

# Vehicle Type Classification Using Hybrid Features and a Deep Neural Network

Sathyantarayana N., Vemana Institute of Technology, India\*

Anand M. Narasimhamurthy, International School of Engineering (INSOFE), India

## ABSTRACT

Currently, considerable research has been done in vehicle type classification, especially due to the success of deep learning in many image classification problems. In this research, a system incorporating hybrid features is proposed to improve the performance of vehicle type classification. The feature vectors are extracted from the pre-processed images using Gabor features, a histogram of oriented gradients, and a local optimal-oriented pattern. The hybrid set of features contains complementary information that could help discriminate between the classes better; further, an ant colony optimizer is utilized to reduce the dimension of the extracted feature vectors. Finally, a deep neural network is used to classify the types of vehicles in the images. The proposed approach was tested on the MIO vision traffic camera dataset and another more challenging real-world dataset consisting of videos of multiple lanes of a toll plaza. The proposed model showed an improvement in accuracy ranging from 0.28% to 8.68% in the MIO TCD dataset when compared to well-known neural network architectures.

## KEYWORDS

Ant Colony Optimizer, Camera Response Model, Deep Neural Network, Gaussian Mixture Model, Image Classification, Object Detection, Vehicle Type Classification

## INTRODUCTION

In recent decades, vehicle classification has played a vital role in intelligent transportation systems, because the usage of vehicles has become increasingly universal in human life due to the rapid development of society. For vehicle detection and classification, existing methodologies utilize various types of information such as radar signal and acoustic signature (Zhou & Cheung, 2016). The performance of these methodologies is vulnerable to several environmental variations such as weather, illumination noise and so on (Jiang et al., 2017). With the proliferation of cameras, an abundance of video data related to vehicles, such as traffic on highways, road intersections, toll booths, etc. has become available for analysis, insights and even real-time actions. For example, in traffic measurement and management, vehicle detection and classification delivers important information and also assists in road planning and maintenance by understanding the distribution of dissimilar vehicle classes (Javadi et al., 2018; Wang et al., 2017). Also, vehicle detection and classification have become a research area in video-based intelligent transportation systems (Mithun et al., 2012). For example, counting the vehicles in busy intersections helps in reducing the level of traffic congestion. Due to all the above-mentioned reasons, there has been a significant amount of research undertaken,

DOI: 10.4018/IJAMC.292518

\*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

which addresses the challenging task of vehicle classification with the help of image/video data related to vehicles.

The research work on vehicles has received significant interest among researchers includes the applications such as fine-grained vehicle classification, vehicle detection, vehicle identification, vehicle classification and so on (Ke & Zhang, 2020; Rachmadi et al., 2018) from image/video data. The fine-grained vehicle detection and classification becomes a challenging problem (Yang & Lei, 2014; Li et al., 2019) due to the low interclass and high intra-class variance of images. Many classifiers have been used for vehicle classification such as AdaBoost algorithm (Chen et al., 2018; Wen et al., 2014), dynamic Bayesian network (Kafai & Bhanu, 2011), support vector machine (Sentas et al., 2018), artificial neural fuzzy inference system (Murugan & Vijaykumar, 2018), invariant Charlier moments (Aqel et al., 2017), etc. The vehicles have unique visual and structural characteristics compared to other objects, and also have small inter-class distances and larger intraclass variation. These factors make the detection and classification of vehicles difficult (Yu et al., 2017; Biglari et al., 2017). In recent decades, Deep Neural Networks (DNNs) have been used extensively in image classification problems. The DNN classifier is the best choice for vehicle type classification when the additional prior information about the images is unavailable. Although DNNs have the advantage that the images can directly be fed as inputs, i.e. DNNs can play the role of feature extractor and classifier combined, this requires a large amount of training data and involves significant training time and computational resources. The requirement of training data size is reduced using strategies like transfer learning, where the deep neural network-based solutions still involve significant computation. While it is hard to come up with a single compact set of powerful descriptive features for complex image classification tasks that global-level descriptive features such as Histogram of Oriented Gradients (HOG) and Gabor filters have been used successfully in object detection tasks. By using these aforementioned feature descriptors instead of the raw images, the amount of training data can be reduced and the performance vehicle image classification can also be improved.

Although there has been a lot of research on vehicle classification and commercial solutions exist, many of these are developed for western scenarios. The Indian scenario presents its specific challenges, including different vehicle types, uncertain lane demarcations, and so on. Due to the presence of highly varied scenarios, there is a high probability that methods that perform well on publicly available datasets, may not perform that efficiently on the dataset obtained from the Indian scenario. Initial experiments were conducted on the data collected from the publicly available MIOvision Traffic Camera Dataset (MIO TCD) and a real-time toll plaza dataset. The proposed approach yielded better performance on the MIO TCD and toll plaza dataset as compared to results obtained from existing methods. In this experiment, a Camera Response Model (CRM) is used to enhance the visual quality of the images. Additionally, Gaussian Mixture Model (GMM) is used to detect the vehicles from the enhanced images. Then, Histogram of Oriented Gradients (HOG), Gabor feature, and Local Optimal Oriented Pattern (LOOP) are applied to extract the feature vectors from the pre-processed images. After extracting the features, an Ant Colony Optimization (ACO) based method is applied to reduce the dimension of the feature set, this in turn yields a hybrid set of features and helps to attain better classification performance. Finally, Deep Neural Network (DNN) classifier is applied to classify the types of vehicles from the images. The performance of the proposed approach was compared with six different Deep Neural Network models, namely, AlexNet, Inception V3, ResNet 50, VGG 19, Xception and DenseNet. Standard measures such as sensitivity, specificity, accuracy, error rate, False Omission Rate (FOR) and False Discovery Rate (FDR) were used for evaluation. This paper is organized as follows: A few recent research papers related to vehicle type classification are surveyed in section 2. The details of the proposed multi-objective ACO with the DNN approach are given in section 3. Section 4 discusses the experimental analysis of the proposed approach. The conclusion of this research work is presented in Section 5.

## Literature Survey

This section provides an overview of relevant literature where the computer vision-based system for vehicle classification involves many components such as preprocessing, feature extraction and classification. Traditionally, many approaches involve pre-processing such as background subtraction and possibly transformations of the image intensities, and subsequently feature extraction followed by classification. More recently, a significant amount of literature relates to the use of Deep Neural Networks for object detection and image classification tasks.

A variety of features have been proposed in the literature for object detection include the Haar-like features used in an AdaBoost framework proposed by Viola and Jones (2001) for face detection, Histogram of Oriented Gradients (HoG) proposed by Dalal and Triggs for pedestrian detection (2005), Local Binary Pattern (LBP) (Ojala et al., 1996), Gabor filters (Gabor, 1946) and Local optimal-oriented pattern (LOOP) (Chakraborti et al., 2018). Typically, many of these feature descriptors are low-level informative features that capture few texture information. These features individually are not discriminative enough but several such features generated from various regions of the image can be combined in a suitable classification framework to provide good discrimination between the object classes. Many of these feature descriptors have also been used for vehicle classification. For instance, Yan et al. (2016) used the Histogram of Oriented Gradients (HoG), along with an AdaBoost classifier framework for vehicle classification. Wang et al. (2019) implemented a new Spatio Temporal Sample Consistency (STSC) algorithm for enhancing the efficiency of background subtraction in complex scenes. The developed algorithm includes two main steps (i) vehicles were identified from the interference of illumination variation and then the shadow of vehicles was recognized after vehicle identification, (ii) feature level fusion method was utilized for classifying vehicle types and pedestrian utilizing vehicle symmetry, area, face feature, and plate number. The developed algorithm performance was validated on three datasets; MIO TCD, Beijing Institute of Technology (BIT) vehicle, and CDnet 2014. The vehicle type classification and motion vehicle detection were helpful in numerous applications such as post-event forensics, anomaly recognition, activity recognition, car counting, and vehicle tracking. However, the developed STSC algorithm attained lower accuracy in multi-class classification.

