

# A ROBUST GA/KNN BASED HYPOTHESIS VERIFICATION SYSTEM FOR VEHICLE DETECTION

Nima Khairdoost<sup>1</sup>, Mohammad Reza Baghaei Pour<sup>2</sup>, Seyed Ali Mohammadi<sup>3</sup> and Mohammad Hoseinpour Jajarm<sup>4</sup>

<sup>1</sup>Department of Computer Engineering, Tabaran Institute of Higher Education, Mashhad, Iran

<sup>2</sup>Department of Computer Engineering, Ferdows Higher Education Institute, Mashhad, Iran

<sup>3</sup>Department of Computer Engineering, Tabaran Institute of Higher Education, Mashhad, Iran

<sup>4</sup>Department of Computer Engineering, Islamic Azad University, Mashhad Branch, Mashhad, Iran

## **ABSTRACT**

*Vehicle detection is an important issue in driver assistance systems and self-guided vehicles that includes two stages of hypothesis generation and verification. In the first stage, potential vehicles are hypothesized and in the second stage, all hypothesis are verified. The focus of this work is on the second stage. We extract Pyramid Histograms of Oriented Gradients (PHOG) features from a traffic image as candidates of feature vectors to detect vehicles. Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are applied to these PHOG feature vectors as dimension reduction and feature selection tools parallelly. After feature fusion, we use Genetic Algorithm (GA) and cosine similarity-based K Nearest Neighbor (KNN) classification to improve the performance and generalization of the features. Our tests show good classification accuracy of more than 97% correct classification on realistic on-road vehicle images.*

## **KEYWORDS**

*PHOG, Gaussian Low-pass Filter, PCA, LDA, GA, KNN*

## **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]. 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.

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 [2].

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 [3,4].

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 [5], 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. [6] 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 [3]. 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) [7,8,9]. In [10], Principle Component Analysis (PCA) was used for feature extraction and linear Support Vector Machine (SVM) for classification of vehicle images. Goerick et al. [11] employed Local Orientation Code (LOC) to extract the edge information of ROI and NNs to learn the characteristics of vehicles. In [12], 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 [13], Gabor filters provide a mechanism to extract the line and edge information by tuning orientation and changing the scale. In [14], an Adaboost classifier [15] trained on Haar features was used to classify detections. Papageorgiou and Poggio [16] have presented by acquisition of Haar wavelet transform for feature extraction and SVMs for classification. In [17], 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 [18], a similar work was performed. Negri et al. [18] 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 [19], 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 [20], a general object detection scheme was proposed using PCA and Genetic Algorithm (GA) for feature extraction and feature subset selection respectively.

The rest of the paper is organized as follows: in section 2, the proposed method is described in detail. Section 3 shows the experimental results and the last section is the conclusions.

## **2. PROPOSED METHOD**

The framework in the hypothesis verification is feature extraction from the hypotheses and classification them into vehicle and non-vehicle classes. Therefore the performance of this stage is directly depended on employing a classifier that is well trained by the appropriate features. For achieving this purpose, we propose a framework that is shown in Figure 1. 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 [20], facial expression recognition [21], human motion classification [22] and image categorization [23]. 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 reduce the dimensionality and improve the classification accuracy more, we also apply PCA and Linear Discriminant Analysis (LDA) as dimension reduction and feature selection tools parallelly to the PHOG features obtained from the original hypothesis images and the images resulted from applying Gaussian low-pass filter on the original images. Then, after fusion of the extracted features from PCA and LDA, we divide the samples into two parts of *Training Data* and *Test Data* as shown in Figure 1.

