View-independent Vehicle Category Classification System

Sara Baghdadi[1], Noureddine Aboutabit[2]
IPIM Laboratory. National School of Applied Sciences, Khouribga
Sultan Moulay Slimane University, USMS
Beni-mellal, Morocco

***Abstract*—Vehicle category classification is important, but it is a challenging task, especially, when the vehicles are captured from a surveillance camera with different view angles. This paper aims to develop a view-independent vehicle category classification system. It proposes a two-phase system: one phase recognizes the view angles helping the second phase to recognize the vehicle category including bus, car, motorcycle, and truck. In each phase, several descriptors and Machine Learning techniques including traditional algorithms and Deep neural networks are employed. In particular, we used three descriptors: HOG (Histogram of Oriented Gradient), LBP (Local Binary Patterns) and Gabor filter with two classifiers SVM (Support Vector Machine) and k-NN (k-Nearest Neighbor). And also, we used the Convolutional Neural Network (CNN, or ConvNet). Three experiments have been elaborated based on many datasets. The first experiment is dedicated to choosing the best approach for the recognition of views: rear or front. The second experiment aims to classify the vehicle categories based on each view. In the third experiment, we developed the overall system, the categories were classified independently of the view. Experimental results reveal that CNN gives the highest recognition accuracy of 94.29% in the first experiment, and HOG with SVM or k-NN gives the best results (99.58%, 99.17%) in the second experiment. The system can robustly recognize vehicle categories with an accuracy of 95.77%.**

Keywords*—Vehicle category classification; view recognition; machine learning; deep learning; convolutional neural network

I. Introduction

Actually, with the increasing number of vehicles and surveillance cameras, it is necessary to automate vehicle recognition to collect traffic data [1]. Automatic vehicle recognition is an integral part of Intelligent Transportation Systems and has become an important subject of study for monitoring and surveillance issues [2]. It presents a particularly challenging problem in contemporary road safety. Various computer systems have been proposed to detect and classify vehicles according to several criteria (category, color, brand, size, views…) where the type (category) of a vehicle is the most important and essential characteristic [3].

Category classification plays an essential role in intelligent traffic systems and can be used extensively for various purposes such as highway surveillance, toll collection, traffic flow statistics [1], automatic parking, preventing heavy trucks from entering city bridges [4]. It can also be used to specify speed limits for each category to monitor traffic rules and regulation systems (sending warnings). For example, trucks have different speed limits than other vehicles. Vehicle category classification can also be used in self-guided vehicles to alert, for example, the driver that a truck is approaching. This work is a part of our project that aims to detect automatically prohibited overtaking of vehicles from a static camera. Thus, the future system must take into account special situations to allow, for example for motorcycles to overtake traffic without sending a warning for violation laws. In fact, the classification of vehicle categories is influenced by changes in viewing angles. This makes this task more challenging. This paper aims to create a system that classifies vehicle categories (Bus, Car, Motorcycle, and Truck) independently of the view angle (Front/ rear angles). The system has two phases: view recognition phase and vehicle category classification phase.

The motivation to take this challenge is that first, many previous studies have worked on classifying vehicles from the side. Unfortunately, there has been relatively little research on classifying vehicles from front and rear views [5]. Classifying vehicles, from their front or/and rear views, is more challenging. Most vehicle categories have the same characteristics in their view structures such as windshield, forward lighting, mirrors, bumper, etc. (Example: Fig. 1/ Front view). Second, most researchers have used just one, two or three methods in their comparative studies. Third, there is not a lot of work being done on vehicle view recognition [2]. Forth, no one has previously developed a view-independent classification system for vehicle categories.

Fig. 1. Example of Front View Structure of Trucks, Buses and Cars.

In this work, Machine Learning and Deep Learning techniques were used to classify the views and categories of vehicles. Object classification is a broad field of computer vision and machine learning research that aims to classify objects in images into meaningful categories [6]. This involves predicting the class of a new object in an image according to the information obtained from the training set of data whose category class is known [7]. This work is constituted of three experiments. The first experiment is devoted to vehicle view classification. The second one is dedicated to vehicle category classification. The third one presents the overall system using the best models we obtained. To accomplish this, we used three descriptors (LBP, HOG and Gabor filter), two classifiers, and Convolutional Neural Networks.

The rest of the paper is organized as follows. Section 2 presents the works that are closely related to vehicle classification. In Section 3, we present an overview of the system. Section 4 describes in detail all techniques used for vehicle view and category classification. The experimental results and comparisons are presented in Section 5. The last section presents the conclusions and future works.

The next section presents the previous research on vehicle classification.

II. Related Work

This section presents some information on some previous works related to vehicle classification. [8] used size and shape measurements as features with SVM and random forest classifiers to classify vehicles into four categories: car, van, bus and bicycle/motorcycle. The results showed that the SVM outperformed RF with an accuracy of 96.26%. [9] proposed a real-time classification of vehicle types from surveillance videos. Vehicles are classified as small cars, large cars or motorbikes using the HOG descriptor with SVM classifier. [3] proposed a vehicle-types recognition based on improved HOG_SVM. The authors used HOG to extract features from images and PCA to reduce the dimension and complexity of feature vectors. SVM is used as a classifier to classify vehicle types. The improved HOG_SVM model has an accuracy of 92.6%. In [4], the authors proposed for the vehicle type classification a cascade ensemble classifier using Multiple Layer Perceptron (MLP) and K-Nearest Neighbor (K-NN) as base classifiers. HOG, LBP and the combined feature HOG-LBP are used as inputs. [10] used two feature extractors the Pyramid Histogram of Oriented Gradients (PHOG) and Curvelet Transform and the classifiers MLP, SVM, kNN and Random Forest to classify the type and make of vehicles. The authors combined the PHOG with Curvelet Transform and applied for classification the RF/MLPs ensemble on a database of 600 images from 21 makes of cars/vans. The method achieved an accuracy of 96,5%. [11] proposed a method based on a linear SVM to solve vehicle make and model classification problems. For the feature extraction step, the authors used SIFT (Scale Invariant Transform Feature). They reported an accuracy of 89%. [12] classified three classes: vehicle taxi, a mini-bus, and a double-decker by estimating the width, length, and height. The model achieved an accuracy of 92.5%. In [13] used Logistic Regression, neural networks, and SVM as classifiers. LR was the best one compared to the other method with an accuracy of 93.4%. [14] proposed a new method of detecting and classifying vehicles into six types, such as SUV type, sedan type, RV type, based on images of visible light and thermal cameras. The authors extracted features by measuring the texture, the contrast, the homogeneity, the entropy and the energy of the images of the front view. The visible image classifier has an accuracy of 92.7% and the thermal image classifier has 65.8%. [15] proposed a multimodal method to address this task of vehicle type classification from traffic scenario videos. The method extracts image and audio features and fuses them to feed an SVM classifier. To extract features from images, the authors used the pre-trained networks AlexNet [16] and GoogleNet [17]. The model classified the vehicles as an armored vehicle, construction vehicle, crane vehicle, rescue vehicle, military vehicle, motorcycle, and rescue vehicle. It achieved an accuracy of 72.1%.

In recent years, Deep learning and especially Convolutional Neural Networks have been used to tackle the task of vehicle type classification but with different approaches and datasets [6]. CNNs have demonstrated better performance in image classification. [18,1,6,5] used Convolutional Neural Networks to classify vehicle types. [18] proposed a method based also on a semi-supervised convolutional neural network with Laplacian Filters for kernels. The authors constructed the BIT-Vehicle dataset of front view images. The method achieved an accuracy of 96.1% in daytime conditions and 89.4% in nighttime conditions. [1] combined two steps. The first is data augmentation and the second is the CNN model training. In [6], the authors used also the CNNs with low-resolution images from a frontal view. The model achieved an accuracy of 93.90%. Also, [5] developed a model using CNN, but, on rear view images. The model achieved an accuracy of 97.88 %.

Unfortunately, there is no previous work on vehicle view classification topic except our last work [2]. We developed a classification system of the vehicle's front and back using two descriptors HOG and LBP with two classifiers SVM and kNN. The system achieved an accuracy of 97,47%.

III. System Overview

This section provides an overview of the overall system, which is summarized in Fig. 2. The system has two phases. The first phase is to recognize the view of the vehicle helping the second phase to recognize the vehicle category, including bus, car, motorcycle, and truck.

We provide above an overview of each phase. So, as we said, we used Machine Learning techniques to recognize vehicle views and categories. Machine Learning is a branch of Artificial Intelligence. It analyzes data to build a model by itself. There are three types of Machine Learning: Supervised Learning, Unsupervised Learning and Reinforcement Learning [19]. The problem in each phase is classified under Supervised Learning, in particular, in the classification category.

The next section presents in detail the vehicle view classification phase.

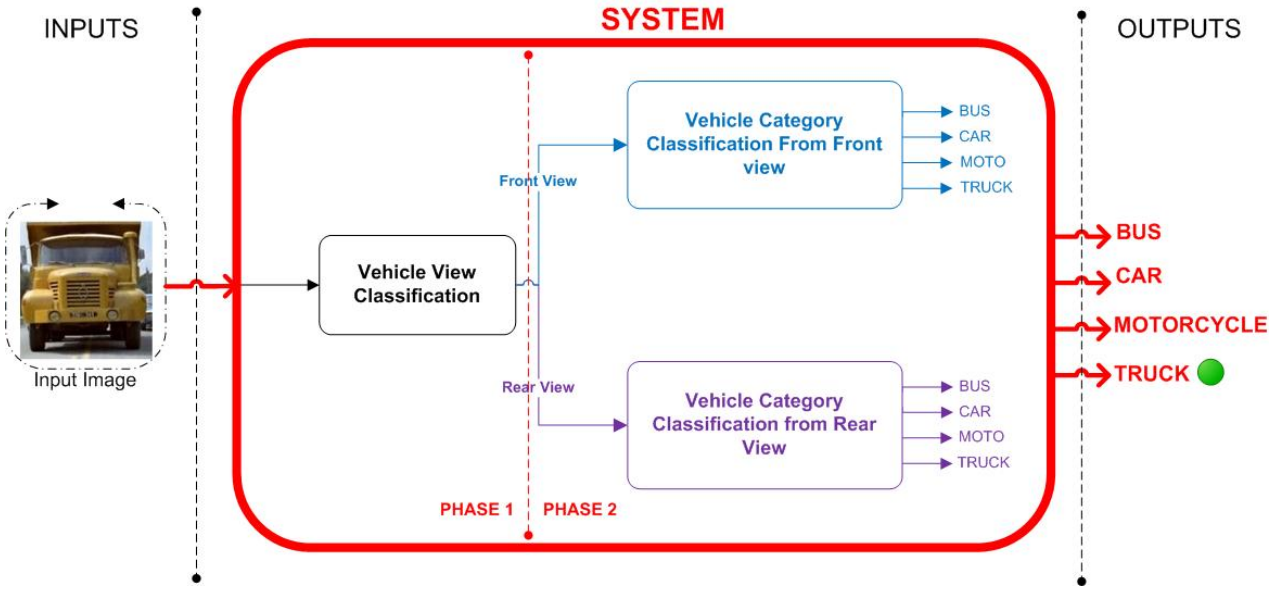


Fig. 2. Overall System.

