Illumination Correction in a Comparative Analysis of Feature Selection for Rear-View Vehicle Detection

S. Baghdadi and N. Aboutabit

Abstract— **One of the most frequent infractions on the road is the act of a vehicle crossing to the wrong side of the road to pass another vehicle traveling in the same direction. Automatic detection of this violation can be a challenging issue. Thus, we aim to develop a computer vision system to robustly detect forbidden overtaking observed from a fixed camera. Our approach is based on two main phases: Line Detection and Vehicle Detection. In this work, we focus on the vehicle detection stage. Here, features are extracted from the image and then classified using machine learning algorithms. In a first experiment, we constructed different models using features such as HOG (Histogram of Oriented Gradient), SURF (Speeded up Robust Features), Gabor filter and LBP (Local Binary Patterns), and machine learning classifiers as SVM (Support Vector Machines), kNN (k-Nearest Neighbor) and Decision Tree. Then, we merged the best descriptors to combine the advantages of their different robustness in order to build a strong vehicle detection model. The performances of all constructed models are evaluated on the GTI database. In a second experiment, illumination normalization techniques were applied to image database for the effective models. Thus, the first experimental results reveal that the combination HOG+ LBP+ Gabor Phase performs well with SVM. The second experimental results show that the Variational Retinex algorithm-based illumination correction provides a significant improvement in the detection rate.**

Index Terms—**Feature extraction, illumination normalization, machine learning, vehicle detection.**

I. Introduction

The number of road accidents increases day by day [1], up to 90% of vehicle accidents are caused by human faults [2]. Overtaking accidents are usually very serious accidents on rural roads [3]. They are due to a vehicle trying to overtake another on a two-lane road. This is usually indicated by a solid white or yellow line painted on the roadway marking the left limit of traffic (centerline). Every vehicle must not exceed this line in order to avoid passing to the opposite direction of the traffic. Thus, prohibited overtaking is especially a hard problem that still remains a challenging task [4].

Nowadays, the technology may reduce the huge number of human casualties by using many techniques based on artificial visual information to increase security and comfort of transport [2]. In doing so, we aim to develop a computer vision system that aims to detect automatically prohibited overtakings where the camera is fixed such as in traffic/driveway monitoring systems (Fig. 1). In this system, robust and reliable vehicle detection is the first step. Thus, correct vehicle detection results give better prohibited overtaking detection [1].

Vehicle detection is one of the most challenging research problems in computer vision due to the different views, lighting variations and complex background. There are many possible views of a vehicle: the side-view, the forward and rear views [5], [6]. The vehicle side-views have obvious and consistent characteristics in their structure such as wheels, oblique windows and bumpers, which provide crucial cues for detection [7]. We find a very active area of research that was focused on side-view vehicle detection. However, less efforts have been devoted in rear view vehicle detection which is an important problem as many road cameras capture rear view images. Rear views are also less discriminative and therefore more challenging [8].

The focus of this work is on rear-view vehicle detection. The rear-view vehicle images are captured by a static road camera from a distance along one track (Fig. 1).

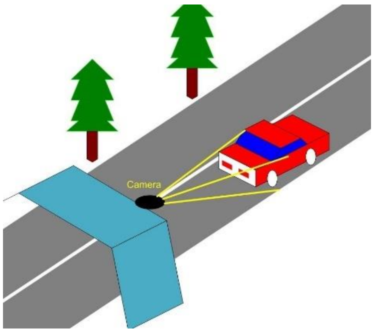


Fig. 1. Camera position.

In almost every vehicle detection system, two main phases can be identified: feature extraction and classification. The first step in vehicle detection is to extract relevant and informative features from the image containing the vehicle to be detected. Many general and robust methods to extract image features have been proposed in the literature. The extracted features measure in the most cases a propriety in the image such as color, texture, shape, motion, location and more. There are many algorithms that are designed to help us in extracting features. These algorithms are divided into two categories: 1- Hand engineered feature extraction methods such as: SURF, HOG, Scale Invariant Feature Transform (SIFT), LBP, GIST. 2- Learn features that are discriminative in the given context (Principal Component Analysis (PCA), Independent Component Analysis (ICA), K-means) [9]. In every object classification system, it is necessary to have a decision that predicts the object's class based on its features. To accomplish this, a classifier utilizes some training data to understand how given input variables relate to the class. We find two types of learners in classification: lazy learners and

Manuscript received March 31, 2019; revised October 25, 2019.

The authors are with the National School of Applied Sciences, Khouribga, Sultan Moulay Slimane University, Morocco (e-mail: sara92.baghdadi@gmail.com, noure049@gmail.com).

doi: 10.18178/ijmlc.2019.9.6.863

eager learners. Lazy learning stores training data and waits until it is given a new instance. For example: Ex. k-nearest neighbor, SVM. Eager learning given a set of training set, constructs a classification model before receiving new (test) data to classify. For example: Decision Tree, Naive Bayes, Artificial Neural Networks [10].

In this project, we realized a comparative study of the most well-known descriptors with some known classifiers in order to select the best combinations descriptor/classifier. Our aim is to improve the performance of the whole system. For this, we divided our work into two parts. The first part is dedicated to the hybridization of these descriptors. The second part is devoted to tackle the illumination and shadow problems in the database images. This is an extremely important step because the performance of the entire system depends on the quality of the data.

Detection systems must cope with a variety of challenges in outdoor scenes such as illumination changes, camouflage and shadows. Thus, the illumination and shadow problems have become a serious outdoors problem. These problems have been addressed in several studies using many illumination enhancement methods. In this work, we tried to resolve them using many methods such as: Morphological Quotient Image, Variational Retinex algorithm-based illumination correction, and Contrast enhancement techniques (Top-Hat filtering, Local Contrast enhancement and Intensity Adjustment). We tried also to remove shadow regions from the vehicle images.

The paper is organized as follows: Section II presents a survey of the main related works. Section III describes the vehicle detection system. This section is divided into three parts. The first part presents the used descriptors and classifiers. The second part presents all methods used to resolve the illumination and shadow problems. The third part defines the evaluation parameters. Section V describes the experiment results and discussion.

II. Related Work

In this section, we discuss the related research to vehicle detection. In fact, the study of this problem differs from one situation to another according to many criteria. For example, driving environments are visually dynamic and feature diverse background and illumination conditions [11]. There is also variability in the size, shape, and color of vehicles encountered on the road [12]. The camera position plays an important role in obtaining different views of the vehicle whatever it is static or mobile. We start by first addressing similar work to our study on vehicle detection from rear-view. Further we address some studies that have dealt with illumination problem on vehicle detection systems.