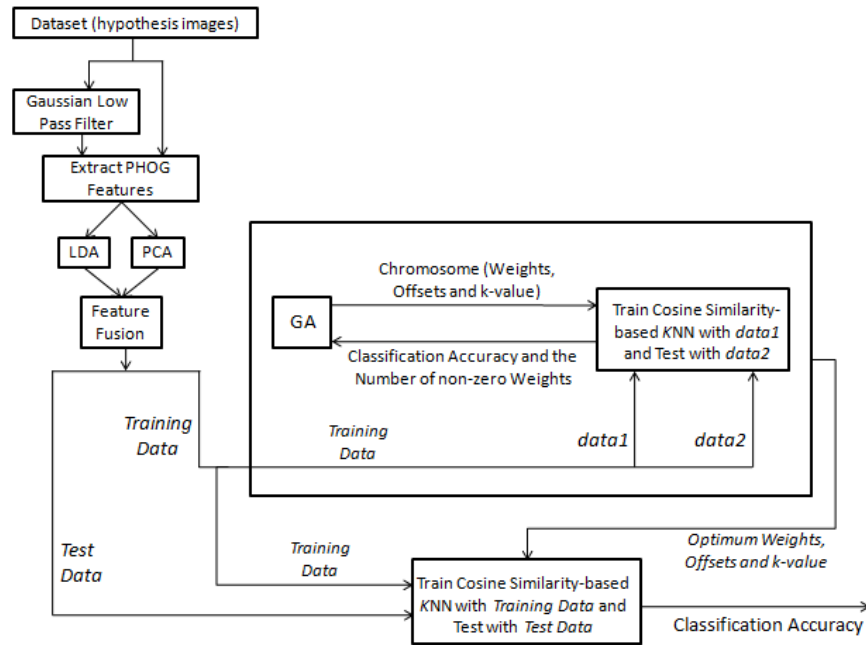


Figure 1. the proposed method

It is well known that feature weighting is effective for pattern classification as shown in [24,25,26]. It is expected that the classification accuracy can be further improved by weighting the obtained features since some local regions are less relevant for vehicle detection than the others. For this purpose, we employ a GA combined with a cosine similarity-based K Nearest Neighbor (KNN) for vehicle/ non-vehicle classification. In another words, we use the GA to find optimum weights and offsets for the features as well as optimum k-value. The *Training Data* is divided into two parts of *data1* and *data2*. The knn classifier 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, the cosine similarity-based knn is trained regarding the *Optimum Weights, Offsets, K-value* 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 proposed method is shown in Figure 1.

It should be mentioned that we performed several experiments to explore the performance of the obtained features and also Euclidian distance-based knn classifier during the steps of the proposed method. Section 3 shows the experimental results related to the experiments and demonstrates

that the proposed method has reached the better performance. In the following of this section, we will describe the steps of the proposed method in detail.

## 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. [27,28, 29]. 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 2, the PHOG descriptor consists of a histogram of orientation gradients over each image sub-region at each resolution.

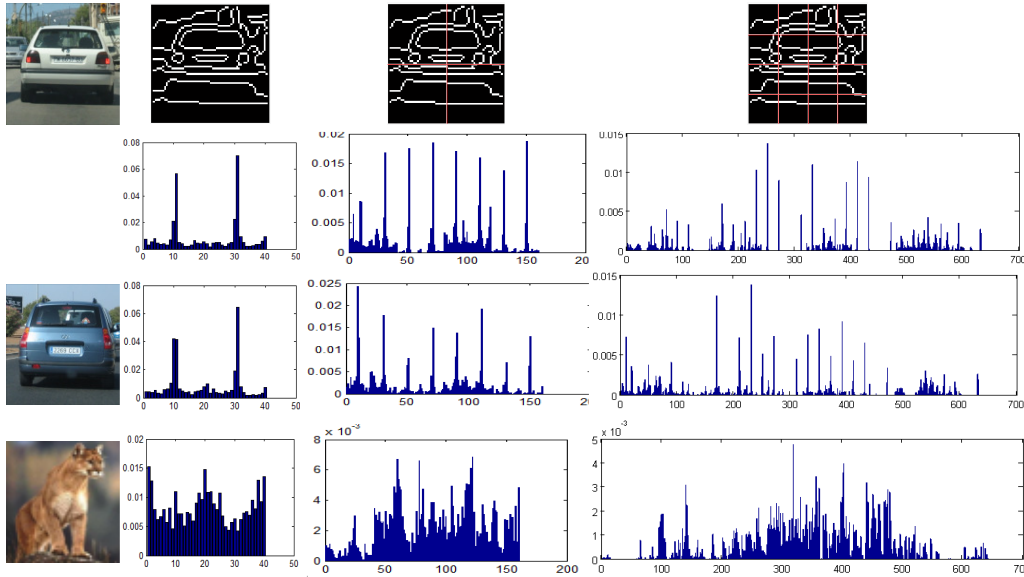


Figure 2. Shape spatial pyramid representation. Top row: a vehicle image and grids for levels  $l = 0$  to  $l = 2$ ; Below: histogram representations corresponding to each level. The final PHOG vector is a weighted concatenation of vectors (histograms) for all levels. Remaining rows: another vehicle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