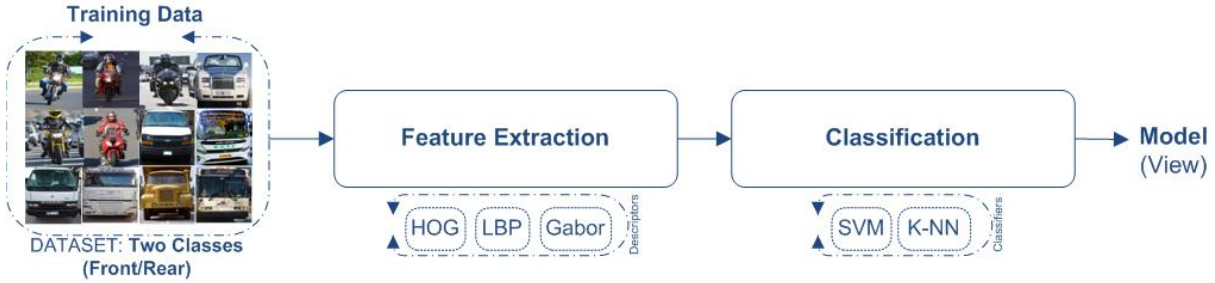


Fig. 3. Flow Chart for Building a Vehicle View Classification Model.

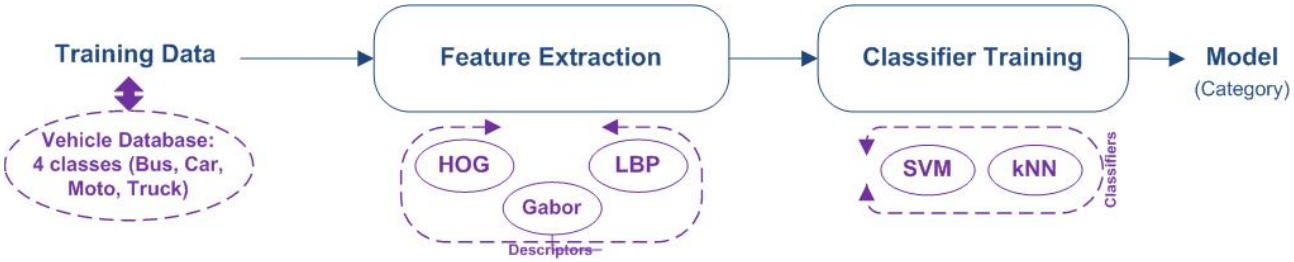


Fig. 4. Flow Chart for Building a Vehicle Category Classification Model.

A. *Vehicle View Classification*

Several factors make the recognition of rear and front views very difficult, including the similarity of shape, size, and color [2]. In this step, we used Machine Learning techniques including traditional algorithms and Convolutional Neural Networks. Fig. 3 shows all the steps involved in building the model from the data. Here, we used a dataset containing two classes (Front/Rear views). Several features are computed to characterize the shape of the vehicle view using descriptors like HOG, LBP or Gabor. For traditional ML techniques, features are extracted manually however CNNs extract features automatically. The feature vector associated with each image enters the classifier such as SVM or k-NN. We described in detail all descriptors and classifiers we used, in Section 4.

B. *Vehicle Category Classification*

This step aims to fit a model classifying the vehicle categories from the front view and another for rear view. So here, we used two databases. Each database contains the images of one view with four classes in which each class contains the images of a category. As explained before, building a model requires two steps (Fig. 4): feature extraction and classification. We extract features from the training data using three different kinds of descriptors: LBP, HOG and Gabor filter. Then, we train a classifier using these features to understand how the given input variables relate to the class, here we used, SVM and k-NN classifiers. We can call the process of constructing a model the training, learning or modeling step.

The following section presents all the methods we used.

IV. METHODOLOGY

In this section, we present all the methods used to resolve the vehicle category classification and view recognition problems. This section is divided into three parts. The first part presents the descriptors (HOG, LBP, and Gabor). The second part describes the classifiers (SVM, k-NN). The third part defines Convolutional Neural Networks in detail.

A. Descriptors

The key to efficient vehicle classification is to select a good feature descriptor that can characterize the vehicle view shape. As we said, we used three types of descriptors. These descriptors are summarized as it's written below.

1) Histogram of Oriented Gradients (HOG): HOG is proposed by Dalal et al. [20]. The HOG feature extraction process is based on the pixel gradient. The gradient corresponds to the first derivative of the image along the vertical and horizontal axes [21]. Features are calculated as follows: the gradient value and gradient direction value ("(1)" & "(2)") of each pixel I(x,y) of a cell are calculated and then grouped into a 9-bins [3] as shown in Fig. 5. Cells are grouped in blocks and then all vectors of all blocks are combined in series to obtain the final feature vector for classification [22].

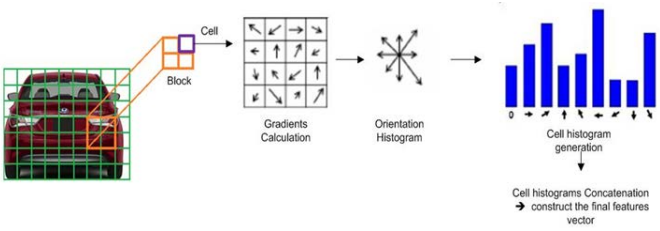


Fig. 5. HOG Features [4].

$$G(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \quad (1)$$

$$\theta(x,y) = \arctan\left(\frac{G_y(x,y)}{G_x(x,y)}\right) \quad (2)$$

$G_x(x,y)$ is the horizontal gradient "(3)" and $G_y(x,y)$ is the vertical gradient "(4)".

$$G_x(x,y) = I(x,y) - I(x-1,y) \quad (3)$$

$$G_y(x,y) = I(x,y) - I(x,y-1) \quad (4)$$

As we can see, the orientation histogram varies from 0 to 180° (20° per bin) [3].

A cell is composed of n × n pixels, with the increase of n, the number of features in the image decreases [22].

2) Local Binary Pattern (LBP): The LBP is a texture descriptor. Its general purpose is to create a feature based on order for each pixel by comparing its intensity value with that of its neighboring pixels [23].

The LBP feature vector is created using the following process:

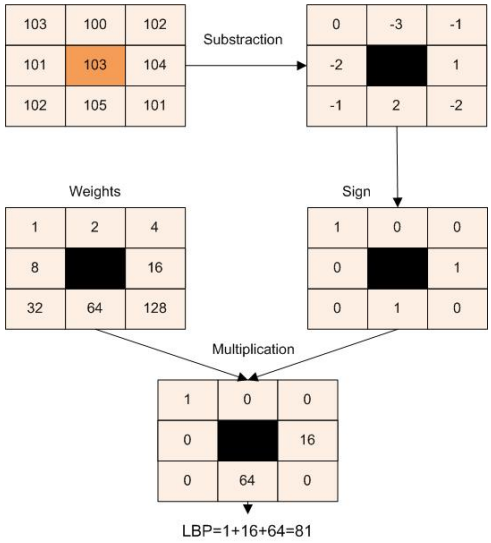


Fig. 6. LBP Operator [25].

The image is divided into cells of size 3x3. LBP labels the center pixel of each cell with a binary number called LBP code. This code is calculated as described in Fig. 6. The central pixel p_c is subtracted from the 8 neighboring pixels p_i (3 × 3 neighborhood, $i = 0, 1, ..., 7$). If the result is negative, the neighboring pixel is coded with "0". If it is positive, it is coded with "1", as in the following formula:

$$s(p_i - p_c) = \begin{cases} 1, if\ p_i \geq p_c \\ 0, if\ p_i < p_c \end{cases} \quad (5)$$

Where p_i and p_c are the grayscale values of the neighbor and center pixels, respectively.

The result code is an 8-bit number, so it is converted into a decimal value (Label), as the following equation shows [24]:

$$LBP(p_c) = \sum_{i=0}^{7} s(p_i - p_c).\, 2^i \quad (6)$$

In addition, the LBP descriptor is calculated in its general form as follows [24]:

$$\text{LBP}(p_c) = \sum_{i=0}^{M-1} s(p_i - p_c).\, 2^i,\ s(d) = \begin{cases} 1, if\ d \geq 0 \\ 0, \quad else \end{cases} \quad (7)$$

where p_c is the gray level value of the central pixel and p_i is the values of the M neighboring pixels.

Then, the obtained decimal values are used to construct the histogram for each cell by calculating the frequencies of the obtained values of all pixels. This histogram is considered as a feature vector with $256 = 2^8$ dimensions. The histograms of all the cells are concatenated to give a feature vector for the whole image.

