

FRONT AND REAR VEHICLE DETECTION USING HYPOTHESIS GENERATION AND VERIFICATION

Nima Khairdoost, S. Amirhassan Monadjemi and Kamal Jamshidi

Department of Computer Engineering, Faculty of Engineering,
University of Isfahan, Isfahan, 81746, Iran
{n.kheirdoost, monadjemi, jamshidi}@eng.ui.ac.ir

ABSTRACT

Vehicle detection in traffic scenes is an important issue in driver assistance systems and self-guided vehicles that includes two stages of Hypothesis Generation (HG) and Hypothesis Verification (HV). The both stages are important and challenging. In the first stage, potential vehicles are hypothesized and in the second stage, all hypotheses are verified and classified into vehicle and non-vehicle classes. In this paper, we present a method for detecting front and rear on-road vehicles without lane information and prior knowledge about the position of the road. In the HG stage, a three-step method including shadow, texture and symmetry clues is applied. In the HV stage, we extract Pyramid Histograms of Oriented Gradients (PHOG) features from a traffic image as basic features to detect vehicles. Principle Component Analysis (PCA) is applied to these PHOG feature vectors as a dimension reduction tool to obtain the PHOG-PCA vectors. Then, we use Genetic Algorithm (GA) and linear Support Vector Machine (SVM) to improve the performance and generalization of the PHOG-PCA features. Experimental results of the proposed HV stage showed good classification accuracy of more than 97% correct classification on realistic on-road vehicle dataset images and also it has better classification accuracy in comparison with other approaches.

KEYWORDS

Vehicle Detection, Hypothesis Generation, Hypothesis Verification, PHOG, PCA, GA, Linear SVM, Feature Weighting

1. INTRODUCTION

Each year, on average, at least 1.2 million people die as a result of worldwide vehicle accidents and they injure at least 10 million people. It is predicted that damage property, hospital bill and other costs associated with vehicle accidents will add up to 1-3 percentage of the world's domestic product [1].

There are at least three reasons for the increasing research in this area: 1) the statistics show that most deaths in vehicle accidents caused by car collision with other vehicles, 2) improved machine vision algorithms, 3) availability of low cost high computational power [1]. Consequently, the development of on-board automotive driver assistance systems with aiming to alert a driver about possible collision with other vehicles and also driving environment has attracted a lot of attention over the last 20 years among vehicle manufactures, safety experts and universities. Several national and international companies have launched over the past several years to research new technologies for reducing accidents and improving safety [2].

Robust and reliable vehicle detection in images is the critical step for these systems and self-guided vehicles as well as traffic controllers. This is a very challenging task since it is not only affected by the size, shape, color, and pose of vehicles, but also by lighting conditions, weather, dynamic environments and the surface of different roads. A vehicle detection system must also

distinguish vehicles from all other visual patterns which exist in the world, such as similar looking rectangular objects [3].

Almost every vehicle detection system includes two basic stages: 1) Hypothesis Generation (HG) which hypothesized all regions in the image that potentially contain a vehicle, and 2) Hypothesis Verification (HV) which verifies the hypotheses [4,5].

Various HG methods have been suggested in the literature and can be classified in three basic categories [1]: 1) knowledge-based, 2) stereo-based and 3) motion-based. Knowledge-based methods employ information about color and vehicle shape as well as general information about the context such as: a) shadow [6,7], b) symmetry [8,9,10], c) horizontal/vertical edges [11,5,12], d) color [13,14], e) texture [15,16], and f) vehicle lights [17,18]. Stereo-based approaches usually employ the Inverse Perspective Mapping (IPM) to estimate the locations of people, vehicles and obstacles [19,20,21] in the images. Motion-based approaches detect object such as people, vehicles and obstacles using optical flow [22,23]. However, generating a displacement vector for each pixel is a time-consuming task and impractical for real time systems. To attack this problem, discrete methods employ the image features such as color blobs [24] or local intensity minima and maxima [25].

In the HV stage, correctness of hypotheses are verified and sorted into vehicle and non-vehicle classes. The HV approaches can be divided into two categories [1]: 1) template-based and 2) appearance-based. The template-based methods employ the predefined patterns of the vehicle class and perform correlation between the template and the image. In [26], a HV algorithm was proposed based on the presence of the license plates and rear windows. This can be considered as a loose template of vehicle class. Handmann et al. [27] attempted to employ the template of 'U' shape which describes the bottom and sides edge of a vehicle. During verification, if they could find the 'U' shape, the image region was considered as a vehicle.

In the appearance-based methods, the characteristics of the vehicle appearance are learned from a set of training images which capture the variability in the vehicle class. Usually, the variability of the non-vehicle class is also modelled to improve performance. To begin, each training image is presented by a set of global or local features [4]. Then, the decision boundary between vehicle and non-vehicle classes is learned either by training a classifier (e.g. Support Vector Machine, Adaboost and Neural Network) or by modelling the probability distribution of the features in each class (e.g. employing the Bayes rule assuming Gaussian distributions) [28,29,30]. In [31], Principle Component Analysis (PCA) was used for feature extraction and linear Support Vector Machine (SVM) for classification of vehicle images. Goerick et al. [32] employed Local Orientation Code (LOC) to extract the edge information of ROI and NNs to learn the characteristics of vehicles. In [33], a multilayer feedforward neural network-based method was proposed with the linear output layer for vehicle detection. Features extraction by application of Gabor filters was investigated in [34], Gabor filters provide a mechanism to extract the line and edge information by tuning orientation and changing the scale. In [35], an Adaboost classifier [36] trained on Haar features was used to classify detections. Papageorgiou and Poggio [37] have presented by acquisition of Haar wavelet transform for feature extraction and SVMs for classification. In [12], multiple detectors were built with employing Haar wavelets, Gabor filters, PCA, truncated wavelets, and a combination of wavelet and Gabor features using SVM and neural networks classifiers. A comparison of feature and classifier performance was presented, the conclusion was the feature fusion of the Haar and Gabor features can result in robust detection. In [38], a similar work was performed. Negri et al. [38] compared the performance of vehicle detectors with Adaboost classification that was trained using the Haar-like features, a histogram of oriented gradient features, and a fusion of them. The conclusion was that a feature fusion can be valuable. A statistical method was used in [39], performing vehicle detection employing PCA and independent component analysis (ICA) to classify on a statistical model and its speed was increased by modelling the PCA and ICA vectors with a weighted Gaussian mixture

model. In [40], a general object detection scheme was proposed using PCA and Genetic Algorithm (GA) for feature extraction and feature subset selection respectively.

In this paper, in the HG stage, three different knowledge-based cues are applied to take the advantages of all the cues and also besides these, the data are prepared for the HV stage. Also, in the HV stage, a four-step method is used to optimize the features and rise the classification accuracy. The rest of the paper is organized as follows: in Section 2, the proposed method is described in detail. Our experimental results and comparisons are mentioned in section 3. The last section is the conclusion and future works.

2. PROPOSED METHOD

As before, the vehicle detection system includes two stages of the hypothesis generation and verification, in this section we describe the solutions that are employed in two stages.

2.1. Hypothesis Generation (HG)

Three different clues of shadow, texture and symmetry are used in this stage. The procedure starts with detecting shadow underneath a vehicle and using the aspect ratio, the ROI is determined. The advantage of shadow is that all of vehicles will be detected [41,42]. In the next step, the ROI is tested whether it has enough entropy or not. For this purpose, the rows with low entropy are removed from the ROI. If too few rows remain, the detected shadow is rejected (the shadow does not belong to a vehicle). Finally, the horizontal symmetry of the remaining rows of the ROI is verified. In this step, asymmetric ROI is classified as background (which means the detected shadow is rejected) and symmetric ROI is considered as a hypothesis. In this step, besides separating the symmetric ROIs from background, the boundaries of the symmetric ROI is modified and it also improves the performance of the HV stage. This stage was inspired by the work of [42]. The three mentioned steps are described in the following.

2.1.1. Shadow

Shadow is the first clue that is searched in the image. Using shadow, the positions of the present vehicles can be indicated that is based on the fact that the shadow underneath a vehicle is darker than the observed surface of the road. After detecting the shadow underneath a vehicle, the region above the shadow is considered as a ROI that will be further analyzed.

The presented work in [7] forms the basis for detecting the shadow underneath a vehicle. There is no lower boundary for the intensity of the shadow underneath a vehicle but based on the intensity distribution of the road surface, an upper boundary can be defined for it, although it will not be fixed. The value of this threshold depends on the color of the surface of the road and its illumination.

