

Classification of vehicles' types using histogram oriented gradients: comparative study and modification

Rania Salah El-Sayed¹, Mohamed Nour El-Sayed²

¹Faculty of Science, Al-Azhar University, Cairo, Egypt

²The Electronics Research Institute, Cairo, Egypt

Article Info

Article history:

Received Mar 17, 2020

Revised Jul 14, 2020

Accepted Nov 2, 2020

Keywords:

Ant colony optimization (ACO)

Features' extraction

Improved histogram of oriented gradients (HOG)

Performance evaluations

Support vector machine (SVM)

Vehicle type classification

ABSTRACT

This paper proposes an efficient model for recognizing and classifying a vehicle type. The model localizes each object in the image then identifies the vehicle type. The features of an image are extracted using the histogram oriented gradients (HOG) and ant colony optimization (ACO). A vehicle type is determined using different classifiers namely: the k-nearest neighbor (KNN), support vector machine (SVM), random forest (RF), and Softmax classifiers. The model is implemented and operated on two datasets of vehicles' images as test-beds. From the comparative study, the SVM outperforms the other adopted classifiers and is also better using HOG than that using ACO. A modification is done on HOG by adding the Laplacian filter to select the most significant image features. The accuracy of the SVM classifier using modified HOG outperforms that one using the traditional HOG. The proposed model is analyzed and discussed regardless the local geometric and photometric transformations like illumination variations.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Rania Salah El-Sayed

Department of mathematics and computer science

Faculty of Science, Al-Azhar University (girls Branch)

Nasr city, yossef Abbas str., Cairo. Egypt. P.O.Pox 11754

Email: rania5salah@azhar.edu.eg, rania5salah@yahoo.com

1. INTRODUCTION

Recognition of a vehicle type is an important task for vision-based sensors. A type of vehicle based access control for outdoor sites, buildings and even housing estates have become an important theme. The various traffic monitoring and control systems that depend on the identification type of vehicles become vital in our life. Recognition of Bus, Microbus, Minivan, Sedan, SUV, and Trucks enables determination of optimal assignment of green time for the particular crossroad approach.

Several research efforts were presented for recognizing and classifying the types of vehicles. Examples of such efforts include; but not limited to; the following:-

Apostolos Psylos, *et. al.*, [1] used scale invariant feature transform (SIFT) to recognize manufacturer logo images and model of a vehicle. Then, neural network (NN) approaches were assessed as classifiers and they achieved an average recognition rate about 85%.

Zhen Dong and Yunde Jia, [2] combined distributions of structural and appearance-based features to classify vehicle type recognition model. The authors mentioned that two types of features are computed to obtain compact and discriminative representation of vehicles.

Jiquan Ngiam, *et. al.*, [3] introduced the sparse filtering algorithm based on only one hyperparameter, the number of features as a proposed algorithm for vehicle recognition model. The authors in their research work used a simple cost function for optimization objectives.

Yu Peng, *et. al.*, [4] presented a method for classifying a vehicle type based on adaptive multi-class principal components analysis (PCA). The authors extracted vehicle front width and the location of license plate. Then, authors generated eigenvectors to represent extracted vehicle fronts, then they applied PCA method with self-clustering to classify a vehicle type. The authors built a database including 4924 images of vehicle front view.

Jesse Jin, *et. al.*, [5] presented a model based on SVM for classifying a vehicle's type. The features of an image were extracted from the license plate and width of vehicle front after subtracting the background. The eigenvector of each vehicle-front-image was calculated. The features of an image were entered to the SVM classifier. The authors's model adopted five types of vehicles namely: truck, minivan, bus, passenger car, and sedan (SUV).

Zhen Dong, *et. al.*, [6] presented a method for vehicle type classification. The authors' method used a semisupervised conventional neural network from vehicle frontal view images. The authors presented sparse Laplacian filter learning to obtain the filters of the network with large amounts of unlabeled data. The softmax classifier was trained by multitask learning with small amounts of labeled data. The authors' method was able to learn good features for classification. The learned features were discriminative enough to work well in complex scenes. The authors built BIT-vehicle dataset including more than nine-thousands of vehicle images. The authors' method was effective from the experimental results on both the authors' dataset and the public dataset.

Heikki Huttunen, *et. al.*, [7] presented and discussed the car type recognition using deep neural methods. The adopted car types were: bus truck, van, and small cars. The authors considered deep neural networks and the SVM with SIFT features. The authors' work was operated on a database with more than 6500 images. The achieved prediction accuracy was more than 97%. The authors' approach outperforms some of the early studies in this direction.

Jie Fang, *et. al.*, [8] presented a fine-grained model for vehicle recognition. The model is based on a coarse-to-fine conventional neural network (CNN) architecture. The authors mentioned that the fine grained vehicle recognition model can be defined by locating discriminative parts. The authors proposed a corresponding course to fine method where the discriminative regions are automatically detected based on feature maps extracted by the CNN. A mapping from feature maps to the input image was established to locate the regions and those regions were repeatedly refined until there are no more qualified ones. The SVM classifier was applied and the authors' work outperforms most of the other adopted approaches.

Wei. Sun, *et. al.*, [9] extracted the local feature from four partitioned key patches by improving Canny edge detection algorithm. A set of Gabor wavelet kernels with five scales and eight orientations were used. The authors introduced two-stage classification that used K-NN as the first stage to recognize a large vehicle or a small vehicle depending on global features and sparse representation as the second stage which was based on local features.

Yassin Kortli, *et. al.*, [10] mentioned that facial recognition is important for several applications. Various facial recognition methods were presented and discussed to reduce the amount of calculation and improve the recognition rate. Such methods can be grouped into three categories namely: local feature approaches, subspace learning approaches, and correlation filtering approaches. The authors in their research work presented a comparative study among some facial recognition algorithms. The adopted algorithms were HOG, SIFT, speedup robust features (SURF), and binary robust independent elementary features (BRIEF). The performance of the adopted algorithms was evaluated in terms of true positive rate, false positive rate, and recognition time. The recognition time of SURF was the smallest one while the recognition accuracy of HOG was the best.

