| ReLU             | 99.42%      | 99.61%      | 0.39%       | 99.24%      | 0.75%     | 1.239981  |
| leakyReLU        | 99.09%      | 98.43%      | 1.57%       | 99.75%      | 0.25%     | 1.241131  |

| Table 2. Results when changing the batch size value |
|------------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                  | Rear        | Front       | Rear        | Front       | Rear        | Front       | Rear        | Front       | Runtime/Image |
| TL.AlexNet       | Overall accuracy | True Classif rate | Misclassif rate | True Classif rate | Misclassif rate | Runtime/Image |
| MiniBatch=32, ReLU | 99.75%      | 100%        | 0%          | 99.49%      | 0.51%      | 1.101806  |
| MiniBatch=64, ReLU | 99.42%      | 99.61%      | 0.39%       | 99.24%      | 0.75%      | 1.239981  |

4.1.2. Training from scratch

To compare the Transfer Learning approach with training from scratch, we tried to keep the same evaluation conditions: same dataset images, same image dimensions (227x227x3), same model structure as in AlexNet. So, before the training process, we configured various
hyperparameters like the number of layers, activation function, batch size, number of epochs, convolution layer parameters (filter size and stride), pooling layer parameters, etc.

First, we show the influence of the quality of the input images on the behaviour of the model. We trained the model for two dimensions (150x150) and (227x227) pixels. Thus, we can easily conclude from Table.3 that the better the image quality, the better the model performs. High quality means that there is more information to provide to the model.

| Size of input images | Overall accuracy | True Classification rate | Misclassification rate | True Classification rate | Misclassification rate |
|---------------------|------------------|--------------------------|------------------------|--------------------------|------------------------|
| 150x150 pixels      | 94.29%           | 92.91%                   | 7.09%                  | 95.69%                   | 4.31%                  |
| 227x227 pixels      | 95.85%           | 94.49%                   | 5.51%                  | 97.21%                   | 2.79%                  |

Table.4 presents the accuracy and error values for CNN-training from scratch.

| Model                  | Overall accuracy | True Classification rate | Misclassification rate | True Classification rate | Misclassification rate | Runtime/Image |
|------------------------|------------------|--------------------------|------------------------|--------------------------|------------------------|----------------|
| CNN-training from scratch | 96.79%          | 94.09%                   | 5.91%                  | 99.49%                   | 0.51%                  | 1.117101       |

As can be seen in Table.4, the model achieved an accuracy of 96.79% and a runtime of 1.117101 seconds per image.

4.1.3. Discussion

The training accuracy and loss function are plotted below for a better understanding. Figure.6 gives the curve of training accuracy and loss against the number of epochs (a- training from scratch, b- TL). In Figure.6, the overall loss decreases with increasing epochs. Thus, three points can be made to differentiate transfer learning from training from scratch (Figure.4.1.3 [15]): Transfer learning has (1) Higher start, (2) Higher slope, and (3) Higher asymptote [15].

From experimental results, we concluded that Transfer Learning learns faster and achieves greater accuracy, it outperformed the training from scratch. We explain this by many reasons: First, in Transfer Learning, we used AlexNet as a starting point for training our new model, we retrained just the last layer of the network. Second, we already said that the pre-trained models are already adapted to natural images. Thus, when applied to a new database. It can easily and efficiently extract features. These two reasons make TL faster and stronger compared to training from scratch.
4.2. Experiment 2: Transfer Learning-AlexNet Model + SVM

As we said, CNN is composed of a feature extractor and a classification neural network. The output of the feature extractor network is transformed into a one-dimensional vector which enters into the classifier network. The classifier is a fully connected neural network consisting of at least one layer [17]. As the pre-trained CNNs were trained with large natural image datasets, they represent a good feature descriptor. Thus, we decided to use the TL-AlexNet as a feature extractor and replace fully connected layers by an SVM classifier (Figure.8/Case1). On the other hand, we combined the SVM with the fully connected layers of TL-AlexNet (Figure.8/Case2). The input to any fully connected layer could be used as the input to an SVM classifier; features pass through one or two fully connected layers and then enter to the SVM classifier which outputs the labels.

As shown in Table.5, TL-AlexNet features+SVM achieved an accuracy of 99.48%. It does not improve the results. However, when the two classifiers are combined, the first case improves the accuracy to 99.87%. The second outperforms all the other models, it achieved an accuracy
Figure 8. Combining AlexNet Features with the classifier SVM

Table 5. Results when combining AlexNet Features with SVM classifier

| TL.AlexNet model | Rear Overall accuracy | True Classif rate | Misclassif rate | Front True Classif rate | Misclassif rate | Runtime/Image |
|------------------|-----------------------|-------------------|-----------------|------------------------|-----------------|---------------|
| TL.AlexNet +SVM  | 99.48%                | 99.21%            | 0.79%           | 99.75%                 | 0.25%           | 0.2822615     |
| TL.AlexNet +(FC6+SVM) | 99.87%            | 100%              | 0%              | 99.75%                 | 0.25%           | 0.829891      |
| TL.AlexNet +(FC7+SVM) | 100%              | 100%              | 0%              | 100%                   | 0%              | 0.791048      |

of 100% and saved time.

