

## Research Article

# Real-Time Vehicle Detection Using Cross-Correlation and 2D-DWT for Feature Extraction

Abdelmoghith Zaarane , Ibtissam Slimani , Abdellatif Hamdoun, and Issam Atouf

*LTI Lab, Laboratory of Information Processing, Department of Physics, Faculty of Sciences Ben M'sik, University Hassan II Casablanca, BP 7955, Casablanca, Morocco*

Correspondence should be addressed to Abdelmoghith Zaarane; [z.abdelmoghith@gmail.com](mailto:z.abdelmoghith@gmail.com)

Received 31 July 2018; Revised 5 November 2018; Accepted 5 December 2018; Published 9 January 2019

Academic Editor: Jar Ferr Yang

Copyright © 2019 Abdelmoghith Zaarane et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Nowadays, real-time vehicle detection is one of the biggest challenges in driver-assistance systems due to the complex environment and the diverse types of vehicles. Vehicle detection can be exploited to accomplish several tasks such as computing the distances to other vehicles, which can help the driver by warning to slow down the vehicle to avoid collisions. In this paper, we propose an efficient real-time vehicle detection method following two steps: hypothesis generation and hypothesis verification. In the first step, potential vehicles locations are detected based on template matching technique using cross-correlation which is one of the fast algorithms. In the second step, two-dimensional discrete wavelet transform (2D-DWT) is used to extract features from the hypotheses generated in the first step and then to classify them as vehicles and nonvehicles. The choice of the classifier is very important due to the pivotal role that plays in the quality of the final results. Therefore, SVMs and AdaBoost are two classifiers chosen to be used in this paper and their results are compared thereafter. The results of the experiments are compared with some existing system, and it showed that our proposed system has good performance in terms of robustness and accuracy and that our system can meet the requirements in real time.

## 1. Introduction

The automatic vehicle detection has gained importance in research for the last fifteen years where the development of a successful system for vehicle detection is the principal step for driver assistance which needs calculation of the distances between vehicles to warn drivers to slow down vehicles to avoid accidents and collisions.

Several methods are used to detect vehicles [1] such as laser-based or radar systems. However, in this paper, we are based on image processing. The majority of the proposed method follows two steps, namely, hypothesis generation and hypothesis verification. In the hypothesis generation step, the localization of vehicles “zones of interest” in the image is hypothesized. In the hypothesis verification, the zones of interest are treated and verified if they are vehicles or not. Several methods are proposed to generate the hypothesis. Yan et al. [1] used the preknowledge shadows

underneath the vehicles to detect the zones where a vehicle can be in the image but that can be suitable just in specific weather and specific time in the day. Soo et al. [2] proposed a monocular symmetry-based vehicle detection system in which the symmetry is one of the most interesting visual characteristics of a vehicle. However, computation of the symmetry values for every pixel is a time-consuming process. Jazayeri et al. [3] detected and tracked vehicles based on motion information, and they relied on temporal information of features and their motion behaviors for vehicle identification, which helps recompensing the complexity in recognizing types, colors, and shapes of a vehicle. A motion-based method is a successful method to detect moving objects. However, it is intensive in terms of calculation and requires analysis of several frames before an object can be detected. It is also sensitive to camera movement and may fail to detect objects with slow relative motion. Gao et al. [4] used color information and edge information to detect

vehicles where the detection method is based on the detection of rear lights by looking for the red representation in the image, and they used the function of symmetry measure to analyse the symmetry of the color distribution and determine the specific position of axis of the symmetry. Afterwards, the pair of edges are determined to rebuild the integrated edges of a vehicle.

After the hypotheses generation step, the generated hypotheses should be classified either as vehicles or not. In this step, two essential operations are needed: feature extraction and classification. Various methods are proposed to overcome this step. The Haar-like feature extraction method is usually used which is a robust and rapid method which uses the integral image, but the problem resides in the huge number of the output features. Usually dimensionality reduction techniques [5, 6] are required for the high-dimensional features. The Haar-like method was a good partner for many classifiers. In [7–9], the Haar-like method was combined with the SVMs classifier. Also, the Haar-like combination method and the AdaBoost classifier have been used in [10–12]. Other famous features extraction methods are also used such as the histogram of oriented gradients (HOG), Gabor filters, and Gradient features. In [1] the AdaBoost and SVMs classifiers are trained by the combined HOG's features. In [13], a new descriptor is proposed for vehicle verification using the alternative family of functions of log Gabor instead of the existing descriptors based on the Gabor filter. Descriptors which are based on Gabor filters have presented good results and showed good performance in extracting features [14]. A system that detects rear of vehicles in real time based on the AdaBoost classification method and the gradient features method for adaptive cruise control application (ACC applications) is presented. The Gradient features method is good at characterizing the objects shape and appearance.