Until now many researchers dedicated their works to rear-view vehicle detection on the road highlighting the steps of machine learning. In [1], the authors proposed a hybrid method to detect rear-view vehicle by extracting the shadow with Haar-like feature combined with Adaboost and then verifying the back of the vehicle with a HOG descriptor and SVM classifier. In [13], the author proposed a new rear-view vehicle detection and tracking method based on camera. The detection has done by features extraction such as color, texture and region of the rear of the vehicles. In [14], the author proposed a way to detect vehicles at different perspective front, back, side, and oblique. For the vehicle's front and back, he used the symmetrical feature of the vehicle. For the vehicle's side, he used a side-view car detection algorithm based on template detection. In [15], the author proposed a rear-view vehicle detection and tracking method based on a high-resolution camera. First, the license plate and rear lamps are localized. Then, an MRF model is constructed by treating vehicle parts as graph nodes. The results of the proposed method are compared with that of rear-view vehicle detection and tracking method (Bin *et al*., 2014) and morphological operation method (Zezhong *et al*., 2013), and found that the proposed algorithm is more effective in terms of accuracy of vehicle detection and cost.

We highlight now the descriptors and classifiers used in the solutions proposed in vehicle detection. [16]-[20] used many descriptors like SIFT, GLOH, SURF, HOG, LBP and GIST. [16] proposed the scale invariant feature transform (SIFT), which uses a rectangular location grid. The SIFT features are invariant concerning illumination, rotation and scaling [17]. [18] proposed gradient location and orientation histogram (GLOH) descriptor similar to SIFT and replaces the rectangular location grid used by SIFT with a log-polar one. SURF features can be compared and computed much faster than SIFT. HOG is similar to both SIFT and GLOH, because it uses both rectangular and log-polar location grids. The main difference between HOG and SIFT is that HOG is computed on a dense grid of uniformly spaced cells, with overlapping local contrast normalization [19]. In [20], the features of the vehicle are extracted by the proposed GIST image processing algorithm and recognized by Support Vectors Machine classifier. [21] used optimized Gabor filter, here, the author concluded that the experimental results illustrate that the set of Gabor filters, specifically optimized for the problem of vehicle detection, yield better performance than using traditional filter banks. In [22], the authors proposed a hybrid method of vehicle detection based on Haar-like feature with Adaboost.

In a few studies, researchers have worked on the illumination problem in the vehicle detection domain. In [23], the proposed method is based on a combination of a quotient image, representing the reflectance ratio between the current frame and a background model, and an edge-density estimation process. In [24], the author presented two problems. In the first problem he tried to create the Retinex effect using by extracting the illumination image proposed by [25]. In the second problem he tried to detect and remove shadows from images. In the Retinex problem, the Euler-Lagrange differential equation is used to minimize the energy function. In the shadow removal problem, he used differential equation to minimize it.
The next section is devoted to the description of the methods used in vehicle detection.

III. Vehicle Detection Framework

In this section, we present the structure of the vehicle detection system shown in Fig. 2. To detect an object in an image, it requires two main stages: a training process and a test process. In the training process, we extract the features from a first data subset reserved to training data to store the discriminative information about each vehicle image, using many descriptors like HOG, SURF, LBP, Gabor. After this step, we have a learning or modeling step, its objective is to fit a model based on machine learning algorithms such as SVM SVM, kNN or Decision-Tree. In the test process, when we have an input image, we extract features and then we use the prepared model to give, as a result, the appropriate class (Vehicle / Non-vehicle).

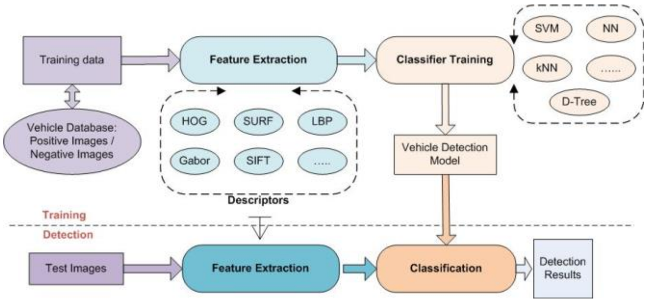


Fig. 2. Vehicle detection system.

The next section presents all methods used in our work. We start by defining the descriptors and then the classifiers.

A. Descriptors

1) *Histogram of Oriented Gradient HOG Features*. Introduced in 2005 by Dalal and Trigg, the HOG is a feature descriptor used to detect objects [22], [26]. As can be observed in Fig. 3, the image is segmented into small regions called cells, which are interconnected. The HOG directions are compiled for each pixel of these cells using gradient extraction operator. A histogram is formed in different orientations according to the accumulated value of the gradient, the HOG features of each cell are extracted to the series, and a one-dimensional feature vector of the vehicle image is obtained [4].

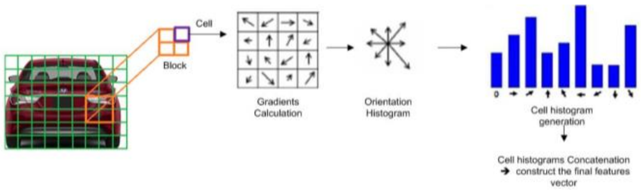

Cell
Block
Gradients Calculation
Orientation Histogram
Cell histogram generation
Cell histograms Concatenation → construct the final features vector

Fig. 3. HOG Descriptor

2) *Speeded Up Robust Features (SURF)*. Operates in a scale space and uses a Hessian based blob detector to find interest points. It is used mainly for object recognition. SURF was built on another feature extraction algorithm, Scale invariant feature transform (SIFT), which was one of first algorithms used in the late 90's. SURF is faster in getting the results than SIFT and also very robust in nature. The feature descriptor used in SURF is based on the sum of the Haar wavelet response around the point of interest, which makes it proficient to be computed with the aid of internal image [9], [22].

SURF descriptors have been used to locate and recognize objects, people or faces, to track objects and to extract points of interest [4].

3) Local Binary Pattern (LBP). The LBP operator is a texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers, as a binary number, the result. The LBP feature vector is created as shown in Fig. 4.

By applying LBP operator to one image, one pattern map can be computed. The pattern map is divided into many blocks and the histogram computed in each block is concatenated together to form the description of the input image [27].