Zhuo et al. (2017) utilized Convolutional Neural Network (CNN) for vehicle classification which consists of two phases such as fine-tuning and pre-training. In the pre-training phase, GoogLeNet was utilized to obtain the initial model with connection weights. In the fine-tuning phase, the initial model was fine-tuned on ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC-2012) dataset for obtaining the final classification model. The ILSVRC-2012 database includes six vehicle categories namely, van, bus, motorcycle, car, truck and minibus. The experimental validation shows that the developed model significantly improves the feature extraction and classification performance. In contrast, CNN models need a huge amount of data for better classification and are also computationally expensive. Liu et al. (2017) presented DNN with a balanced sampling model to classify the imbalanced data or different vehicle categories bus, truck, car, etc. which were collected from the visual traffic surveillance sensors. The developed model includes two phases, (i) data augmentation with balanced sampling was utilized for alleviating the unbalanced dataset concern, and (ii) an ensemble of CNN models with dissimilar architecture was constructed for training and testing the database. The balanced sampling data augmentation method was used to decrease the classification bias. In this paper, the performance of the CNN model was analyzed on the MIO TCD dataset. In a few instances, the CNN model failed to attain better classification performance, due to missing object details and blurred images.

Wang et al. (2019) developed a new vehicle type classification model based on deep learning; faster Region CNN (R-CNN). In this work, the convolutional layer was sharpened using a region proposal network that significantly diminishes the computational time. This literature study report accuracy of 90.65 and 90.51% to cars and trucks respectively. The developed method not only time-saving but also has more robustness and higher accuracy based on experimental results. Derrouz et

al. (2019) implemented a 2-layer vehicle type classification system based on 3D parameters and local features of the vehicle. The disparity map developed from stereo images was utilized to extract the 3D feature vectors of the vehicles in the first layer. The extracted 3D feature vectors were utilized to calculate the length, width, and height of the vehicles. Then, a gradient-based methodology was used to extract the 2D features and a dimensional reduction algorithm was used for decreasing the extracted feature vectors size in the second layer. Both the 2D and 3D feature vectors were fused to construct the final feature vector, which was given as the input to the classifiers to classify the vehicle types. In this work, Moroccan and BIT databases were used to validate the performance of the developed model. The 2D feature extraction was accomplished using gradient-based features, which were not enough to classify the dissimilar type of vehicles. Also, Luo et al. (2018) focused on motor vehicle localization and classification by introducing a new MIO TCD database. The MIO TCD database is a large database that includes 11 traffic object classes like background, car, work van, pedestrian, bus, pickup truck, single-unit truck, bicycle, non-motorized vehicle, articulated truck and motorcycle.

Dong et al. (2015) implemented a semi-supervised CNN model for vehicle type classification. In this work, a sparse Laplacian filtering technique was used to capture the discriminative and rich information of the vehicles from a large amount of unlabeled data. Additionally, a softmax classifier was trained by multi-task learning with a limited amount of labeled data. Unlike existing models, the developed model automatically learns better features for the classification task, where the learned features were discriminative enough to attain better performance in complex scenes. They used the BIT vehicle database to validate the performance developed model. The experimental outcome shows that the developed model attained significant performance in vehicle type classification in light of accuracy. Due to the factors like illumination conditions, the CNN model does not encode the orientation and position of the vehicles, which was a major concern in vehicle type classification. Liu et al. (2018) presented a semi-supervised pipeline model for vehicle type classification, which was the combination of DNN and data augmentation based on Generative Adversarial Nets (GANs). The developed model includes three phases, at first GANs were employed to generate the adversarial samples. In the second phase, the CNN model was used for training the original imbalanced dataset and then sample selection was carried out for filtering the lower quality adversarial samples. Finally, an ensemble model was used for vehicle type classification. In the experimental phase, the MIO TCD dataset was used to demonstrate the efficiency of the developed model. This proposed model was compared with two networks namely ResNet and Inception.

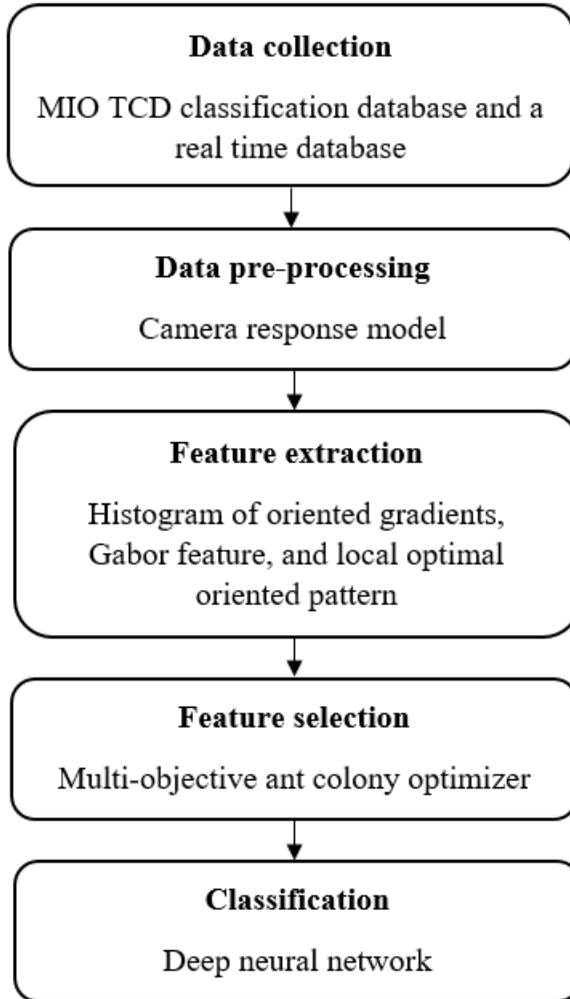
Akilan, et al. (2017) has analysed the effect of combining high level features with multi deep ConvNets for scene or object classification. In this literature, three pre trained ConvNets were exploited as the feature extractors, and an individual hidden layer was adopted for transforming the higher level feature space into lower dimension feature space, and finally the feature vectors were combined for harnessing the rich cues of the individual features. In addition, Akilan, et al. (2018) extract CNN features from three pre-trained models such as Inception-V3, VGG-16, and AlexNet. The feature vectors from the single CNN model were mapped into a common sub-space, and then embedded by utilizing arithmetic rules for constructing fused feature vectors. Experimental analysis showed that the multimodal CNN feature fusion technique was well suited for image classification tasks. However, the pre-trained model selects fixed data sometimes that may leads to poor classification performance. In order to address the above stated concerns, a new model is proposed in this paper for vehicle type classification.

## Proposed Approach

The implementations of the deep neural networks is demonstrated in various image classification tasks, but the limitation of the requirement of a significantly large amount of training data is still not addressed. While data augmentation techniques could be used to generate more data from unlabeled images, this will still not completely overcome the limitation with respect to providing a sufficient number of training images for a deep neural network. A methodology is proposed in this research to

address the aforementioned concerns in vehicle type classification from images. The workflow of the proposed model is given in Figure 1.

Figure 1. Workflow of the proposed model



### Datasets Description

In this research work, the MIO TCD classification database and a toll booth dataset are used for experimental investigation and the detailed explanation of the datasets is given as follows:

#### *MIO-TCD Dataset*

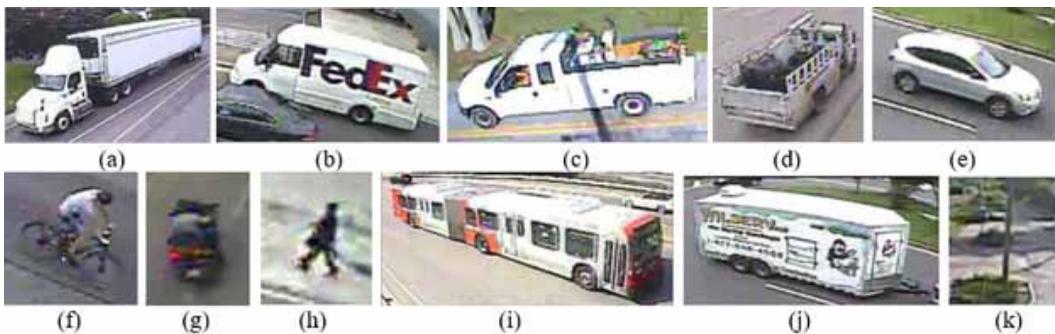
The MIO-TCD classification dataset comprises 648,959 lower resolution images, which includes 11 categories like background, car, work van, pedestrian, bus, pickup truck, single-unit truck, bicycle, non-motorized vehicle, articulated truck, and motorcycle, where the statistics of MIO-TCD classification dataset is given in Table 1. In MIO TCD dataset, objects are pictured in different sizes and recorded

in different view angles and time periods (Luo et al., 2018). The sample images of MIO-TCD dataset is shown in Figure 2.