For extracting the PHOG features, the edge contours are extracted by acquisition of the Canny edge detector for entire image as shown in Figure 2. 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  $3 \times 3$  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 [27]. 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 2 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.

## 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:

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

where  $x$  is the distance from the origin in the horizontal axis,  $y$  is the distance from the origin in the vertical axis, and  $\sigma$  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 3 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.

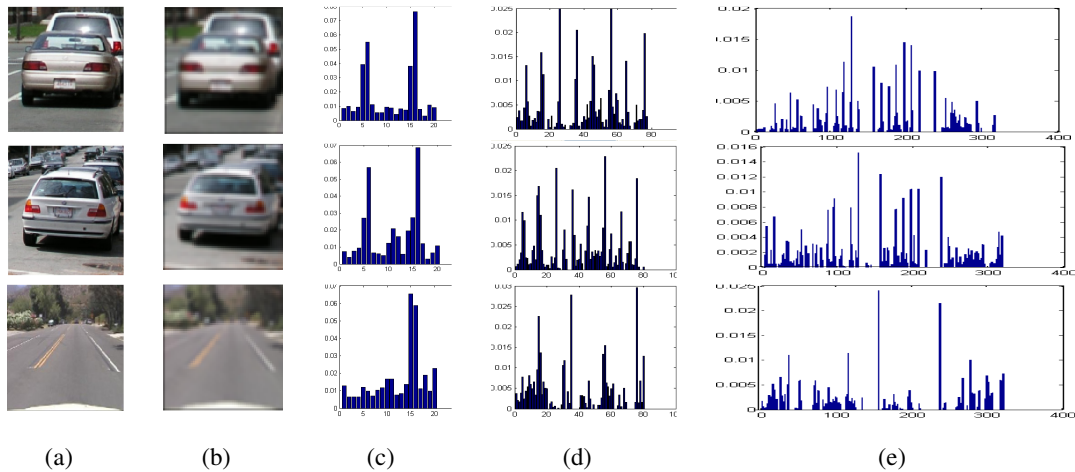


Figure 3. the PHOG features representation for sample vehicle and non-vehicle images after applying Gaussian low-pass filter. Column (a): the original vehicle and non-vehicle images; Column (b): the results of applying Gaussian low-pass filter on the images of column (a); Columns (c),(d) and (e): the PHOG features representation of the corresponding filtered images of column (b).

## 2.3. 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 [30, 31] 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  $\Sigma$  that is defined as follows:

$$\Sigma = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^T \quad (2)$$

By solving the eigen equations of the covariance matrix  $\Sigma$ , the optimum projection matrix  $U$  is obtained

$$\Sigma U = \Lambda U, (UU^T = I) \quad (3)$$

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.

$$y = U^T(x_i - \bar{x}) \quad (4)$$

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

## 2.4. Linear Discriminant Analysis (LDA)

Undoubtedly, one of the most popular and experienced research trends in recent years is using different versions of the LDA in the problems such as face recognition [32], vehicle type recognition [33] and pedestrian detection[34]. In this method, we seek to find  $W_{LDA}$  vectors that can be able to separate different classes. The purpose is to produce a space as inside class scattering to be minimum and outside class scattering to be maximum. In this method, the first two-dimensional images with the dimensions of  $a \times b$  are converted into vectors with  $a \times b$  elemental. Two scatter matrixes inside the class,  $S_w$ , and outside the class,  $S_b$ , are used.  $S_w$  and  $S_b$  are defined as follows:

$$S_w = \frac{1}{n} \sum_{j=1}^c \sum_{x \in c_j} (X - \bar{X}_j)(X - \bar{X}_j)^T \quad (5)$$

$$S_b = \frac{1}{n} \sum_{j=1}^c (\bar{X}_j - \bar{X})(\bar{X}_j - \bar{X})^T \quad (6)$$

where  $c$  is the number of classes,  $\bar{X}_j$  is average of class  $j$ ,  $n_j$  is the number of samples of class  $j$ ,  $n$  is the total number of samples, and  $\bar{X}$  is the average of all samples.