The remaining part of this paper will be as follows: Section 2 presents the architecture of the main building blocks of the proposed model. Section 3 presents the preprocessing operations done on any input image. Section 4 presents two methods of feature extraction while sections 5 analyzes the adopted classifiers. Section 6 presents the implementation work which involves some modifications on the feature extraction process by applying the Laplacian filter. A comparative study and discussion of results are presented in section 7. Finally, Section 8 concludes the whole work.

2. ARCHITECTURE OF THE PROPOSED MODEL

A robust system and/or model for vehicle type recognition is proposed and presented. After making some sort of preprocessing operations, the object detection starts. In other words, the robustness of local geometric and photometric transformations like illumination variations are achieved. An object can be detected from an input image as shown in section 3. The feature extraction process and/or method can be developed using two new concepts: the HOG and the ACO. Features of the input image can be extracted using HOG descriptors which are based on laplacian filter for extracting the edge gradients and orientations.

The concept of ACO is used as an optimization approach for the feature selection process. After that the SVM classifier is applied to predict and identify the vehicle type. The adopted vehicle types in this work are: Bus, Microbus, Minivan, Sedan, SUV, and Trucks. The main building blocks are briefly shown in Figure 1.

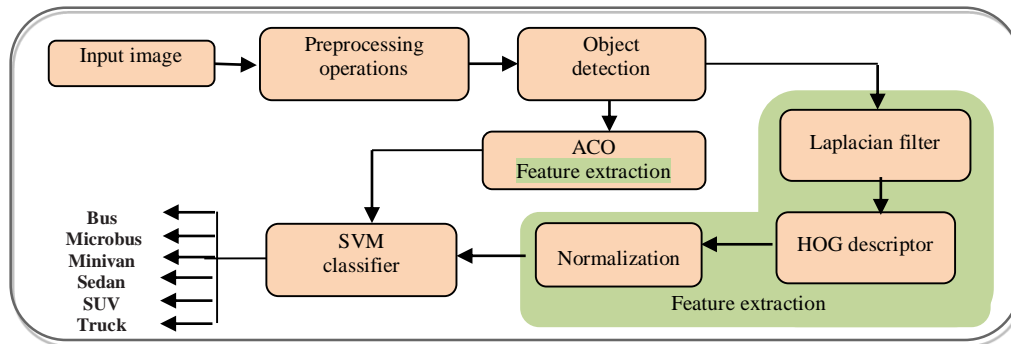


Figure 1. Architecture of the proposed vehicle type recognition system

3. PREPROCESSING OPERATIONS

Several pre-processing operations have been performed on the input image. Step 1, read the input image as a colored image i.e. in RGB. Step 2, convert the RGB values to grayscale values by forming a weighted sum of the R, G, and B components using (1) [11]:

$$G_g(x,y) = 0.299 I_{R(x,y)} + 0.587 I_{G(x,y)} + 0.114 I_{B(x,y)} \quad (1)$$

Where I is the image and x and y are representing the pixel coordinates. Step 3, test the image; if the image contains two objects then we crop region of each object. Step 4, resize the cropped image to an optimal size (128x128) because it can be of several sizes. Processing of a big sized image makes the operation slow. It also requires unnecessary power and computation time. Step 5, rebuild a database containing images with the same size and each image contains only one object. Figure 2 briefly presents the preprocessing operations done on some input images taken from the adapted dataset.



Figure 2. An example of vehicle detection approach with every step's output

4. FEATURE EXTRACTION

In this section, we mainly describe two well-known feature extraction approaches that can be used for vehicle type recognition system.

4.1. Histogram of oriented gradients (HOG)

The image is divided into small connected regions called cells of size $N \times N$ pixels. HOG [12] is able to provide the edge direction and shape of the object. HOG specifies the magnitude $m_{x,y}$ and orientation $\Theta_{x,y}$ parameters of the feature regions ($N \times N$) in an image as shown respectively in (2-3) [10].

$$m_{x,y} = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \quad (2)$$

$$\theta_{x,y} = \tan^{-1} \frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)} \tag{3}$$

Where L be the intensity (grayscale) and x, y are the gradients of each cell in both the horizontal and vertical directions respectively. The amplitude of the gradient and the orientation of each pixel in the cell are voted in 9 bins. To improve the accuracy, the local histograms can be contrast-normalized by calculating a measure of the intensity across a larger region of the image, called a block, and then using this value to normalize all cells within the block. This normalization results in better invariance to the changes in illumination and shadowing. The number of cells in a block is specified as a 2-element vector. A large block size value reduces the ability to suppress local illumination changes. Because of the number of pixels in a large block, these changes may get lost with averaging. Reducing the block size helps to capture the significance of local pixels. Smaller block size can help suppressing the illumination changes of HOG features. Figure 3 presents the HOG descriptor that was applied on a single image of size 128x128. The image was divided into cells of size 8x8 for each and block of size 2x2 for each (i.e no. of cells in block 2x2) and finally, the HOG would generate a histogram for each cell. The histograms are created using the magnitude and orientations of pixel values in cell 8x8. The histogram is a vector of 9 bins corresponding to angles 0, 20, 40, 60 ... 160. Figure 4 illustrates the contribution of one pixel value in the cell. It has a direction 114 and a magnitude 89. So, it adds 26.7 to 6th bin and 62.3 to 7th bin. Then, contributions of all the pixels in the 8x8 cells are added up to create the 9-bin histogram. As the block has four cells; each cell is represented by a matrix 9x1 so, the block is considered a single matrix 4x9x1 i.e 36x1. Hence, the total number of features for an image would be 15 x 15x 36x1= 8100 features.