Table 1. Size of each category in the MIO TCD database

Category	Training	Testing
Background	160,000	40,000
Car	260,518	65,131
Work van	9,679	2,422
Pedestrian	6,262	1,565
Bus	10,316	2,579
Pickup truck	50,906	12,727
Single unit truck	5,120	1,280
Bicycle	2,284	571
Non-motorized vehicle	1,751	438
Articulated truck	10,346	2,587
Motorcycle	1,982	495
Total	519, 164	129,795

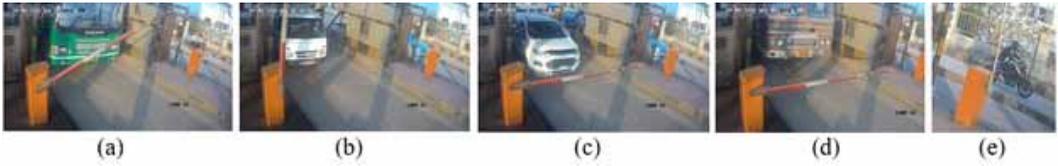
Figure 2. MIO TCD database, a) articulated truck, b) work van, c) pickup truck, d) single-unit truck, e) car, f) bicycle, g) motorcycle, h) pedestrian, i) bus, j) non-motorized vehicle, k) background



### Toll Plaza Dataset

The Toll plaza dataset includes multiple video sequences of different lanes of a toll plaza on a highway. The total length of each video sequence is around two minutes and the frame rate is 25 frames per second. This dataset includes five categories which are tagged as bike, car, bus, pickup truck and lorry. Sample images of different vehicles are shown in Figure 3. In this dataset, 80% training (12000 images) was used for training and 20% (3000 images) for testing.

Figure 3. Tollbooth dataset, a) bus, b) pickup truck c) car d) lorry, e) bike



### Data Pre-processing and Object Detection

After data collection, image pre-processing is accomplished using the Camera Response Model (CRM) proposed in (Ying et al., 2017) to enhance the visual quality of the images. The CRM is applied to help in extracting the relevant information from the images for discriminating between the classes. The CRM includes two components; brightness transformation and camera response function. Initially, the two frames  $p_0$  and  $p_1$  are selected from the video sequences to calculate the brightness transformation function. Further, gamma value is utilized to represent the linear amplification of the image pixel values, which are closely related to real exposed image pixels that are mathematically stated in equation (1).

$$i_1 = g(i_0, k) = \beta i_0^\gamma \quad (1)$$

where,  $\gamma$  and  $\beta$  are denoted as the brightness transformation parameters, which are closely related to the exposure ratio  $k$ . Then, the camera response function  $f(E)$  is employed to identify the relation between  $\gamma$  and  $\beta$  values, which is mathematically indicated in equation (2).

$$f(kE) = \beta f(E)^\gamma \quad (2)$$

If  $\gamma \neq 1$ ,  $f(E)$  becomes a two-parameter, non-linear function. If  $\gamma = 1$ ,  $f(E)$  becomes a power function, which perfectly fits the camera. After pre-processing the images, a Gaussian Mixture Model (GMM) based approach is used to detect the vehicles from the images. The GMM is a parametric probability density function, which represents a weighted sum of Gaussian component density (Zhang et al., 2016). Usually, GMM models continuous features in image classification systems. A pre-processed and vehicle-detected images are graphically stated in Figures 4 & 5.

Figure 4. a) Input image, b) pre-processed image and c) vehicle detected image



Figure 5. a) Image with multiple vehicles in a frame and b) Multiple vehicles detected image



## Feature Extraction

After detecting the vehicles, the region corresponding to the detected vehicle is cropped and resized to 64 x 64. The feature extraction is carried out on the resized 64x64 images by utilizing Gabor features (Khan et al., 2016), HOG (Liu et al., 2018) and LOOP (Dong et al., 2015). The Gabor features are useful for analyzing the texture patterns of the images. Also due to a more general set of parameters (i.e. more degrees of freedom), the Gabor features could provide higher flexibility in extracting the shape of the vehicles. Additional features were to capture low-level information relevant to classification. The Histogram of Oriented Gradients (HoG) feature descriptor operates on localized cells and upholds the geometric and photometric transformations within a local region, this could help in extracting the structural and visual properties of the objects of interest (vehicles in our case). The LOOP is a binary descriptor that is derived from Local Binary Pattern, where it encodes the rotation invariance of the objects. Gabor features were run at multiple scales and orientations that yielded 10240 features, along with the supplementation of 1764 HoG features and 256 LOOP features, which yields a total of 12260 features for each candidate image. After extracting the feature vectors from the pre-processed images, feature selection is carried out using a multi objective Ant Colony Optimization (ACO) approach.

## Feature Selection

Ant Colony Optimization (Parsons, 2005) is a metaheuristic optimization algorithm which originally inspired by the behavior of ants seeking a path between their colony and a source of food. ACO is originally proposed in the early 1990s, and since then has been used for many discrete optimization problems. ACO-based heuristics are quite popular for the Travelling Salesman Problem and similar problems such as the shortest route problem (e.g. (Sun et al., 2010)). Most implementations utilize a distributed positive feed-back parallel computing process, due to the ease of combining with other algorithms.

In this research work, a multi-objective version of ant colony optimization (ACO) is used for feature selection. At every cycle, each ant constructs a solution and then pheromone trails are updated in multi objective ACO by using negative ratio association and ratio cut (objective functions). Here, every extracted feature vector is associated with an amount of heuristic information based on the dependency degree. Practically, the heuristic information values are equal to zero or smaller than the largest one. On the basis of transient probability  $P_{r,s}^k(t)$ , the features are selected probabilistically from the extracted feature vectors. The algorithm stops the iteration, when the termination condition is met. Generally, the termination condition is a maximum number of cycles or a given time limit. As mentioned earlier, the multi objective ACO algorithm select the relevant features from the extracted

feature vectors that helps in improving the classification performance. The fitness function of multi objective ACO algorithm is  $fit_i = \frac{1}{1 + N_i}$ ,  $N_i = \frac{1}{D_{ratio}} \sum_{i,j=1}^{D_{ratio}} d(k_j, \xi_i^{Cluster_{Known}(k_j)})$ , where  $D_{ratio}$  represents ratio cut,  $k_j$  indicates negative ratio association, and  $\xi_i = 100$  denotes initial population size.

The steps involved in the multi objective ACO algorithm are given below, Initialize the population of ant colony on each edge  $\tau_i(0) = C$ , where all the solutions are generated with random probability, as shown in equation (3).

$$C = (\tau_1, \tau_2, \dots, \tau_l), \tau_i = \begin{cases} 0 & \text{Randomly placed, } i = 1, 2, \dots, l \\ 1 & \end{cases} \quad (3)$$

where,  $\tau_i$  is indicated as pheromone value, which is updated based on negative ratio association and ratio cut.

In the solution construction phase  $s$ , the  $k_{th}$  ant at time  $t$  is positioned  $r$  using equation (4).

$$s = \begin{cases} \arg \begin{cases} \max \{ \tau_{r,u}(t), \eta_{r,u}(t) \} \\ u \in N_1^k(t) \end{cases} & \text{while } q \leq q_0 \\ S & \text{else} \end{cases} \quad (4)$$

where,  $\eta_{r,u}(t)$  is stated as problem specific heuristic information,  $\tau_{r,u}(t)$  is denoted as pheromone trail at time  $t$ ,  $q$  is denoted as uniformly distributed random number, which has the range of  $[0, 1]$ ,  $q_0$  is indicated as pre-specified parameter ( $0 < q_0 < 1$ ),  $N_r^K(t)$  is represented as set of optimal features, which are not assigned by ant  $k$  at time  $t$ , and  $S$  is denoted as index of features selected from  $N_r^K(t)$  based on the probability distribution as given in the equation (5).

$$P_{r,s}^k(t) = \begin{cases} \frac{\tau_{r,s}(t), \eta_{r,s}(t)}{\sum k, N_1^k \tau_{r,s}(t), \eta_{r,s}(t)} & j \in N_r^K(t) \\ 0 & \text{else} \end{cases} \quad (5)$$

Step 3: Terminate if termination criteria is reached, or else go to the next step.