3) Gabor features: The Gabor Filter (named after the English physicist Dennis Gabor) is a linear filter used for texture analysis, edge detection, feature extraction, etc. It is defined by the product of a complex sinusoid (known as the carrier) and a Gaussian function (known as the envelope) [26]. Then, in a spatial domain of dimension 2 (if it is an image), the Gabor function is presented as follows:

$$G(x,y) = s(x,y).g(x,y) \quad (8)$$

with *s(x,y)* is the complex sinusoid. It is defined as follows:

$$s(x,y) = \exp(2j\pi.(u_0.x + v_0.y) + \varphi) \quad (9)$$

(u_0, v_0) and φ define the spatial frequencies and phase of the sinusoid, respectively. The real and imaginary parts of this sinusoid are [26] :

$$Re\big(s(x,y)\big) = \cos(2\pi(u_0.x + v_0.y) + \varphi) \quad (10)$$

$$Im\big(s(x,y)\big) = \sin(2\pi(u_0.x + v_0.y) + \varphi) \quad (11)$$

g(x,y) is the Gaussian function. Its formulation is given by :

$$g(x,y) = \exp\left(-\left(\frac{(x-x_0)^2}{\sigma_x^2} + \frac{(y-y_0)^2}{\sigma_y^2}\right)\right) \quad (12)$$

where *(x0,y0)* is the peak Gaussian envelope *g*, σ_x (respectively σ_y) is the standard deviation of *g* from the *x*-axis (resp. *y*-axis) [26].

The Gabor function can be represented by a real component "(13)" and an imaginary component "(14)" whose directions are perpendicular (phase shift of 90 degrees).

$$Gr(x,y) = \cos(2\pi(u_0.x + v_0.y) + \varphi).\exp\left(-\left(\frac{(x-x_0)^2}{\sigma_x^2} + \frac{(y-y_0)^2}{\sigma_y^2}\right)\right) \quad (13)$$

$$Gi(x,y) = \sin(2\pi(u_0.x + v_0.y) + \varphi).\exp\left(-\left(\frac{(x-x_0)^2}{\sigma_x^2} + \frac{(y-y_0)^2}{\sigma_y^2}\right)\right) \quad (14)$$

This Gabor function is applied to a convolution mask, to define a convolution filter called Gabor filter. The application of this filter *G* to an image *I* is therefore the convolution of the image with the Gabor mask or kernel *N*, as shown in the following formula:

$$G(I) = I * N \quad (15)$$

where "*" is the convolution operator.

B. Classifiers

In the classification stage, we used two types of classifiers SVM and k-NN. We describe each classifier below:

1) Support Vector Machine (SVM): SVM is a classifier proposed by Cortes and Vapnik in 1995 [27], their main idea was separating the training data by a hyperplane without error. SVM classification is essentially a binary classification technique with category labels having only two values, 1 and -1 respectively. It is modified to handle multiclass tasks in real-situations. Two methods are proposed for this adaptation include techniques "One Against One" 1A1 and "One Against All" 1AA [28]. The 1AA approach represents the oldest and most common SVM multiclass approach [29] and involves the division of an N class dataset into two classes. In contrast, the 1A1 approach involves constructing a machine for each pair of classes, giving N(N-1)/2 machines. When applied to a test point, each classification gives a vote to the winning class and the point is tagged with the class with the most votes. This approach can be further modified to weight the voting process. According to the theory of machine learning, it is accepted that the 1AA approach has more disadvantages than 1A1; its performance can be compromised due to unbalanced learning data sets. However, the 1A1 approach requires more computation because the results of the SVM pairs have to be calculated [28].

2) k-Nearest Neighbor (kNN): K-Nearest Neighbor (kNN) is the extension of the minimum distance and the nearest neighbor classifiers. It calculates the distance between the test sample *x* and all the training samples and defines the distances in decreasing order. Then, it selects the *k* training samples closest to the test sample *x* and counts the category of *k* selected training samples. The test sample belongs to the category with the highest number of votes in the same category [30] [31].

3) Convolutional neural networks: As known, Deep Learning has recently attracted a great deal of interest especially in the field of computer vision and in particular image classification. Briefly, Deep Learning is a machine learning technique that uses deep neural networks [19]. It refers to Artificial Neural Networks (ANN) with multiple layers [14]. This section briefly describes the Convolutional Neural Network (CNN), one of the most popular deep neural networks. It is specialized in image recognition. Its name means that the network uses a linear mathematical operation called convolution.

In fact, CNN is an old technique that was developed in the 1980s and 1990s. However, it has been forgotten for some time because it was not practical for real-world applications [19]. There was no GPU to help training, and even CPUs were slow. Increasingly, data was becoming more and more available and accessible to people by making the databases public like ImageNet, CIFAR and MNIST. And CPUs were getting faster, and GPUs became a multipurpose computing tool. Both data and computing power made the CNN revived in 2012.

Using directly the original images for recognition gives poor results. Thus, feature maps are provided instead of the original images. As illustrated in Fig. 7, CNN consists of a feature extractor and a classification neural network. The feature extractor consists of pairs of convolution and pooling layers. The output from the convolution layer passes through an activation function like ReLU, Sigmoid or Tanh. The classifier is a fully connected neural network consisting of at least one layer. The final results of this part are transformed into a one-dimensional vector and then enter into the classifier network that generates the output [19].

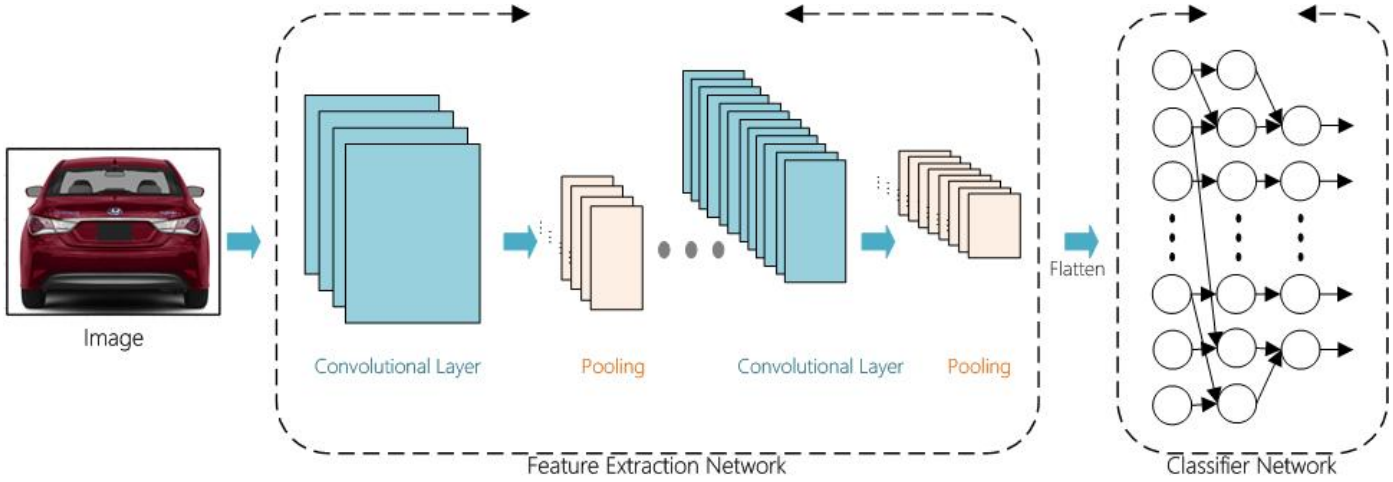


Fig. 7. Typical CNN Architecture.

The convolutional layer converts the images via the convolution operation, and the pooling layer reduces the dimensions of the image. It combines the neighboring pixels into a single pixel [19]. We will define these two layers in detail in the following sections.

- Convolution Layer

The convolution layer generates new images called feature maps using convolution filters. The feature map is a 2D grid of features; it highlights the features of the original image. The number of feature maps is identical to the number of convolution filters. The filters are two-dimensional matrices whose elements are the hidden weights. As shown in Fig. 8, the filter starts with the upper-left region. Each region is a submatrix which has the same size as the convolutional filter.

Thus, for each submatrix, the convolution result is the sum of the products of values having the same position with the filter kernel values [19]. In the same way, the filter slides across the image from left to right and top to bottom until the feature map is produced. The feature map dimension is presented as follows:

$$f = \frac{(D-K)}{S} + 1 \quad (16)$$

D and K are the dimensions of the image and filter respectively. S is the stride which is the number of pixels by which the filter is allowed to slide on the image.

- Activation function

It is necessary to use Activation functions in CNNs and artificial neural networks. Without them, the CNNs would be just a series of linear operations. An activation function is a non-linear transformation of the output of a neuron in a layer, like ReLU, Tanh, and Sigmoid [32].

a) Rectified Linear Unit (ReLU): ReLU is the most used activation function nowadays. It is presented by the equation "(17)". As shown in Fig. 9, when the input is equal to or greater than zero, the output is identical to the input. When it is less than zero, the output is zero [32].

$$f(x) = \max(0, x) \quad (17)$$

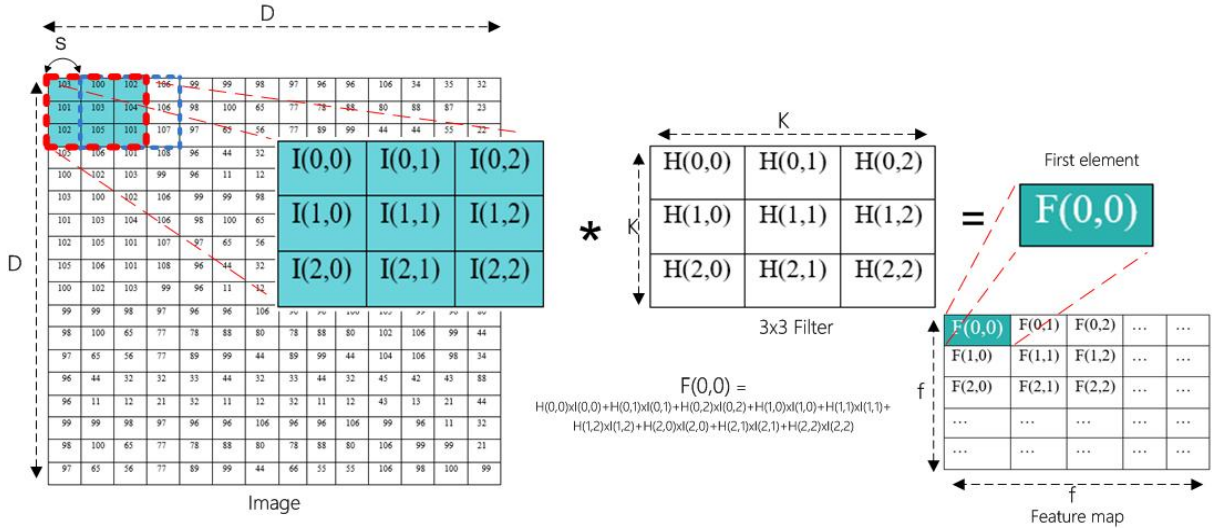


Fig. 8. Convolution Operation.