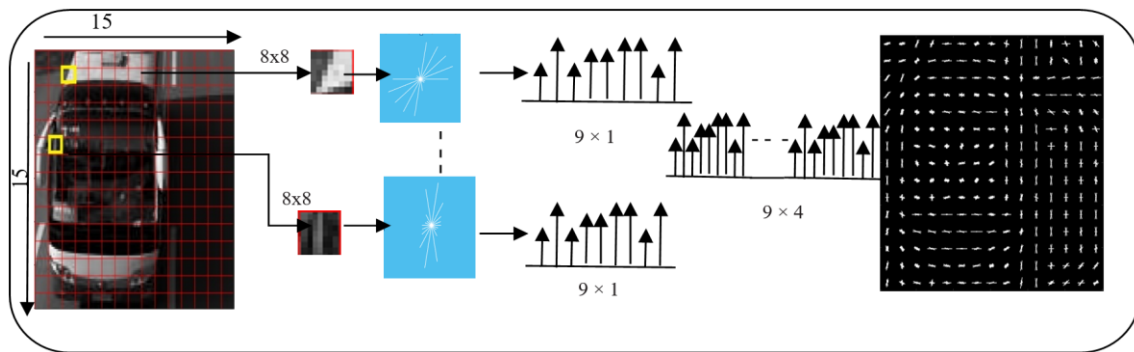


Figure 3. HOG features extraction process: image is divided into cells of size 8x8 pixels. The orientation of all pixels is computed and accumulated in a 9-bins histogram of orientations. All cell histograms are concatenated in order to construct the final features vector.

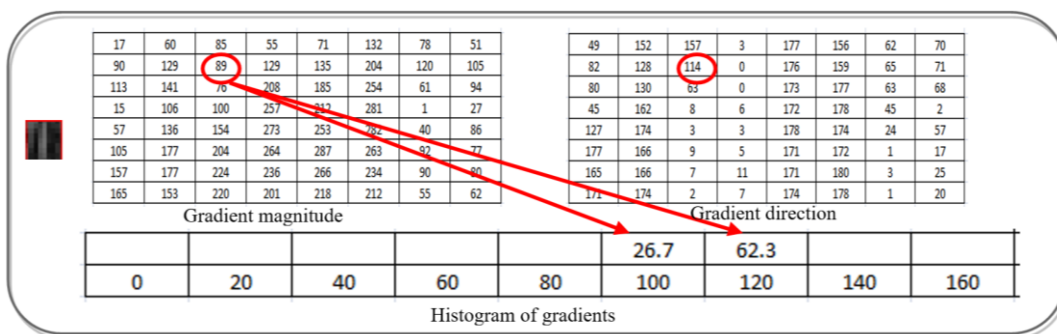


Figure 4. Contributions of one pixel value in the 8x8 cell

4.2. Ant colony optimization (ACO)

ACO was introduced by Marco Dorigo and Thomas Stützle [13] as a nature-inspired metaheuristic approach for the solution of hard combinatorial optimization (CO) problems. The ACO [14-15] can be used

to extract the features of an image. This can be done by a well form of representation of the image. The problem of image recognition can be represented by a graph with a set of nodes connected with edges. The nodes and edges represent respectively the features and weights. To find the optimal feature subset, it is important to consider an ant traversal within the graph with a minimum number of nodes which are visited and then satisfy a stopping criterion using K-NN function. Figure 5 represents the construction of subset by one ant in ACO for feature selection. The ant on feature f6 and constructs one subset features {f6, f1, f2, f3} from all features f1, f2, ..., f6 according to stopping criteria.

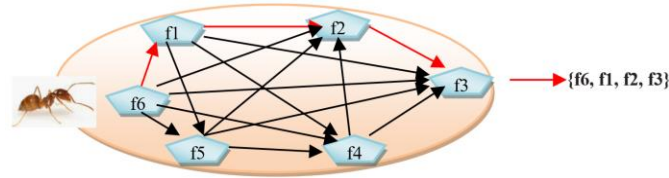


Figure 5. An example, one ant placed in feature f6 and constructed one subset {f6, f1, f2, f3}

The conventional probabilistic transitions of an ant k at feature i choosing to travel to feature j at time t can be defined as in (4-5).

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{u \in U} [\tau_{iu}(t)]^\alpha \cdot [\eta_{iu}]^\beta} & (if j \in U) \\ 0 & otherwise \end{cases} \quad (4)$$

$$\tau_{ij}(t) = \rho \tau_{ij}(t-1) + \Delta\tau_{ij}(t) \quad (5)$$

Where, u is a feasible feature, U is feasible neighborhood of ant k , τ_{ij} is the pheromone value associated with edge (i,j) , η_{ij} is the heuristic desirability of choosing feature j when at feature i , ρ is the evaporation coefficient, $\Delta\tau_{ij}(t)$ is the sum of the contributions of all ants that used moving (ij) to construct their solution by (6) and α, β parameters determine respectively the relative pheromone value and heuristic information.

$$\Delta\tau_{ij}(t) = \sum_{k=1}^n \Delta\tau_{ij}(t)^k \quad (6)$$

Where n is the number of ants and $\Delta\tau_{ij}(t)^k$ is the amount of pheromone value laid on edge (ij) by ant k .

5. ANALYSIS OF SOME ALGORITHMS FOR CLASSIFYING VEHICLES' TYPES

As mentioned in literature, there are several types of classification algorithms and/or methods. Such classifiers can be used to assign and predict a predefined class label to an input instance. The instance here represents the features of input vehicle image and the class label is the type of vehicle. The classifiers are based respectively on statistical approach, structural approach, semantic approach ... etc.

In this research work four different classification methods are adopted, analyzed, operated and tested. The classifiers are: K-NN, SVM, RF and Softmax classifiers. The adopted classifiers are briefly mentioned in the following subsections.

5.1. K-nearest neighbors (K-NN) classifier

The K-NN supervised machine learning algorithm can be used for both classification and regression predictive problems. In K-NN classification [16], the output is a class membership. An object is classified based on voting of its neighbors, with the object being assigned to the class most common among its K-NN. The K-NN algorithm is briefly described as shown below. For more the reader can refer to [16-17].

K-NN Algorithm

1. Load data as a matrix where each row represents an image.
 2. Select K to be the specified number of neighbors.
 3. Calculate the Euclidean distance between test data x^* and each row of the training data $[x_1, x_2, \dots, x_n]$. The Euclidean distance is calculated as in (7).
-

$$D(x, x^*) = \sqrt{\sum_1^n (x_n - x^*)^2} \quad (7)$$

4. Select the K rows which have the shortest distance.
5. Select the most frequent class of these rows

5.2. Support vector machine (SVM) classifier

SVM is a discriminative supervised learning classifier defined by a separating hyperplane. SVM was introduced by Cortes and Vapnik [18] for two class classification problems. Authors in [19-20] introduced the modified SVM classifier to handle the multi-classes problem. They constructed the SVM classifier for each pairwise (one vs. one) of classes and a voting system aided to elect the predicted class when an unseen item is tested. If N be the number of classes, constructs $N(N-1)/2$ decision functions for all the combinations of class pairs. The training data for corresponding two classes were used to implement a decision function for a class pair. For more details about the SVM classifier, the reader can refer to [12, 21-22].