Update the pheromone intensity by a mutate operator as indicated in the equations (6) and (7);

$$\tau(t) \rightarrow \tau(t+1).$$

$$\tau(t) = (\tau_1, \tau_2, \dots, \tau_l), \tau_i = \begin{cases} 0 \\ 1 \end{cases} \text{ randomly placed} \quad (6)$$

$$\tau(t+1) = (\tau_1, \tau_2, \dots, \tau_{r-1}, \tau_r', \dots, \tau_l), \tau_r' = \begin{cases} 0 & \tau_r = 1 \\ 1 & \tau_r = 0 \end{cases} \quad (7)$$

where,  $r$  is represented as random number, which has the range of  $[0.1]$  and then go to step 2. The number of features computed for each frame is 12260. Using the multi-objective, the number of these features is reduced to 3678. The parameter setting of multi objective ACO algorithm is indicated as follows; number of ants is 100, maximum uncovered cases is 10, evaporation rate is 0.15, number of rules converged is 10,  $\alpha=1$ , and  $\beta=1$ .

## Classification

The topmost ranked 3678 features (30% of total) selected by multiobjective ACO are fed as inputs to a classifier. These derived features reduce the risk of overfitting and speeding up training (and testing). In this experiment, a deep neural network of dimensions 3678 - 500 - 250 - 6 was used for classification, the output layer was a softmax layer with 6 nodes corresponding to 6 classes (5 vehicle classes + background). Multiple autoencoders were trained and used to initialize the layers of the deep neural network. In the Matlab toolbox, these multiple autoencoders were called stacked autoencoders. However, a 'stacked autoencoder neural network' has different meanings with respect to different contexts, therefore in this research paper, the term Deep Neural Network (DNN) itself will be used for further discussion.

The deep neural network; stacked autoencoder consistently performed better than all other classifiers. In another set of experiments, different optimizers were used for feature selection and then fed as inputs to the stacked autoencoder deep neural network. All the above experiments are described in detail in section 4. Here, a brief general description of neural networks and autoencoders is given below. A neural network typically comprises an input and output layer along with multiple hidden layers and can be described mathematically by the equations (8) and (9).

$$Z^{(l)} = y^{(l-1)}W^{(l)} + b^{(l)} \quad (8)$$

$$y^{(l)} = g\left(Z^{(l)}\right) \quad (9)$$

where,  $l \in [1, \dots, L]$  is the  $l^{th}$  layer,  $y^{(l-1)}$  is represented as input to layer  $l$  and output of the previous layer  $l - 1$ ,  $g(\cdot)$  is a nonlinear activation function,  $y^{(0)}$  is denoted as input to a neural network model,  $y^{(L)}$  is the final layer output,  $W^{(l)} \in R^{n_1 \times n_2}$  is a matrix of learnable weights and biases and  $y^{(l)} \in R^{n_0}$  is denoted as the output layer. In this experiment, ReLU is used as an activation function for hidden layer nodes and softmax is used as activation for the output layer to provide a probability interpretation of the classifier output. If  $z = [z_1^{(l)}, z_2^{(l)}, \dots, z_K^{(l)}]$  is the output vector of layer  $L$  (the output layer), the softmax transformed outputs are mathematically in equation (10).

$$\text{softmax}\left(Z^{(L)}\right) = \frac{\exp Z_k}{\sum_{k=1}^K \exp Z_k} \quad (10)$$

where  $K$  is indicates as number of output classes. Cross entropy loss is used in this DNN classification model. Mathematically, the cross-entropy loss is written as in (11).

$$C = -\sum_{k=1}^K y \wedge k^{\log} \left( y_k^{(L)} \right) \quad (11)$$

where,  $y^{(L)}$  is indicated as model output, and  $y \wedge k \in \{0,1\}^k$  is stated as the encoded label. An autoencoder (Hinton & Salakhutdinov, 2006) is a neural network, which is designed to learn a low dimensional encoding of the input. An autoencoder usually employs a bottleneck architecture and can be considered to have two phases, namely encoder and decoder. A standard autoencoder with a single hidden layer can be specified by the equations (12)

$$h_e = a_1 (W_e x) \text{ and } \hat{x} = a_2 (W_d h_e) \quad (12)$$

where  $x$  is stated as the input feature vector and  $\hat{x}$  is the reconstructed output,  $W_e$  and  $W_d$  are the matrices representing the linear combination of inputs for the encoding and decoding sections respectively. The  $h_e$  is the output of the bottleneck layer and this can be considered as the low dimensional representation of the input feature vector.

The autoencoders are typically used in an unsupervised setting for dimensionality reduction and also used for the purposes such as pre-training layers of a neural network. A Stacked autoencoder refers to an architecture where multiple autoencoders are trained and the feature vectors thus obtained are used as inputs to the next layer of the neural network. For instance, the first autoencoder is trained by input data and the learned feature vector obtained is used as the input for the first hidden layer, and so on. After all the hidden layers are trained this way, the backpropagation algorithm is used to minimize the cost function and update the weights with labeled training set to achieve fine-tuning. The parameter setting of DNN is given as follows; a number of classes are 11 and 6 for the two datasets, sparsity regularization is four, sparsity proportion is 0.15, L2 weight regularization is 0.004, number of input and output layers is one each and number of iterations is 500.

## EXPERIMENTAL RESULTS

In this research, MIO-TCD and toll booth datasets are used in the experiments. The performance evaluation is done using an 80/20 train/test split. The MIO-TCD dataset consists of 648,959 lower resolution images in that 519,164 images (80% of data) are utilized for training, and 129,795 images (20% of data) are used for testing. In toll booth dataset, the total number of images are 15000 in that 12000 images are used for training and 3000 images are used for testing.

On MIO-TCD database, without feature selection, the hybrid feature extraction achieved 96.26% of accuracy, whereas the individual features, namely HOG, Gabor and LOOP achieved accuracies of 91%, 87.89% and 90.44%, respectively. On toll booth dataset, the hybrid feature extraction without feature selection achieved 84.48% of accuracy, where the individual features; HOG, Gabor and LOOP attained 81.4%, 76.96% and 82% of classification accuracy. These results suggested that the HoG and LOOP features improves classification accuracy, but the combined set of features is better than any individual features. For all remaining experiments, the Gabor, HoG and LOOP features are combined and then feature selection is carried out by multi-objective Ant Colony Optimization (ACO) approach. The top 30% of features are selected based on ranking provided by the optimizer instead of the full set of combined features. Additionally, the performance of DNN is compared with the performance of other classifiers using the same set of hybrid features and these results are discussed in section 4.1. The DNN classifier is used with different optimizers for feature selection (same set of initial hybrid features) and these results are described in section 4.2. Further, the results of the proposed model on MIO-TCD dataset is compared with the existing methods like ensemble deep learning approach

(Liu et al., 2017), AlexNet (Luo et al., 2018), Inception V3 (Luo et al., 2018), ResNet 50 (Luo et al., 2018), VGG 19 (Luo et al., 2018), Xception (Luo et al., 2018), DenseNet (Luo et al., 2018) and GAN based deep ensemble model (Liu et al., 2018), which is described in section 4.3.

In this research work, MATLAB 2019a environment is used for all experiments, which are run on a machine with 128 GB RAM, 3 TB hard disk, Windows 10-64 bit operating system and Intel i9 processor. The MIO TCD classification and toll booth datasets are used for validating the proposed model performance in terms of sensitivity, specificity, accuracy, error rate, FDR and FOR. The mathematical expressions for sensitivity, specificity, accuracy, error rate, FDR and FOR are denoted in the equations (13) - (18).