The intensity distribution of the surface of the road is estimated without pre-knowledge about the location of the road in the image. It is achieved by means of a simple algorithm for detecting the free-driving-space. The free-driving-space is defined as the part of the observed road directly in front of the camera. For estimation of the free-driving-space, to begin, edges in the image are estimated, then the space is discovered, defined by the lowest central homogeneous region in the image and delimited by the edges (Figure 1).

We assume that the intensity values of the road surface are normally distributed and estimate the mean value m and variance σ of the distribution. The upper bound for the shadows can be considered $\text{Thresh}_{\text{sh}} = m - 3\sigma$. Although there are alternative approaches to estimate the free-driving-space with more refinement but this procedure is sufficient to estimate m and σ for our application.

For detecting the shadows underneath vehicles we need to estimate the edges that correspond to the transitions from the road surface to the dark areas underneath the vehicles. These edges are

usually horizontal. For this purpose, to begin, the points of the image are found that their intensity values are lower than the threshold (the dark areas). Next, within the image is searched for vertical transitions from brighter intensity values to the darker ones (scanning the image bottom-up). This operation can be done efficiently by simply shifting the image vertically and then subtracting the two images (Figure 2). By this implementation, the obtained image has fewer horizontal edges than in the case of using a normal edge detector.

The equation which gives the thresholded image is as follows.

$$D(u, v) = \begin{cases} 1: I(u, v) \leq \text{Thresh}_{sh} \wedge |I(u, v+1) - I(u, v)| \geq \text{Thresh}_v \\ 0: \text{else} \end{cases} \quad (1)$$

where Thresh_v is the intensity-value difference for the vertical edge estimation. The horizontal line segments are discovered in two successive lines. By thresholding the length of a line segment, the lines belonging to potential vehicles can be roughly separated and the remaining line segments are considered the background. Finally, the region above detected shadow is considered as a ROI. The ROI is estimated by the aspect ratio and slightly wider regions are used for the further analysis of each ROI (Figure 2.c).

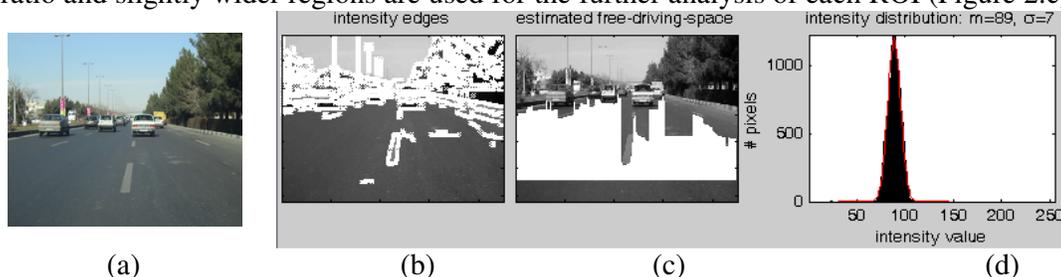


Figure 1. Estimation procedure for the intensity distribution of the 'free-driving-space'. (a) the original image. (b) the edges estimation in the image. (c) the rough estimation of the free-driving-space. (d) the histogram of the intensity distribution of the free-driving-space. A Gaussian is fit to this histogram.

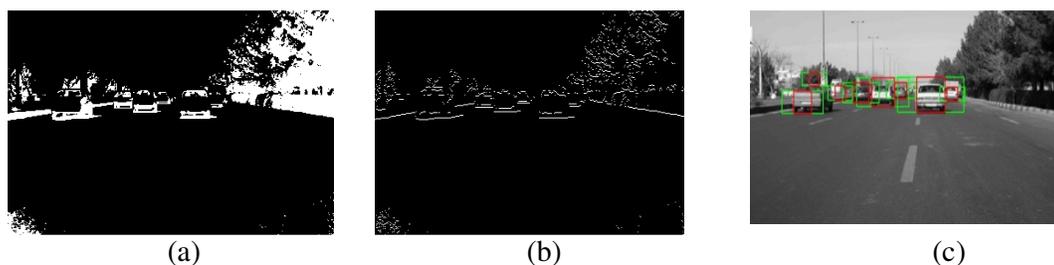


Figure 2. Hypotheses generated based on shadow. (a) pixels identified as shadow based on Thresh_{sh} . (b) the shadow points based on Thresh_{sh} and Thresh_v that exhibit a vertical transitions from a brighter intensity values to the darker ones (scanning the image bottom-up). (c) scanning the image bottom-up, a ROI is defined above horizontal detected shadow presented in Figure b. A ROI is only further analyzed if its size indicates the possibility to contain a vehicle and in this case slightly wider regions (green boxes) are used for the ROIs for the further analysis.

2.1.2. Texture

Local entropy based on the information theory presented by Shannon in [43] can be considered as a measure for the information content. In [15], texture was employed on the grounds that their vehicle detection algorithm intended to focus on the parts in the image with high information content.

For our application, the entropy is employed to investigate the ROIs that are detected by the previous step (shadow). If a vehicle is observed in the ROI, it is expected in horizontal direction in between boundaries of the ROI, the estimated entropy will be high. For this purpose, the local entropy is computed along the lines of the ROI and the lines that indicate lack of entropy are removed from the ROI. If the number of remaining lines is lower than a threshold, the detected shadow line will be removed which means, it does not belong to a vehicle. In this case, we do not remove the whole of the segment lines in the ROI since a part of another potential vehicle may exist in the ROI and by removing its segment lines, the vehicle may be not detected by the shadow step in the further analysis.

The entropy is defined as follows:

$$H(r_x) = -\sum_k p(r_x) \log p(r_x) \quad (2)$$

where $p(r_x)$ is a probability distribution of r_x that in here is local intensities. The intensity distribution $p(r_x)$ of a line in a ROI is derived from its histogram of intensities.

2.1.3. Symmetry

Good horizontal symmetry can be obtained in uniform background areas. Therefore, estimating their symmetry would be useless. The preceding steps of the HG procedure already reject the uniform regions before entering to the symmetry step. Besides this, these steps prepare the data for the symmetry step in such a way that this step can be done efficiently. Furthermore, employing the preceding steps causes the data which should be analyzed in the symmetry step are decreased. This makes the process of finding the axis and width of the symmetry efficient.

In here, the estimation method of the local horizontal symmetry will be explained briefly. The notation is as follows:

$G(u)$: a one dimensional function defined over a line of the ROI (that obtained from the texture step)

w_{max} : the width of the ROI

w : the width of the symmetry interval

x_s : location of a potential symmetry axis

Any function $G(u)$ can be broken into a sum of its even part $G_e(u)$ and its odd part $G_o(u)$:

$$G(u) = G_e(u) + G_o(u)$$

$$G_e(u) = \frac{G(u) + G(-u)}{2} \quad (3)$$

$$G_o(u) = \frac{G(u) - G(-u)}{2}, \quad u \in [-w_{max}/2, w_{max}/2]$$

For shifting the origin of the function $G(u)$ to any potential symmetry axis x_s , substitution $u = x - x_s$ is used. The even function of $G(x) = G(x - x_s)$ for a given interval of width w about symmetry axis x_s is defined as:

$$E(x, x_s, w) = \begin{cases} (1/2)(G(x - x_s) + G(x_s - x)) & \text{if } |x - x_s| \leq w/2 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

The odd function of $G_o(x)$ is defined as:

$$O(x, x_s, w) = \begin{cases} (1/2)(G(x - x_s) - G(x_s - x)) & \text{if } |x - x_s| \leq w/2 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

For any pair $\{x_s, w\}$, the values of either $E(x, x_s, w)$ or $O(x, x_s, w)$ are expressed appropriately by their energy content. However, there is a problem in this case since the mean value of the odd function is always zero, whereas the mean value of the even function in general is some positive number. Therefore a normalized even function that its mean value is zero is defined as follows:

$$E_n(x, x_s, w) = E(x, x_s, w) - \frac{1}{w} \int_{-w/2}^{w/2} E(x, x_s, w) dx \quad (6)$$

The normalized measure is constructed with E_n and O for the degree of symmetry $S(x_s, w)$:

$$S(x_s, w) = \frac{\int E_n(x, x_s, w)^2 dx - \int O(x, x_s, w)^2 dx}{\int E_n(x, x_s, w)^2 dx + \int O(x, x_s, w)^2 dx} \quad (7)$$

$S(x_s, w)$ indicates the measure of symmetry for any symmetry axis x_s with the symmetry width w which has the following property:

$$-1 \leq S(x_s, w) \leq 1 \quad (8)$$

Furthermore, $S = 1$ in the case of ideal symmetry, $S = 0$ for asymmetry and $S = -1$ for ideal anti-symmetry.