5.3. Random forest (RF) classifier

The RF classifier is an ensemble learning approach proposed by [23-24]. RF operates by constructing a multitude of decision tree at training time and outputting the class. Each individual tree in the random forest can be used as class prediction and class classification with the most votes which become our model's prediction [25-26].

5.4. Softmax classifier

Softmax is often used in neural networks and convolution neural network [8, 27] to map the non-normalized output of a network to a probability distribution over predicted output classes. i.e We have one input X and a corresponding value Y which can be predicted after passing it to the network/layers. A softmax function outputs a vector that represents the probability distributions of a list of potential outcomes as follows:

$$P(Y = k|X = x_i) = \frac{e^{s_i}}{\sum_i e^{s_i}} \quad (8)$$

Where s_i is an intermediate variable for the distribution, $x \in \mathbb{R}^{k \times 1}$ represents the input feature and k is the number of vehicle types that used in our standard scoring function. The scoring function is,

$$s_i = f(x_i, W) = W^T x_i \quad (9)$$

Where W are the parameters or weights, $i = 1, 2, \dots, k \in \mathbb{R}^k$.

6. IMPLEMENTATION AND PRACTICAL WORK

6.1. Datasets description

All the experimental sessions have been carried out on two datasets as test-beds. The first dataset is called BIT-Vehicle dataset [6] and it includes 9850 vehicle images. Two cameras have been used in different time and places to construct images in the dataset. The image taken by the cameras may be of size 1600x1200 or 1920x1080. The proportion of nightlight images in the whole dataset is about 10%. In addition, the images contain variation in the illumination condition, the scale, the color of vehicles, and the viewpoint. The top or bottom parts of some vehicles are not included in the images and there may be one or two vehicles in one image. All of these challenges can be overcome. All vehicles in the dataset are divided into six categories: Bus, Microbus, Minivan, Sedan, SUV, and Truck with the corresponding number of vehicles for each type 558, 883, 476, 5922, 1392, and 822, respectively. Figure 6 illustrates some examples of the BIT-Vehicle dataset.

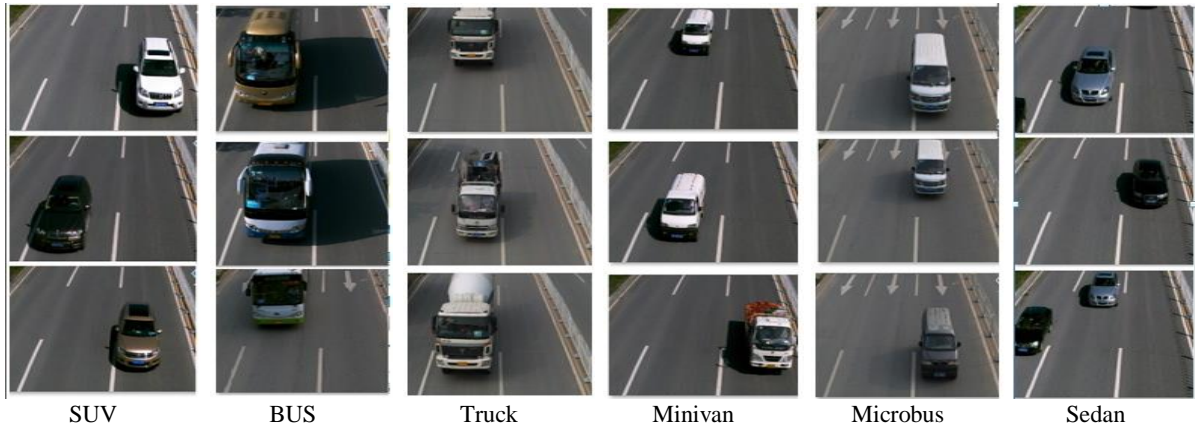


Figure 6. Examples of vehicle images

The proposed model was also tested on the second dataset which is called vehicles-nepal-dataset [28]. It includes 4,800 vehicles' images cropped from videos taken from the streets of Kathmandu. The images of vehicles are divided into five categories namely: Bus, Microbus, Minivan, Sedan, and Truck. Figure 7 briefly illustrates some images of the vehicles-nepal-dataset.

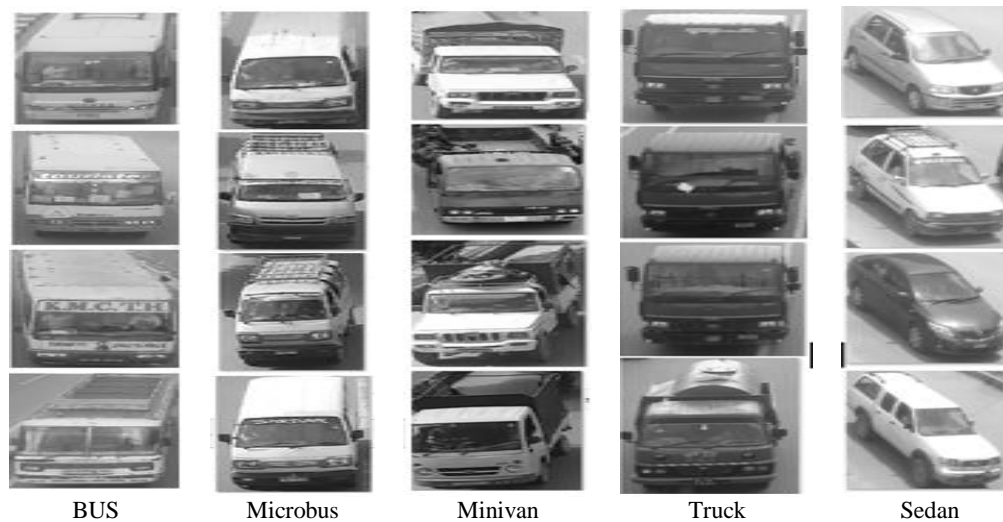


Figure 7. Examples of vehicle images

We have used an Intel Core TM i7 processor (1.80 GHz) with 12 GB of RAM to perform and execute all our experiments. The vehicle type recognition system is implemented using MATLAB R2018a on a 64-bit Microsoft Windows 10 operating system.