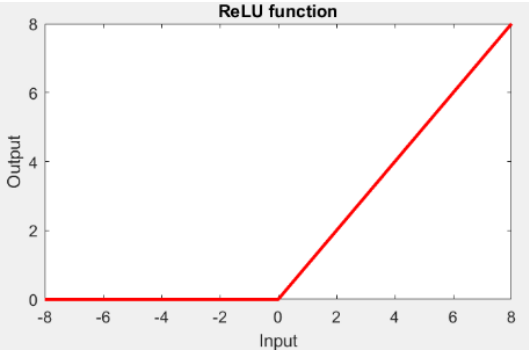


Fig. 9. Rectified Linear Unit.

b) Sigmoid: The Sigmoid or logistic function is presented by the equation "(18)" [32]. As shown in Fig. 10, Sigmoid has the output between the values of 0 and 1.

$$f(x) = \frac{1}{1+e^{-x}} \quad (18)$$

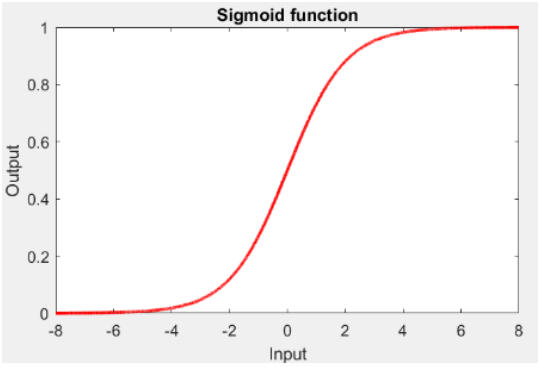


Fig. 10. Sigmoid Function.

Recently, it has lost favor because of several disadvantages, such as the problem of vanishing gradient.

c) Tanh: The hyperbolic tangent function (Tanh) is presented by the "(19)" [32]. As we see in Fig. 11, Tanh is very similar to the sigmoid function, it centered around 0. Tanh has the output between the -1 and 1.

$$f(x) = \frac{1-e^{-2x}}{1+e^{-2x}} \quad (19)$$

In practice, Tanh is usually preferred to the sigmoid, but it still suffers from the problem of the vanishing gradient.

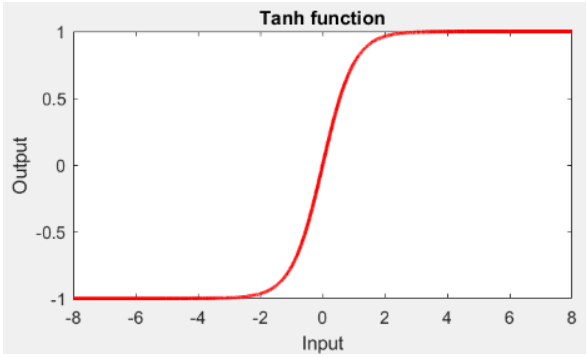


Fig. 11. Hyperbolic Tangent.

- Pooling Layer

The pooling layer reduces the size of feature maps. It replaces neighboring pixels by their maximum or average value. Fig. 12 illustrates the two operations.

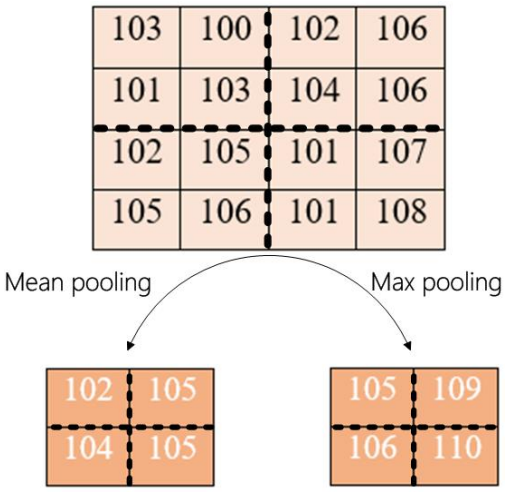


Fig. 12. Pooling Operations.

a) Mean Pooling: Mean pooling operation compresses a feature map by taking the average value of each block "(20)" [33]. Pooling is performed on non-overlapping blocks.

$$P_{i,x,y} = \frac{1}{d^2}\sum_{v,h} F_{i,(x+v),(y+h)} \quad (20)$$

$F_{i,x,y}$ is the value at the position *x,y* of the i^{th} feature map. *v* is a vertical index in the local neighborhood. *h* is a horizontal index in the local neighborhood. *d* is the dimension of the block.

b) Max pooling: Max-pooling operation is similar to the mean pooling, except that instead of taking the average, we take the max presented by the following equation "(21)" [33]:

$$P_{i,x,y} = \max_{v,h}(F_{i,(x+v),(y+h)}) \quad (21)$$

$F_{i,x,y}$ is the value at the position *x,y* of the i^{th} feature map. *v* is a vertical index in the local neighborhood. *h* is a horizontal index in the local neighborhood.

V. Result and Discussion

Various experiments are conducted to evaluate the performance of the methods we described below. In this section, we present the experimental results.

A. Dataset Construction & Implementation Details

In practice, several factors make the classification of vehicle categories very challenging. Among them: the unavailability of databases; none of the existing datasets are suitable for our work. We find only a few databases/top view or only databases for cars/side and rear views. Thus, we have collected, from many websites, thousands of images containing buses (Appendix/ Fig. 23 & Fig. 27), cars (Appendix/ Fig. 24 & Fig. 28), motorcycles (Appendix/ Fig. 25 & Fig. 29), and trucks (Appendix/ Fig. 26 & Fig. 30) from front and rear views. We built three datasets. The first is for the view

classification. It contains 4000 images with two classes: one for front view images and the other for rear view images. The two other datasets are used for the classification of vehicle categories: a dataset for front vehicle images (2182 images), the other for rear vehicle images (2000 images). Each of these two datasets contains four classes: each class for one category (500 images under each class).

All the images were resized to a fixed size of 150*150 pixels. From each dataset, two sets are created with a random split: a training set and a test set. Eighty percent of the images were included in the training set. The remaining 20% were included in the test set. These images are selected under different weather conditions.

We executed the algorithms on a Lenovo ThinkPad with a processor Intel® Core™ i5 7th Generation CPU @ 2.50GHz 2.71GHz, RAM 8Go. We implemented the algorithms in Matlab.

B. Performance Metrics

To evaluate the performance of the constructed models, we calculated many metrics using the confusion matrices extracted for each model. The rows and columns show the true and predicted classes, respectively. The values in the diagonal represent the correct classification rates while those outside the diagonal represent classification errors.

In this work, we have two problems: binary classification (Rear/ Front) and multi-class classification (Displayed in Fig.13) with four classes in the order of Bus, Car, Moto, and Truck. Thus, for a confusion matrix with more than two-classes, we calculated the metrics (True Positives TP, False Positives FP, False Negatives FN) of each class as follows:

- The TP for each class is the corresponding value in the main diagonal. For example, the TP of the Motorcycle class is: TPM.
- The FN for each class is the sum of the values of the corresponding row excluding the main diagonal element TP. For example, the FN of the Truck class is:

FNT = ERTB+ ERTC + ERTM (22)

- TP+FN=100%
- The FP for each class is the sum of the values of the corresponding column excluding the TP. For example, the FP of the Bus class is:

FPB = ERCB+ ERMB+ ERTB (23)

C. Experiments

This section aims to compare the constructed models to find which of these models provides significant results. The best models in each classification have been selected to create a robust system. To accomplish this, three experiments are conducted. The first experiment is devoted to the recognition of the vehicle view: back or front. The second experiment aims to classify the vehicle categories according to each view. In the third experiment, we created the global system; the categories were classified independently of the view.

In the learning stage, each model is trained on three subsets of training data using the resampling method 4-fold cross-validation. In total, we generated 12 models for each method making different prediction errors on test data. And then we calculated the average performance. This process leads to a more stable prediction than those of any individual member model.

Among the traditional ML algorithms, we have used k-NN. As explained before, to classify a sample the k-NN starts by finding all the closest k training samples, and then predict the class by majority vote [10]. So here in experiments, we simply chose k=1 and the distance metric=Euclidean. On the other hand, in front and rear vehicle classification, we used SVM with the linear kernel. For the category classification, we used the SVM multiclass approach “1A1”.

The features of HOG have been extracted from the front and rear vehicle images as shown in Fig. 14 [2]. The inputs are 150 × 150 color images. Firstly, we converted the color images into gray images. Every input RGB three-channel image is converted into a single-channel image, the transformation formula is as follows:

$$Gry = 0{,}3.R + 0{,}59.G + 0{,}11.B \quad (24)$$

The feature vector is obtained by using the histograms of each block. Here, each block is represented by 8x8 cells, and each cell is represented by 9 bins. Thus, each block is represented by the vector 8x8x9=576 features. By combining all the vectors of all the blocks in series, we obtain the final HOG feature vector for the whole image for use in the classification step.

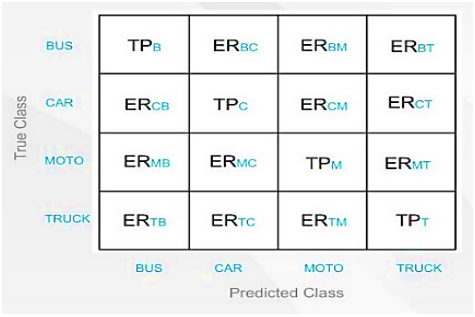


Fig. 13. Multi-Class Confusion Matrix (4 Classes).