$S(x_s, w)$ provides us the measure for symmetry regardless of the width of the interval being considered. For example, in the case of estimating the same measure of symmetry for two different axis of symmetry, one should select the estimation corresponding to the largest width w . To account the width, the measure $SA(x_s, w)$ is defined:

$$SA(x_s, w) = \frac{w}{2w_{\max}} (S(x_s, w) + 1), \quad w < w_{\max} \quad (9)$$

For detecting two dimensional symmetry, $SumSA(x_s, w)$ should be computed as follows:

$$SumSA(x_s, w) = \sum_{l=1}^n SA_l(x_s, w) \quad (10)$$

where n is the number of the ROI lines. For each symmetry axis, w is increased to w_{\max} and the maximum value of $SumSA(x_s, w)$ is recorded. After doing this for each symmetry axis, we find the maximum value of the recorded values. The x_s and w corresponding to the maximum value will be the symmetry axis of the ROI and the interval width respectively which are denoted as $\{\hat{x}_s, \hat{w}\}$. Then, we look for more accurate location of the symmetrical part observed within the ROI and its corresponding measure of symmetry. For example, more areas close to the boundaries of the ROI belong to the background instead of the vehicle. For this purpose, we remove the lines in the lower and upper quart of the ROI which exhibit relatively low symmetry values $SA(x_s, w)$.

With denoting the height of the ROI by h , we determine the upper and lower boundaries for the refined symmetry region:

$$i_{upper} = \max_{i \in [\frac{3}{4}h, h]} \left(\sum_{j=\frac{3}{4}h}^i \frac{SA_j(\hat{x}_s, \hat{w})}{i - \frac{3}{4}h + 1} \right) \quad \text{and} \quad i_{lower} = \max_{i \in [1, \frac{1}{4}h]} \left(\sum_{j=i}^{\frac{1}{4}h} \frac{SA_j(\hat{x}_s, \hat{w})}{\frac{1}{4}h - i + 1} \right) \quad (11)$$

Finally, we modify the estimation of the interval width. For this purpose, \hat{w} is increased to w_{\max} and the w corresponding to the maximum value of $SA(\hat{x}_s, w)$ will be the modified symmetry width. This provides us with the smallest box that contains the symmetrical part observed within the ROI. We call this box *Sym.ROI*. We use (12) to find the symmetry measure of the *Sym.ROI* that is denoted as *SM.ROI*.

$$SM.ROI = \sum_{j=i_{lower}}^{i_{upper}} \frac{SA_j(\hat{x}_s, \hat{w})}{i_{upper} - i_{lower} + 1} \quad (12)$$

If $SM.ROI$ value is lower than a pre defined threshold $Thresh_{sym}$, the detected shadow line will be removed that means $Sym.ROI$ contains no vehicle and otherwise $Sym.ROI$ will be considered as a hypothesis. Figure 3 shows a detailed block diagram of the proposed HG stage.

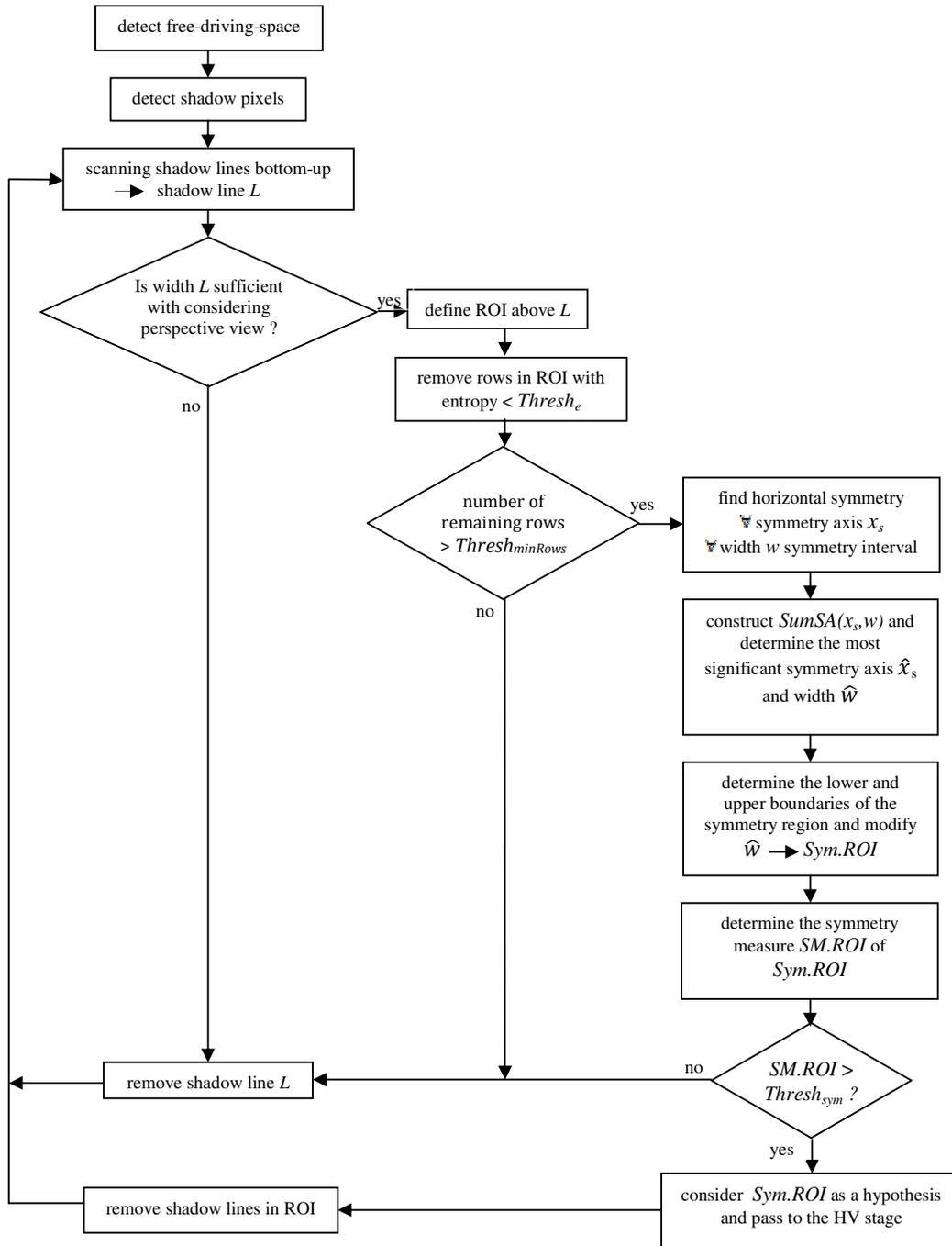


Figure 3. the HG stage (including the shadow, texture and symmetry steps)

2.2. Hypothesis Verification (HV)

The framework in this stage is feature extraction from the hypotheses and classification them into vehicle and non-vehicle classes. Therefore the performance of this stage is directly depended to employ a classifier that is well trained by the appropriate features. For achieving this purpose, we propose a framework that is shown in Figure 4. Pyramid Histograms of Oriented Gradients (PHOG) features are extracted from an image dataset as the primitive features since they have shown good results in object detection [40], facial expression recognition [44], human motion classification [45] and image categorization [46]. Then Gaussian low-pass filter is applied on the image. Following this, the size of the obtained image is reduced and the PHOG features are extracted again from this image. This work will be to improve the classification accuracy since it leads to extract other effective features from the image. To improve the classification accuracy more and reduce the dimensionality, we also apply PCA to these PHOG features to generate what we call the PHOG-PCA feature vector. Then, we divide the samples into two parts of *Training Data* and *Test Data* as shown in Figure 4.

It is well known that feature weighting is effective for pattern classification as shown in [47,48,49]. It is expected that the classification accuracy can be further improved by weighting the proper first PHOG-PCA features since some local regions are less relevant for vehicle detection than the others. For this purpose, we use a GA feature weightener. The *Training Data* is divided into two parts of *data1* and *data2*. We employ linear SVM for vehicle/ non-vehicle classification which is trained with the *data1* and then the *data2* is used for validation of the classifier. The classification accuracy is returned to the GA as one of the fitness factors. After the convergence of the GA, linear SVM is trained regarding the *Optimum Weights* and the *Training Data*. Next, we test it with the *Test Data* and the classification accuracy of the proposed HV stage is obtained. The overview of the HV stage is shown in Figure 4.