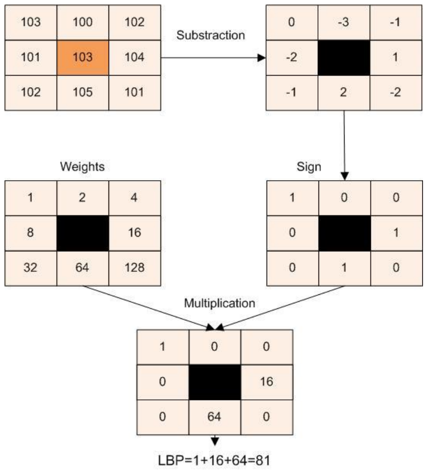


Fig. 4. LBP operator.

4) *Gabor Features.* The Gabor wavelet representation of an image is the convolution of the image with Gabor kernels [27]:

$$G_{u,v}(z) = I(z) * K_{u,v}(z) \qquad (1)$$

where *I* is the input image, and "*" is the convolution operator; *z* denotes the pixel *z*=(*x*, *y*), and *K* denotes the Gabor kernel with orientation *u* and scale *v*.

For each Gabor kernel, at every image pixel *z*, a complex number containing two Gabor parts, i.e., real part Re(*z*) and imaginary part Im(*z*), can be generated. Based on these two parts, magnitude A and phase *P* can be computed by (2) and (3), respectively

$$A_{u,v}(z) = \sqrt{\mathrm{Im}^2_{u,v}(z) + \mathrm{Re}^2_{u,v}(z)} \qquad (2)$$

$$P_{u,v}(z) = \arctan(\mathrm{Im}_{u,v}(z) / \mathrm{Re}_{u,v}(z)) \qquad (3)$$

B. Classifiers

1) Support Vector Machines (SVM). Was initially proposed by Cortes and Vapnik [26]. The principle of SVM classification is to find a hyper plane, which makes the classification of multidimensional space maximized. In the linear binary classification, the category labels are only two values, respectively, 1 and -1. Assuming that the new data points are classified according to the equation, the margin between samples to a hyperplane is defined as:

$$M_i = y_i.(w.x_i + b) \qquad (4)$$

where x_i represents the eigenvector of sample *i*, y_i represents the class label of sample *i*, *w* represents a spatial conversion of one function, *b* represents the offset parameter. After normalizing *w* and *b*, the margin can be expressed as

$$M_i = |w.x_i + b| / \|w\| \qquad (5)$$

The geometrical margin is shown in Fig. 5

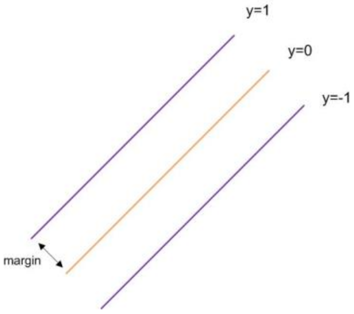


Fig. 5. Geometrical margin of SVM classifier.

In Fig. 5, y = 0 is the Classification-plane, both y = 1 and are y = -1 parallel to y = 0, and they are through the sample point which is nearest to y = 0. The interval between y = 1or y = -1 and y = 0 is the geometrical margin. In order to minimize errors, it is necessary to maximize the geometrical margin [4].

Multi-class classification can be regarded as a combination of a number of binary classifications.

2) K-nearest neighbor (kNN). Was proposed by Cover and Hart [26]. It is the extension of the minimum distance method and the nearest neighbor method. It calculates the distance between testing sample x and all the training samples and makes the distances in descending order. Then it selects k training samples which are nearest to the testing sample x and counts the category of the k training samples selected. The testing sample belongs to the category which has the largest number of votes in the same category [4], [26].

3) Decision-Tree. Is a classifier based on hierarchy. it compares the data to a range of properly selected features. The selection involves a binary question (yes / no answer) about certain features of each node of the tree. The leaves represent the class labels and the branches represent the conjunctions of features leading to these class labels. To predict an answer, follow the decisions in the tree from the root node (beginner) to a leaf node [28]. There are two types of decision trees: classification trees and regression trees. The classification tree is when the expected result is a class. The regression tree is when the expected result can be considered as a real number.

C. Illumination and Shadow Problems

Lighting conditions change the global appearance of vehicles along the road depending on the time of the day and the weather. Then, the illumination problem is a challenging issue in the outdoor scenes. The large variation in illumination would probably affect the performance of vehicle detection algorithms. So, it's helpful to enhance the visual effect of images. Although classical descriptors such as Local Binary Pattern (LBP) and Gabor are commonly believed to be robust to slight illumination change, their performance would drop when the lighting condition becomes severe [29]. Various advanced methods have been proposed by extracting the reflectance component and the illumination component such as Retinex algorithms. Contrast enhancement techniques and Quotient Image algorithms can be used to address this issue. Removing shadows from database images can also enhance the detection rate.

1) Illumination normalization methods

In this work, we used the methods: Variational Retinex algorithm-based illumination correction, Morphological Quotient Image, and Contrast enhancement techniques (Top-Hat filtering, Local Contrast enhancement and Intensity Adjustment). The descriptions of these methods are given below:

Variational Retinex algorithm

Retinex theory addresses the problem of separating illumination from reflectance in a given image and thus compensating non-uniform illumination [25]. It supposes the image intensity value $I(x, y)$ of a pixel (x, y) is composed of two components: the illumination $L(x, y)$ and reflectance $R(x, y)$ images. The relationship between the three can be defined as:

$$I(x, y) = L(x, y) \times R(x, y) \tag{6}$$

where I, L, R represent the intensity, illumination component and reflectance component respectively.

The advantages of such a decomposition include the ability to suppress the illumination effects of the back / front lighting and to improve shots with spatially varying illumination, such as images containing indoor and outdoor areas.

A first step taken by most of the Retinex algorithms is the conversion in the logarithmic domain by i = log$I$, l = log$L$, r = log$R$, and hence i = l + r. The different Retinex algorithms generally have the same diagram as that shown in Fig. 6.

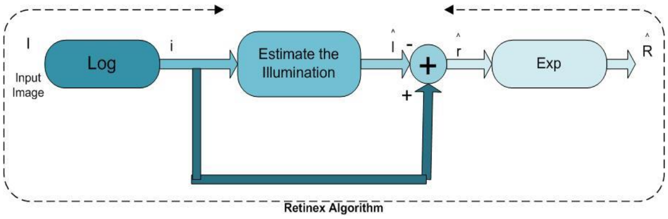


Fig. 6. Retinex diagram.