For each test on the BIT-Vehicle dataset, we randomly selected 3522 images containing 6-classes where 2400 samples were dedicated for training (400 for each class) and 1122 samples were used for testing. For the vehicles-nepal-dataset, we randomly selected 2850 images containing 5-classes. The number of images or instances were 1850 (370 images for each class) and 1000 for training and testing respectively. In order to give a better estimation of the generalization performance, the reported results of the datasets are the averages of 10 independent experiments because cross-validation in such experiments was adopted.

6.2. Experimental results of classifying vehicles' types

In this section, we evaluate the adopted four classification algorithms and demonstrate the effectiveness and feasibility of the best classifier for vehicle type recognition. First, the K-NN classifier is applied using (7) of the Euclidean distance where the number of neighbors $K=5$. Secondly, we apply the

random forest classifier (RF) based on creating random subsets of training data using the bag ensemble method with randomly subsampling of training features to build the decision trees [26, 29]. The CART model is then trained on each sample. Thirdly, the Softmax classifier is trained by multitask learning with small amounts of labeled data. For a given vehicle image, the network can provide the probability of each type to which the vehicle belongs using W in (9) with parameters λ , $\eta = 0.1$, $\mu = 0.4$, and the threshold is 10^{-4} . Fourthly, the SVM has been applied using one-vs.-one multiclass methods [12, 30] based on the cubic kernel function. The average classification accuracy for a vehicle's type for the four classifiers are summarized as shown in Table 1.

Table 1. Classification accuracies for the different classifiers on BIT-vehicle dataset

Classifier	Average accuracy
K-NN	75.5%
RF	76.9%
Softmax	71.1%
SVM	82.3%

Table 1 shows that the SVM is the best classifier for vehicle recognition system. Table 2 shows the confusion matrix for the 6-classes vehicle types when operating the SVM classifier on the input images of the dataset.

Table 2. Confusion matrix of the 6-classes of vehicle types using the SVM classifier

Vehicle type	Bus	Microbus	Minivan	Sedan	SUV	Truck
Bus	90.37%	1.60%	2.67%	0.00%	0.00%	5.34%
Microbus	1.06%	83.42%	3.74%	2.67%	8.02%	1.06%
Minivan	1.60%	9.62%	62.03%	0.53%	0%	11.22%
Sedan	0.00%	6.42%	1.06%	87.70%	4.81%	0.00%
SUV	0.00%	8.02%	1.06%	13.90%	77.00%	0.00%
Truck	2.67%	4.27%	12.83%	0.00%	0.00%	80.21%

6.3. Improving the classification performance

The next step aims at improving the classification performance. Extraction of features that represent an image is more suitable for pattern recognition applications. These features are used to represent an object in the image. The HOG descriptor and ACO have been used to extract a feature vector for each image after the preprocessing operations. Firstly, the HOG operator is adopted for extracting the features using the setting shown in Table 3. The values of classification accuracy and recognition time for the four adopted classifiers using the HOG descriptor for extracting features are shown in Table 4. Table 5; on other hand; shows the confusion matrix for the best classification accuracy of the SVM classifier for the 6-classes vehicle types.

Table 3. The parameters' setting for HOG

Parameter	Value
Image size	128x128
cell size	8x8
block size	2x2
Number of orientation histogram bins	9

Table 4. Classification accuracies for the different classifiers on BIT-vehicle dataset using HOG feature extraction

Feature Extraction	Classifier	Average accuracy	Recognition time (s)
HOG	K-NN	85.7%	14.152 sec
	RF	80.7%	22.115 sec
	Softmax	78.4%	5.0243 sec
	SVM	89.3%	74.021 sec

Table 5. Confusion matrix of 6-classes of vehicle types with SVM classifier based on HOG

Vehicle type	Bus	Microbus	Minivan	Sedan	SUV	Truck
Bus	94.55%	0.00%	4.85%	0.00%	0.00%	0.61%
Microbus	0.00%	81.82%	4.24%	5.45%	8.48%	0.00%
Minivan	0.00%	1.82%	76.36%	0.61%	0.00%	6.06%
Sedan	0.00%	2.42%	0.61%	90.91%	6.06%	0.00%
SUV	0.00%	6.67%	1.82%	5.45%	86.06%	0.00%
Truck	0.61%	0.00%	6.67%	0.00%	0.00%	92.73%

Secondly, the ACO extracted the optimal features subset from an object in the image using the parameters' setting shown in Table 6. To determine the best number of features in an image different number of features were selected and tested. This was done by using and running the ACO in several experiments. Figure 7 shows the obtained number of features in the run experiments. By changing the threshold value the number of features is also changing. The SVM classifier was run for the different number of features in each experiment. Figure 8 shows the percentage of accuracy values for each experiment. It is shown that the percentage of accuracy values increasing by increasing the number of features. This is clear until a certain number of features; after that the accuracy values decrease by increasing the number of features. This means increasing the number of features than that number is no longer effective. This is shown in Figure 8.

Table 6. The parameters' setting for ACO

Parameter	Value
Number of ants n	10
Maximum number of iterations	100
Control coefficients α	1
Control coefficients β	1
Pheromone τ	0.2
evaporation coefficient ρ	0.5
fitness function	K-NN with (K=5)