$$Sensitivity = \frac{TP}{FN + TP} \times 100 \quad (13)$$

$$Specificity = \frac{TN}{FP + TN} \times 100 \quad (14)$$

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} \times 100 \quad (15)$$

$$Errorrate = 100 - accuracy \quad (16)$$

$$FDR = \frac{FP}{FP + TP} \times 100 \quad (17)$$

$$FOR = \frac{FN}{TN + FN} \times 100 \quad (18)$$

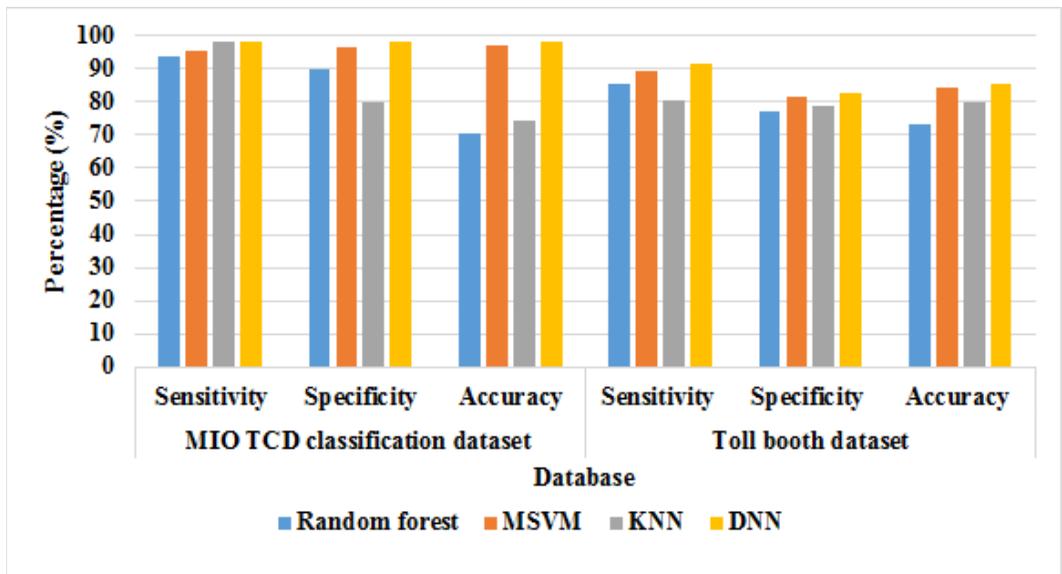
Here, true negative is indicated as  $TN$ , false negative is represented as  $FN$ , true positive is stated as  $TP$  and false-positive is indicated as  $FP$ .

### Quantitative analysis with multiple classifiers

The performance of the proposed model is analyzed on both MIO TCD classification database and toll plaza dataset. The proposed model performance is compared to multiple classification techniques; random forest, Multi Support Vector Machine (MSVM), K-nearest neighbor (KNN) and DNN. The performance metrics; sensitivity, specificity, accuracy, FDR, and FOR of the proposed model on both MIO-TCD and toll booth datasets are reported in Table 2. The performance of multiple classifiers concerning sensitivity, specificity, accuracy, FDR, and FOR are graphically represented in the Figures 6 and 7.

Table 2 shows that the DNN classifier attained 97.88% of classification accuracy on MIO TCD dataset which is higher compared to other classification techniques. Correspondingly, DNN classifier obtained better performance in vehicle type classification in terms of sensitivity and specificity related to other classification techniques. On toll booth dataset, DNN classifier achieved accuracy of 85.45%, sensitivity of 91.42% and specificity of 82.56% on toll booth dataset, which are higher compared to other classification techniques like random forest, MSVM, and KNN. Table 2 shows that the DNN classifier achieved 1.12% of FDR and 0.52% of FOR on MIO TCD database, which is significantly lower compared to other classification techniques such as random forest, MSVM, and KNN. In toll booth dataset, DNN classifier attained 12.65% FDR, and 0.87% FOR. The performance of the DNN; stacked autoencoder neural network classifier is better than that of other classification techniques. The proposed model consumed 0.88 seconds to detect vehicles from each frame on MIO TCD dataset. In addition, the proposed model consumed 0.94 seconds to detect vehicles from each frame on toll booth dataset, and the memory footprint of dissimilar classifiers is denoted in figure 8. The confusion matrix of MIO-TCD and toll booth datasets are graphically represented in the figures 11 and 12.

Figure 6. Comparison of multiple classifiers



### Quantitative Analysis of the Proposed Model with Multiple Optimizers

In this section, the performance of the proposed hybrid features + multi-objective ACO is compared to the multiple existing optimization techniques such as firefly optimizer (Chou & Ngo, 2017), Particle Swarm Optimizer (PSO), and Grey Wolf Optimizer (GWO). In Table 3, the proposed model performance is analyzed using sensitivity, specificity, and accuracy on both MIO-TCD and toll booth datasets. As shown in Table 3, the multi-objective ACO algorithm with DNN classifier attained a classification accuracy of 97.88%; 98.2% sensitivity and 97.9% specificity, which are higher compared to other optimization algorithms on MIO-TCD database. On toll booth dataset, the multi-objective ACO algorithm with the DNN classification technique achieved classification accuracy of 85.45%, 91.42% sensitivity, and 82.56% specificity. The performance of multiple optimizers in terms of sensitivity, specificity, and accuracy is graphically represented in Figure 9. In the confusion matrix shown in Figure 12, the percentage of misclassification of buses as cars is more compared to the other

Figure 7. Graphical analysis of multiple classifiers by means of FDR and FOR

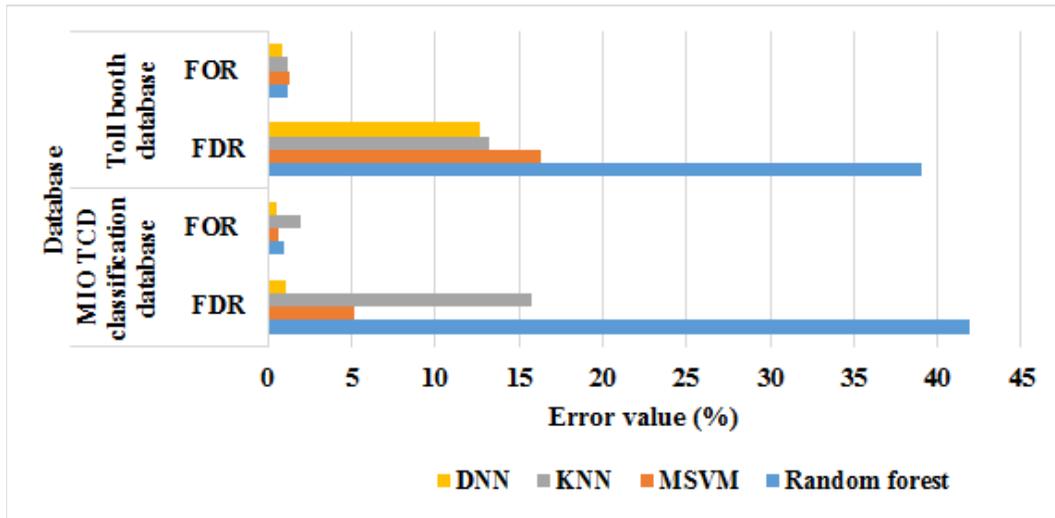


Table 2. Performance analysis of multiple classifiers in terms of sensitivity, specificity, accuracy, FDR, FOR

Database	Performance measures	Random forest	MSVM	KNN	DNN
MIO TCD classification dataset	Sensitivity (%)	93.50	95.20	98.09	98.21
	Specificity (%)	90.00	96.36	80.02	97.92
	Accuracy (%)	70.45	96.92	74.09	97.88
	FDR (%)	41.94	5.23	15.79	1.12
	FOR (%)	1.01	0.64	1.99	0.52
Toll booth dataset	Sensitivity (%)	85.54	89.45	80.38	91.42
	Specificity (%)	77.19	81.45	78.67	82.56
	Accuracy (%)	73.18	84.23	79.65	85.45
	FDR (%)	39.03	16.28	13.28	12.65
	FOR (%)	1.19	1.26	1.22	0.87

misclassifications, and one such image is shown in Figure 13. The proposed model showed 2%-25% improvement in vehicle type classification compared to other optimizers. The parameter settings of firefly optimizer, PSO, and GWO algorithms are represented as follows. The population size and the number of iteration fixed in GWO and firefly optimization algorithm are 30, and 100 along with  $\alpha=0$ ,  $\beta=1$ , and  $\gamma=0.2$ . Further, the PSO algorithm has population size of 20, number of iteration is 100, and the initial weight size is 0.2.