The algorithm we used in our study is the Variational Retinex algorithm proposed by [25]. This algorithm used Euler-Lagrange differential equation derived from a related energy function. Like all other Retinex algorithms, the Variational Retinex algorithm is based on some assumptions detailed in [24]. Thus, the Euler-Lagrange equation becomes:

$$\Delta E = \alpha.(l - i) - \Delta l - \beta.\Delta(l - i) = 0 \tag{7}$$

where α and β are positive constants.

The Variational Retinex algorithm uses this equation to estimate the illumination image l.

So far, we have dealt with only one channel. When our image is colored, there are two ways to extract the illumination image using the Variational Retinex algorithm. The RGB Retinex and the HSV Retinex. In the RGB Retinex, the colored image is decomposed into three channels: Red R, Green G and Blue B. Then, the Variational Retinex algorithm is applied on each color channel separately to obtain lR, lG, lB. Then, these three images are combined to obtain the colored illumination l [24]. For the HSV Retinex, the colored image is decomposed into three layers: Hue H, Saturation S, and Value V. The layer V corresponds to the brightness of the pixel; Therefore, it makes sense to apply Retinex Variation to V only, as we are trying to extract the illumination image. Here, we used the RGB Retinex.

Illumination Correction Algorithm

As shown in Fig. 7, we start by separating the illumination and reflectance images using the Variational Retinex algorithm. Then, we correct the illumination by a Gamma factor to obtain a new illumination image and multiply it by R, that gives the enhanced version of the original image I'.

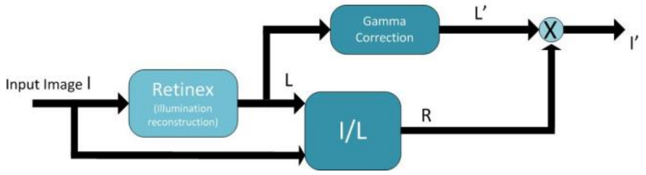


Fig. 7. Illumination correction algorithm.

The Gamma factor is given by:

$$L' = W.\left[\frac{L}{W}\right]^{\frac{1}{\delta}} \tag{8}$$

where W is the White value. The output image I' is:

$$\begin{aligned} I' &= L'.R \\ &= \frac{L'}{L}.I \\ &= W.\frac{(L/W)^{1/\delta}}{L}.I \\ &= \frac{I}{(L/W)^{1-1/\delta}} \end{aligned} \tag{9}$$

Morphological Quotient Image (MQI)

Recently, quotient image-based methods are reported to be a simple and efficient solution to illumination variances problem. [30] proposed a method based on Morphological Quotient Image (MQI). It employs mathematical morphological theory and quotient image technique for illumination normalization.

The MQI is defined by the relationship:

$$MQI = I(x, y)/M(x, y) \tag{10}$$

where $I(x, y)$ is the original image and $M(x, y)$ is its smoothed version. We use Morphological operation to smooth the image. Here, we used the opening operator, which is defined as an erosion followed by a dilation with the same structuring element.

Contrast Enhancement Techniques

There are some existing contrast enhancement methods to deal with the illumination problems:

The Top-Hat transformation. is one of the morphology-based contrast enhancement techniques. In mathematical morphology and digital image processing, the top-hat transformation is an operation that extracts small elements and details of given images. Top-Hat filtering calculates the morphological opening of the image (opening is the dilation of the erosion), and subtracts the result from the original image [31].

Local contrast enhancement. We preprocess the original images by exploiting the local contrast enhancement. The intensity range is very wide for an image illuminated by an uneven light source. And its data value distribution is unbalanced. The local contrast for a pixel (m, n) with the luminance value $I(m, n)$ is given by

$$Y(m,n) = \begin{cases} \log(I(m,n)/\overline{I(m,n)}), I(m,n) > \theta, \overline{I(m,n)} > \theta \\ 0, otherwise \end{cases} \tag{11}$$

θ is a predefined threshold, $I(m, n)$ indicates the mean luminance value of the neighborhoods of the pixel (m, n) [32].

Intensity Adjustment. is an image enhancement technique that maps the intensity values of an image to a new range. It returns an image of equal size to I with its adjusted intensity values, usually in order to increase the contrast of the image. Here, we adjust the intensity values in our database images using the Imadjust function [33].

2) Shadow removal

The existence of vehicle shadow has a negative effect on the vehicle detection. If the shadow is not eliminated, it will merge with the vehicle and affect vehicle's detection. Since shadows occur because of lack in light in certain regions. we try to remove shadows by supplying more light to the shadowed regions. We detect and separate shadow from non-shadow region and then we try to produce light for the shadowed region.

We use the effect of minimizing energy function in removing shadows. The needed light is assumed to be a constant. So, the constant light is a three-component vector $\vec{c}=(c_R, c_G, c_B,)$, one component for each light: red, green, and blue. When the value of the vector c is added to the shadow region, it minimizes the norm of the difference between the average light inside and the average light outside of the shadow region. the energy function is given by:

$$E(\vec{c}) = \left[(c_R.u_{in}^R - u_{out}^R)^2 + (c_R.u_{in}^G - u_{out}^G)^2 + (c_B.u_{in}^B - u_{out}^B)^2\right] \tag{12}$$

The solution of this equation is:

$$\vec{c} = \left(\frac{u_{out}^R}{u_{in}^R}, \frac{u_{out}^G}{u_{in}^G}, \frac{u_{out}^B}{u_{in}^G}\right) \tag{13}$$

The method was presented in [24] with more details.

3) Evaluation criteria

To evaluate the accuracy of the constructed models, we use the criteria below. For this, we started by calculating True Positive, False Positive, False Negative and True Negative. A true positive is one that detects the vehicle when the vehicle is present. A true negative is one that does not detect the vehicle when the vehicle is absent. A false positive is one that detects the vehicle when the vehicle is absent. A false negative is one that does not detect the vehicle when the vehicle is present.

From this, Runtime rate, True Detection rate, Extra Detection rate, and Missed Detection rate are calculated as follows:

a) True Detection = (TP + TN)/(TP + TN + FP + FN)

b) Extra Detection = FN

c) Missed Detection = FP

IV. Experiment Results

In this section we conducted a series of experiments to evaluate the performance of various vehicle detection methods. Two scenarios were considered in our experiment. The first scenario aimed to compare the constructed models using the classifiers SVM, kNN and Decision Tree and the descriptors HOG, SURF, LBP, Gabor Magnitude and Gabor Phase, in order to find which of these models provides significant and faster results. The second scenario aimed to compare the models which are constructed after an illumination normalization step of the database images.