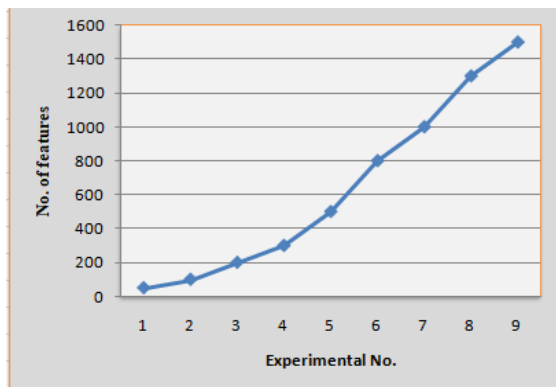


Figure 7. No. of experiments with different No. of features

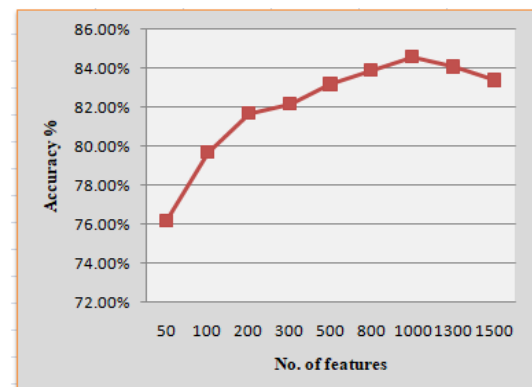


Figure 8. Classification accuracy with different No. of features using ACO

Table 7 shows the classification accuracies and recognition time for the four classifiers after extracting features using the ACO. The confusion matrix for the best classification accuracy of the SVM for the 6-classes of vehicle types is shown in Table 8.

Table 7. Classification accuracies for the different classifiers on BIT-vehicle dataset using ACO feature extraction

Feature Extraction	Classifier	Average accuracy	Recognition time (s)
ACO	K-NN	77.4%	2.0601 sec
	RF	78.4%	5.3811 sec
	Softmax	72.6%	0.3051 sec
	SVM	84.8%	8.0857 sec

Table 8. Confusion matrix of 6-classes of vehicle types with SVM classifier based on ACO

Vehicle type	Bus	Microbus	Minivan	Sedan	SUV	Truck
Bus	90.58%	0.00%	4.35%	0.00%	0.00%	5.07%
Microbus	0.73%	82.48%	4.38%	3.65%	5.11%	3.65%
Minivan	3.42%	7.69%	75.21%	0.00%	0.85%	12.82%
Sedan	0.00%	4.38%	0.73%	89.05%	5.84%	0.00%
SUV	0.00%	6.52%	1.45%	10.14%	81.88%	0.00%
Truck	0.74%	0.00%	7.35%	0.74%	2.21%	88.97%

Tables 4, 7 show that the HOG is the better approach for extracting the vehicle features. The SVM classifier using HOG achieves better accuracy values than the corresponding values using the ACO. On the other hand, the time consumed in recognition using HOG is greater than the corresponding values using ACO. From the experiments, there are some sort of trade-offs between achieving better accuracy and better recognition. In this case we are adopting such approach (i.e HOG) which achieves better accuracy. So, a filter is applied to improve the performance of HOG descriptor and classification accuracy.

6.4. Improving the feature extraction based on HOG descriptor

Local sparse Laplacian filtering is a computationally intensive algorithm. The Laplacian filtering is often applied to an image to calculate the gradient image that extracts the edge gradients and orientations. The feature extraction based on HOG descriptor has been improved after enhancing the edge detection using this appreciated filter. Then, based on their gradients and orientations, a grid of histograms is created for HOG descriptor. Sigma 0.4 and alpha 0.5 parameters of the laplacian filter are involved to process the details and increase the contrast respectively. Figure 9 shows the percentage values of classification accuracy for ten experiments for the adopted classifiers. This was done using the HOG based on the Laplacian filter. The performance of the SVM is better than the other adopted classifiers. Moreover, the average percentage accuracy for the SVM (supported with HOG descriptor using Laplacian filter) outperforms the other ones as shown in Table 9. Table 10, shows the confusion matrix of 6-classes vehicle types when using the SVM classifier after improving the HOG descriptor with the laplacian filter.

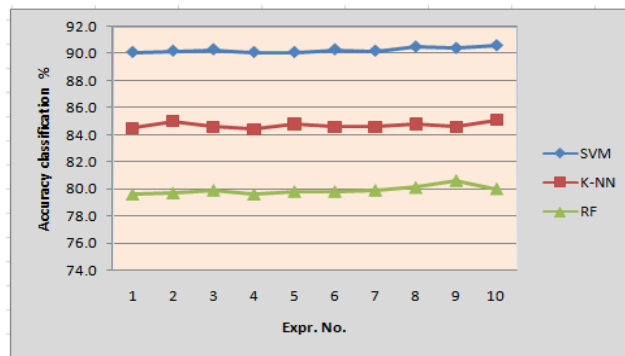


Figure 9. Classification accuracies for the different classifiers of independent experiments after using HOG feature extraction based on laplacian filter

Table 9. Classification accuracies for the different classifiers on BIT-vehicle dataset after using HOG feature extraction based on laplacian filter

Feature Extraction	Classifier	Average accuracy
Lap + HOG	K-NN	84.7%
	RF	79.9%
	SVM	90.3%

Table 10. Confusion matrix of 6-classes of vehicle types with SVM classifier based on HOG after applying the Laplacian filter

Vehicle type	Bus	Microbus	Minivan	Sedan	SUV	Truck
Bus	96.97%	0.00%	1.82%	0.00%	0.00%	1.21%
Microbus	0.00%	91.52%	5.45%	0.61%	2.42%	0.00%
Minivan	0.00%	4.29%	86.43%	0.71%	0.00%	8.57%
Sedan	0.00%	4.85%	0.61%	87.88%	6.67%	0.00%
SUV	0.00%	12.73%	1.21%	6.67%	79.39%	0.00%
Truck	0.00%	0.00%	1.21%	0.00%	0.00%	98.79%

7. COMPARATIVE STUDY AND DISCUSSION OF RESULTS

The proposed architectural model for vehicles’ recognition and classification was implemented, tested and evaluated. The model was run on two chosen databases: the BIT-vehicle dataset and the vehicles-nepal-dataset respectively. Four types of classifiers were adopted. From the results in Tables 1, 4 and 7 it is

shown that the performance of the SVM classifier outperforms the other three ones. This is clear from the average accuracy. From Tables 4, 5, 7 and 8 it is important to mention that the average accuracy for the proposed model using SVM classifier is better for the HOG than that of the ACO. The average accuracy values for the SVM using the feature extraction HOG-based and ACO-based were respectively 89.3% and 84.8%. The proposed approach based on amalgamating and/or combining the HOG and the Laplacian filter gave better results compared with that HOG approach without using filtering. The filtering effect was important as it enhanced the edge detection of a vehicle. Table 11, on the other hand; illustrates a comparative study between the proposed models with other six architecture approaches. The proposed model achieved average accuracy about 90.3% for vehicle type classification for the first dataset.