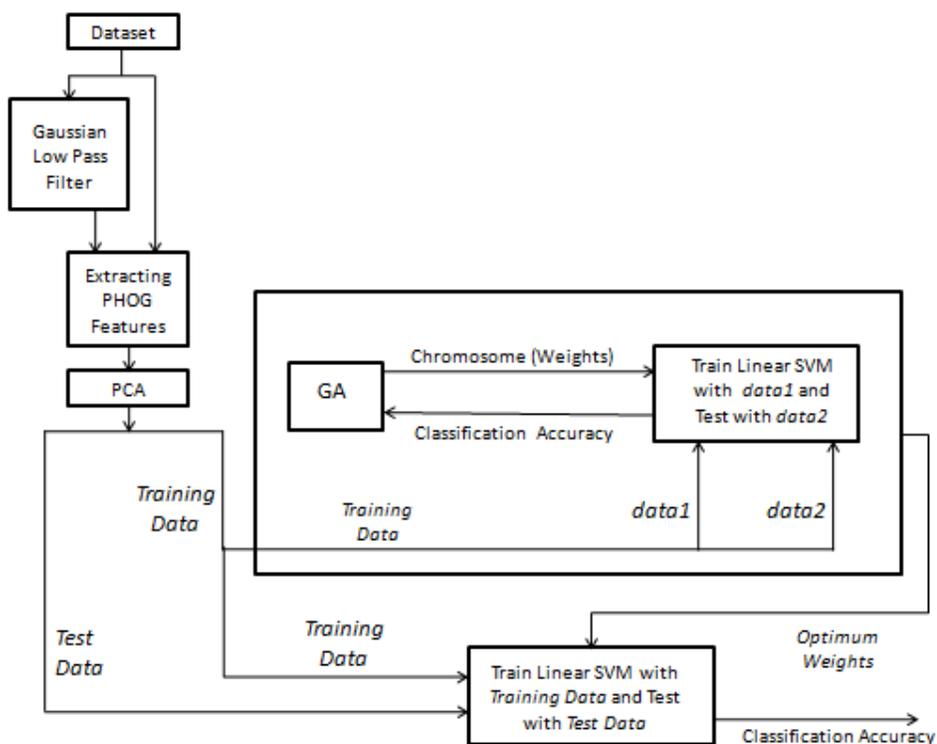


Figure 4. the HV stage

2.2.1. Pyramid Histograms of Oriented Gradients (PHOG)

PHOG descriptor is a spatial pyramid representation of HOG descriptor, and reached good performance in many studies, e.g. [50,51,52]. In this paper, the PHOG features are extracted from vehicle and non-vehicle samples to represent by their local shape and spatial layout. As illustrated in Figure 5, the PHOG descriptor consists of a histogram of orientation gradients over each image sub-region at each resolution.

For extracting the PHOG features, the edge contours are extracted by acquisition of the Canny edge detector for entire image as shown in Figure 5. Following this, each image is divided into cells at several pyramid level. The grid at resolution level l has 2^l cells along each dimension. The orientation gradients are computed using a 3×3 Sobel mask without Gaussian smoothing. Histogram of edge orientations within each cell is quantized into K bins. Each bin in the histogram represents the number of edges that have orientations within a certain angular range. The histograms of the same level are concatenated into one vector. The final PHOG descriptor for an image is a concatenation of all vectors at each pyramid resolution that introduces the spatial information of the image [50]. Consequently, level 0 is represented by a K -vector corresponding to the K bins of the histogram, level 1 by a $4K$ -vector, and the PHOG descriptor of the entire image is a vector with dimensionality $K \sum_{l \in L} 4^l$. The PHOG descriptor is normalized to sum to unity that ensures images with more edges are not weighted more strongly than others. Figure 5 shows the PHOG descriptor procedure and the PHOG features of the example images. As can be seen, vehicle images have similar PHOG representations whereas non-vehicle images have different PHOG representations far enough from the vehicle ones.

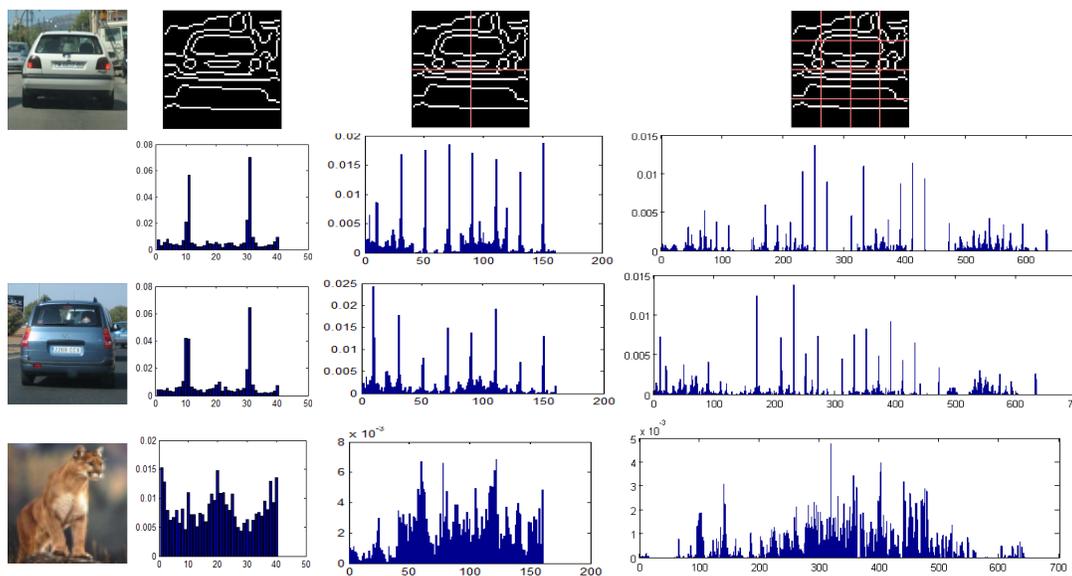


Figure 5. Shape spatial pyramid representation. Top row: a vehicle image and grids for levels $l = 0$ to $l = 2$; Below: histogram representations corresponding to each level. The final PHOG vector is a weighted concatenation of vectors (histograms) for all levels. Remaining rows: another vehicle image and a non-vehicle image, together with their histogram representations.

2.2.2. Gaussian Low-pass Filter

By eliminating the high frequencies of the image, the image is blurred. Gaussian low-pass filter does this by beneficiary of a Gaussian function. It is widely used to reduce the image noise and detail. It is also used in computer vision as a pre-processing step in order to enhance image structures at different scales. In two dimensions, the Gaussian low-pass filter can be expressed as:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (13)$$

where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution. To construct Gaussian low-pass filter, two parameters x and s are used. The values of x and s indicate the size of the filter mask and the sigma respectively, and the sigma indicates the filter frequency. Any value of x is more, the filter mask will be bigger and also any value of s is more, the filtered frequency will be increased. Figure 6 shows the results of applying Gaussian low-pass filter on three sample vehicle and non-vehicle images and also the PHOG features representation of the filtered images.

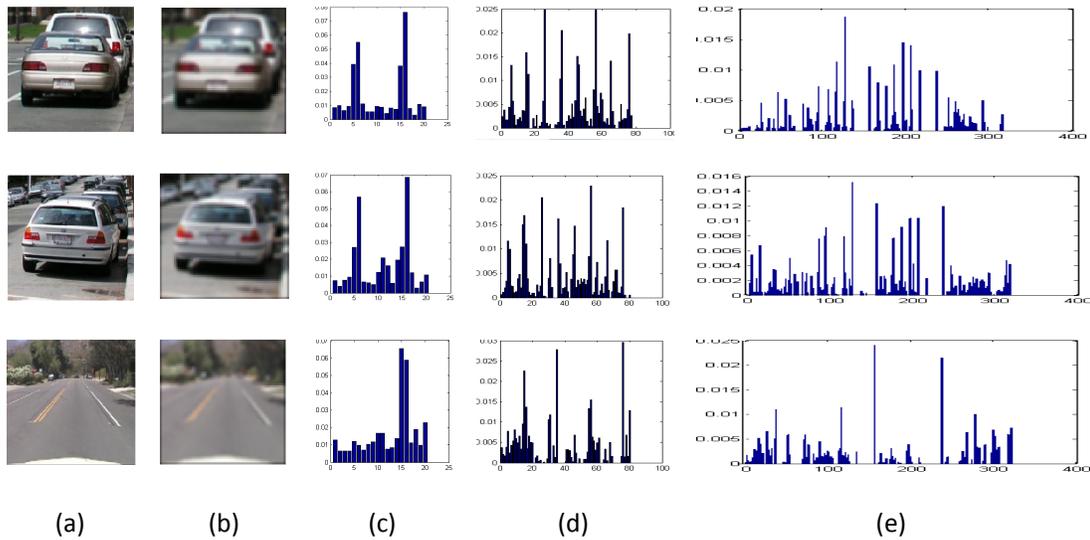


Figure 6. the PHOG features representation for sample vehicle and non-vehicle images after applying Gaussian low-pass filter. Column (a): the original vehicle and non-vehicle images; Column (b): the results of applying Gaussian low-pass filter on the images of column (a); Columns (c),(d) and (e): the PHOG features representation of the corresponding filtered images of column (b).