In our proposition, at the hypothesis generation step, vehicle candidates are determined by using cross-correlation after preprocessing using edge detection to improve the results. The cross-correlation is a common method which has been used to evaluate and compute the similarity degree between two compared images. In the step of hypothesis verification, the generated candidates in the previous step are verified. Two major operations are needed in this step: feature extraction and classification. For feature extraction, the third level of 2D-DWT is utilized which is a powerful technique for representing data at different scales and frequencies. For classification, two classifiers are used: support vector machines (SVMs) classifier and AdaBoost classifier, and then their results are compared to get a reliable result. We have tested these classifiers using real data. However, it needs a large training set. Currently, we concentrate on the daytime detection for various vehicle models and types. In our approach, the vehicle candidates are generated using the highly correlated zones. These possible vehicle candidates are then classified with AdaBoost and SVMs to remove nonvehicle objects. Figure 1 shows the overall flow diagram of the method.

The organization of the paper is as follows. Section 2 describes the hypothesis generation. In Section 3, the

hypothesis verification method is presented. The experimental results are presented in Section 4 followed by the conclusion in Section 5.

## 2. Hypothesis Generation

The principal step in the vehicle detection system is the generation of hypothesis; where in this step, we should look in the image for the places where vehicles may be found (zones of interest). In our proposition, we perform first a preprocessing method using edge detection which acts an important role in the performance of our method. After performing the preprocessing, the cross-correlation is used to detect the zones of interest, which is an algorithm that calculates the similarity between a template and an image. The use of edge detection improves the result of the cross-correlation and also reduces the processing time.

In this section, the preprocessing and the cross-correlation techniques for initial candidate generation are treated.

*2.1. Edge Detection.* The best features that can be extracted from vehicles in the detection systems are corners, color, shadows, and horizontal edges and vertical edges. The shadows are good features to extract that can be utilized to facilitate the hypothesis of vehicles. However, they are very dependent on image intensity that depends also on weather conditions. The corner features can be found easily. However, they can be corrupted easily due to the noise.

In this paper, the edge detection is used where the horizontal edges and vertical edges are good features to extract. Looking at the edges reduces the required information because they replace a color image by a binary image in which objects and surface markings are outlined. These image parts are the most informative ones.

The first step is to generate a global contour image from the input gray-scale image using the Canny edge detector [15]. The selection of the threshold values for the Canny edge detector is not so critical as long as it generates enough edges for the symmetry detector. The edge detection was performed on the image and on the template. Figure 2 shows the result of edge detection performed on a typical road scene captured by the forward looking camera.

This technique improves the choice quality of the vehicle candidates, and it optimizes the processing time.

*2.2. Cross-Correlation.* The purpose is to identify areas in the image that are probably vehicles. However, the problem is to detect the pattern position in images. The cross-correlation is utilized to achieve this purpose which is a standard method of estimating the degree of similarity, in other words to estimate how much two images are correlated [16]. Therefore, the vehicle hypotheses in the images are found based on the similarity degree between template images and test images. Figure 3 shows a template image example.

The function of cross-correlation between the image and the template is defined as:

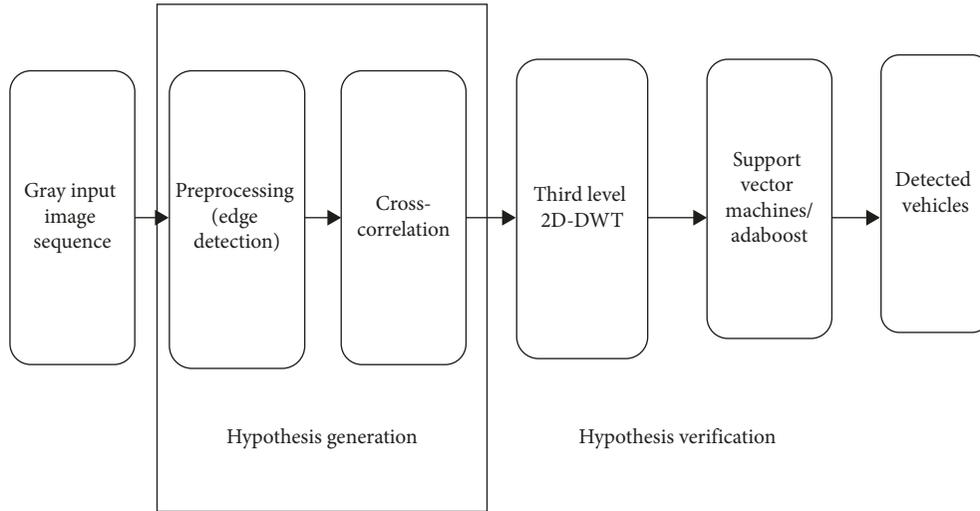


FIGURE 1: Overall flow diagram of our vehicle detection algorithm.