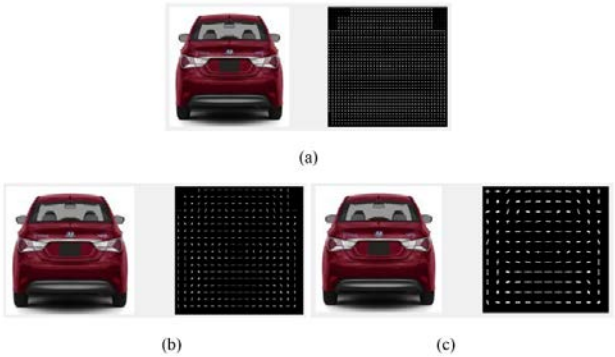


Fig. 14. (a). HOG Features of Rear-View Vehicle (Cell Size 4*4), (b). HOG Features (Cell Size 8*8).

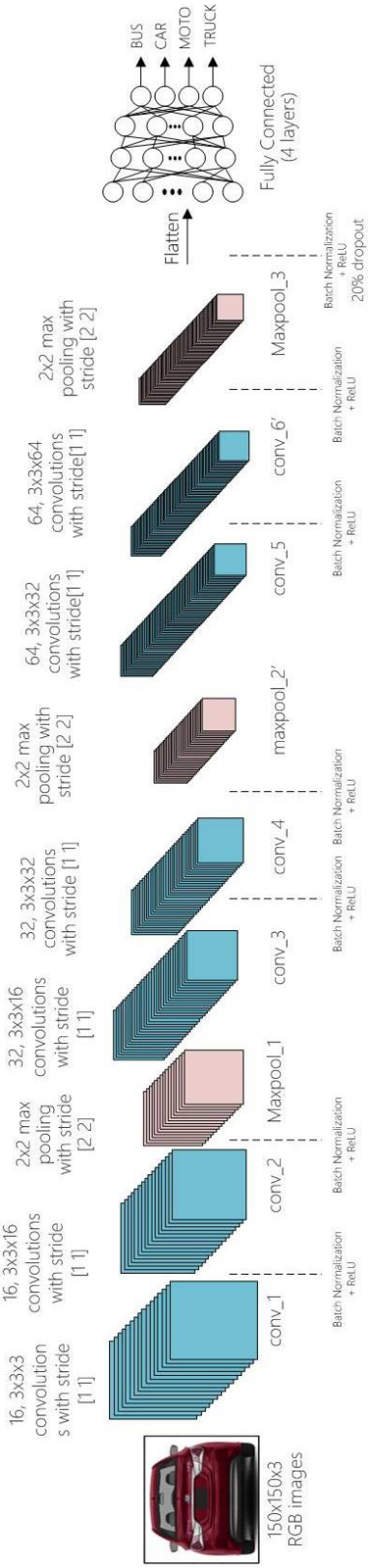


Fig. 15. CNN Architecture for Category Classification.

For CNNs, the input image of a convolution layer has a dimension D × D × c, where D is the height and width, and c is the number of channels. For RGB images, c = 3 [34]. In the experiments, we have D=150 and c=3. As shown in Fig. 15, the image passes through the convolution layer, the pooling layer, and then the fully connected layer. The network consists of six convolution layers using the ReLU activation function. The second and fourth convolution layers are followed by max-pooling layers. We add the Dropout layer at the end to regularize the network. In the classification network, we used a fully connected layer followed by a Softmax classifier. The size of the output layer is identical to the number of classes we expect.

CNNs automatically learn filter values (weights) by training the network on large amounts of data using a concept called backpropagation, continuing to have a minimal classification error. In the first layers, the network tries to recognize certain aspects of images, such as edges, shapes (squares, triangles, circles), and colors. More and more layers learn complex patterns until all of these patterns can help the network to learn the classification of the input [32].

1) Experiment 1 (Vehicle view classification): Fig. 16 presents a summary of the comparison of the obtained accuracy and error results. From the graph, we can easily see that CNN outperformed the other methods. It reached an accuracy of 94.29%, higher than LBP+SVM by a difference of 30.65%. We can also notice that HOG with SVM and k-NN obtained the same accuracy.

From Table I, we can notice that CNN produced 92.91% rear view images. But for the class/Front, the TP rate of HOG+kNN was higher than CNN (accuracy difference 1.01%).

When we trained the Convolutional Neural Networks, we monitored the training progress by plotting various metrics. Thus, we can determine if and how quickly the accuracy and error of the network are improving. Fig. 17 and Fig. 18 show these metrics at every iteration. Each iteration estimates the gradient and updates the network parameters.

Fig. 17 presents the training progress of one CNN model. It illustrates the evolution of training accuracy and its smoothed version according to the number of epochs. Each epoch is marked using a shaded background. An epoch is a full pass through the entire dataset.

We can observe that the accuracy increases progressively according to the number of epochs (total 24 epochs) to reach its best level. The model has become more generalized, especially from epoch 7. The graph shows also the classification accuracy across the entire validation set.

In the following Fig. 18, we show the curve evolution of the training loss and its smoothed version according to the number of epochs. As shown, the loss function decreases as the number of epochs increases.

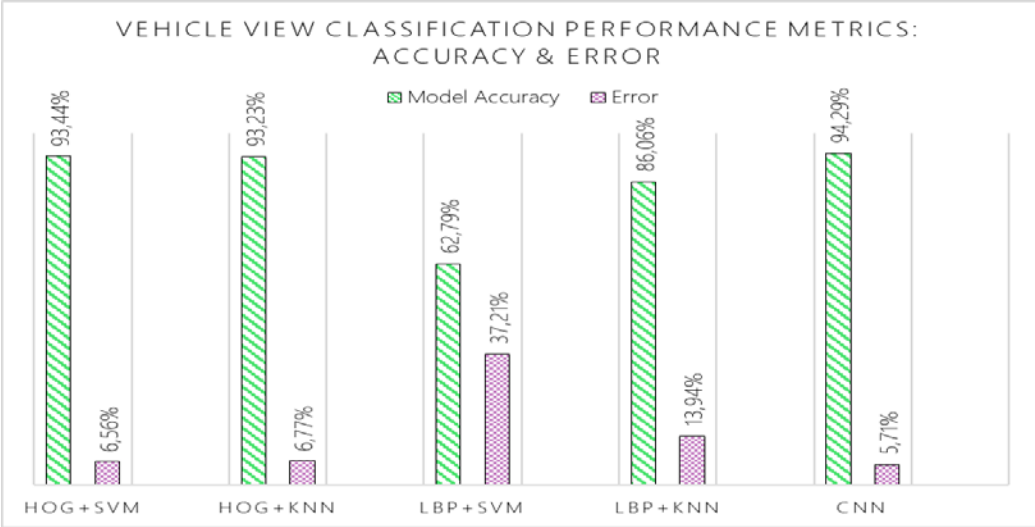


Fig. 16. Vehicle View Classification Accuracy.

TABLE I. VIEW CLASSIFICATION RESULTS

Models	Class : Rear		Class : Front	
	TP	FN	TP	FN
HOG+SVM	90.94%	9.06%	95.64%	4.06%
HOG+KNN	89.76%	10.24%	96.70%	3.30%
LBP SVM	32.68%	67.32%	92.89%	7.11%
LBP+KNN	85.83%	14.17%	86.29%	13.71%
CNN	92.91%	7.09%	95.69%	4.31%

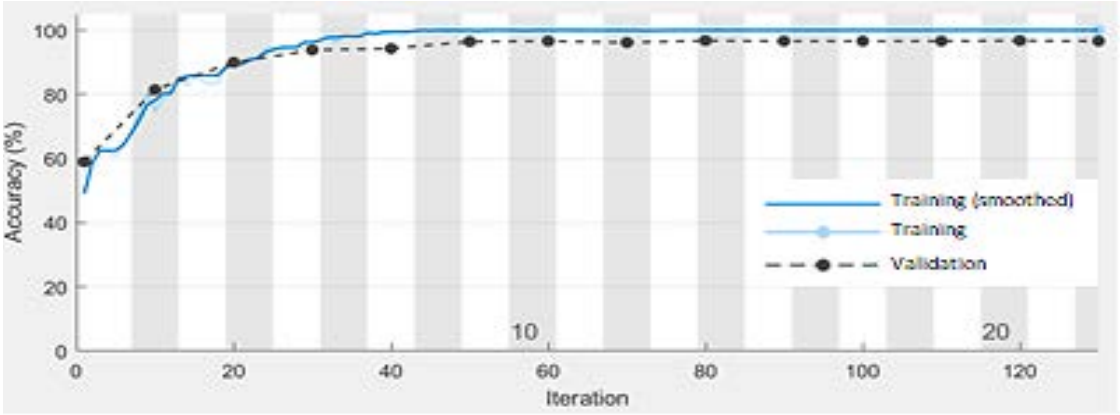


Fig. 17. The Evolution of Training Accuracy.

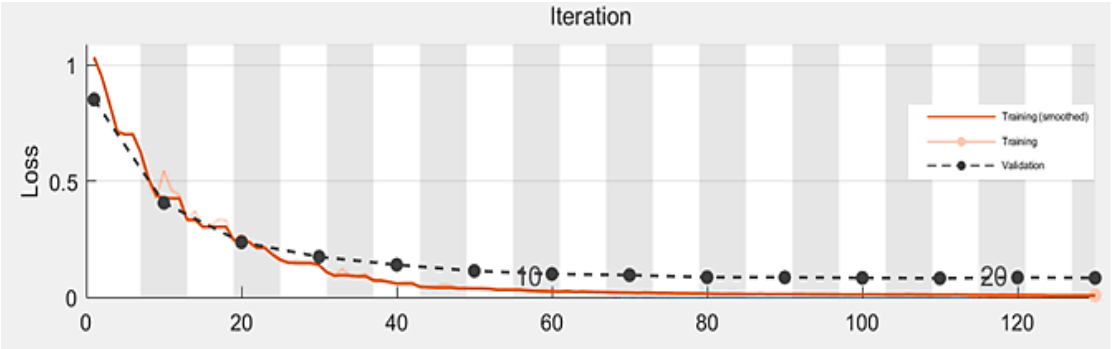


Fig. 18. Error Function.