Intuitively, if the data of each class are concentrated and the class masses are far from each other as far as possible, designing a learning machine for classifying the data will be easy and the learning machine will have a higher potential for classification of the data as well.

Inside class scattering is an index for the concentration of the data in each class and the outside class scattering is an index for the amount of distances between the class masses.

One way to obtain the conversion matrix  $W_{LDA}$ , is using the fisher criterion that is obtained by maximization of (7):

$$J(W_{LDA}) = \frac{W_{LDA}^T S_b W_{LDA}}{W_{LDA}^T S_w W_{LDA}} \quad (7)$$

$W_{LDA}$  will be obtained by a set of eigenvectors corresponding to the largest eigen values of  $S_w^{-1} \cdot S_b$ . The maximum number of separatable spaces is  $c-1$ , in other words, one unit less than the number of classes. Thus, after dimension reduction the image dimension will be  $c-1$ :

$$Reduced_{matrix} = Image_{matrix} * W_{LDA} \quad (8)$$

$W_{LDA}$  is matrix with  $C-1$  rows of the eigenvectors corresponding to the largest eigen values of  $S_w^{-1} \cdot S_b$ . Therefore, the final image vectors size changes from  $a \times b$  to  $c-1$ .

The main idea of PCA is retaining directions of large variance, it does not take into account the actual classes. Since large variance is not always best for classification, while LDA searches the directions for maximum discrimination of classes in addition to dimensionality reduction and selects the features that are most effective for class separability.

## 2.5. Feature Fusion

As before, feature reduction is done using PCA and LDA algorithms. In feature fusion plan, feature vectors extracted from the mentioned algorithms are concatenated to construct a new feature vector to be used for the post-processing as shown in Figure 1.

Suppose  $F_1$  and  $F_2$  are the feature vectors extracted using PCA and LDA algorithms, respectively. The feature vectors will be defined as follows:

$$V_{PCA} = \left[ \frac{F_1}{\|F_1\|} \right] \quad (9)$$

$$V_{LDA} = \left[ \frac{F_2}{\|F_2\|} \right] \quad (10)$$

$$V_{Fusion} = \frac{[V_{PCA} V_{LDA}]}{\|[V_{PCA} V_{LDA}]\|} \quad (11)$$

where  $\|\cdot\|$  is the second norm. Because the ranges of the values in the feature vectors extracted from the two different algorithms are not same, the feature vectors  $F_1$  and  $F_2$  are normalized as shown in (9)-(11), respectively, to make sure that the influences of the two different algorithms to the feature vectors are similarly weighted.  $F_{Fusion}$  is the final feature vector generated by concatenating the two feature vectors obtained using the algorithms.

## 2.6. K-NN Classification

### 2.6.1. Euclidian Distance-based K-NN Classification

In  $knn$  classification,  $d$ -dimensional space is considered for training patterns, where  $d$  is the number of features. These patterns are plotted with respecting to their observed feature values and labeled according to their known class. An unlabelled test pattern is plotted in the same space and classified with respect to the most frequently existing class among its  $k$ -most similar training patterns. The most common similarity measure for  $knn$  classification is the Euclidian distance metric, defined between feature vectors  $\vec{x}$  and  $\vec{y}$  as follows:

$$euc(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^f (x_i - y_i)^2} \quad (12)$$

where  $f$  is the number of features. Smaller distance values indicate greater similarity.