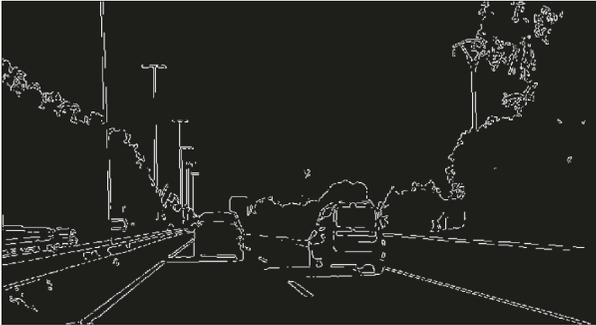


FIGURE 2: Resulting image using the Canny edge detector.

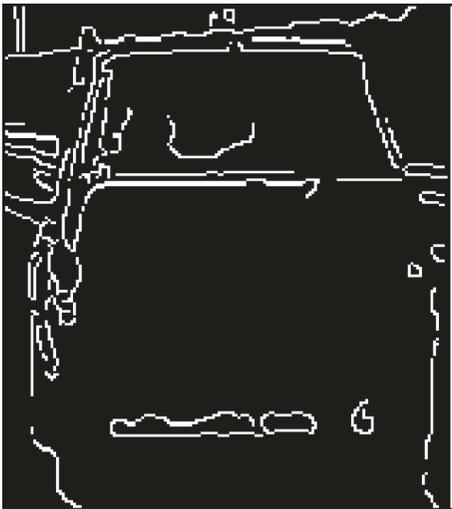


FIGURE 3: Example of a template.

$$\rho = \sum_{i,j} \frac{(x(i,j) - \bar{x})(y(i,j) - \bar{y})}{\sigma_x \sigma_y}, \quad (1)$$

where  $x(i, j)$  is the part of the image shared by template and  $\bar{x}$  is the mean of  $x(i, j)$ ;  $y(i, j)$  is the template and  $\bar{y}$  is the mean of  $y(i, j)$ ; and  $\sigma_x$  and  $\sigma_y$  are the standard deviations of

$x(i, j)$  and  $y(i, j)$ , respectively. The function  $\rho$  varies between  $-1$  and  $+1$ , where the good correlation state is found when the  $\rho$  function takes values near  $+1$  (i.e., when first function increases, the second one does too in proportion); the uncorrelated state is found when the  $\rho$  function takes values near  $0$  (i.e., no relation between variation in the first function and the second one); and the anti-correlated state is detected when the  $\rho$  function takes values near  $-1$  (i.e., when the first function decreases, the second increases in proportion). The best match occurs when templates and test images have maximum  $\rho$ . Multiple candidate locations can be found by using this technique.

The problem of matching using cross-correlation is that it detects the similarity between template and a part of the image only if they have almost the same size or a little bit bigger or smaller size which means that we can detect vehicles just in a predefined distance; in other words, we can detect only far vehicles or near vehicles. In our proposition, to overcome this problem, we chose to work with four different template's sizes. Two smaller sizes are used to detect far and very far vehicles, and two bigger sizes are used to detect close and very close vehicles. We do not need various sizes because the farthest vehicles are not that important. Different hypotheses of different vehicles are generated using few templates even if they have different shapes or types compared with the templates using the edge detection; therefore, there is no need to use templates for each vehicle type, shape, or texture. In our case, three templates in four sizes are enough to generate the hypotheses following the three vehicles categories, template for cars, template for buses, and template for trucks. Figure 4 shows an example of cross-correlation result that generates the hypothesis of far vehicles "red bounding box" and hypothesis of nearby vehicles "green bounding box."

### 3. Hypothesis Verification

The hypothesis verification step acts an important role for vehicle detection. The results of the previous step are the

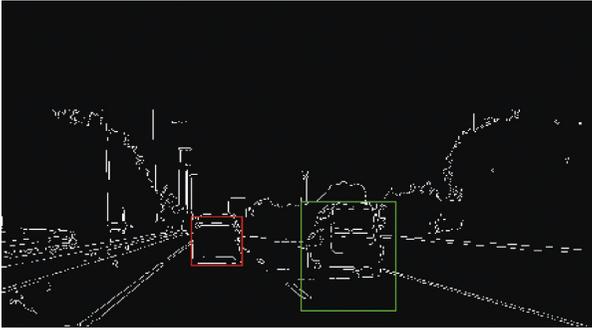


FIGURE 4: The result of the cross-correlation.

positions in the image where vehicles may be found. However, not all positions detected on the image belong to vehicles. Therefore, further verification is needed. In the verification step, two major methods are needed: feature extraction method and classification method. The classifier is used to classify the extracted features if they correspond to vehicles or not. Seeking the solutions to improve the vehicle detection accuracy and reduce the false detection rate while considering the real time, we propose to use the two-dimensional discrete wavelet transform for feature extraction, AdaBoost, and SVMs to classify these extracted features. The discrete wavelet transform (DWT) has a good location property in frequency and time domains, and it is an efficient method for features extraction. The AdaBoost and SVM classifiers are used in several studies, and they showed a very good result.