During the execution of the algorithms, we observed that the CNNs take a very long time to be trained compared to the other models. If the number of layers increases, the training and testing time increase. On the other hand, the performance of the network becomes better [14]. In CNN, the most important layer is the convolution layer. Thus, this latter takes the most training time. Moreover, we have already worked on the classification of front and rear views but only for cars [2] (97.47% obtained by HOG+kNN). However, in this experiment, we classified the views of the four vehicle categories. This classification is more difficult, especially for motorcycles. Both views (front and rear) have the same shape.

2) Experiment 2 (Vehicle category classification)

a) Front view: In this experiment, we tested the algorithms with the front view dataset. The evaluation was based on the metrics we presented. The results were demonstrated in Fig. 19 and Table II.

Fig. 19 presents the overall accuracy and error of each model. Table II summarizes the comparative performance between all models; it shows the percentage of the accuracies and errors FP and FN of each category. We present the results of vehicle type classification basing on test data. Several observations can be made from this table. First, the combination of HOG and k-NN performs much better than the other combinations. It outperforms LBP and CNN. [3] also found that the accuracy of HOG features is higher than the convolutional neural network's accuracy (by 2,3%). Because HOG is a dense grid; it is used as low-level features, so it can extract richer information from the images [10]. Second, the Gabor filter is least efficient among all descriptors we tested. Gabor and LBP perform less efficiently with SVM, but when they are combined with k-NN, they provide better results. All descriptors work very well in combination with the k-NN classifier.

As shown in Fig. 19, the order of the best models, in this experiment, according to the accuracy is as follows: HOG+kNN, HOG+SVM, CNN, and LBP+kNN.

For each category, we calculated TP, FN, and FP as shown in Table II. We can see that all models (except Gabor+SVM and LBP+SVM) classified very well the three classes MOTO, CAR, and TRUCK. The MOTO class was obviously the easiest to be correctly classified.

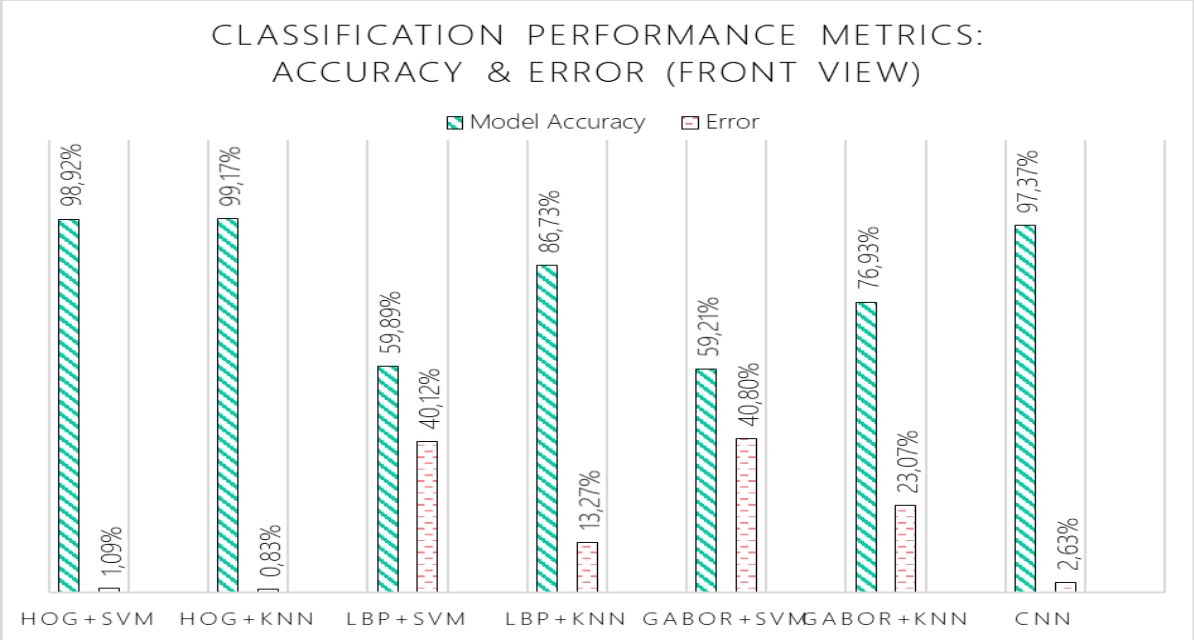


Fig. 19. The Accuracy and Error Obtained for Each Model (Front View).

TABLE II. CLASSIFICATION METRICS FOR EACH CATEGORY (FRONT VIEW)

Models	Class BUS			Class CAR			Class MOTORCYCLE			Class TRUCK		
	TP (%)	FN (%)	FP (%)	TP (%)	FN (%)	FP (%)	TP (%)	FN (%)	FP (%)	TP (%)	FN (%)	FP (%)
HOG+SVM	95.835	4.17	0	**99.8225**	0.1775	0	**100**	0	0	**100**	0	4.3475
HOG+kNN	**97.40**	2.60	0.26	**99.8225**	0.1775	0.257	**100**	0	0	**99.485**	0.515	2.7825
LBP+SVM	39.8475	60.16	7.99	95.1775	4.8175	127.04	78.75	21.2525	3.02	25.775	74.4	22.405
LBP+kNN	71.355	28.65	7.6705	92.8575	7.1425	16.89.5	94.585	5.42	4.115	88.145	11.8575	22.85
Gabor+SVM	70.05	29.9575	50.695	75	25	53.36	41.2525	58.7525	7.83	50.515	49.48	51.305
Gabor+kNN	63.5425	36.4625	16.6825	82.32	17.6775	33.9885	75	25	9.005	86.8575	13.1475	32.615
CNN	91.494	8.5049	0.402466	**99.2275**	0.772466	0.7741	**99.72166**	0.27833	0.0866	**99.0558**	0.9441	9.23663

The experimental results show that HOG+SVM produced errors $FN_{BUS} = 4.17\%$, $FN_{CAR} = 0.1775\%$ and $FP_{TRUCK} = 4.3475\%$. We observe that $FN_{BUS} + FN_{CAR} = FP_{TRUCK}$, this means that 4.17% of buses and 0.1775% of cars are classified as trucks. The same observation for the model HOG+kNN, 2.6% of buses and 0.1775% of cars are classified as trucks, but also 0.515% of trucks are classified as buses (0.26%) and cars (0.257%). This misclassification is due to the similarity of the appearance of these categories from the front view.

The next experiment aims at testing the same algorithms on the rear-view dataset.

b) Rear view: The second experiment was elaborated to verify the performance of the algorithms for vehicle category classification from the rear-view. Fig. 20 shows the classification performance of every constructed model. As observed from the figure, the models HOG+SVM, HOG+kNN and CNN, in the testing sets, achieved higher classification accuracies 99.58%, 99.47%, and 97.43%, respectively. The metrics of each class are listed in Table III. As concluded in the first experiment, HOG achieved the best results and k-NN outperforms SVM. For example, for the class BUS, when LBP is combined with SVM, it provided an accuracy of 9.09% whereas when it is combined with k-NN it provided an accuracy of 93.1825%. However, when k-NN and SVM are combined with HOG, they work slightly similar.

Table III shows the TP, FN, and FP per class.

In this experiment/both views, it is easy to see that the difficult category to classify was the BUS, especially from the front view. This is because most buses contain the same details and information as trucks from the front view compared to the rear-view. In both experiments, the HOG descriptor was the key to this classification. It delivered better performance in characterizing the shape and appearance of vehicles. HOG is considered as one of the most accurate feature descriptors for visual classification problems [35].

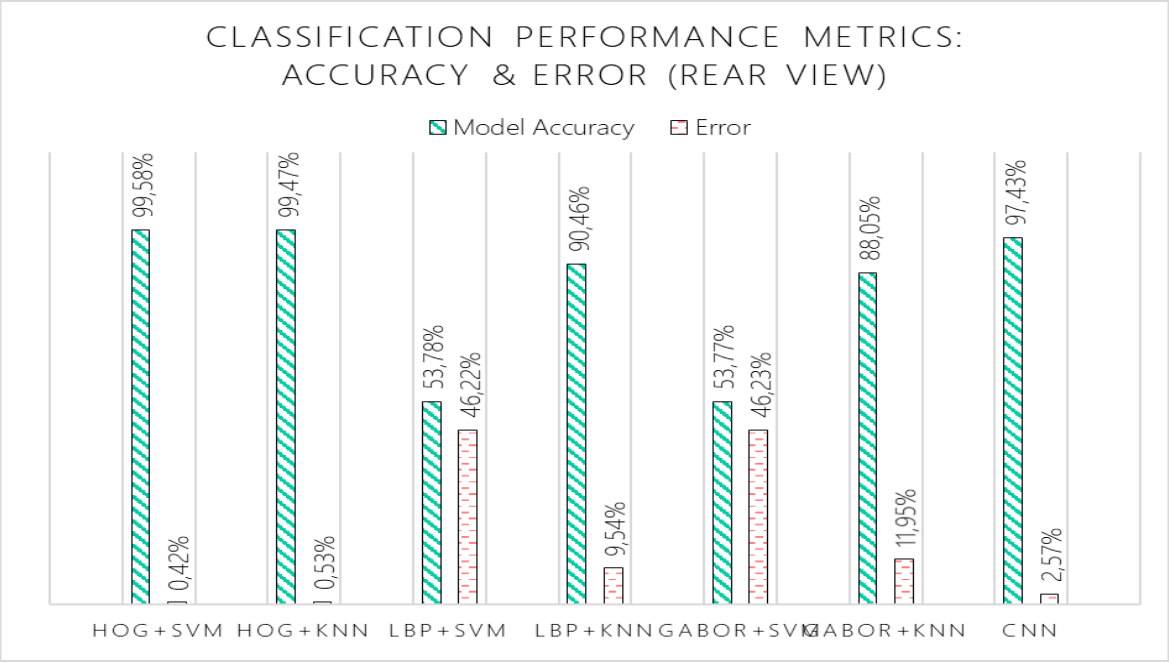


Fig. 20. The Accuracy and Error Obtained for Each Model (Rear View).