In most cases, the feature concatenation provides better results than using a single descriptor, it combines complementary advantages for each method. Then, we merged the best descriptors in one descriptor to build a strong vehicle feature model.

The Fig. 8 presents the descriptor fusion system.

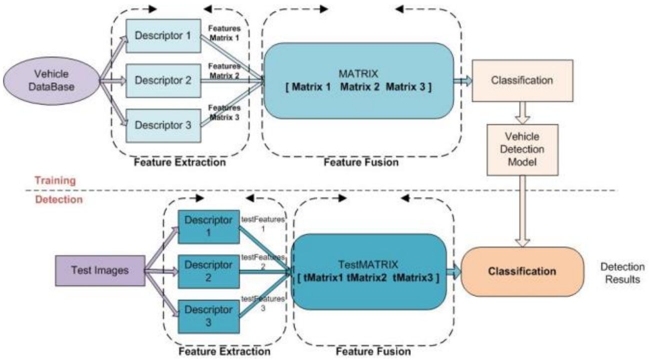


Fig. 8. Feature fusion system.

Then, the obtained features are concatenated and provided as an input of the classification step (SVM, KNN and Decision-Tree are used). Finally, we calculated the evaluation criteria.

A. Experiment Database and Materials

We have conducted an experiment using a GTI database [34] having 7785 positive images of vehicles (rear-view) and 8968 negative images that do not contain vehicles.

Fig. 9. Samples from Vehicle class.

The dataset included two classes of images, the first class contains the positive images (Fig. 9), and the other class contains the negative images. From the dataset two sets are created; a training set and a test set. Eighty percent of these positives were included in the training set. The remaining 20% were included in the test set. To accomplish this, we implemented the algorithms in Matlab.

The complete set of images is selected under different weather conditions. In fact, 20% of these images were taken in a sunny weather, 20% during cloudy days, 20% in medium conditions (neither very sunny nor cloudy), 20% with poor illumination, 10% with light rain, 5% with bad resolution cameras, and 2,5% in tunnels (with artificial light) [34].

We executed our algorithms on a Lenovo ThinkPad with a processor Intel® Core™ i5 7th Generation CPU @ 2.50GHz 271GHz, RAM 8Go.

B. Results and Discussion

1) Experiment 1

To evaluate the performance of the algorithms we used, we tested all of them by the same image list to keep the same evaluation conditions. This evaluation was based on runtime and detection rate. The average runtime per image for each model is shown in Fig. 10.

The descriptors HOG, LBP, Gabor filter are implemented in Matlab. In our experiment, we set the default values for all parameters. For Gabor filter, each pixel has a group of features containing multiple magnitudes and corresponding phase values. Then, we calculated two types of the features. Phase and magnitude values; the phase feature is calculated as the standard deviation of the selected phase values. The magnitude feature is calculated as the mean of selected magnitude values.

However, for classifiers, the parameters are set as follows: the kernel function of SVM is the linear kernel, the parameter k of kNN classifier is set to 4 and 1 (by default) and the distance metric is either Euclidean (by default) or cosine.

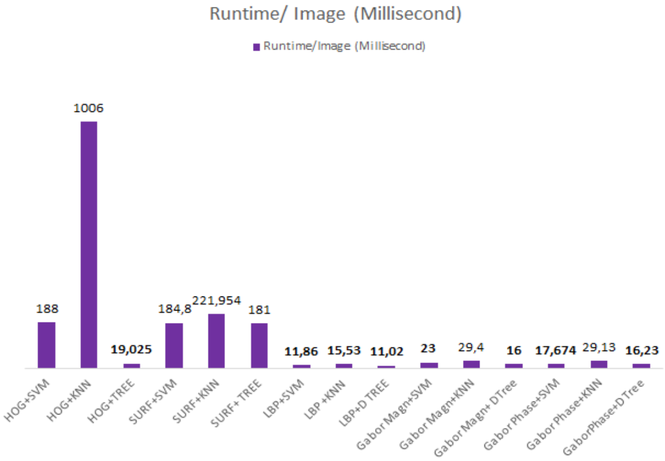


Fig. 10. Comparison of runtime averages of the models.

As shown in Fig. 10, the model LBP + Decision Tree is the fastest one. The Decision-Tree classifier is extremely simple. Since the complexity of this method depends on the depth and the depth of this tree, here, is 1 (A depth-1 decision tree is called a decision stump) then, it is simpler than a linear classifier. For the other classifiers, we see that SVM, the linear classifier is a fair bit faster than kNN classifier. As the parameter k increases, the classifier is averaging over more neighbors, and then more time required classifying data. The same observation if we modify the kNN classifier to use cosine distance instead of the default.

LBP, Gabor Magnitude and Gabor Phase features need less time to be extracted than the other features HOG and SURF. Because, histograms need more time to compute. SURF descriptor is the slowest one, because it is based on a bag of features which contains different phases such as the extraction of features, the coding of features, the pooling of features, and the spatial information preservation.

We note that the fastest models are, in descreasing order: LBP+ DecisionTree, LBP +SVM, LBP+kNN (D=Euclidean), GaborMagnitude+DecisionTree, GaborePhase+DecisionTree, GaborPhase +SVM and HOG+DecisionTree.

The Fig. 11 summarizes the comparative performance between all models, it shows the percentage of true detection, the number of extradetection followed by the number of missed vehicles. As shown, the descriptors LBP, HOG, and Gabor Phase perform much better than the other descriptors. Because texture features like LBP are robust to illumination changes and shadows. Furthermore, HOG is a dense grid; it is used as low-level features. For Gabor Phase, Phase information has the advantage of being very resistant to changes in illumination [35].

All descriptors work very well in combination with SVM and kNN classifier except Gabor Magnitude.

For the classifier kNN, if we increase slightly the parameter *k*, the classifier performs slightly better. The kNN classifier with the cosine distance performs slightly similarly with the one using the Euclidean distance function. We note that the Classifier kNN with the parameters (*k*=1 & Distance=Euclidean) or (*k*=4 & Distance=Cosine) achieves better results.