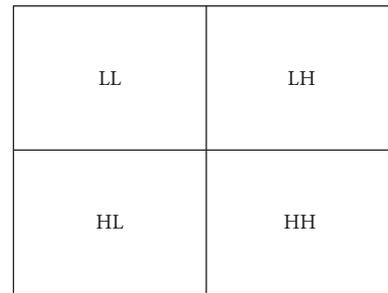
In this section, the discrete wavelet transform and SVMs and AdaBoost classifiers are treated.

**3.1. Discrete Wavelet Transform.** Wavelet transform is widely used in many applications because it reduces the computation cost and provides sharper time/frequency localization [17] in contrary to the Fourier transform. The discrete wavelet transform (DWT) is any wavelet transform for which the wavelets are discretely sampled. The principal of DWT is to decompose the input signal into two sub-signals: the detail and the approximation. The approximation corresponds to the low frequency of the input signal which is the most energy of a signal, and the detail corresponds to the high frequency of the input signal. This technique can be repeated at multiple levels by taking the approximation as an input signal. The same principal is applied for images, and the DWT decomposes the image into four subband images: LL, LH, HL, and HH subband images [18] as shown in Figure 5. The LL subband image contains the low-frequency component of the input image which corresponds to the approximation, and HL, LH, and HH subband images contain the high-frequency components of the input image which are the details.

As shown in Figure 6, the low-pass filter and the high-pass filter are used first on the lines of the input image, “i.e., vertically” and then on the columns, “i.e., horizontally.” Furthermore, after each filtering operation a down sampling is used to reduce the overall number of computation. This technique can be repeated at multiple levels until obtaining the desired result as shown in Figure 7.



(a)



(b)

FIGURE 5: (a) First level of DWT. (b) Subbands of the first level of DWT.

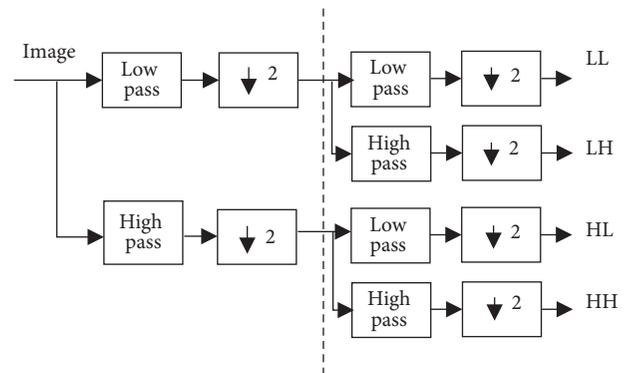


FIGURE 6: The structure of forward two-dimensional DWT.

In this study, we have concentrated on the third level of the 2D-DWT. This technique is applied on each generated candidate and on the dataset images to extract features. We extract the important features that we need, and it helps us to improve the result of the classification.

**3.2. Support Vector Machines (SVMs).** SVM is a popular machine learning algorithm for classification. It is a distinctive classifier that defines a separation hyper plane based on training data with its label (supervised learning). This algorithm generates the best hyper plane that classifies new examples. The SVM algorithm principle is used to find the hyper plane that maximizes the distance between the training example classes which is called the margin.

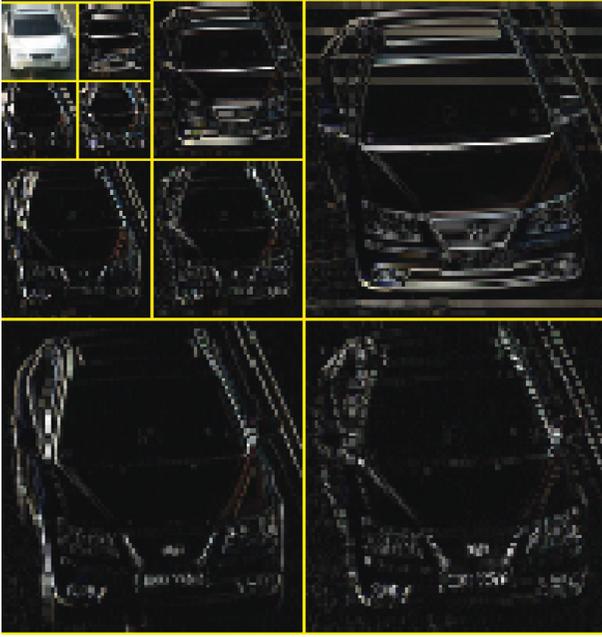


FIGURE 7: Third level of 2D-DWT.

Therefore, the optimal separating hyper plane maximizes the margin of the training data.

The separating hyper plane is defined as

$$f(x) = (\omega \cdot x) + b, \quad (2)$$

where  $\omega$  is known as the weight and  $b$  is called the bias.

The margin is given as

$$M = \frac{2}{\|\omega\|}. \quad (3)$$

According to this expression, it is necessary to minimize  $\omega$  to maximize the margin.