TABLE III. CLASSIFICATION METRICS FOR EACH CATEGORY (REAR VIEW)

Models	Class BUS			Class CAR			Class MOTORCYCLE			Class TRUCK		
	TP(%)	FN(%)	FP(%)	TP(%)	FN(%)	FP(%)	TP(%)	FN(%)	FP(%)	TP(%)	FN(%)	FP(%)
HOG+SVM	**99.4325**	0.5675	0.2875	**99.7125**	0.2875	0	**99.5825**	0.4175	0.425	**99.5975**	0.425	0.985
HOG+kNN	**98.295**	1.705	0.425	**100**	0	0	**100**	0	0	**99.5975**	0.425	1.705
LBP+SVM	9.09	90.9075	4.3175	95.1125	4.8875	135.01	63.335	36.6625	15.1725	47.5825	52.4225	30.38
LBP+kNN	93.1825	6.8175	11.195	93.675	6.325	10.495	96.25	3.75	3.58	90.7275	9.2925	4.6725
Gabor+SVM	17.0425%	82.955	8.975	85.9225	14.08%	97.22	50.4175	49.585	61.2725	61.6925	38.3075	17.46
Gabor+kNN	82.9525%	17.045	6.2675%	86.21%	13.8	21.445	86.665	13.3325	13.545	96.3725	3.6275	6.5475
CNN	96.0216%	3.9775%	4.04716%	96.8375%	3.16232	1.923326	**99.5825**	0.41746	0.47916	**97.3125**	2.687166	3.7948

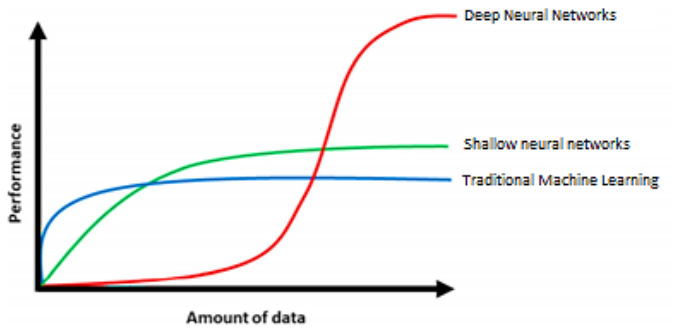


Fig. 21. Impact of Data Availability on the Algorithms [36].

CNN outperformed other methods in the vehicle view classification experiment. However, it was surpassed by HOG with k-NN and SVM in the vehicle category classification experiment. This is due to the impact of data availability. In the first experiment, we used a larger database than in the second one. Fig. 21 illustrates, in general, this impact on the performance of traditional machine learning methods and neural networks. Traditional methods reach better results with small amounts of data. However, deep learning can outperform all other methods with a large amount of data [36].

It was difficult to compare our results with those of other works on vehicle category recognition since no reference image set contains the categories from their front and rear views. And also, most of the works did not use the same methods we used. However, the results of previous works are as follows (Table IV). [8] achieved an accuracy of 96.26%. The author in [1] achieved accuracy of 93.90% using CNNs on vehicle frontal-view images. In [5], the CNN model achieved an accuracy of 97.88% on rear view images. In [3] achieved an accuracy of 92,6% on the BIT vehicle dataset. In [18], the author used the same dataset and reached 96.1% in daytime conditions and 89.4% in nighttime conditions. Unfortunately, this dataset has been removed from the website [37].

TABLE IV. COMPARISON OF VEHICLE CATEGORY CLASSIFICATION RESULTS WITH THE EXISTING WORKS

Study	View	Approach	Performance
[8]	Side	SVM and Random Forest	96.26%
[1]	Front	CNN	93.90%
[5]	Rear	CNN	97.88%
[3]	Front	Improved HOG_SVM	92,6%
[18]	Front	Semi-supervised CNN	96.1% / daytime 89.4% / nighttime
Proposed	Front and Rear	HOG, LBP, Gabor, SVM, k-NN, CNN	99.58% / Rear 99.17% / Front

3) Experiment 3 (Overall system): After building the models, we chose the best ones to create the global system (Fig. 22) in order to know which category an input image belongs to.

Experimental results demonstrated that CNN model achieved the best performance to recognize vehicle views, the HOG+kNN model has the highest accuracy in classifying vehicle categories from the front view, and the HOG+SVM model for the rear-view.

To evaluate the performance of the system, we tested it with a list of 209 images (different from those used in the datasets). This evaluation was based on many metrics: overall accuracy, accuracy for each view, runtime for each image. The evaluation results are presented in Table V. As shown, the system achieved an accuracy of 95.7746% and a runtime of 12.46 seconds per image.

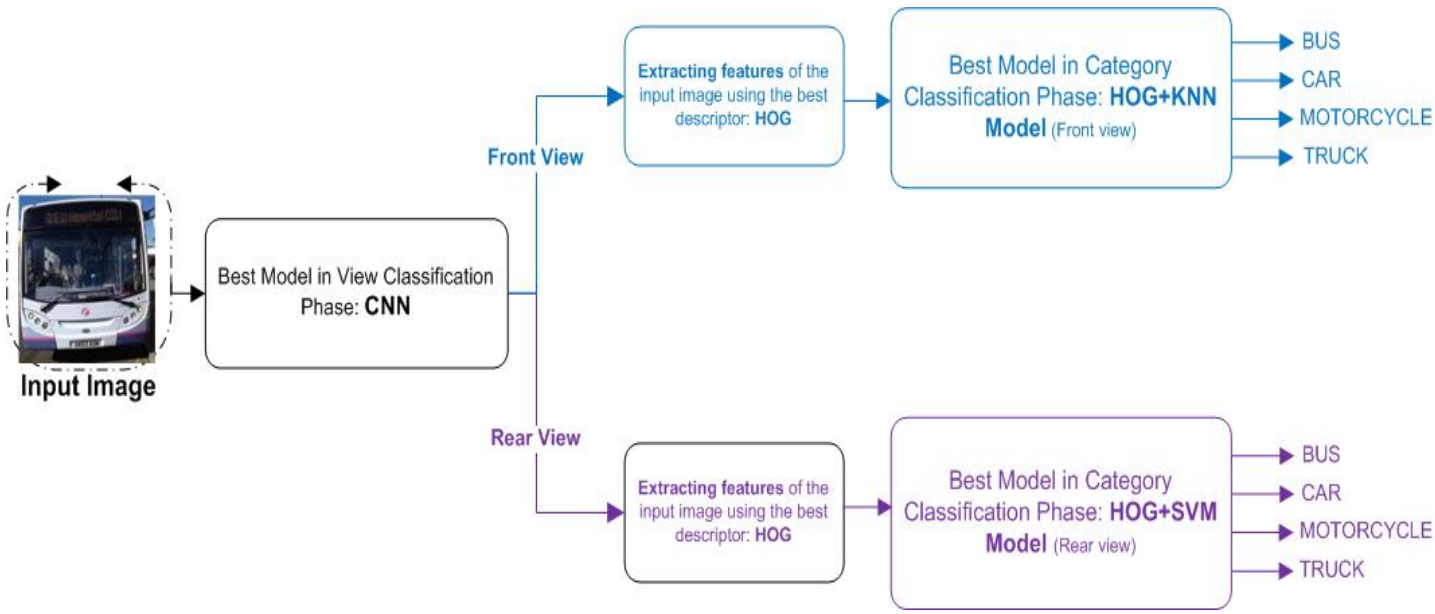


Fig. 22. System Flowchart.

TABLE V. SYSTEM PERFORMANCE METRICS

	Overall Accuracy		**Error**		**Time /Image (Second)**
System	**95.7746%**	Front : 98.29% Rear : 92.5%	4.22%	Front : 1.709% Rear : 7.5%	12.46

VI. CONCLUSION

In this paper, we have proposed a view-independent classification system for vehicle categories (Bus, Car, Motorcycle, and Truck) using traditional machine learning algorithms and CNNs. The system uses a two-phase approach. The first phase is used to recognize the vehicle view. The second recognizes the categories. To evaluate the built models, we used the three databases we built ourselves. The first dataset includes 4000 images, the second contains 2182 front vehicle images and the third contains 2000 rear vehicle images. The results showed that CNN provides the highest accuracy with the lowest error rate in the view classification step. But at the vehicle category classification step, the HOG features were efficient compared to Gabor and LBP features. they characterized well the vehicle's view orientations, making it more resistant to geometric and lighting variations. In fact, in this classification, it was easy to see that Motorcycle class was the easiest to be correctly classified with an accuracy of 100% / for both views. However, buses and trucks were misclassified due to their similar shapes. HOG achieved an accuracy of 99,58% with SVM, for the rear-view and 99,17% with k-NN for the front view. Finally, the overall system achieved an accuracy of 95,7746%. We can notice that, in classification, two main factors affect the performance of the system: the characteristics selected and the availability of data. The results we have obtained show that the system can be successfully exploited for many applications.

In future work, we will train CNNs with larger datasets, and detect and classify vehicles in traffic videos.

REFERENCES

[1] B., Hicham, A. Ahmed, M. Mohammed (2018). Vehicle Type Classification Using Convolutional Neural Network. *2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*, 313-316. Doi: 10.1109/CIST.2018.8596500.

[2] S. Baghdadi, N. Aboutabit, Front and Rear Vehicle Classification, *Advanced Intelligent Systems for Sustainable Development (AI2SD'2019)*, vol.4, no.142. doi:10.1007/978-3-030-36674-2_28.

[3] Ge., Penghua and Hu., Yanping, Vehicle Type Classification based on Improved HOG_SVM. *Proceeding . 3rd International Conference on Mechatronics Engineering and Information Technology (ICMEIT 2019)*. Doi: 10.2991/icmeit-19.2019.102.