In Table 4, the proposed model performance is analyzed using error rate, FDR and FOR on MIO TCD and toll booth dataset. On MIO TCD database, the multi-objective ACO algorithm with DNN classification technique attained 2.12% of error rate, 1.12% of FDR and 0.5% FOR, which has a lower error value compared to other optimization approaches. Meanwhile, multi-objective ACO with DNN classifier showed better performance on toll booth dataset by means of error rate, FDR and

Figure 8. Memory footprint of dissimilar classifiers

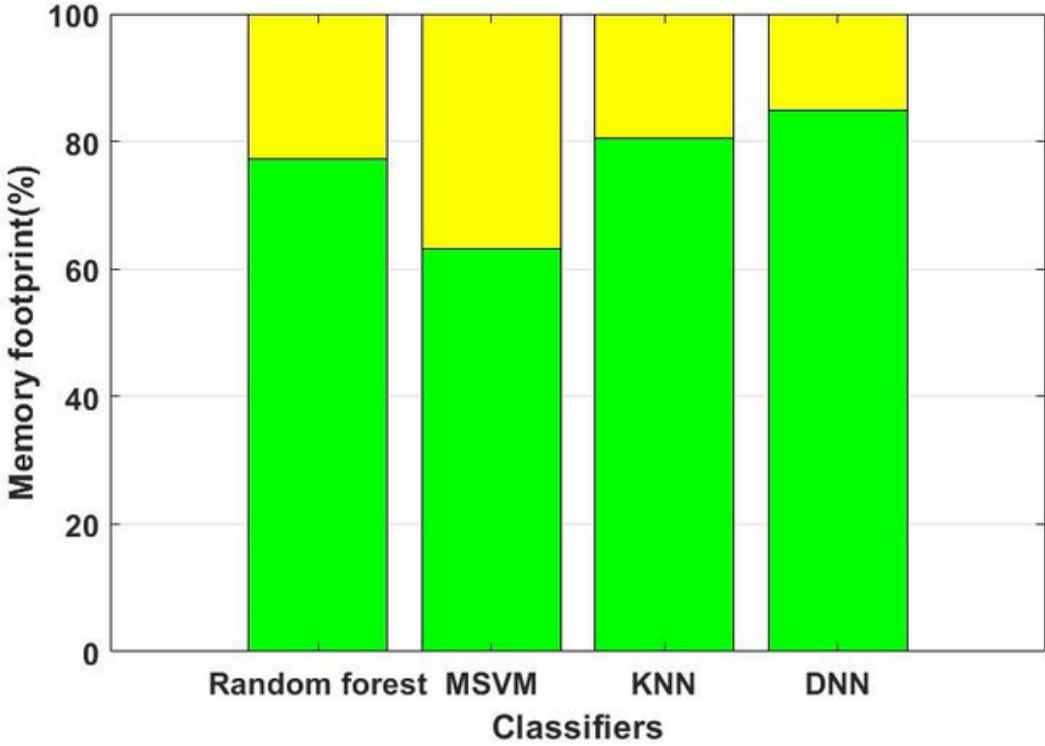


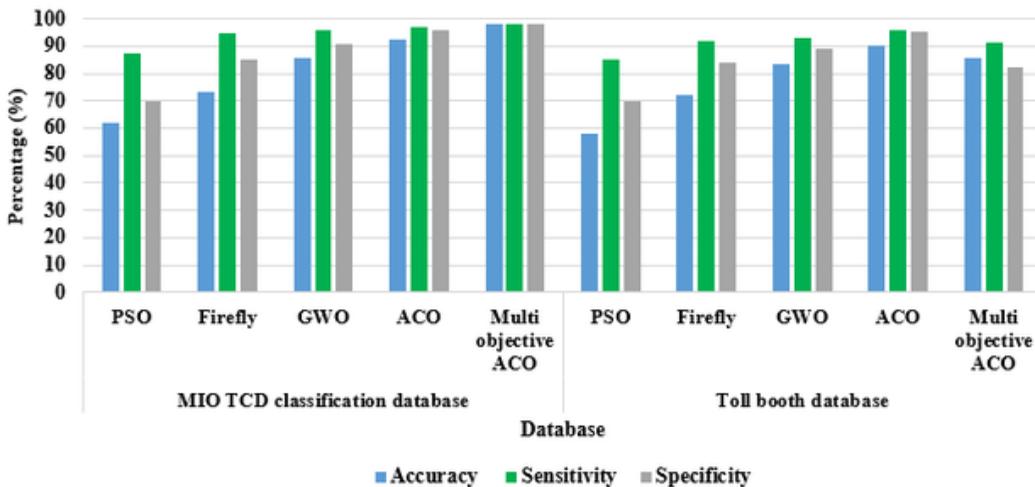
Table 3. Performance analysis of multiple optimizers with respect to accuracy, sensitivity, specificity

Feature Optimization Algorithms	Datasets	Performance measures (average value)		
		Accuracy (%)	Sensitivity (%)	Specificity (%)
PSO	MIO TCD classification database	62.27	87.50	70.00
Firefly		73.18	94.50	85.34
GWO		85.45	96.00	90.53
ACO		92.27	97.23	95.78
Multi objective ACO		97.88	98.21	97.92
PSO	Toll booth dataset	58.12	85.19	69.88
Firefly		72.15	91.76	84.18
GWO		83.26	93.12	89.22
ACO		90.45	96.13	95.28
Multi-objective ACO		85.45	91.42	82.56

Table 4. Performance analysis of multiple optimizers with respect to error rate, FDR and FOR

Feature optimization algorithms	Datasets	Performance measures (average value)		
		Error rate (%)	FDR (%)	FOR (%)
PSO	MIO TCD classification database	37.73	64.10	3.31
Firefly		26.82	39.29	1.56
GWO		14.55	30.77	1.03
ACO		7.73	9.96	0.82
Multi objective ACO		2.12	1.12	0.52
PSO	Toll booth dataset	41.88	60.12	3.12
Firefly		27.85	35.70	1.43
GWO		16.74	28.47	1.26
ACO		19.55	19.78	1.66
Multi objective ACO		14.55	18.65	1.07

Figure 9. Performance of multiple optimizers by means of sensitivity, specificity, and accuracy



FOR. The performance of multiple optimizers concerning error rate, FDR, and FOR is graphically indicated in Figure 10.

### Comparative analysis of proposed approach and standard neural network architectures

Table 5 indicates the comparative analysis of the proposed multi-objective ACO with the DNN approach and existing neural network architectures. Liu et al. (2017) developed DNN with a balanced sampling model to classify different vehicle categories. In this literature, the performance of the developed model was investigated on the MIO TCD database. The developed model attained 88.4% recall and 97.76% precision on average. Luo et al. (2018) focused on motor vehicle localization and classification by introducing a new dataset; MIO TCD. In this literature, the introduced database was validated on six standard systems; AlexNet, Inception V3, ResNet 50, VGG 19, Xception, and

Figure 10. Graphical analysis of optimizers with respect to error rate, FDR and FOR

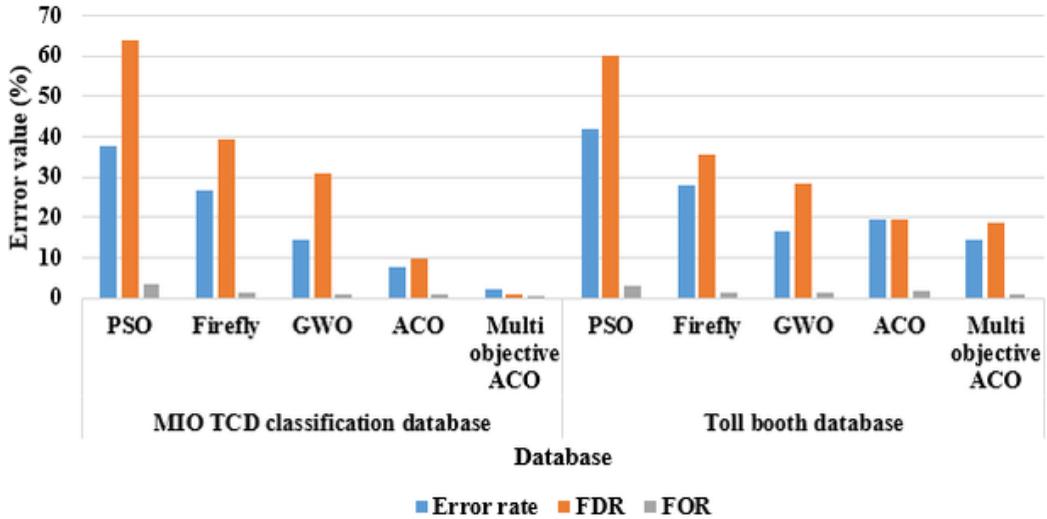
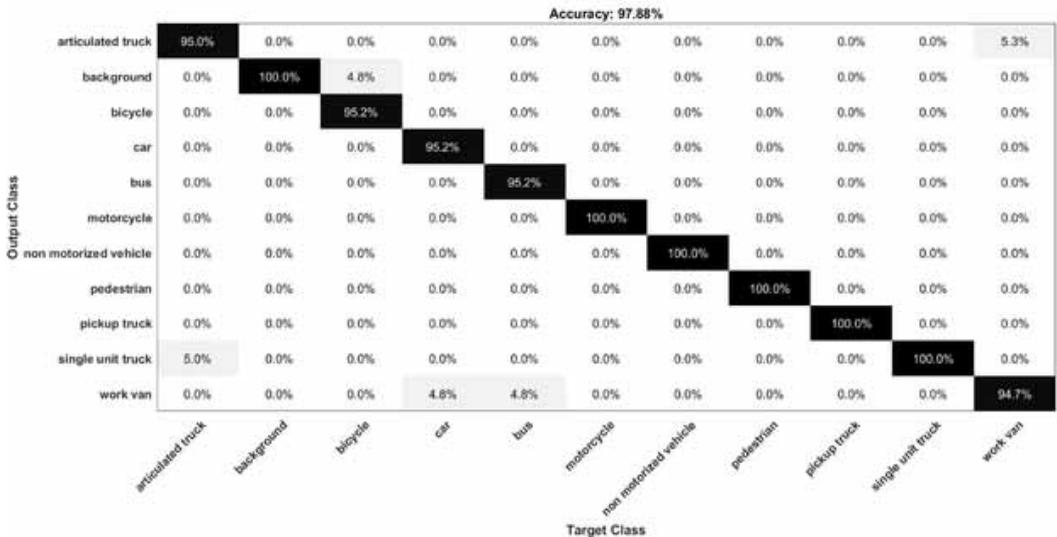