2.2.3. Principal Component Analysis (PCA)

The total number of the extracted PHOG features is rather high. Also, these features are likely irrelevant and redundant. PCA was applied in [53,54] for reducing the dimensionality of the feature vectors. PCA can be defined as the orthogonal projection of the input data onto a lower dimensional linear subspace, such that the variance of the projected samples is maximized. Dimension reduction and noise reduction are two advantages of employing PCA. In this paper, we utilize this idea to reduce the dimensionality of the feature vectors. The PCA algorithm can be summarized in the following:

Let $\{x_i | i = 1, \dots, N\}$ be a set of M -dimensional vectors. We compute the mean vector of input vectors that is defined as $\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$ and then we compute the covariance matrix Σ that is defined as follows:

$$\Sigma = \frac{1}{N} \sum_{n=1}^N (x_i - \bar{x})(x_i - \bar{x})^T \quad (14)$$

By solving the eigen equations of the covariance matrix Σ , the optimum projection matrix U is obtained

$$\Sigma U = \Lambda U, \quad (UU^T = I) \quad (15)$$

and then the PCA scores for any PHOG features can be computed by using the following equation. We called these new features PHOG-PCA features.

$$y = U^T(x_i - \bar{x}) \quad (16)$$

In order to reduce the dimensionality, we just keep the first d principal axis that they keep the significant discriminant information.

2.2.4. Genetic PHOG-PCA Feature Weighting

GA is a probabilistic optimization algorithm and a branch of evolutionary algorithms. In the past, it has been used to solve different problems such as object detection [40], face recognition [55,56], vehicle detection [4], image annotation [57], gender classification [58] and target recognition [59].

In this study, we utilized the GA for the PHOG-PCA features weighting to reduce the classification error of the classifier. Thus, we formed a population consisting of the chromosomes representing the weights for the features of the two classes of vehicle and non-vehicle and used them in the GA process. The best chromosome is the one leading to the lowest test classification error. The procedure in finding the optimum weights via the GA is as follows:

- 1) *Feature weighting encoding*: Let the number of the PHOG-PCA features be L , so each chromosome will be represented with L genes that each gene take values from the range of [0-5], which in our study is divided into 10 discrete levels.
- 2) *Calculating the fitness of these chromosomes*: We forced weights 0.5 in value to 0 during our trial. These embellishments resulted in GA-optimized classifiers with reduced feature sets. With some training data and regarding non-zero weights, the linear SVM is trained using the chromosome whose fitness value is to be calculated. Then, some test data is presented to the trained classifier and classification accuracy is calculated in percentage form. The fitness function is as follows:

$$\text{Fitness}(c) = CA^4(c) - \alpha \left(\frac{N(c)}{L} \right) \quad (17)$$

where c is the chromosome, $CA(c)$ is the classification accuracy using the linear SVM classifier. α represents the tradeoff between the two criteria (using $\alpha = 0.01$). $N(c)$ is the number of non-zero weights. Finally, L is the total number of the features (which is fixed at 315 for all experiments). In our experiments, the classification accuracy is often more than 75%. So we used $CA^4(c)$ instead of $CA(c)$ because it can be more distinctive fitter chromosome than others.

- 3) *Initial population*: All the genes of the first chromosome are '5', which means the weights of all the features are equal. The other chromosomes are generated randomly. In all of our experiments, we used 1000 generations and a population size of 800. In most cases, the GA converged in less than 1000 generations.
- 4) *Crossover*: We used uniform crossover, in this case each bit of the offspring is selected randomly from the corresponding bits of the parents. The crossover rate used in all of our experiments was 0.9.
- 5) *Mutation*: We choose uniform mutation that is, each bit has the same mutation probability. The mutation rate used in all of our experiments was 0.08.
- 6) *Elitism*: We used the elitism strategy to prevent fitness of the next generation be smaller than the largest fitness of the current generation, the best 40 chromosomes are preserved for the next generation automatically.

3. EXPERIMENTAL RESULTS

3.1. Dataset

The vehicle dataset used contains 1646 non-vehicle images and 1648 front and rear view vehicle images. Some of these images are from the MIT vehicle dataset and the Caltech-101 dataset, while the rest images have been gathered with different types, poses and colors (although all images were converted to grayscale). Some of the images contain the vehicle and other background objects. We converted all images to jpg format and normalized size of each image to 128×128 pixels (see Figure 7).



Figure 7. Some vehicle and non-vehicle training sample images

3.2. Experiments

In our experiments, we used the linear SVM classifier and the extracted PHOG features from all collected images with 3 levels of pyramids and 40 orientation bins in the range of [0, 360] in each level. Therefore, the 3 level PHOG descriptor of an image is an 840-vector.

Also, we used a 7-fold cross-validation to estimate both the accuracy and generality of the linear SVM classifier. In this case, all of the examples are partitioned into 7 subsamples and the 7th subsample is retained as *Test data* while the remaining 6 subsamples are used as *Training Data*. The cross-validation is then repeated 7 times with all of the 7 subsamples used exactly once as the *test data*. It should be mentioned that with the aim of comparing the results of the different steps of the HV stage, we used the same folds in all following experiments for the cross validation. In the first experiment, we applied the PHOG descriptors and the linear SVM classifier. Table 1 shows the result.

Table 1. the classification results with the PHOG features extracted from the dataset images

Number of Features	True Positive (%)	True Negative (%)	Classification Accuracy (%)
840	96.06	92.59	94.32

In second experiment, to improve the classification accuracy, the Gaussian low-pass filter is applied on the dataset images and then the size of the obtained images is reduced to 64×64 pixels. Next, the PHOG features are extracted again from these images. Table 2 compares the results of the classification using the linear SVM classifier and the extracted PHOG features from the dataset images as well as the filtered images. In this table, K , L , X and S are the number of bins, the number of pyramid levels, the size of the filter and the standard deviation respectively.

According to Table 2, we find that employing the PHOG features extracted from the dataset images and also the filtered images will climb the classification accuracy compared with employing only the PHOG features extracted from the original dataset images. As another result of this table, exploiting the features extracted with the 3 levels of pyramids has always better performance compared with using the 2 levels of pyramids. It is also observed while using the 3

levels of pyramids, it is often better to use $k=20$ rather than $k=40$. As final result in this step, the best classification accuracy (95.69 %) is obtained using $k=20$, $L=2$, $X=5$ and $S=5$. In this case, the total number of features is equal to $(840+420=)1260$ and we apply these features in the next step.

Table 2. the classification results with the PHOG features extracted from the dataset images and also the filtered images

K	L	X	S	True Positive (%)	True Negative (%)	Classification Accuracy (%)
20	2	10	15	97.45	93.68	95.57
20	1	10	15	96.84	93.01	94.93
20	2	10	10	97.51	93.68	95.60
20	1	10	10	96.72	92.89	94.81
40	2	10	15	97.52	93.59	95.56
40	1	10	15	96.97	92.95	94.96
40	2	10	10	97.21	94.11	95.66
40	1	10	10	96.91	92.77	94.84
20	2	5	5	97.51	93.86	95.69
20	1	5	5	97.09	92.77	94.93
40	2	5	5	97.27	94.06	95.67
40	1	5	5	96.84	92.89	94.87
20	2	5	10	97.51	93.74	95.63
20	1	5	10	97.21	93.30	95.26
40	2	5	10	97.33	93.80	95.57
40	1	5	10	97.27	93.13	95.20
20	2	5	2	97.21	94.11	95.66
20	1	5	2	96.78	93.01	94.90
40	2	5	2	97.09	94.20	95.65
40	1	5	2	96.72	93.07	94.90

It should be mentioned that for increasing the classification accuracy more, the Gaussian low-pass filter was applied again on the filtered images and then the PHOG features were extracted from them but in this case, the classification accuracy was not increased and even it often led to reduction of the classification accuracy so we avoid mentioning these experiments and their results.