Classification is done after verifying the  $k$ -most similar training points to the query. Instead of employing a standard voting scheme, following the scheme used by [35], the algorithm classifies the query point using a weighted plan based on each neighbor's proximity to the query. If the data contains only two positive and negative classes, then the query point  $\vec{x}$  is labelled according to the  $q$  measure:

$$q = \sum_{i=1}^n euc(\vec{x}_i, \vec{x})c(\vec{x}_i) \quad (13)$$

where

$$c(\vec{x}_i) = \begin{cases} 1 & : \text{if } \vec{x}_i \in \text{the positive class} \\ -1 & : \text{otherwise} \end{cases} \quad (14)$$

and  $n$  is the number of training points and  $\vec{x}_i$  represents the  $i^{\text{th}}$  training point. Because smaller distances indicate greater similarity, the query point is assigned to the positive class if  $q$  is negative, otherwise it is assigned to the negative class. For weight optimization, given a weight vector  $\vec{w}$ , each feature vector  $\vec{x}$  is transformed by  $\vec{w}$  such that the transformed feature vector  $\vec{x}' = x_i * w_i$ . Euclidian distance is invariant to offset shifting, so offset optimization is not performed for the Euclidian distance knn classifier.

### 2.6.2. Cosine Similarity-based K-NN Classification

For the knn classifier considering the cosine similarity measure, the cosine similarity between feature vectors  $\vec{x}$  and  $\vec{y}$  is defined as follows:

$$\cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \|\vec{y}\|} \quad (15)$$

where "." represents the dot product between the vectors, and  $\|\vec{z}\|$  is the length of the vector. Unlike the Euclidian distance, larger cosine values indicate a greater similarity between vectors. Classification is done after identifying the  $k$ -most similar neighbors by using a  $q$ -score, replacing cosine similarity for Euclidian distance. Since larger values indicate greater similarity, the query is assigned to the positive class if the  $q$ -score is positive, otherwise the negative class is chosen. While, feature weighting affects cosine-based knn classification, shifting the feature offset provides an additional opportunity for optimization. For cosine similarity the angle between two vectors is taken relative to the origin; a GA may perform feature extraction by shifting the location of the origin relative to the data. Independently shifting each feature positively or negatively changes the angular point of reference, affecting the classification. For combination of weight and offset optimization, given a weight vector  $\vec{w}$  and an offset vector  $\vec{o}$ , each feature vector  $\vec{x}$  is transformed such that the transformed feature vector  $\vec{x}' = (x_i - o_i) * w_i$ .

## 2.7. The Genetic Algorithm Configuration

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 [20], face recognition [36,37], vehicle detection [3], image annotation [38], gender classification [39] and target recognition [40]. Moreover, in some studies GA has been combined with knn to improve the classification performance [41,42].

In this study, we utilized the GA for finding optimum weight and offset for each feature obtained from the feature fusion step and optimum k-value simultaneously to reduce the classification error of the classifier. Thus, the best chromosome is the one leading to the lowest test classification error. The configuration of the GA is as follows:



### 2.7.1. Chromosome structure

Let the number of the features be  $L$ , so each chromosome will be represented with  $2*L+1$  genes. The first  $L$  genes are weights that take values from the range of 0.0 to 20.0. The second  $L$  genes represent offsets which range from -5.0 to +5.0 and the last gene is k-value for classification which takes integer values from 1 to 25.

### 2.7.2. Calculating the fitness of these chromosomes

We forced weights  $<0.4$  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  $knn$  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:

$$\text{Fitness}(c) = CA^4(c) - \alpha \left( \frac{N(c)}{L} \right) \quad (16)$$

Where  $c$  is the chromosome,  $CA(c)$  is the classification accuracy using the cosine similarity-based  $knn$  classifier.  $\alpha$  represents the tradeoff between the two criteria (using  $\alpha = 0.01$ ).  $N(c)$  is the number of non-zero weights. As before,  $L$  is the total number of the features. 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.

### 2.7.3. Initial population

All the offsets and weights of the first chromosome are 0 and '20' respectively which means the weights of all the features are equal. The other chromosomes are generated randomly. In all of our experiments, we used 400 generations and a population size of 600. In most cases, the GA converged in less than 300 generations.

### 2.7.4. 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.

### 2.7.5. Mutation

We choose uniform mutation which means each bit has the same mutation probability. The mutation rate used in all of our experiments was 0.1.