Figure 11. Confusion matrix of MIO TCD



DenseNet in light of mean precision and accuracy. Liu et al. (2018) developed a semi-supervised pipeline model for motor vehicle type classification, which was the combination of DNN and data augmentation based on GANs. In the experimental phase, the MIO TCD database was utilized for validating the efficiency of the developed model in light of mean precision and mean recall, where the developed model achieved 93.55% of mean precision and 90.74% of mean recall. The proposed model achieved good performance in vehicle type classification compared to the above-stated existing models. The proposed model; multi-objective ACO with DNN significantly reduces the dimension of extracted feature vectors and helps in attaining better classification performance.

Figure 12. Confusion matrix of toll booth dataset

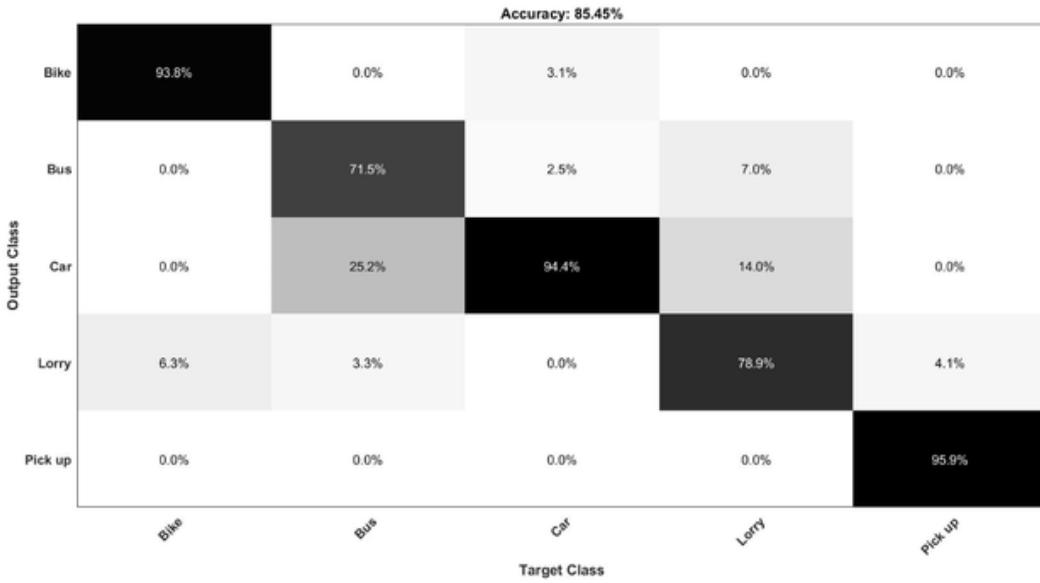


Figure 13. a) Input image and b) Image showing Misclassification of the bus as the car



**Table 5. Comparative analysis of proposed model and results reported in the literature on the MIO TCD dataset**

Methodology	Mean precision (%)	Mean recall (%)	Accuracy (%)
Ensemble deep learning approach (Liu et al., 2017)	97.76	88.4	-
AlexNet (Luo et al., 2018)	89.9	82.6	89.2
Inception V3 (Luo et al., 2018)	88.9	88.8	96.2
ResNet 50 (Luo et al., 2018)	90.3	89.2	96.9
VGG 19 (Luo et al., 2018)	85.3	82.4	95.6
Xception (Luo et al., 2018)	90	90.8	97.6
DenseNet (Luo et al., 2018)	91.6	89.6	97
GAN based deep ensemble model (Liu et al., 2018)	93.55	90.74	-
Proposed model	98.09	98.21	97.88

## CONCLUSION

In this research, a multi-objective ACO with DNN approach is proposed to detect and classify vehicle types from images. The approach employs pre-processing using a camera reference model and Gaussian Mixture Model to localize the vehicle. The proposed approach utilizes a hybrid feature set consisting of HOG, Gabor features and LOOP features, which are extracted from the pre-processed images. Then, a feature selection is performed using an Ant Colony Optimization (ACO) approach and the selected features serve as inputs to Deep Neural Network. In the experiments, the proposed approach showed an improvement in classification accuracy ranging from 0.28% to 8.68% as compared to well-known existing neural network architectures ensemble deep learning approach, AlexNet, Inception V3, ResNet 50, VGG 19, Xception, DenseNet, and GAN based deep ensemble model on a publicly available benchmark dataset. Although, deep neural networks consider the raw image data as input and act as feature extractors and classifiers. In this proposed method, the hybrid set of features helps in improving the performance of the feature extraction process and gives better performance than existing models using individual features. The proposed approach on the MIO TCD dataset is validated on toll booth dataset that contained images extracted from videos of a toll plaza located in the state of Karnataka, India. In future work, the feature extraction and feature selection can be analyzed by comparing deep learning methods to improve the challenges on a real-world dataset.

## FUNDING

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

## REFERENCES

- Akilan, T., Wu, Q. J., Safaei, A., & Jiang, W. (2017). A late fusion approach for harnessing multi-CNN model high-level features. In *2017 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 566-571). IEEE. doi:10.1109/SMC.2017.8122666
- Akilan, T., Wu, Q. J., & Zhang, H. (2018). Effect of fusing features from multiple DCNN architectures in image classification. *IET Image Processing*, *12*(7), 1102–1110. doi:10.1049/iet-ipr.2017.0232
- Aqel, S., Hmimid, A., Sabri, M. A., & Aarab, A. (2017). Road traffic: Vehicle detection and classification. In *2017 Intelligent Systems and Computer Vision (ISCV)* (pp. 1-5). IEEE.
- Biglari, M., Soleimani, A., & Hassanpour, H. (2017). A cascaded part-based system for fine-grained vehicle classification. *IEEE Transactions on Intelligent Transportation Systems*, *19*(1), 273–283. doi:10.1109/TITS.2017.2749961
- Chakraborti, T., McCane, B., Mills, S., & Pal, U. (2018). Loop descriptor: Local optimal-oriented pattern. *IEEE Signal Processing Letters*, *25*(5), 635–639. doi:10.1109/LSP.2018.2817176
- Chen, W., Sun, Q., Wang, J., Dong, J. J., & Xu, C. (2018). A Novel AdaBoost and CNN Based for Vehicle Classification. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 60445–60455. doi:10.1109/ACCESS.2018.2875525
- Chou, J. S., & Ngo, N. T. (2017). Modified firefly algorithm for multidimensional optimization in structural design problems. *Structural and Multidisciplinary Optimization*, *55*(6), 2013–2028. doi:10.1007/s00158-016-1624-x
- Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)* (vol. 1, pp. 886-893). IEEE. doi:10.1109/CVPR.2005.177
- Derrouz, H., Elbouziady, A., Abdelali, H. A., Thami, R. O. H., El Fkihi, S., & Bourzeix, F. (2019). Moroccan Video Intelligent Transport System: Vehicle Type Classification Based on Three-Dimensional and Two-Dimensional Features. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 72528–72537. doi:10.1109/ACCESS.2019.2920740
- Dong, Z., Wu, Y., Pei, M., & Jia, Y. (2015). Vehicle type classification using a semisupervised convolutional neural network. *IEEE Transactions on Intelligent Transportation Systems*, *16*(4), 2247–2256. doi:10.1109/TITS.2015.2402438
- Gabor, D. (1946). Theory of communication. Part 1: The analysis of information. *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, *93*(26), 429–441. doi:10.1049/ji-3-2.1946.0074
- Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507. doi:10.1126/science.1127647 PMID:16873662
- Javadi, S., Rameez, M., Dahl, M., & Pettersson, M. I. (2018). Vehicle classification based on multiple fuzzy c-means clustering using dimensions and speed features. *Procedia Computer Science*, *126*, 1344–1350. doi:10.1016/j.procs.2018.08.085
- Jiang, L., Li, J., Zhuo, L., & Zhu, Z. (2017). Robust vehicle classification based on the combination of deep features and handcrafted features. In *2017 IEEE Trustcom/BigDataSE/ICSS*. IEEE Computer Society.
- Kafai, M., & Bhanu, B. (2011). Dynamic Bayesian networks for vehicle classification in video. *IEEE Transactions on Industrial Informatics*, *8*(1), 100–109. doi:10.1109/TII.2011.2173203
- Ke, X., & Zhang, Y. (2020). Fine-grained Vehicle Type Detection and Recognition Based on Dense Attention Network. *Neurocomputing*, *399*, 247–257. doi:10.1016/j.neucom.2020.02.101
- Khan, S., Hussain, M., Aboalsamh, H., Mathkour, H., Bebis, G., & Zakariah, M. (2016). Optimized Gabor features for mass classification in mammography. *Applied Soft Computing*, *44*, 267–280. doi:10.1016/j.asoc.2016.04.012
- Li, X., Yu, L., Chang, D., Ma, Z., & Cao, J. (2019). Dual cross-entropy loss for small-sample fine-grained vehicle classification. *IEEE Transactions on Vehicular Technology*, *68*(5), 4204–4212. doi:10.1109/TVT.2019.2895651