The classification function is given as

$$C_f = \sum_i \omega_i \cdot k(x, x_i) + b, \quad (4)$$

where  $x_i$  is the support vector selected from training samples,  $x$  is the input vector,  $k(x, x_i)$  is the kernel function, and  $\omega_i$  is the support vector weight ( $x_i$ ) which is determined in the training process.

In our paper, radial basis function kernel (RBF kernel) is used, and it gives good results compared to the other kernels. The RBF kernel function is given as

$$k(x, x_i) = \exp\left(-\left(\frac{\|x - x_i\|^2}{2\delta^2}\right)\right). \quad (5)$$

The SVMs are trained using the positive samples and negative samples. The positive and negative vectors are trained to be classified with the SVMs.  $X$  is considered to be a member of class one only if  $C_f \geq 0$ ; otherwise,  $x$  is considered a member of class two. The flowchart that illustrates the SVM classification is shown in Figure 8.

**3.3. AdaBoost Classifier.** AdaBoost (Adaptive boosting) was proposed by Freund and Schapire in 1996 [19]. It is a

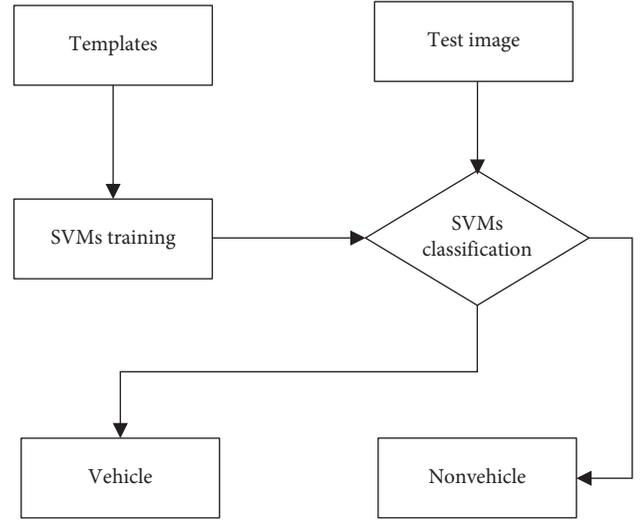


FIGURE 8: The flowchart of SVM classification.

supervised learning algorithm that classifies between positive and negative examples, and it aims at converting an ensemble of weak classifiers into strong classifier; a single classifier may classify the objects poorly. However, when multiple classifiers are combined with selection of the training set at every iteration and assigning right amount of weight in final voting, we can have good accuracy score for the overall classifier. The algorithm's input is a set of labeled training examples  $(x_i, y_i)$ ,  $i = 1, \dots, m$ , where  $x_i$  is an example and  $y_i$  is its label that indicates if  $x_i$  is a positive or negative example. Every weak classifier is noted as function  $h_t(x)$  that returns one of the two values  $\{+1, -1\}$ .  $h_t(x)$  is  $+1$ , if  $x$  is classified as a positive example, and  $h_t(x)$  is  $-1$ , if  $x$  is classified as a negative example. The AdaBoost algorithm is shown in Algorithm 1 according to [20].

Concerning training examples, we give  $m$  labeled examples  $(x_1, y_1), \dots, (x_m, y_m)$  whither the  $x_i \in X$ , and the labels  $y_i \in \{-1, +1\}$ .  $D_t$  is a distribution calculated on the  $m$  training examples of each value of  $t = 1, \dots, T$ , and to find a weak hypothesis  $h_t: X \rightarrow \{-1, +1\}$ , a weak learning algorithm is applied. Where the weak learner purpose is to look for a weak hypothesis that has a low-weighted error  $\epsilon_t$  relative to  $D_t$ . The weighted combination sign of the weak hypotheses is computed to determine  $H$  the final hypothesis.

### 3.4. Preparation of Input Data

**3.4.1. Training Process.** To train the classifier, we should prepare the templates first by normalizing them to  $158 \times 154$  grayscale images, then extracting the features using the third level of 2D-DWT, and finally, setting them in labeled vectors.

**3.4.2. Classification Process.** To classify the generated candidates (zones of interest), we should normalize them to  $158 \times 154$  grayscale images and then extract the features using the third level of 2D-DWT, and finally, we construct a vector using the extracted features which will be the input of the trained classifier and then obtain the results of the classification.