### 2.7.6. 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 50 chromosomes are preserved for the next generation automatically.

### 3. EXPERIMENTAL RESULTS

#### 3.1. Dataset

We used the vehicle dataset as hypotheses, which contains 1400 non-vehicle images and 1400 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 4).

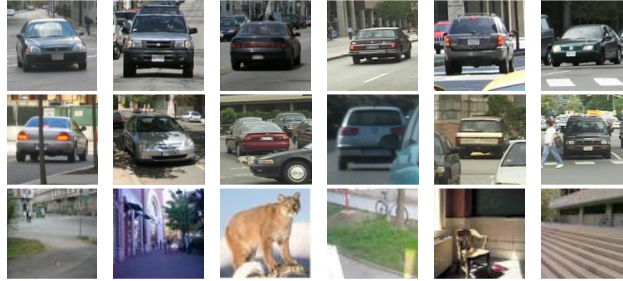


Figure 4. Some vehicle and non-vehicle training sample images

#### 3.2. Experiments

In our experiments, we used the Euclidian distance-based and cosine similarity-based *knn* classifiers 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.

Also, we used a 7-fold cross-validation to estimate both the accuracy and generality of the *knn* classifiers. 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 proposed method, we used the same folds in all following experiments for the cross validation. In the following of this section, "TP", "TN" and "CA" represent true positive, true negative and classification accuracy respectively. In the first experiment, we applied the *knn* classifiers to the PHOG descriptors. The best results for each classifier are shown in bold, Table 1 shows the results.

Table 1.the Classification Results with the PHOG Features Extracted from the Dataset Images

Euclidian Distance-Based <i>K</i> -NN						Cosine Similarity-Based <i>K</i> -NN					
<i>K</i> =5			<i>K</i> =15			<i>K</i> =5			<i>K</i> =15		
TP(%)	TN(%)	CA(%)	TP(%)	TN(%)	CA(%)	TP(%)	TN(%)	CA(%)	TP(%)	TN(%)	CA(%)
<b>90.13</b>	<b>86.07</b>	<b>88.10</b>	89.64	85.36	87.50	85.28	81.14	83.21	<b>86.22</b>	<b>81.92</b>	<b>84.07</b>

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 classification is done using the *knn* classifiers and the extracted PHOG features from the dataset images as well as the filtered images. it should be mentioned that to improve the

performance of the Gaussian low-pass filter, we performed several tests over employing different parameters of the filter and the PHOG feature extractor. As a result in this step, the best classification accuracy was obtained using 20 bins, 2 levels of pyramids, 5 for size of the filter and 5 for the standard deviation. Table 2 presents the results of the *knn* classifiers.

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 (Table 1).

Table 2. the Classification Results with the PHOG Features Extracted from the Dataset Images and also the Filtered Images

Euclidian Distance-Based <i>K</i> -NN						Cosine Similarity-Based <i>K</i> -NN					
<i>K</i> =5			<i>K</i> =15			<i>K</i> =5			<i>K</i> =15		
TP(%)	TN(%)	CA(%)	TP(%)	TN(%)	CA(%)	TP(%)	TN(%)	CA(%)	TP(%)	TN(%)	CA(%)
<b>90.71</b>	<b>87.57</b>	<b>89.14</b>	90.21	86.49	88.35	86.92	82.92	84.92	<b>87.85</b>	<b>84.35</b>	<b>86.10</b>

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 four different methods to the PHOG features extracted from the original dataset images and also the filtered images for improving the classification accuracy more (see Table 3). We employed PCA and LDA as dimension reduction tools separately. For more research on the performance of PCA and LDA, in the third method, a sequential combination of them was applied. In the last method, we utilized PCA and LDA as dimension reduction tools parallelly and next, the feature fusion step occurred. According to Table 3, our results show that the fusion method outperforms PCA, LDA and also the combination of them in the two separate stages.

According to Table 1, Table 2 and Table 3, as a general result, the Euclidian distance-based *knn* classifier outperforms the cosine similarity-based *knn*.