- Liu, W., Luo, Z., & Li, S. (2018). Improving deep ensemble vehicle classification by using selected adversarial samples. *Knowledge-Based Systems, 160*, 167–175. doi:10.1016/j.knsys.2018.06.035
- Liu, W., Zhang, M., Luo, Z., & Cai, Y. (2017). An ensemble deep learning method for vehicle type classification on visual traffic surveillance sensors. *IEEE Access: Practical Innovations, Open Solutions, 5*, 24417–24425. doi:10.1109/ACCESS.2017.2766203
- Luo, Z., Branchaud-Charron, F., Lemaire, C., Konrad, J., Li, S., Mishra, A., Achkar, A., Eichel, J., & Jodoin, P. M. (2018). MIO-TCD: A new benchmark dataset for vehicle classification and localization. *IEEE Transactions on Image Processing, 27*(10), 5129–5141. doi:10.1109/TIP.2018.2848705 PMID:29994117
- Mithun, N. C., Rashid, N. U., & Rahman, S. M. (2012). Detection and classification of vehicles from video using multiple time-spatial images. *IEEE Transactions on Intelligent Transportation Systems, 13*(3), 1215–1225. doi:10.1109/TITS.2012.2186128
- Murugan, V., & Vijaykumar, V. R. (2018). Automatic moving vehicle detection and classification based on artificial neural fuzzy inference system. *Wireless Personal Communications, 100*(3), 745–766. doi:10.1007/s11277-018-5347-8
- Ojala, T., Pietikäinen, M., & Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition, 29*(1), 51–59. doi:10.1016/0031-3203(95)00067-4
- Parsons, S. (2005). Ant Colony Optimization by Marco Dorigo and Thomas Stützle. *The Knowledge Engineering Review, 20*(1), 92–93. doi:10.1017/S0269888905220386
- Rachmadi, R. F., Uchimura, K., Koutaki, G., & Ogata, K. (2018). Single image vehicle classification using pseudo long short-term memory classifier. *Journal of Visual Communication and Image Representation, 56*, 265–274. doi:10.1016/j.jvcir.2018.09.021
- Sentas, A., Tashiev, I., Kucukayvaz, F., Kul, S., Eken, S., Sayar, A., & Becerikli, Y. (2018). Performance evaluation of support vector machine and convolutional neural network algorithms in real-time vehicle type classification. In *6th International Conference on Emerging Internet, Data & Web Technologies* (pp. 934-943). Springer.
- Sun, X., You, X., & Liu, S. (2010). Multi-objective ant colony optimization algorithm for shortest route problem. In *2010 International Conference on Machine Vision and Human-machine Interface* (pp. 796-798). IEEE. doi:10.1109/MVHI.2010.67
- Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition (CVPR)* (vol. 1, pp. I-I). IEEE. doi:10.1109/CVPR.2001.990517
- Wang, R., Guo, S., Li, Y., & Zhang, Y. (2017). Fisher discriminative dictionary learning for vehicle classification in acoustic sensor networks. *Journal of Signal Processing Systems for Signal, Image, and Video Technology, 86*(1), 99–107. doi:10.1007/s11265-016-1105-x
- Wang, X., Zhang, W., Wu, X., Xiao, L., Qian, Y., & Fang, Z. (2019). Real-time vehicle type classification with deep convolutional neural networks. *Journal of Real-Time Image Processing, 16*(1), 5–14. doi:10.1007/s11554-017-0712-5
- Wang, Y., Ban, X., Wang, H., Wu, D., Wang, H., Yang, S., Liu, S., & Lai, J. (2019). Detection and classification of moving vehicle from video using multiple spatio-temporal features. *IEEE Access: Practical Innovations, Open Solutions, 7*, 80287–80299. doi:10.1109/ACCESS.2019.2923199
- Wen, X., Shao, L., Fang, W., & Xue, Y. (2014). Efficient feature selection and classification for vehicle detection. *IEEE Transactions on Circuits and Systems for Video Technology, 25*(3), 508–517.
- Yan, G., Yu, M., Yu, Y., & Fan, L. (2016). Real-time vehicle detection using histograms of oriented gradients and AdaBoost classification. *Optik (Stuttgart), 127*(19), 7941–7951. doi:10.1016/j.ijleo.2016.05.092
- Yang, B., & Lei, Y. (2014). Vehicle detection and classification for low-speed congested traffic with anisotropic magnetoresistive sensor. *IEEE Sensors Journal, 15*(2), 1132–1138. doi:10.1109/JSEN.2014.2359014
- Ying, Z., Li, G., Ren, Y., Wang, R., & Wang, W. (2017). A new low-light image enhancement algorithm using camera response model. In *Proceedings of the IEEE International Conference on Computer Vision Workshops* (pp. 3015-3022). IEEE. doi:10.1109/ICCVW.2017.356

Yu, S., Wu, Y., Li, W., Song, Z., & Zeng, W. (2017). A model for fine-grained vehicle classification based on deep learning. *Neurocomputing*, 257, 97–103. doi:10.1016/j.neucom.2016.09.116

Zhang, Y., Zhao, C., He, J., & Chen, A. (2016). Vehicles detection in complex urban traffic scenes using Gaussian mixture model with confidence measurement. *IET Intelligent Transport Systems*, 10(6), 445–452. doi:10.1049/iet-its.2015.0141

Zhou, Y., & Cheung, N. M. (2016). Vehicle classification using transferable deep neural network features. arXiv preprint arXiv:1601.01145.

Zhuo, L., Jiang, L., Zhu, Z., Li, J., Zhang, J., & Long, H. (2017). Vehicle classification for large-scale traffic surveillance videos using convolutional neural networks. *Machine Vision and Applications*, 28(7), 793–802. doi:10.1007/s00138-017-0846-2

*Sathyanarayana N. was born in Karnataka, India in 1983. He received the Bachelor degree in Medical Electronics in the year 2007, Master's Degree in Electronics in the year 2012 and now pursuing Ph.D in Computer Vision domain from Visvesvaraya Technological University, Karnataka, India. He is currently working as an Assistant Professor in Electronics and Communication Engineering Department, Vemana Institute of Technology, Bangalore, Visvesvaraya Technological University and has ten years of experience of teaching Under-graduate students. He is actively engaged in research activities and guiding UG students. He has contributed 4 research papers in international conferences/journals. His areas of interest are Computer Vision, Internet of Things, Renewable energy sources and Artificial Intelligence.*

*Anand Narasimhamurthy has an undergraduate degree in Electronics and Communication. He completed his Master's and Ph.D. both in Computer Science and Engineering at Penn State University in 2003 and 2006 respectively. His research interests include machine learning, image analysis, numerical optimization and graph-based methods.*