In third experiment, we applied PCA to the PHOG features for improving the classification accuracy more and reducing the dimensionality in two different cases. In the first case, PCA was applied to the PHOG features extracted from the original dataset images. In the second case, PCA was applied to the PHOG features extracted from the original dataset images and also the filtered images. Table 3 shows the results corresponding to the second case with employing different number of the first PHOG-PCA features. The best classification accuracy in the first case was obtained equal to 96.05 % with employing the 370 first PHOG-PCA features (from 840 features). Since the second case resulted in the better results, we avoid mentioning other results for the first case.

The results of the two cases show that employing PCA in general will amplify the classification accuracy. In the second case (Table 3) with employing the 315 first PHOG-PCA features (from 1260 features), we achieve the best classification accuracy (96.84 %) that is better than the best classification accuracy in the first case. In the second case, besides increasing classification accuracy, another significant point is, even though the number of features is more than the first case but the number of effective features has decreased from 370 to 315 features that is shown employing the features extracted from the filtered images is a good idea so we use these features in the next step.

Table 3. the classification results with employing PCA to reduce the dimensionality of the PHOG descriptors extracted from the dataset images as well as the filtered images

Number of Features	True Positive (%)	True Negative (%)	Classification Accuracy (%)
100	97.63	94.71	96.17
200	97.88	95.44	96.66
250	98.12	95.38	96.75
270	98.12	95.38	96.75
300	98.12	95.50	96.81
315	98.18	95.50	96.84
330	98.18	95.50	96.84
350	98.18	95.50	96.84
400	98.18	95.50	96.84
450	98.05	95.52	96.79
500	98.05	95.52	96.79
600	98.05	95.46	96.76

In fourth experiment, we used the GA with the configuration mentioned in section 2.2.4 to optimize the weights of these features. We also used the same folds in the previous experiments for the cross validation and in this case, we used 5 folds of *Training Data* (that we called *Data1*) for training of linear SVM and 1 fold of *Training Data* (that we called *Data2*) for validation of the learned classifier to guide the GA during the weight optimization. After the convergence of the GA, we trained the linear SVM with the *Training Data* regarding the *optimum weights* and tested it by application of *Test Data*. Applying the feature weighting led to decline in the number of the features from 315 to 303 as well as improvement of the classification accuracy by 0.92 %. Table 4 shows the result.

Table 4. the classification results with employing the GA to weight the PHOG-PCA features (proposed HV method)

Number of Features	True Positive (%)	True Negative (%)	Classification Accuracy (%)
303	98.48	97.03	97.76

Figure 8 shows the results of applying our proposed method on some sample images.

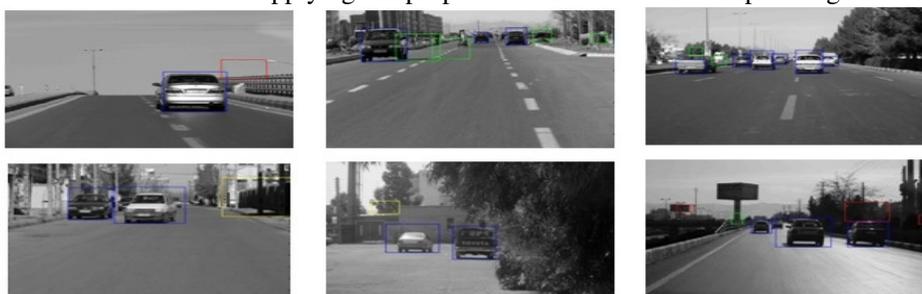


Figure 8. Applying our proposed method on some sample images to detect on-road vehicles. Red rectangle: The ROI has been considered as non-vehicle by the entropy step; Yellow rectangle: The ROI has passed the entropy step but it has been considered as non-vehicle by the symmetry step; Green rectangle: The ROI has passed the symmetry step too but it has been considered as non-vehicle by the classifier; Blue rectangle: The ROI has passed all the steps and it has been classified as a vehicle

The results of applying our proposed method on another sample image is shown in Figure 9. As can be seen, the vehicle in the right side has been considered as non-vehicle mistakenly by the entropy step since the vehicle does not have a rich texture, (it has few edges) and also a non-vehicle ROI in the left side has been considered as vehicle mistakenly.

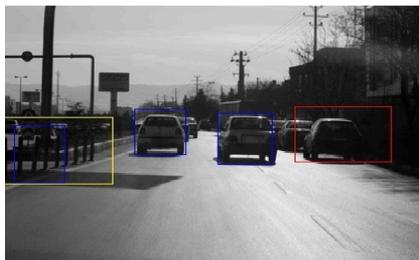


Figure 9. Two examples of false detections

With the aim to investigate more about the accuracy of the proposed HV stage, we compare it with the methods presented in [40,60]. We implemented the method presented in [40]. Next, we applied it on our dataset images. In [40], to begin, the features are extracted from the dataset images by PCA and then the images are classified by the SVM classifier. Table 5 shows the results of classifying with the different number of the first PCA features.

Table 5. the classification results using the different number of the first PCA features

Number of Features	True Positive (%)	True Negative (%)	Classification Accuracy (%)
50	88.77	87.31	88.04
100	89.02	88.56	88.79
150	89.62	89.06	89.34
200	90.47	89.35	89.91
250	91.09	89.80	90.45
300	90.47	89.77	90.12

In [40], after extracting the features, the genetic algorithm is employed to select features from the proper first PCA features. According to Table 5, the 250 first PCA features shows the better performance and since still other useful features may be between 250 to 300 first features so we used the GA to select an optimum subset of the 300 first PCA features. It should be mentioned that we also used the same folds in the previous experiments for the cross validation. Table 6 shows the result.

Table 6. the classification result using the selected PCA features by the GA

Number of Features	True Positive (%)	True Negative (%)	Classification Accuracy (%)
82	95.45	94.41	94.93

It can be seen in Table 6, the classification accuracy using the selected features by the GA has reached to 94.93 % while our proposed method shows 97.76 %. So the classification accuracy has been increased by 02.83 % according to our approach.

In the following, we also compare our HV stage with the method presented in [60]. The dataset used in [60] is shown in Table 7.

Table 7. the vehicles dataset [60]

Dataset	#vehicle images	#non-vehicle images	Source
Training	1154	1154	CALLTECH2001
Validation	155	256	CALLTECH1999
Testiing	120	180	GRAZ + INRIA + their own images

In [60], an analysis of ways to integrate the features and classifiers has been presented for vehicle recognition in presence of illumination noise unseen in the training stage. They found that employing their ensemble method has better performance than the others. In this case, Local Receptive Field (LRF) features are classified by a Multi Layer Perceptron (MLP) classifier and once again by a SVM classifier and also HOG features are classified by a SVM classifier. Next, to integrate the classifiers, they used Heuristic Majority Voting (Heuristic MV) approach. To construct the effect of penumbra and white saturation (the two effects prone to happen in outdoor environments) two artificially light transformations over the dataset was applied (see Figure 10). Then, the classification accuracy was obtained over the testing datasets. Table 8 summarizes their results. Consequently, the highest classification accuracy belongs to Heuristic MV approach with an average accuracy 91.4 %.



Figure 10. 32×32 pixel image samples. In the left column, some original samples under normal condition; in the middle and right columns, the respective images under white saturation and penumbra

Table 8. the classification accuracy over the testing dataset [60]

Classifier	Normal (%)	Penumbra (%)	Saturation (%)	Average (%)
LRF/MLP	87.0	88.0	85.7	86.9
LRF/SVM	90.0	90.0	83.7	87.9
HOG/SVM	92.0	78.0	84.3	84.76
Heuristic MV	94.3	91.7	88.3	91.4

We applied our proposed HV method over the datasets, Table 9 shows the results.

Table 9. the classification accuracy of the proposed HV method over the same testing dataset

Classifier	Normal (%)	Penumbra (%)	Saturation (%)	Average (%)
Proposed HV method	96.0	92.7	90.3	93.0

Table 9 shows that by employing the proposed HV method, the classification accuracy has been enhanced by 1.7, 1.0 and 2.0 % compared to Heuristic MV approach over normal, Penumbra and white saturation datasets respectively. This shows the proposed HV method has better performance over noisy images in addition to normal images. The proposed HV method gained an average accuracy of 93.0 %, which means the average accuracy has been grown by 1.6 % compared to Heuristic MV approach.