For further exploration, in fourth experiment using the obtained fusion features, the GA with the mentioned configuration was applied on the distance-based *knn* classifier to find its optimum weights and *k*-value and once again on the cosine similarity-based *knn* classifier to discover its optimum weights, offsets and *k*-value. Table 4 compares the averages of the optimum *k* and the final results for each optimized classifier. As can be seen, the optimized cosine *knn* outperforms the optimized Euclidian *knn* significantly.

Table 3. the Classification Results with Employing Four Different Methods to Climb the Performance of the *knn* classifiers Using the PHOG Descriptors Extracted from the Dataset Images as well as the Filtered Images

Classifier	Euclidian Distance-Based <i>K</i> -NN						Cosine Similarity-Based <i>K</i> -NN					
	<i>K</i> =5			<i>K</i> =15			<i>K</i> =5			<i>K</i> =15		
Method	TP(%)	TN(%)	CA(%)	TP(%)	TN(%)	CA(%)	TP(%)	TN(%)	CA(%)	TP(%)	TN(%)	CA(%)
PCA	91.6 4	88.2 0	89.9 2	<b>93.1</b> <b>4</b>	<b>89.7</b> <b>0</b>	<b>91.4</b> <b>2</b>	88.2 8	84.1 4	86.2 1	<b>89.1</b> <b>4</b>	<b>85.4</b> <b>2</b>	<b>87.2</b> <b>8</b>
LDA	<b>93.5</b> <b>0</b>	<b>90.7</b> <b>0</b>	<b>92.1</b> <b>0</b>	93.5 0	90.6 4	92.0 7	<b>89.8</b> <b>5</b>	<b>86.7</b> <b>9</b>	<b>88.3</b> <b>2</b>	89.2 1	86.4 3	87.8 2
PCA-LDA	<b>93.9</b> <b>2</b>	<b>91.4</b> <b>2</b>	<b>92.6</b> <b>7</b>	94.3 5	89.7 1	92.0 3	<b>91.6</b>	<b>89.3</b> <b>2</b>	<b>90.4</b> <b>6</b>	90.7 8	88.4 2	89.6 0
Fusion of PCA and LDA	<b>95.0</b> <b>0</b>	<b>93.2</b> <b>0</b>	<b>94.1</b> <b>0</b>	94.9 2	92.8 6	93.8 9	91.6 4	89.6 4	90.6 4	<b>92.1</b> <b>4</b>	<b>90.4</b> <b>2</b>	<b>91.2</b> <b>8</b>

Table 4. the Classification Results with Employing the GA to Optimize the Classifiers

Optimized Euclidian Distance-Based <i>K</i> -NN Average of Optimum <i>K</i> =8			Optimized Cosine Similarity-Based <i>K</i> -NN Average of Optimum <i>K</i> =13		
TP(%)	TN(%)	CA(%)	TP(%)	TN(%)	CA(%)
96.42	94.78	95.60	<b>98.54</b>	<b>97.1</b>	<b>97.82</b>

#### 4. CONCLUSION AND FUTURE WORK

Generally, in this paper, an approach has been proposed to robustly verify vehicle hypotheses to classify into vehicle and non-vehicle classes. To begin, we extracted the PHOG features from an image dataset as well as the image obtained from applying the Gaussian filter on the image as the primitive features. Next, we utilized the fusion of the features extracted from PCA and LDA as dimension reduction tools. Our results have shown that these features improve the performance of Euclidian distance-based and cosine similarity-based *knn* classifiers remarkably. Finally, we benefited from the GA to find the optimum weights, offsets and *k*-value for cosine similarity-based *knn* to improve the performance and generalization of the fusion features. The experimental results indicate that the optimized cosine similarity-based *knn* outperforms the optimized Euclidian distance-based *knn* significantly while unoptimized Euclidian *knn* performed better than unoptimized cosine *knn*. The proposed system achieved 97.82% classification accuracy on the realistic on-road vehicle images.

In hypothesis verification 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 or Wavelet can also be extracted and then feature selection or weighting is applied on concatenation of their normalized vectors. 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.