During the execution of our algorithms, we observed that with the increase of training samples, the accuracy of the model rises and the error rate seems to decrease. For kNN classifier, if we increase *k*, the error on training data starts to increase. And we also observed that the kNN works well with the new data because it just uses the training data as a look up table and search to classify the new data.

SURF descriptor is least efficient among all descriptors we tested. SURF performs less efficiently with kNN, but when it is combined with SVM and Decision Tree classifiers, it provides better results. We observed that the Decision Tree classifier works well in combination with GaborPhase.

Basing on the results with test data (Fig. 11), the order of ten first best models according to the accuracy in descending is as follows: HOG+SVM, HOG+kNN (*k*=1), LBP+kNN (*k*=1), LBP+kNN (*k*=4 & *D*=Euclidean), GaborPhase+SVM, GaborPhase+kNN (*k*=4 & *D*=Euclidean), HOG+kNN (*k*=4&D=Euclidean), GaborPhase+*k*NN (*k*=1 & *D*=Euclidean), LBP+SVM and GaborPhase +DecisionTree.

When there is a large impact on image from surroundings light and noise, single feature detection will be weakened, so multi-feature combination is able to improve the accuracy of vehicle detection because each feature has its strengths and weaknesses against each challenge. The best descriptors HOG, LBP and Gabor Phase are used to combine the advantages of their different robustness in order to build a strong vehicle detection model. HOG is a robust descriptor but it has drawbacks when illumination changes. The LBP and Gabor Phase descriptors can compensate the shortage of the HOG due to their robustness to changing illumination [35]. As shown in Fig. 12, the new model achieves the best result on testdata with an accuracy of 92.15% with SVM, 91.47% with kNN and 83.65% with Decision Tree. There is approximately a 8.85% accuracy difference between SVM and Decision Tree classifier.

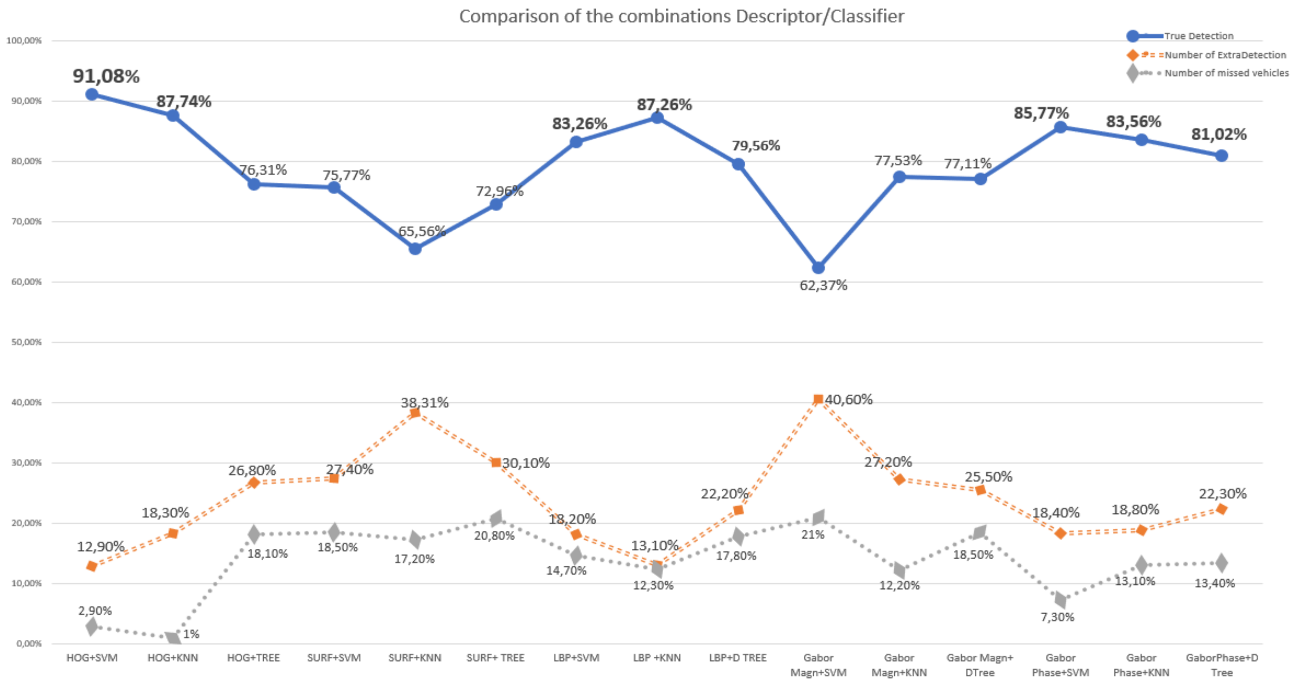


Fig. 11. Comparison of the constructed models (test data).

All used features produce false positive detections, caused by illumination changes and shadows, and false negative detections due to camouflage (when background and foreground share similar colors). Then, the combination of these features reduces the missed vehicle detections (the false positive detections).

Since the road environment undergoes variables such as weather conditions, tunnel, light changes throughout the day, it is vital to develop image processing algorithms that can robustly and reliably detect vehicles and extract effective information.

2) *Experiment 2*

In this section, our aim is to improve the performance of the vehicle detection system. Then, we started by enhancement of database images in order to improve the visual effects of the image and eliminate the impact of uneven illumination. We applied the illumination normalization methods on the database. We start by the variational Retinex algorithm based-illumination correction and the Morphological Quotient Image. Furthermore, we used three contrast enhancement techniques: Top-Hat filtering, Local Contrast enhancement and Intensity Adjustment. This process can clear the gap between the dark road and other vehicles on the road in order to extract easily the vehicle regardless of the light and road conditions. For the shadow problems, we try to remove the shadow regions based on the concept of minimizing the energy function.

The performance of illumination normalization is evaluated according to the detection rate of the best combinations selected in the first experiment (LBP HOG Gabor Phase) +SVM, (LBP HOG Gabor Phase) +KNN (*K*=1 & Distance=Cosine).