4. CONCLUSION AND FUTURE WORKS

Generally, in this paper, a two-stage approach has been proposed to robustly detect preceding vehicles (front and rear vehicle view). The first stage is the hypothesis generation (HG) which contained a combination of the three clues of shadow, entropy and symmetry without prior knowledge about the position of the road. In the hypothesis verification (HV) stage, all the hypotheses were verified by a strong classifier. For this purpose, we have proposed a four-step method for classifying the vehicle candidate images into vehicle and non-vehicle classes. To begin, we extracted the PHOG features from an image dataset as well as the image obtained from employing the Gaussian filter on the image as the primitive features. Next, we applied the PCA to reduce the dimensionality of the PHOG descriptors and produce the reduced PHOG-PCA features. Finally, we used the GA to find the optimum weights for these features with respect to both the classification accuracy and the number of used features to improve their performance and generalization. Our tests showed the HG stage has the ability to detect the approximate location of the vehicles with appropriate accuracy and also the HV stage achieved 97.76 % classification accuracy on the realistic on-road vehicle images.

Combining multiple cues can be useful for developing more robust and reliable systems. We have also used the combination of the shadow, entropy and symmetry cues in the HG stage in this study. In past, combining multiple cues has produced promising results (e.g., combination of Local Orientation Coding (LOC), entropy, and shadow [27], color and shape [61], shape, symmetry, and shadow [62] and motion with appearance [63]). For future work, combining different cues can be explored to achieve effective fusion algorithm as well as cues which are fast to compute.

In HV stage, in previous works, concentration has been usually on feature extraction while many of the features are not relevant and it has great impact on the classification accuracy. Therefore, applying a feature selection or weighting strategy seems to be beneficial. For future work, the other features such as Gabor, Haar-Like, Wavelet can also be extracted and then feature selection or weighting is applied on concatenation of their normalized vectors. Moreover, the proposed HV stage can also be exerted on the concatenation. In this study, we used the PHOG features as a spatial descriptor and by beneficiary of Gabor features as a frequency descriptor besides the PHOG features, better results can be achieved since in this case, the image is described in both spatial and frequency domain. As a result, we can benefit from the advantage of both. In another case, after extracting features with different types (e.g. Gabor, PHOG), the features of each type can be classified by separated classifiers and then the outputs of the classifiers can be integrated (by investigating different methods) and finally the classification is done.

To complete the proposed system, detecting the passing vehicles can be added by acquisition of moving information. Moreover, detecting vehicles in night time can also be appended [64]. For this purpose, to begin, by determining the light condition of above region of the image, the day or night can be realized. Then, certain algorithm is employed to detect vehicles in each case. Finally, we also pointed out adding detection of other traffic objects such as pedestrians, motor cycles and traffic signs to the proposed system for its completeness.

REFERENCES

- [1] Z. Sun, G. Bebis, and R. Miller, "On-road Vehicle Detection: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 694-711, 2006.
- [2] W. Jones, "Building safer cars," *IEEE Spectrum*, vol. 39, no. 1, pp. 82-85, Jan. 2002.
- [3] F. Han, Y. Shan, R. Cekander, H. S. Sawhney, and R. Kumar, "A Two-Stage Approach to People and Vehicle Detection with HOG-Based SVM," in *PerMIS proceeding*, 2006, pp. 133-140.
- [4] Z. Sun, G. Bebis, and R. Miller, "On-Road Vehicle Detection Using Evolutionary Gabor Filter Optimization," *IEEE Trans. Intelligent Transportation Systems*, vol. 6, no. 2, pp. 125-137, 2005.
- [5] Z. Sun, R. Miller, G. Bebis, and D. DiMeo, "A Real-Time Precrash Vehicle Detection System," *Proc. IEEE Int'l Workshop Application of Computer Vision*, Dec. 2002.
- [6] E. Dickmanns et al., "The Seeing Passenger Car 'Vamors-P'," in *Proc. Int'l Symp. Intelligent Vehicles*, 1994, pp. 24-26.
- [7] C. Tzomakas and W. Seelen, "Vehicle Detection in Traffic Scenes Using Shadows," Technical Report 98-06, Institut fur Neuroinformatik, Ruht-Universitat, Bochum, Germany, 1998.
- [8] T. Zielke, M. Brauckmann, and W. V. Seelen, "Intensity and Edge-Based Symmetry Detection with an Application to Car-Following," *Computer Vision, Graphics, and Image Processing: Image Understanding*, vol. 58, no. 2, pp. 177-190, 1993.
- [9] A. Benshair, M. Bertozzi and A. Broggi, "A Cooperative Approach to Vision-based Vehicle Detection," in *IEEE Intelligent Transportation Systems Conference Proceedings*, 2001.
- [10] A. Broggi, P. Cerri and P. C. Antonello, "Multi-Resolution Vehicle Detection using Artificial Vision," in *Proceedings of IEEE Intelligent Vehicles Symposium*, 2004.
- [11] N. Matthews, P. An, D. Charnley and C. Harris, "Vehicle detection and recognition in greyscale imagery," *Control Eng. Pract.*, vol. 4, no. 4, pp. 473-479, 1996.
- [12] Z. Sun, G. Bebis, and R. Miller, "Monocular Precrash Vehicle Detection: Features and Classifiers," *IEEE Transactions on Image Processing*, vol. 15, no. 7, pp. 2019-2034, 2006.
- [13] T. Xiong, and C. Debrunner, "Stochastic car tracking with line and color-based features," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 324-328, 2004.
- [14] D. Guo, T. Fraichard, M. Xie and C. Laugier, "Color modeling by spherical influence field in sensing driving environment," in *IEEE Intelligent Vehicle Symp.*, Dearborn, MI, Oct. 2000, pp. 249-254.
- [15] T. Kalinke, C. Tzomakas, and W. von Seelen, "A texture-based object detection and an adaptive model-based classification," in *Proc. IEEE Int. Conf. Intelligent Vehicles*, Stuttgart, Germany, Oct. 1998, pp. 143-148.
- [16] T. Bucher, C. Curio, J. Edelbrunner, et al., "Image processing and behavior planning for intelligent vehicles," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 1, pp. 62-75, 2003.
- [17] R. Cucchiara and M. Piccardi, "Vehicle Detection under Day and Night Illumination," in *Proc. Int'l ICSC Symp. Intelligent Industrial Automation*, 1999.
- [18] J. Firl, M. H. Hoerter, M. Lauer, and C. Stiller, "Vehicle detection, classification and position estimation based on monocular video data during night-time," In *Proceedings of 8th International Symposium on Automotive Lighting*, Darmstadt, Sept. 2009.
- [19] H. Mallot, H. Bulthoff, J. Little and S. Bohrer, "Inverse perspective mapping simplifies optical flow computation and obstacle detection," *Biol. Cybern.*, vol. 64, no. 3, pp. 177-185, 1991.
- [20] M. Bertozzi and A. Broggi, "Gold: A parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Trans. Image Process.*, vol. 7, pp. 62-81, Jan. 1998.
- [21] A. Broggi, M. Bertozzi, A. Fascioli, C. Guarino Lo Bianco and A. Piazzi, "Visual perception of obstacles and vehicles for platooning," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, pp. 164-176, Sep. 2000.
- [22] A. Giachetti, M. Campani and V. Torre, "The use of optical flow for road navigation," *IEEE Trans. Robot. Autom.*, vol. 14, pp. 34-48, Feb. 1998.
- [23] W. Kruger, W. Enkelmann and S. Rossle, "Real-time estimation and tracking of optical flow vectors for obstacle detection," in *Proc. IEEE Intelligent Vehicle Symp.*, Detroit, MI, Sep. 1995, pp. 304-309.
- [24] B. Heisele and W. Ritter, "Obstacle Detection Based on Color Blob Flow," *Proc. IEEE Intelligent Vehicle Symp.*, 1995, pp. 282-286.
- [25] D. Koller, N. Heinze, and H. Nagel, "Algorithmic Characterization of Vehicle Trajectories from Image Sequence by Motion Verbs," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 1991, pp. 90-95.
- [26] P. Parodi and G. Piccioli, "A Feature-Based Recognition Scheme for Traffic Scenes," *Proc. IEEE Intelligent Vehicles Symp.*, 1995, pp. 229-234.