## 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] 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.
- [3] 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.
- [4] 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.
- [5] P. Parodi and G. Piccioli, "A Feature-Based Recognition Scheme for Traffic Scenes," *Proc. IEEE Intelligent Vehicles Symp.*, 1995, pp. 229-234.
- [6] 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.
- [7] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. CVPR*, 2001, pp. 511-518.
- [8] M. Weber, M. Welling, and P. Perona, "Unsupervised learning of models for recognition," in *Proc. ECCV*, 2000, pp. 18-32.
- [9] 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.
- [10] Q.B. Truong, and B.R. Lee, "Vehicle Detection Algorithm Using Hypothesis Generation and Verification," in *Proc. ICIC* (1), 2009, pp.534-543.
- [11] 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.
- [12] 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.
- [13] Thiang, R. Lim, and A. T. Guntoro, "Car Recognition Using Gabor Filter Feature Extraction," *Circuits and Systems, APCCAS'02.* (2), pp.451-455, 2002.
- [14] 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.
- [15] P. Viola and M. Jones, "Robust real-time object detection," in *International Journal of Computer Vision*, 2001.
- [16] C. Papageorgiou and T. Poggio, "A trainable system for object detection," *Int. J. Comput. Vis.*, vol. 38, no. 1, pp. 15-33, 2000.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] Z. Sun, G. Bebis, and R. Miller, "Object Detection Using Feature Subset Selection," *Pattern Recognition*, vol. 37, pp. 2165-2176, 2004.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] 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.

- [25] 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.
- [26] 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.
- [27] 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.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] E. Kamran and C. Rama, "Discriminant Analysis for Recognition of Human Face Images (Invited Paper)," in Proceedings of the First International Conference on Audio- and Video-Based Biometric Person Authentication: Springer-Verlag, 1997.
- [33]I. Zafar, E. A. Edirisinghe, S. Acar and H. E.Bez,"Two Dimensional Statistical Linear Discriminant Analysis for Real-Time Robust Vehicle Type Recognition,"Proc. SPIE 6496 Real-Time Image Processing, 2007.
- [34] S. Paisitkriangkrai, C. Shen and J. Zhang,"Fast Pedestrian Detection Using a Cascade of Boosted Covariance Feature,"IEEE Trans. Circuits and Systems for Video Technology,vol. 18, no. 8, pp. 1140-1151, 2008.
- [35] M. Kuramochi and G. Karypis, "Gene classification using expression profiles: a feasibility study," In Proceedings 2nd IEEE Int. Symp. onBioinf: &Bioeng., 2001.
- [36] 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.
- [37] 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.
- [38] 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.
- [39] 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.
- [40] 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.
- [41] X. Yan, W. Li, W. Chen, W. Luo and C. Zhang, "Weighted K-Nearest Neighbor Classification Algorithm Based on Genetic Algorithm," TELKOMNIKA Indonesian journal of Electrical engineering, vol. 11, 2013.
- [42] N. Suguna, and K. Thanushkodi, "An Improved k-Nearest Neighbor Classification Using Genetic Algorithm," International Journal of Computer Science, vol. 7, July 2010.

#### 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. He currently is a lecturer in the Engineering Department of Tabaran Institute of Higher Education, Mashhad, Iran. His research interests include image processing, machine vision and pattern recognition as well as evolutionary algorithms.



**Mohammad Reza Baghaei Pour**, received the BS and MS degrees in computer engineering from Azad University of Quchan and University of Tehran, Iran in 2008 and 2013, respectively. He currently is a lecturer in the Engineering Department of Ferdows Higher Education Institute, Mashhad, Iran. His research interests include image processing, machine vision, pattern recognition, evolutionary algorithms and network as well as cryptography.



**Seyed Ali Mohammadi**, received the BS and MS degrees in computer engineering from Islamic Azad University of Maybod and Islamic Azad University of Mashhad, Iran in 2007 and 2011, respectively. He currently is a lecturer in the Engineering Department of Tabaran Institute of Higher Education, Mashhad, Iran. His research interests include image processing, machine vision, pattern recognition and machine learning.



**Mohammad Hoseinpour Jajaram**, received the BS and MS degrees in computer engineering from Azad University of Mashhad, Mashhad, Iran in 2008 and 2014, respectively. His research interests include image processing and machine vision.