Input:  $(x_1, y_1), \dots, (x_m, y_m)$  is a set of labeled examples where  $x_i$  belongs to  $X$ ,  $y_i \in \{-1, +1\}$ .  
Initialization:  $D_1(i) = 1/m$  for  $i = 1, \dots, m$ .  
For  $t = 1, \dots, T$ :  
Train the weak learner based on distribution  $D_t$   
Obtain the weak hypotheses  $h_t: X \rightarrow Y\{-1, +1\}$   
Select  $h_t$  with low weighted error:  

$$\epsilon_t = \sum_{i: h_t(x_i) \neq y_i} D_t(i)$$
If  $\epsilon_t > 1/2$ , then set  $T = t - 1$  and abort loop  
Choose  $\beta_t = \epsilon_t / (1 - \epsilon_t)$   
Update  $D_t$ :  $D_{t+1}(i) = D_t(i) / Z_t \times \begin{cases} \beta_t, & \text{if } h_t(x_i) = y_i \\ 1, & \text{otherwise} \end{cases}$   
Where  $Z_t$  is a factor of normalization (chosen in a way that  $D_{t+1}$  is a distribution)  
The final hypothesis is given as  

$$H(x) = \text{sign} \left( \sum_{t=1}^T \ln(1/\beta_t) h_t(x) \right)$$

ALGORITHM 1

## 4. Experiment Results

**4.1. Experimental Datasets.** The database used in the experiments contains two parts. The first part was done by combining the Caltech car database [21] and some images that are captured manually from different situations, which were used to train the classifier. The second part was collecting the videos in real traffic scenes which are utilized to test the hypothesis generation step and hypothesis verification step. Some of the images contain vehicles and others contain background objects. All images are normalized to  $158 \times 154$  pixels. This paper uses MATLAB R2015b as the software development tool to test the proposed method. The device configuration is 4.0 GB memory DDR4 and 3.40 GHz Intel(R) Core(TM) i5 CPU.

The Caltech car database included 1155 vehicle images from the rear and 1155 nonvehicle images. The real traffic scenes are captured by a camera mounted on the car windshield. The real traffic scenes contain much interference, such as traffic lines, trees, and billboards. Figure 9 shows some examples of the database.

**4.2. Performance Metrics.** To test the proposed system, we collected real traffic videos using a camera mounted on front of a car. The vehicle detection was tested in various environments, and it showed a good rate especially on the highways.

Some results of hypothesis generation using cross-correlation from different image sequences are shown in Figure 10. The trees beside the road and the rear window of a car generate some false hypothesis. However, the purpose of this step was to detect the potential vehicles location regardless of the amount of false candidates generated where the false candidates would be removed in the hypothesis verification step as shown in Figure 11.

To evaluate the performance of the proposed method, the statistical data and the accuracy of various testing cases were recorded and are listed in Table 1. The accuracy is defined as follows:



FIGURE 9: Some vehicle training sample images.

$$\text{accuracy} = \frac{Td}{Td + Mv + Fd} \times 100\%, \quad (6)$$

where  $Td$  is the number of true detections,  $Mv$  is the number of missed vehicles, and  $Fd$  is the number of false detections.

In order to get the best results, we have to look for an efficient classifier where the classification step is the most important step in detection systems. Therefore, we have used and compared two classifiers: SVMs and AdaBoost which are two efficient methods of classification, which have been used to verify and classify the extracted features by using 2D-DWT of the generated hypothesis. The use of these two classifiers gave really efficient results. However, the AdaBoost classifier gave a high accuracy of classification and showed more advantages than the SVM classifier that also showed an important accuracy of generated hypothesis classification. The most missed vehicles are missed due to the overlapping. However, the detection of overlapping vehicles is done successfully based on the percentage of vehicle parts hidden behind other vehicles. If only small part of a vehicle is hidden, it will be generated in the hypothesis generation step and will be detected otherwise it will not be detected. This problem is not very important, and the most important