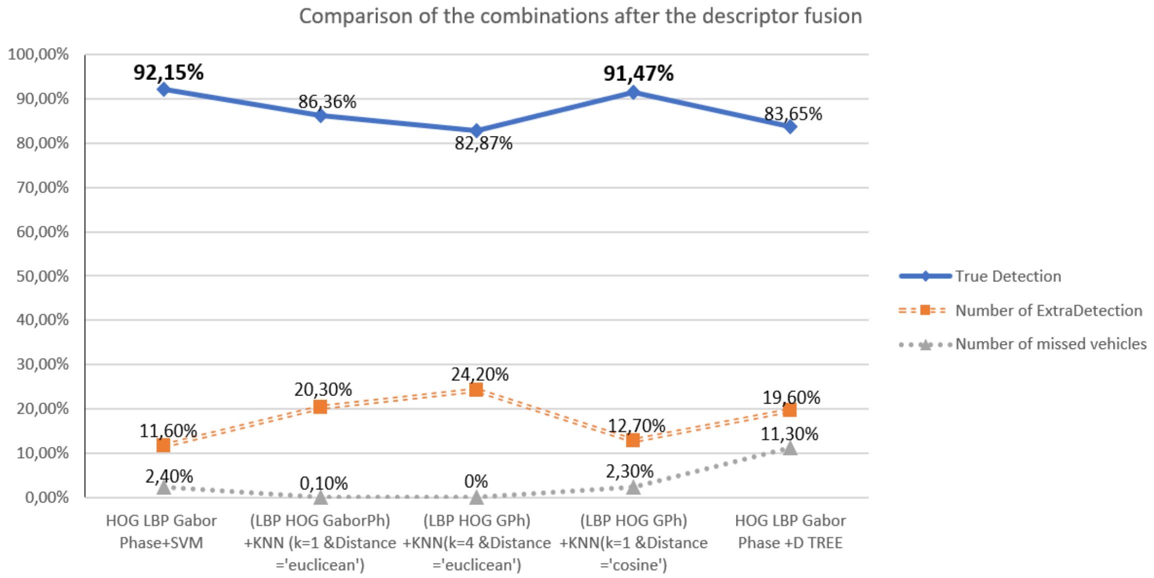


Fig. 12. Comparison of the models after the descriptor fusion step.

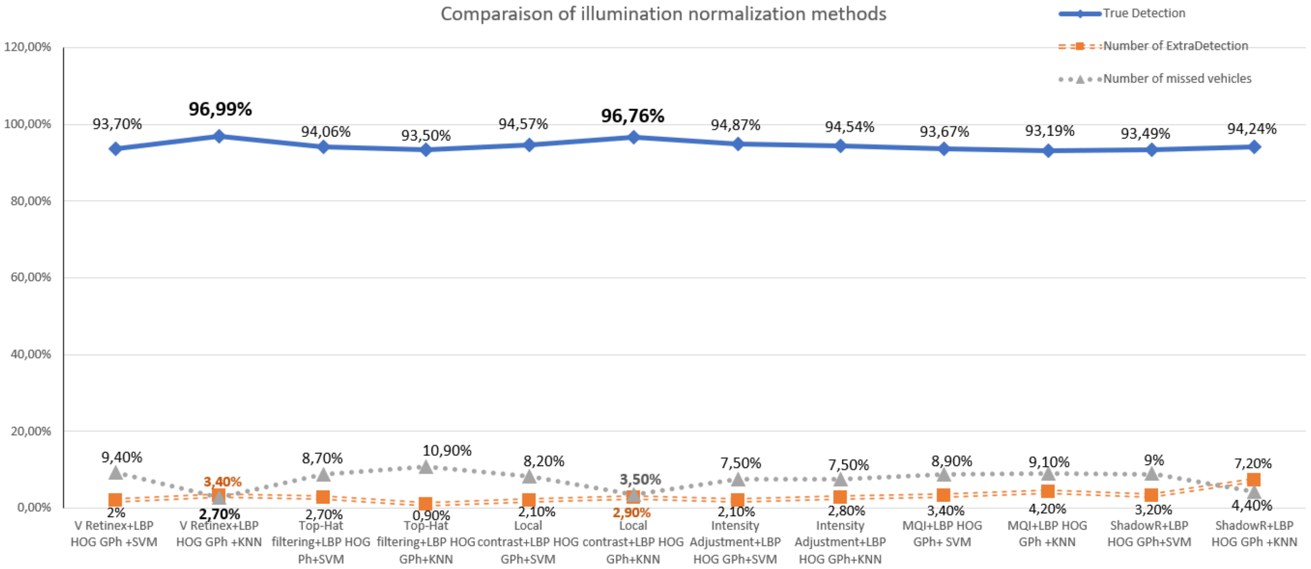


Fig. 13. Comparison of illumination normalization methods.

The Fig. 13 presents the results of our experiment. After enhancement, it can be seen that the evaluating indicators of detection have been improved a lot. All methods achieve the best performances compared to the Fig. 12. The methods Variational Retinex algorithm-based illumination correction and local contrast enhancement have better results than the other ones (96.99%, 96.76% and average enhancements of 5.52% and 5.29%, Extra detections 3.4%, 2.9%, Missed vehicle detections 2.7%, 3.5% respectively) with LBP-HOG-GaborPhase + kNN model. These two methods efficiently avoid the detection errors. Based on our future system objective, it's allowed to have extra-detections against having missed vehicle detections. Especially when the main objective of the system is to detect the vehicles crossing the continuous line. Besides, the most suitable method for our system is the Variational Retinex algorithm-based illumination correction.

V. Conclusion

This paper presented a comparative study of vehicle detection methods. We compared various models using the descriptors: HOG, LBP, SURF, Gabor Magnitude and Phase. From the comparison results, we can conclude that the method based on HOG and SVM provides the best results on test data with an accuracy and runtime per frame of about 91.08% and 0.188s respectively. We improved the performance of the system by combining many descriptors.

The best combination of features is HOG+ LBP+ Gabor Phase. It performs well with SVM classifier with an accuracy of 92.15%. In transportation, one of the most important problems of vehicle detection is illumination variation. In our approach to resolve this problem, we used several methods to correct illumination. Comparing the results, evaluating indicators of detection are all improved, the two methods Retinex and local contrast are better, they improve very well the accuracy of the system (96.99%) and can be very helpful for our global detection system. For our next work, we will work on the both views: rear and forward views of vehicles.

References

[1] H. Wang and H. Zhang, "A hybrid method of vehicle detection based on computer vision for intelligent transportation system," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 9, no. 6, pp. 105–118, 2014.

[2] M. Aly, "Real time detection of lane markers in urban streets," in *Proc. Intelligent Vehicles Symposium IEEE*, 2008, pp. 7-12.