[4] J., Lian, J., Zhang, T., Gan and S., Jiang. Vehicle Type Classification using Hierarchical Classifiers. *Journal of Physics: Conference Series*. 2018. Vo.1069.doi: 10.1088/1742-6596/1069/1/012099.

[5] Y. Chen, W. Zhu, D. Yao and L. Zhang, "Vehicle type classification based on convolutional neural network," *2017 Chinese Automation Congress (CAC)*, Jinan, 2017, pp. 1898-1901. doi: 10.1109/CAC.2017.8243078.

[6] N., Roecker, Max, M. G., Yandre Costa, L. R., João Almeida and H. G., Gustavo Matsushita. "Automatic Vehicle type Classification with Convolutional Neural Networks." *2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP)* (2018): 1-5.

[7] S. Kul, S. Eken and A. Sayar, "A concise review on vehicle detection and classification," *2017 International Conference on Engineering and Technology (ICET)*, Antalya, 2017, pp. 1-4. doi: 10.1109/ICEngTechnol.2017.8308199.

[8] Z. Chen, T. Ellis and S. A. Velastin "Vehicle type categorization: A comparison of classification schemes", *14th IEEE Annual Conference on Intelligent Transportation Systems*, the George Washington University, Washington, DC, USA. pp. 74-79, Oct. 5-7, 2011.

[9] Y. C. Wang, C. C. Han, C. T. Hsieh, and K. C. Fan, Vehicle type classification from surveillance videos on urban roads, in Ubi-Media Computing and Workshops (UMEDIA), 2014 7th International Conference on, 2014, pp. 266–270.

[10] Z., bailing and Z., Yifan. Vehicle type and make recognition by combined features and rotation forest ensemble. *International Journal of Pattern Recognition and Artificial Intelligence*. Vol. 26, No. 03, 1250004 (2012). Doi : 10.1142/S0218001412500048.

[11] M. A. Manzoor, Y. Morgan ,"Vehicle Make and Model Classification System using Bag of SIFT Features", 7th IEEE Annual Conference on Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA. pp. 572 577, 02 March 2017.

[12] A.H.S. Lai , G.S.K. Fung , N.H.C. Yung, "Vehicle Type Classification from Visual-Based Dimension Estimation", Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE, Oakland, CA, USA, pp. 201-206, 25-29 Aug. 2001.

[13] D. Kleyko, R. Hostettler, W. Birk, E. Osipov, "Comparison of Machine Learning Techniques for Vehicle Classification Using Road Side Sensors", 2015 IEEE 18th Int. Conf. Intell. Transp. Syst., pp. 572-577, 2015.

[14] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET)*, Antalya, 2017, pp. 1-6. doi: 10.1109/ICEngTechnol.2017.8308186.

[15] B. Selbes and M. Sert, Multimodal vehicle type classification using convolutional neural network and statistical representations of MFCC. *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, (2017) 1-6. Doi: 10.1109/AVSS.2017.8078514.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 2015.

[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, August 2015. doi:10.1109/TITS.2015.2402438..

[19] K., Phi, book, Matlab Deep Learning: with Machine learning, neural network and artificial intelligence. 2017. ISBN-13 (pbk): 978-1-4842-2844-9. ISBN-13 (electronic): 978-1-4842-2845-6. DOI 10.1007/978-1-4842-2845-6.

[20] N., Dalal, B., Triggs. Histograms of oriented gradients for human detection, in: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 886–893. 2005.

[21] R., Scherer. Computer Vision Methods for Fast Image Classification and Retrieval. 2020. ISSN 1860-949X. ISSN 1860-9503 (electronic). https://doi.org/10.1007/978-3-030-12195-2.

[22] Y., Sinha, P., Jain, N. Kasliwal: Comparative Study of Preprocessing and Classification Methods in Character Recognition of Natural Scene Images. In: Machine Intelligence and Signal Processing. Advances in Intelligent Systems and Computing, vol. 390. Springer, New Delhi (2016). doi: 10.1007/978-81-322-2625-3_11.

[23] A. I. Awad and M. Hassaballah, Image Feature Detectors and Descriptors, Foundations and Applications, Switzerland: Springer, 2016.

[24] M., Bereta, W., Pedrycz, M., Reformat. Local descriptors and similarity measures for frontal face recognition: A comparative analysis. *J. Visual Communication and Image Representation*, vol. 24, pp. 1213-1231. 2013. doi: 10.1016/j.jvcir.2013.08.004.

[25] S., Baghdadi, N., Aboutabit: 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 .2019. doi: 10.18178/ijmlc.2019.9.6.863.

[26] J. R. Movellan, "Tutorial on Gabor Filters".

[27] T.-J., Alhindi, S., Kalra, K.-H., Ng, A., Afrin, H.-R., Tizhoosh: Comparing LBP, HOG and deep features for classification of histopathology images. In: International Joint Conference on Neural Networks (IJCNN) (2018). doi: 10.1109/IJCNN.2018.8489329.

[28] C.-W., Hsu and C.-J., Lin. A Comparison of Methods for Multi-class Support Vector Machines. *IEEE Transactions on Neural Networks* 13(2):415-25. 2002. doi: 10.1109/72.991427.

[29] F., Melgani, and L., Bruzzone. Classification of hyperspectral remote sensing images with Support Vector Machines. IEEE Transactions on Geoscience and Remote Sensing, 42, 1778 – 1790. 2004.

[30] S., Baghdadi, N., Aboutabit: A Comparative Study of Vehicle Detection Methods, Advanced Intelligent Systems for Sustainable Development (AI2SD2018), Springer, vol. 5, pp. 111 (2019). doi: 10.1007/978-3-030-11928-7_83.

[31] Sh., Lin, Ch., Zhao, and X., Qi: Comparative Analysis of Several Feature Extraction Methods in Vehicle Brand Recognition. In: 10th International Conference on Sensing Technology (ICST) (2016). doi: 10.1109/ICSensT.2016.7796337.

[32] M., Salvaris, D., Dean, W., Hyong Tok. *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.

[33] K., Jarrett, K., Kavukcuoglu, M.A., Ranzato, and Y., LeCun. What is the best multi-stage architecture for object recognition? 2009 IEEE 12th International Conference on Computer Vision. doi:10.1109/iccv.2009.5459469.

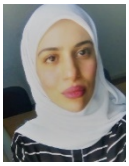
[34] F., Özyurt, T., Tuncer, E., Avci, *et al.* A Novel Liver Image Classification Method Using Perceptual Hash-Based Convolutional Neural Network. *Arab J Sci Eng* 44, 3173–3182 (2019). https://doi.org/10.1007/s13369-018-3454-1.

[35] L., Xuesong, J., Jo, S., Youngbo and D., Stantic. “Detection and Classification of Vehicle Types from Moving Backgrounds.” *RiTA* (2017).

[36] P. Savadjiev, J. Chong, A. Dohan, M. Vakalopoulou, C. Reinhold, N. Paragios, B. Gallix. Demystification of AI-driven medical image interpretation: past, present and future.

[37] http://iitlab.bit.edu.cn/mcislab/vehicledb/.

AUTHORS’ PROFILE



Sara Baghdadi is a Ph.D. Student at the National School of Applied Sciences in Khouribga (Sultan Moulay Slimane University, Morocco). She received her engineering diploma in Networking and Telecommunications in 2015 from the same school.

Her main research interest includes Machine Learning and Image Processing.

Noureddine Aboutabit is actually an Associate Professor in Telecommunications and multimedia processing at the National School of Applied Sciences, Kh-ouribga (Sultan Moulay Slimane University, Morocco) since October 2011. He teaches in the Department of Networking and Telecommunication primarily in the areas of computer vision, signal processing and image processing. In 2007, he received his Ph. D. degree in Signal Image Speech Telecom from Grenoble INP (France). He received his M.S. degree in 2004 from the same institute. In 2003, he obtained his engineering diploma from Ecole Normale Suprieure d’Ingnieurs Electriciens de Grenoble (ENSIEG).

His current research interests include computer vision, machine learning, artificial intelligence, and multimedia processing.

APPENDIX

Some samples of the datasets containing four types: Bus, Car, Motorcycle, and Truck.

Fig. 23. Samples from Bus Class (Front View).

Fig. 24. Samples from Car Class (Front View).

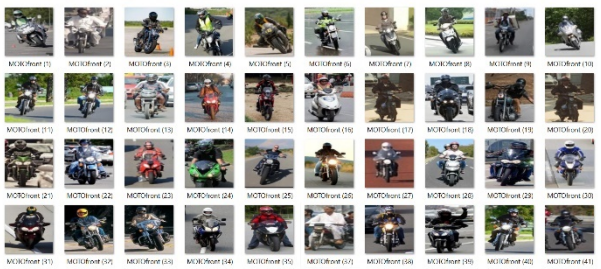


Fig. 25. Samples from Motorcycle Class (Front View).

Fig. 26. Samples from Truck Class (Front View).

Fig. 27. Samples from Bus Class (Rear View).

Fig. 28. Samples from Car Class (Rear View).

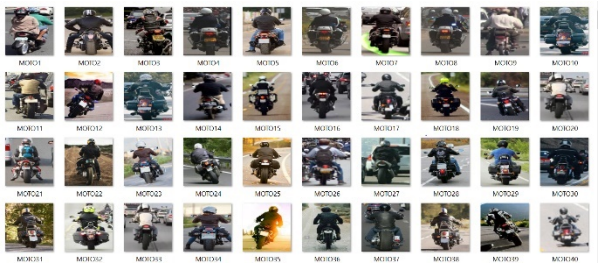


Fig. 29. Samples from Motorcycle Class (Rear View).

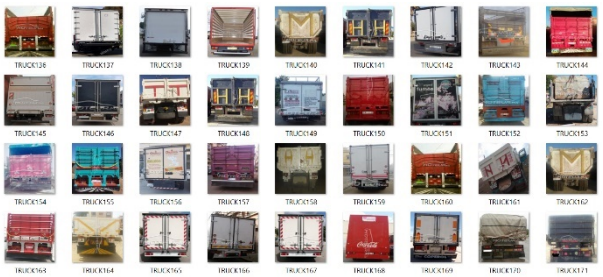


Fig. 30. Samples from Truck Class (Rear View).