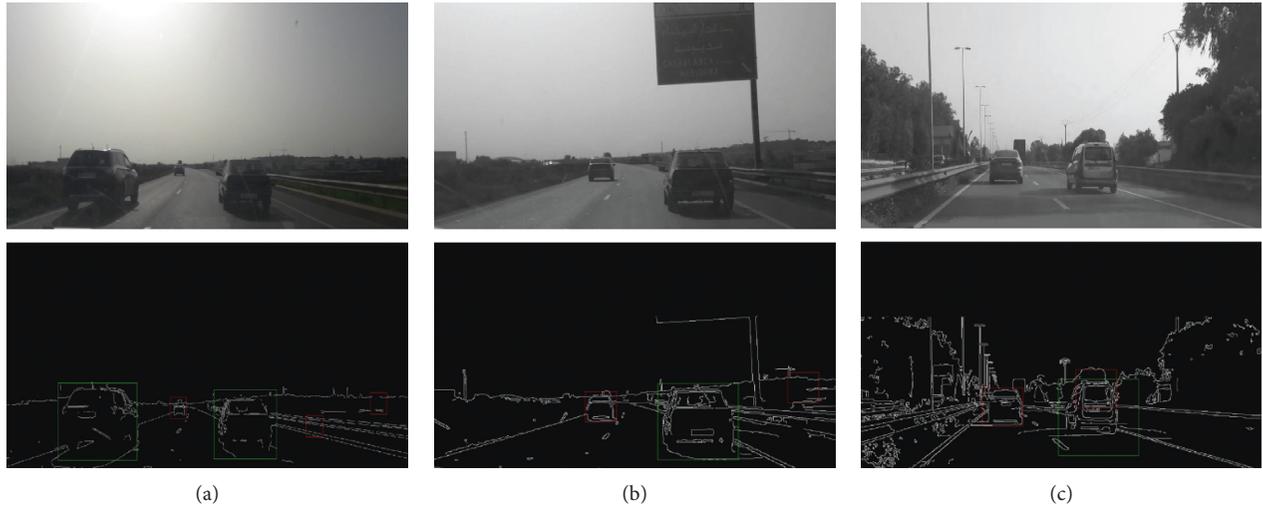


FIGURE 10: Hypothesis generation result after cross-correlation: (a) very close and very far generated candidates; (b) very close and far generated candidates; (c) close and far generated candidates.

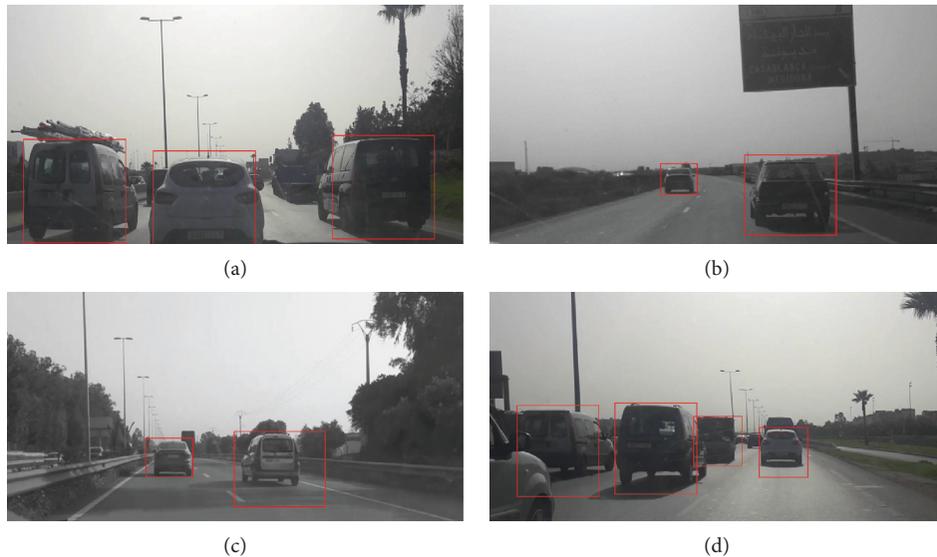


FIGURE 11: The results of hypothesis verification step of the generated candidates.

problem is to detect vehicles directly in front of the current vehicle.

Table 1 shows the results of our vehicle detection system.

**4.3. Evaluation Results.** To evaluate our proposed work, we use three methods to compare with. Yan et al. [1] are based on shadow under vehicle to detect the region of interest and then used histograms of oriented gradients and the AdaBoost classifier for vehicle detection. Tang et al. [7] are based on the Haar-like features and the AdaBoost classifier to detect vehicles which is a very popular method. Ruan et al. [22] focused on wheel detection to detect vehicles. They are based on the HOG extractor and MB-LBP (multiblock local binary pattern) with AdaBoost to detect vehicle's wheels. Table 2 shows the results of three different methods from

TABLE 1: Vehicle detection rates.

Methods	Video sequences				
	1	2	3	4	
Cross-correlation + 2D-DWT + SVMs	TD	98	102	115	121
	MV	2	2	2	4
	FD	2	2	1	2
	Accuracy (%)	96.08	<b>96.23</b>	97.46	95.27
Cross-correlation + 2D-DWT + AdaBoost	TD	99	101	116	123
	MV	1	2	1	2
	FD	1	2	1	1
	Accuracy (%)	<b>98.02</b>	96.19	<b>98.31</b>	<b>97.62</b>

different scenes in different conditions compared to our proposed method results, and this comparison shows that the proposed method has the highest accuracy and confirms

TABLE 2: Evaluation results of 4 vehicle detection methods.

Methods	Video sequences			
	1 (%)	2 (%)	3 (%)	4 (%)
Yana et al. [1]	97.03	95.24	97.48	96.82
Tang et al. [7]	96.08	94.29	96.64	96.06
Ruan et al. [22]	94.23	92.45	94.22	93.79
Proposed method	<b>98.02</b>	<b>96.19</b>	<b>98.31</b>	<b>97.62</b>

that it is able to detect vehicles in different conditions with a high accuracy and efficiency.

## 5. Conclusion

A real-time vehicle detection system using a camera mounted on front of a car is proposed in this paper. We have proposed a solution based on the cross-correlation method. The proposed system included two steps: the hypothesis generation and hypothesis verification steps. Firstly, in the hypothesis generation step, the initial candidate selection is done by using the cross-correlation technique after applying the edge detection to improve the result and reduce the processing time. Then, in the hypothesis verification step, the two-dimensional discrete wavelet transform has been applied on both selected candidates and dataset to extract features. Two famous classifier SVM and AdaBoost have been trained using these extracted features. Based on a comparison of these two classifier results, it was concluded that the AdaBoost classifier performed better in terms of accuracy than SVMs that has also showed an interesting accuracy. The experimental results presented in this paper showed that the proposed approach have good accuracy compared to other methods.