[3] T. Richter, S. Ruhl, J. Ortlepp, and E. Bakaba, "Causes, consequences and countermeasures of overtaking accidents on two-lane rural roads," in *Proc. World Conference on Transport Research – WCTR*, Shanghai, 10-15 July, 2016.

[4] S. Baghdadi and N. Aboutabit, "A comparative study of vehicle detection methods," *Advanced Intelligent Systems for Sustainable Development (AI2SD'2018)*, vol. 5, pp. 1–11, 2019.

[5] M. Gessica, R. Meneses, and L. Nogueira, *Vehicle Shape Recognition Using SVM and ASM*, 2015.

[6] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 694-711, May 2006.

[7] V. Vinoharan *et al.*, "A wheel-based side-view car detection using snake algorithm," in *Proc. 6th International Conference on Information and Automation for Sustainability (ICIAFS)*, 2012, pp. 185–189.

[8] M. Kafai and B. Bhanu, "Dynamic bayesian networks for vehicle classification in video," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, pp. 100–109, 2012.

[9] S. N Raj *et al*, "Comparison study of algorithms used for feature extraction in facial recognition," *International Journal of Computer Science and Information Technologies*, vol. 8, no. 2, pp. 163-166, 2017.

[10] S. Asiri. (Jun. 2018). Machine learning classifiers. *Towards Data Science*. [Online]. Available: https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623

[11] S. Sivaraman and M. Trivedi, "Vehicle detection by independent parts for urban driver assistance," *IEEE Trans. Intell. Transp. Syst.*, 2013.

[12] D. S. Pae, I. H. Choi, T. K. Kang, and M. T. Lim, "Vehicle detection framework for challenging lighting driving environment based on feature fusion method using adaptive neuro-fuzzy inference system," *International Journal of Advanced Robotic Systems*, 2018.

[13] G. Subbammal, M. S. Nisha, and J. A. Jevin, "A new rear-view vehicle detection and tracking for driverless vehicle assistance system," *International Journal of Advance Research in Science and Engineering*, no. 4, 2015.

[14] B. Cai, F. Tan, Y. Lu, and D. Zhang, "Knowledge template based multiperspective car recognition algorithm," *International Journal of Information Engineering and Electronic Business*, vol. 2, no. 2, pp. 38-45, 2010.

[15] B. Tian, Y. Li, B. Li, and D. Wen, "Rear-view vehicle detection and tracking by combining multiple parts for complex urban surveillance," *IEEE Trans. Intell. Transp. Syst*., vol. 15, no. 2, pp. 597-606, Apr. 2014.

[16] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 2, no. 60, pp. 91–110, 2004.

[17] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. 525-531.

[18] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2004, pp. 506-513.

[19] R. Rani, R. Kumar, and A. P. Singh, "A comparative study of object recognition techniques," in *Proc. 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, 2016.

[20] I. El Jaafari, M. El Ansari, L. Koutti, A. Ellahyani, and S. Charfi, "A novel approach for on-road vehicle detection and tracking," *Int. J. Adv. Comput. Sci. Appl*., vol. 7, no. 1, pp. 594-601, 2016.

[21] Z. Sun, G. Bebis, and R. Miller, "Evaluationary gabor filter optimization with application to vehicle detection," in *Proc. the 3rd IEEE International Conference on Data Mining*, Florida, 2003.

[22] H. Wang and H. Ahang, "Hybrid method of vehicle detection based on computer vision for intelligent transportation system," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 9, no. 6, pp. 105-118, 2014.

[23] M. Vargas, S. L. Toral, J. M. Milla, and F. Barrero, "A shadow removal algorithm for vehicle detection based on reflectance ratio and edge density," in *Proc. 13th International IEEE, Annual Conference on Intelligent Transportation Systems*, Madeira Island, Portugal, September 19-22, 2010.

[24] N. Almoussa, *Variational Retinex and Shadow Removal*, 2009.

[25] R. Kimmel, M. Elad, D. Shaked, R. Keshet, and I. Sobel, "A variational framework to retinex," *Int'l J. Computer Vision*, vol. 52, no. 1, pp. 7-23, 2003.

[26] S. Lin, C. Zhao, and X. Qi, "Comparative analysis of several feature extraction methods in vehicle brand recognition," in *Proc. 10th International Conference on Sensing Technology (ICST)*, 2016.

[27] S. Xie, S. Shan, X. Chen, and J. Chen, "Fusing local patterns of Gabor magnitude and phase for face recognition," *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1349-1361, May 2010.

[28] B. L. Zhang, "Classification and identification of vehicle type and make by cortex–like image descriptor HMAX," *Int J Comput. Vision Robot*, vol. 4, no. 3, pp. 195–211, 2014.

[29] J. Y. Zhu, W. S. Zheng, and J. H. Lai, "Logarithm gradient histogram: A general illumination invariant descriptor for face recognition," in *Proc. 10th IEEE Int. Conf. Autom. Face Gesture Recognit. (AFGR)*, April 2013, pp. 1-8.

[30] Y. Zhang, J. Tian, X. He, and X. Yang, "MQI based face recognition under uneven illumination," in *Proc. ICB*, 2007.

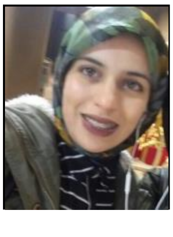
[31] S. Thapar, S. Garg, "Study and implementation of various morphology based image contrast enhancement techniques," *International Journal of Computing Business Research*, pp. 2229-6166, 2012.

[32] S. L. Zhang, H. H. Zhong, Y. J. Kuang, and J. W. Mo, "Illumination robust face recognition based on image fusion of local contrast enhancement and adaptive smoothing," in *Proc. International Conference on Information Science and Technology*, 2016.

[33] Image Processing Toolbox User's Guide. [Online]. Available: http://matlab.izmiran.ru/help/toolbox/images/enhanc17.html#12121

[34] The GTI-UPM Vehicle Image Database. [Online]. Available: http://www.gti.ssr.upm.es/data

[35] T. Bouwmans, C. Silva, C. Marghes, M. Zitouni, H. Bhaskar, and C. Frelicot, "On the role and the importance of features for background modeling and foreground detection," *CoRR*, vol. abs/1611.09099, pp. 1-131, Nov. 2016.



Sara Baghdadi is a Ph.D. student at the National School of Applied Sciences, 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, 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 Supérieure d'Ingénieurs Electriciens de Grenoble (ENSIEG). His current research interests include computer vision, machine learning, artificial intelligence, and multimedia processing.