We can conclude that Deep Learning using Convolutional Neural Networks is a very good approach for vehicle view classification, and the use of Transfer learning is a smart choice being improved by adding the SVM classifier.
5. Conclusion

The front and rear view of vehicles is an integral part of a work that aims to build a traffic monitoring system. We used the Transfer Learning approach using a pre-trained CNN called AlexNet. To show the efficiency of this approach, we constructed another model from scratch inspired from AlexNet architecture to compare both models. Best scores have been achieved by the Transfer Learning technique. Some reasons for the success of the TL: the pre-trained model is already trained on a very large dataset. This makes the model adapted to natural images; it can easily detect edges, colors, squares, etc. These features are represented by the weights. Thus, in TL, these weights are frozen to transfer this learned knowledge. However, final layers are reset to classify the new categories. So, the main advantages of TL are: first it saves time, needs just a small dataset, and achieves good results. Finally, we decided to use the TL-AlexNet as a feature extractor, and replace/combine fully connected layers by an SVM classifier. In fact, our achieved results show that the front and rear vehicle classification system can be successfully integrated into traffic monitoring applications.

References

[1] Meng Lu 2016 Evaluation of Intelligent Road Transport Systems, Methods and Results, ISBN 978-1-78561-172-8 (hardback) ISBN 978-1-78561-173-5.
[2] Khalifa O, Balfaqih M, Bashir S and Albagoul A 2012 Automatic car identification system Proc. 2nd. Int. Conf. on Mathematical Applications in Engineering, https://doi.org/10.5829/idosi.wasj.2013.21.mae.99933.
[3] Baghdadi S and Aboutabit N 2019 Illumination Correction in a Comparative Analysis of Feature selection for Rear-View Vehicle Detection, International Journal of Machine Learning and Computing (IJMLC), vol 9 n.6. p 712-720. doi: 10.18178/ijmlc.2019.9.6.863.
[4] Krizhevsky A, Sutskever I and Hinton G E 2012 ImageNet Classification with Deep Convolutional Neural Networks, Advances In Neural Information Processing Systems, pp 19.
[5] Huo Z, Xia Y and Zhang B 2016 vehicle type classification and attribute prediction using multi-task RCNN, Proc. of 9th international congress on image and signal processing, biomedical engineering and informatics (CISP-BMEI)(Datong), p 564569.
[6] https://towardsdatascience.com/10-machine-learning-methods-that-every-data-scientist-should-know-3cc96e00ee9.
[7] Oztel I, Yolcu G and Oz C 2019 Performance Comparison of Transfer Learning and Training from Scratch Approaches for Deep Facial Expression Recognition, 4th Int. Conf. on Computer Science and Engineering (UBMK) (Samsun, Turkey) p 1-6, doi: 10.1109/UBMK.2019.8907203.
[8] Baghdadi S and Aboutabit N 2019 Front and rear vehicle classification, Advanced Intelligent Systems for Sustainable Development (AI2SD2019) Springer. doi:10.1007/978-3-030-36674-2_28.
[9] Santos D and Correia P -L 2009 Car recognition based on back lights and rear view features, Proc. of the IEEE Workshop Image Analysis for Multimedia Interactive Services p 137140. https://doi.org/10.1009/wiamis.2009.5031451.
[10] Manzoor M A and Morgan Y 2017 Vehicle Make and Model Classification System using Bag of SIFT Features, 7th IEEE Annual Conf. on Computing and Communication Workshop and Conference (CCWC) (Las Vegas, NV, USA) p 572-577.
[11] Ribeiro E et al. 2017 Exploring texture Transfer Learning for Colonic Polyp Classification via Convolutional Neural Networks, IEEE 14th International Symp. on Biomedical Imaging (ISBI 2017), (Melbourne, VIC), 2017, p 1044-1048, doi: 10.1109/ISBI.2017.7950695.
[12] Weiss K R and Khoshgoftaar T M 2017 Comparing Transfer Learning and Traditional Learning Under Domain Class Imbalance, 16th IEEE International Conference on Machine Learning and Applications (ICMLA), 337-343.
[13] Sarkar D, Bali R, Ghosh T and Panwar N Hands on Transfer learning with python, ISBN: 978-78883-130-7.
[14] Mathew S, Danielle D and Wee H T Deep Learning with Azure, ISBN-13 (pbk): 978- 1-4842-3678-9. ISBN-13 (electronic): 978-1-4842-3679-6. https://doi.org/10.1007/978-1-4842-3679-6.
[15] Rohan S, Debaruna S and Sudan J 2018 Transfer learning a comparative analysis.
[16] Chollet F Deep Learning with Python, ISBN 9781617294433.
[17] Phi K Matlab Deep Learning: with Machine Learning, neural network and artificial intelligence, ISBN-13 (pbk): 978-1-4842-2844-9. ISBN-13 (electronic): 978-1-4842-2845-6.
[18] Training A Model From Scratch , MATLAB & Simulink.
[19] Zhang Y, Allem JP, Unger JB and Boley Cruz T Automated Identification of Hookahs (Waterpipes) on Instagram: An Application in Feature Extraction Using Convolutional Neural Network and Support Vector Machine Classification., doi:10.2196/10513.

[20] Khvostikov A, Aderghal K, Benois-Pineau J, Krylov A and Catheline G, 3D CNN-based classification using sMRI and MD-DTI images for Alzheimer's disease studies, arXiv preprint arXiv: 1801.05968, (2018).