## Data Availability

The data used to support the findings of this study are included within the article [21].

## Additional Points

Our perspectives include the improvement in hypothesis verification step by updating the AdaBoost classifier in order to reduce the processing time and the distance measurement between the detected vehicles and the camera.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## References

- [1] G. Yana, Y. Ming, Y. Yang, and L. Fan, "Real-time vehicle detection using histograms of oriented gradients and AdaBoost classification," *Optik*, vol. 127, no. 19, pp. 7941–7951, 2016.
- [2] S. T. Soo and T. Bräunl, "Symmetry-based monocular vehicle detection system," *Machine Vision and Applications*, vol. 23, no. 5, pp. 831–842, 2012.

- [3] A. Jazayeri, H. Cai, Y. Jiang Zheng, and M. Tuceryan, "Vehicle detection and tracking in car video based on motion model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 583–595, 2011.
- [4] L. Gao, C. Li, T. Fang, and X. Zhang, "Vehicle detection based on color and edge information," in *ICIAR 2008*, Vol. 5112, Springer, Berlin, Germany, 2008.
- [5] J. Li and D. Tao, "Simple exponential family PCA," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 3, pp. 485–497, 2013.
- [6] P. J. Cunningham and Z. Ghahramani, "Linear dimensionality reduction:survey, insights, and generalizations," *Journal of Machine Learning Research*, vol. 16, pp. 2859–2900, 2015.
- [7] Y. Tang, C. Zhang, R. Gu, P. Li, and B. Yang, "Vehicle detection and recognition for intelligent traffic surveillance system," *Multimedia Tools and Applications*, vol. 76, no. 4, pp. 5817–5832, 2015.
- [8] X. Wen, H. Zhao, N. Wang, and H. Yuan, "A rear-vehicle detection system for static images based on monocular vision," in *Proceedings of 9th International Conference on Control, Automation, Robotics and Vision*, pp. 2421–2424, Singapore, March 2006.
- [9] W. Liu, X. Wen, B. Duan et al., "Rear vehicle detection and tracking for lane change assist," in *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 252–257, Istanbul, Turkey, June 2007.
- [10] M. M. Moghimi, M. Nayeri, M. Pourahmadi, and M. K. Moghimi, "Moving vehicle detection using AdaBoost and haar-like feature in surveillance videos," *International Journal of Imaging and Robotics*, vol. 18, no. 1, pp. 94–106, 2018.
- [11] X. Wen, L. Shao, Y. Xue, and W. Fang, "A rapid learning algorithm for vehicle classification," *Information Sciences*, vol. 295, pp. 395–406, 2015.
- [12] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Proceedings of IEEE International Conference on Image Processing*, pp. 900–903, Rochester, NY, USA, January 2002.
- [13] J. Arrospe and L. Salgado, "Log-gabor filters for image-based vehicle verification," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2286–2295, 2013.
- [14] A. Khammari, F. Nashashibi, Y. Abramson, and C. Laugeau, "Vehicle detection combining gradient analysis and AdaBoost classification," in *Proceedings of 2005 IEEE Intelligent Transportation Systems*, pp. 66–71, Vienna, Austria, September 2005.
- [15] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.
- [16] S.-D. Wei and S.-H. Lai, "Fast template matching based on normalized cross correlation with adaptive multilevel winner update," *IEEE Transactions on Image Processing*, vol. 17, no. 11, pp. 2227–2235, 2008.
- [17] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Information Theory Society*, vol. 36, no. 5, pp. 961–1005, 1990.
- [18] I. Slimani, A. Zaarane, and A. Hamdoun, "Convolution algorithm for implementing 2D discrete wavelet transform on the FPGA," in *Proceedings of Computer Systems and Applications (AICCSA), 2016 IEEE/ACS 13th International Conference of IEEE*, pp. 1–3, Agadir, Morocco, November-December 2016.
- [19] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of online learning and an application to boosting," in

*Computational Learning Theory, (Eurocolt)*, Vol. 904, Springer, Berlin, Germany, 1995.

- [20] Y. Freund and E. S. Robert, "Experiments with a new boosting algorithm," in *Proceedings of Thirteenth International Conference*, pp. 148–156, Bari, Italy, July 1996.
- [21] Caltech Cars Dataset: <http://www.robots.ox.ac.uk/~vgg/data3.html>.
- [22] Yu-S. Ruan, I.-C. Chang, and H.-Y. Yeh, "Vehicle detection based on wheel part detection," in *Proceedings of IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, pp. 187-188, Taipei, Taiwan, June 2017.