- [27] U. Handmann, T. Kalinke, C. Tzomakas, M. Werner, and W. Seelen, "An Image Processing System for Driver Assistance," *Image and Vision Computing*, vol. 18, no. 5, 2000.
- [28] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. CVPR*, 2001, pp. 511-518.
- [29] M. Weber, M. Welling, and P. Perona, "Unsupervised learning of models for recognition," in *Proc. ECCV*, 2000, pp. 18-32.
- [30] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE PAMI*, 26(11):1475-1490, Nov. 2004.
- [31] Q.B. Truong, and B.R. Lee, "Vehicle Detection Algorithm Using Hypothesis Generation and Verification," in *Proc. ICIC (I)*, 2009, pp.534-543.
- [32] C. Goerick, N. Detlev, and M. Werner, "Artificial Neural Networks in Real-Time Car Detection and Tracking Applications," *Pattern Recognition Letters*, vol. 17, pp. 335-343, 1996.
- [33] O. L. Junior and U. Nunes, "Improving the generalization properties of neural networks: An application to vehicle detection," in *Proc. IEEE Conf. Intell. Transp. Syst.*, Oct. 2008, pp. 310-315.
- [34] Thiang, R. Lim, and A. T. Guntoro, "Car Recognition Using Gabor Filter Feature Extraction," *Circuits and Systems, APCCAS'02. (2)*, pp.451-455, 2002.
- [35] G. Y. Song, K. Y. Lee, and J. W. Lee, "Vehicle detection by edge-based candidate generation and appearance-based classification," *Intelligent Vehicles Symposium*, pp. 428-433, June 2008.
- [36] P. Viola and M. Jones, "Robust real-time object detection," in *International Journal of Computer Vision*, 2001.
- [37] C. Papageorgiou and T. Poggio, "A trainable system for object detection," *Int. J. Comput. Vis.*, vol. 38, no. 1, pp. 15-33, 2000.
- [38] P. Negri, X. Clady, S. M. Hanif, and L. Prevost, "A cascade of boosted generative and discriminative classifiers for vehicle detection," *EURASIP J. Adv. Signal Process.*, vol. 2008, pp. 1-12, 2008.
- [39] C. Wang and J.-J. J. Lien, "Automatic vehicle detection using local features-A statistical approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 83-96, Mar. 2008.
- [40] Z. Sun, G. Bebis, and R. Miller, "Object Detection Using Feature Subset Selection," *Pattern Recognition*, vol. 37, pp. 2165-2176, 2004.
- [41] M.B. van Leeuwen and F.C.A. Groen, "Vehicle detection with a mobile camera: Spotting midrange, distant, and passing cars," *IEEE Robotics and Automation Magazine*, vol. 12, no. 1, pp. 37-43, 2005.
- [42] M.B. van Leeuwen, "Motion estimation and interpretation for in-car systems" Ph.D. dissertation, University of Amsterdam, 2002.
- [43] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, 27:379-423,623-656, 1948.
- [44] Z. Li, J.-i. Imai, and M. Kaneko, "Facial-component-based bag of words and phog descriptor for facial expression recognition," in *Proceedings of the 2009 IEEE international conference on Systems, Man and Cybernetics*, ser. SMC'09, 2009.
- [45] L. Shao and L. Ji, "A Descriptor Combining MHI and PCOG for Human Motion Classification," in *Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR)*, Xi'an, China, July 2010.
- [46] X. H. Han and Y. W. Chen, "Image Categorization by Learned PCA Subspace of Combined Visual-words and Low-level Features," in *Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2009.
- [47] S. Ozşen, and S. Guneş, "Attribute weighting via genetic algorithms for attribute weighted artificial immune system (AWAIS) and its application to heart disease and liver disorders problems," *Expert Systems with Applications*, vol. 36, pp 386-392, Jan. 2009.
- [48] F. Hussein, N. Kharma, R. Ward, "Genetic Algorithm for Feature Selection and Weighting, a Review and Study," *6th Int. Conf. on Document Analysis and Recognition*, Sept. 2001, pp. 1240-1244.
- [49] B. T. Ongkowijaya, and X. Zhu, "A New Weighted Feature Approach Based on GA for Speech Recognition," in *7th International Conference on Signal Processing (ICSP)*, 2004 , pp. 663-666.
- [50] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proceedings of the International Conference on Image and Video Retrieval*, 2007.
- [51] B. Zhang, Y. Song, and S. U. Guan, "Historic Chinese Architectures Image Retrieval by SVM and Pyramid Histogram of Oriented Gradients Features," in *International Journal of Soft Computing*, vol. 5, issue 2, pp. 19-28, 2010.
- [52] Y. Bai, L. Guo, L. Jin, and Q. Huang, "A Novel Feature Extraction Method Using Pyramid Histogram of Orientation Gradients for Smile Recognition," in *16th IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 3305 - 3308.

- [53] T. Kobayashi, A. Hidaka, and T. Kurita, "Selection of Histograms of Oriented Gradients Features for Pedestrian Detection," in *Proc. ICONIP (2)*, 2007, pp.598-607.
- [54] N.G.Chitaliya and A.I.Trivedi, "An Efficient Method for Face Feature Extraction and Recognition based on Contourlet Transform and Principal Component Analysis using Neural Network," *International Journal of Computer Applications*, vol. 6, No. 4, September 2010.
- [55] C. Liu and H. Wechsler, "Evolutionary Pursuit and Its Application to Face Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, pp. 570-582, June 2000.
- [56] G. Bebis, S. Uthiram and M. Georgiopoulos, "Face detection and verification using genetic search," *Int. J. Artif. Intell. Tools*, vol. 9, pp. 225–246, 2000.
- [57] T. Zhao, J. Lu, Y. Zhang, and Q. Xiao, "Image Annotation Based on Feature Weight Selection," in *International Conference on Cyberworlds*, 2008, pp. 251–255.
- [58] Z. Sun, G. Bebis, X. Yuan and S. Louis, "Genetic feature subset selection for gender classification: A comparison study," in *IEEE Int. Workshop Application Computer Vision*, Orlando, FL, Dec. 2002, pp. 165–170.
- [59] A. J. Katz and P. R Thrift, "Generating Image Filters for Target Recognition by Genetic Learning," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.16, pp. 906-910, 1994.
- [60] L. Oliveira and U. Nunes, "On Integration of Features and Classifiers for Robust Vehicle Detection," in *IEEE Conference on Intelligent Transportation Systems*, 2008, pp. 414-419.
- [61] K. She, G. Bebis, H. Gu, and R. Miller, "Vehicle Tracking Using On-Line Fusion of Color and Shape Features," *Proc. IEEE Int'l Conf. Intelligent Transportation Systems*, 2004.
- [62] J. Collado, C. Hilario, A. de la Escalera, and J. Armingol, "Model-Based Vehicle Detection for Intelligent Vehicles," *Proc. IEEE Intelligent Vehicles Symp.*, 2004.
- [63] J. Wang, G. Bebis, and R. Miller, "Overtaking Vehicle Detection Using Dynamic and Quasi-Static Background Modeling," *Proc. IEEE Workshop Machine Vision for Intelligent Vehicles*, 2005.
- [64] S. Y. Kim, S. Y. Oh, J. K. Kang, Y. W. Ryu, K. Kim, S. C. Park and K. H. Park, "Front and Rear Vehicle Detection and Tracking in the Day and Night Times Using Vision and Sonar Sensor Fusion," *IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, Edmonton, Canada, 2005, pp.3616-3621.

Authors

Nima Khairdoost received the BS and MS degrees in Computer engineering from Ferdowsi University of Mashhad and University of Isfahan, Iran in 2008 and 2011, respectively. His research interests include Image processing, machine vision and pattern recognition as well as evolutionary algorithms.



S. Amirhassan Monadjemi, born 1968, in Isfahan, Iran. He received his BS degree at computer hardware engineering from Isfahan University of Tech., in 1991, his MS degree at computer engineering, machine intelligence and robotics from Shiraz University, Shiraz, in 1994 and his PhD degree in computer science, image processing and pattern recognition from University of Bristol, Bristol, UK. in 2004. He is now working as assistant professor at the Department of Computer Engineering, Faculty of Engineering, University of Isfahan, Isfahan, Iran. His research interests include image processing, computer vision and pattern recognition, computer aided learning and physical detection and elimination of viruses.



Kamal Jamshidi received the MS and PhD degrees in electrical engineering from Anna University of India in 1990 and I.I.T University of India in 2003, respectively. He currently is an assistant professor in the Engineering Department of University of Isfahan. His research interests include wireless sensor network and vehicular ad hoc networks as well as fuzzy systems and microprocessor based systems.