Authors in [1] used the SIFT to extract features and the NN was used as classifier for recognition. The reported result was (69.32%) when using this architecture for vehicle type recognition. In [2] the authors combined the distributions of the structural features and appearance-based features together and reported accuracy about (82.16%). The authors in [4] used the multi-class PCA as a classifier after extracting the location of the license plate. The reported accuracy was about (83.89%) when using this architecture. The authors in [5] used the SVM classifier after extracting the location of license plate from vehicle front which reported accuracy about (86.23%) for the same architecture. In [3] the authors applied sparse filtering algorithm based on L2 – normalized features which reported accuracy (86.82%). The research in [6] used semisupervised convolutional neural network classifier based on sparse laplacian filter and the reported accuracy was about (87.23%). Therefore, our proposed model is effective in classifying the vehicle types.

Table 11. Comparative Results of different architectures on the BIT-vehicle dataset (6- classes)

Method	Average accuracy
Apostolos Psyllos, <i>et. al.</i> [1]	69.32%
Zhen Dong and Yunde Jia [2]	82.16%
Yu Peng, <i>et. al.</i> [4]	83.89
Jesse Jin, <i>et. al.</i> [5]	86.23
Jiquan Ngiam, <i>et. al.</i> [3]	86.82%
Zhen Dong, <i>et. al.</i> [6]	87.23%
Our proposed model	90.3%

The proposed model was also tested using the vehicles-nepal-dataset. The vehicles' images; in this dataset; are categorized into five classes: Bus, Microbus, Minivan, Truck and Sedan. The features were directly extracted as each image contains only one vehicle. Table 12 illustrates a comparative study between the proposed model and the other adopted ones using this dataset. The proposed model achieved an average accuracy about 98.84% for classifying a vehicle' type. i.e. the proposed model is significant and effective when applied on two different datasets as test-beds. Moreover, the proposed model is expected also to be effective and reliable for classifying the other datasets.

Table 12. Comparative Results of different architectures on the vehicles-nepal-dataset (5- classes)

Method	Average accuracy
Yu Peng, <i>et. al.</i> [4]	85.86%
Jesse Jin, <i>et. al.</i> [5]	87.98%
Jiquan Ngiam, <i>et. al.</i> [3]	88.09%
Zhen Dong, <i>et. al.</i> [6]	92.67%
Our proposed model	98.84%

8. CONCLUSION

This research work presented a proposed model for vehicle recognition and classification. The model was developed, operated and tested using four different classifiers. The model was tested using two chosen image datasets as test-beds. The performance of the proposed model with the SVM classifier was the best. Also, the feature extraction approach HOG-based was better than that based on ACO. The performance of the proposed model HOG-based combined with the laplacian filter was better than that without filtering. This was clear from the accuracy values. The datasets were used under some varying conditions such as different viewpoints, vehicle covered parts, illumination conditions and similar appearances between two classes: "SUV" and "Sedan". The Laplacian filter with the HOG descriptor improved the edge detection. The values of classification accuracy were promising and efficient for the proposed model with the feature extraction HOG-based with filtering. The recognition time of vehicles' images for the model using HOG and

ACO was different. The recognition time also was different for the adopted classifiers. The recognition time using HOG approach was greater than its corresponding time using the ACO approach. This is due to the number of selected features by the ACO which was less than that corresponding number selected by the HOG approach.

REFERENCES

- [1] Apostolos Psyllos, Christos Nikolaos, and Eleftherios Kayafas, "Vehicle Model Recognition from Frontal View Image Measurements", *Computer Standards & Interfaces*, Vol. 33, No. 2, pp. 142-151, Jun 2011.
- [2] Zhen Dong and Yunde Jia, "Vehicle Type Classification Using Distributions of Structural and Appearance-Based Features", *The Proceedings of the IEEE International Conference on Image Processing, ICIP*, pp. 4321-4324, 2013.
- [3] Jiquan Ngiam, Zhenghao Chen, Sonia Bhaskar, Pangwei Koh and Andrew Ng, "Sparse Filtering", *Proceedings of The International Conference on Neural Information Processing Systems*, pp. 1125-1133, 2011.
- [4] Yu Peng, Jesse S. Jin, Suhuai Luo, Min Xu and Yue Cui, "Vehicle Type Classification Using PCA with Self-Clustering", *Proceedings of the IEEE International Conference on Multimedia and Export Workshops, ICMEW*, pp. 384-389, 2012.
- [5] Yu Peng, Jesse S. Jin, Suhuai Luo, Min Xu, Sherlock Au, Zhigang Zhang, and Yue Cui, "Vehicle Type Classification Using Data Mining Techniques", *The Era of Interactive Media*, New York, NY, USA: Springer-Verlag, pp. 325-335, 2013.
- [6] Zhen Dong, Yuwei Wu, Mingtao Pei, and Yunde Jia, "Vehicle Type Classification Using a Semisupervised Convolutional Neural Network", *The Proceedings of the IEEE International Conference on Image Processing, ICIP*, pp. 313-316, October 2018.
- [7] Heikki Huttunen, Fatemeh Shokrollahi Yancheshmeh, and Ke Chen, "Car Type Recognition with Deep Neural Networks", *The IEEE Proceedings of Intelligent Vehicles Symposium*, pp. 1115-1120, August 2016.
- [8] Jie Fang, Yu Zhou, Yeo Yu, and Sidan Du, "Fine-Grained Vehicle Model Recognition using A Coarse-to-Fine Conventional Neural Network Architecture", *The IEEE Transactions on Intelligent Transportation Systems*, pp. 1782-1792, October 2018.
- [9] Wei Sun, Xiaorui Zhang, Shunshun Shi, Jun He and Yan Jin, "Vehicle Type Recognition Combining Global and Local Features Via Two-Stage Classification", *Hindawi, Mathematical Problems in Engineering*, pp. 1-14, 2017.
- [10] Yassin Kortli, Maher Jridi, Ayman Al-Falou, and Mohamed Atri, "A Comparative Study of CFs, LBP, HOG, SIFT, SURF, and BRIEF Techniques for Face Recognition", *The Proceedings of SPIE: Pattern Recognition and Tracking*, Vol. 10649, PP. 106490M-1-106490M-7, April 2018.
- [11] Eduardo Gastal and Manuel Oliveira, "Domain Transform for Edge-Aware Image and Video Processing", *ACM Transactions on Graphics, ToG*, Vol. 30, No. 4, pp. 1-11, 2011.
- [12] Xiangbo Kong, Zelin Meng, Naoto Nojiri, Yuji Iwahori, Lin Meng and Hiroyuki Tomiyama, "A HOG-SVM Based Fall Detection IoT System for Elderly Persons Using Deep Sensor", *Procedia Computer Science, International Conference on Identification, Information and Knowledge in the Internet of Things, IIKI*, Vol. 147, pp. 276-282, 2019.
- [13] Marco Dorigo and Thomas Stützle, "Ant Colony Optimization", MIT Press: Cambridge, MA, USA, 2004.
- [14] Hamidreza Kanan, Karim Faez, and Sayyed Mostafa Taheri, "Feature Selection Using Ant Colony Optimization (ACO): A New Method and Comparative Study in the Application of Face Recognition System", *Proceedings of the IEEE International Conference on Data Mining, ICDM*, pp. 63-76, 2007.
- [15] Sahar Ebadinezhad, Ziya Dereboylu and Enver Ever, "Clustering-Based Modified Ant Colony Optimizer for Internet of Vehicles (CACIOV)", *MDPI, Sustainability (Switzerland)*, pp. 1-22, 2019.
- [16] Jianping Gou, Wenmo Qiu, Zhang Yi, Qirong Mao and Yonghao Zhan, "A Local Mean Representation-based K-Nearest Neighbor Classifier", *Journal ACM Transactions on Intelligent Systems and Technology, TIST*, Vol. 10, No. 3, May 2019.
- [17] Samah Osama and Mohamed Nour, "Feature Selection Methods for Predicting the Popularity of Online News: Comparative Study and a Proposed Method", *The Journal of Theoretical and Applied Information Technology*, Vol. 96, No. 19, pp. 6969-6980, 2018.
- [18] Corinna Cortes and Vladimir Vapnik, "Support-Vector Networks", *Springer, Machine Learning*, Vol. 20, No. 3, pp. 273-297, 1995.
- [19] S. Knerr, L. Personnaz and G. Dreyfus, "Single-Layer Learning Revisited: A Stepwise Procedure For Building and Training A Neural Network", *Springer, Neurocomputing*, Vol. 68, pp. 41-50, 1990.
- [20] Xin Wang, Hao Zhang and Yichao Wu, "Multiclass Probability Estimation With Support Vector Machines", *Journal of Computational and Graphical Statistics*, Vol. 28, No. 3, pp. 586-595, Jun 2019.
- [21] Utpal Barmana and Ridip Choudhury, "Soil Texture Classification Using Multi Class Support Vector Machine", *ScienceDirect, Journal of Information Processing in Agriculture*, pp. 1-15, August 2019.
- [22] Fereshteh Chamasemani and Yashwant Singh, "Multi-class Support Vector Machine (SVM) Classifiers : An Application in Hypothyroid Detection and Classification", *International Conference on Bio-Inspired Computing: Theories and Applications, BIC-TA*, PP. 27-29, September 2011.
- [23] Leo Breiman, "Random Forests", *Springer, Machine Learning*, Vol. 45, No. 1, pp. 5-32, October 2001.
- [24] Xiang Gao, Junhao Wen and Cheng Zhang, "An Improved Random Forest Algorithm for Predicting Employee Turnover", *Journal of Mathematical Problems in Engineering*, Vol. 2019, pp. 1-12, April 2019.

- [25] Muhammad Manzoor, Yasser Morgan and Abdul Bais, “Real-Time Vehicle Make and Model Recognition System”, *MDPI, Machine Learning and Knowledge Extraction*, pp. 611-629, 2019.
- [26] Mauro Castelli, Leonardo Vanneschi and Alvaro Largo,” Supervised Learning: Classification”, *Journal of Encyclopedia of Bioinformatics and Computational Biology*, Vol. 1, pp. 342-349, 2019.
- [27] Abhishek Kumar and Hal Daume, “Learning Task Grouping and Overlap in Multi-Task Learning”, *Proceedings of the International Conference on Machine Learning, ICML*, pp. 1383-1390, 2012.
- [28] <https://github.com/sdevkota007/vehicles-nepal-dataset>
- [29] Madden Michael and Munroe Daniel,” Multi-Class and Single-Class Classification Approaches to Vehicle Model Recognition from Images”, *Proceedings of the Irish Conference on Artificial Intelligence and Cognitive Science, AICS*, September 2005.
- [30] Mengwan Jiang and Haoliang Li, “Vehicle Classification Based on Hierarchical Support Vector Machine”, Springer, *Proceedings of the International Conference on Computer Engineering and Networking*, Vol. 1, pp. 593-600, 2014.

BIOGRAPHIES OF AUTHORS



Rania Salah El-Sayed is lecturer of Computer Science at Faculty of science, Al-Azhar University, Cairo. Egypt. She received her Ph.D and M.Sc in Pattern Recognition and Network security from Al-Azhar University in 2013 and 2009 respectively. Her B.Sc degree in Math & Computer Science was received in 2004 from Al-Azhar University. In 2012, she received CCNP security certification from Cisco. Her research interests include pattern recognition, machine learning & network security.



Prof. Dr. Mohamed Nour Elsayed is a professor of computer engineering at the Electronics Research Institute, Cairo. He was graduated from the Computer Engineering Department at the Faculty of Engineering, Ain Shams University, Cairo. He obtained his M.Sc. and Ph.D. in 1987 and 1993 respectively. He taught more than twenty-years ago at the American University in Cairo as a part-time instructor. He taught also five years ago at Princess Nourah University, Riyadh, Saudi Arabia. He was the head of the Informatics Department and Vice-President of the Electronics Research Institute, Cairo. He is an IEEE member and a reviewer of some national and international computer journals. The areas of his interest include; but not limited to; high performance computing, computational linguistics, and artificial intelligence.