Original scientific article
http://dx.doi.org/10.59456/afts.2023.1528.039J

STEP TOWARDS INTELLIGENT TRANSPORTATION SYSTEM WITH VEHICLE CLASSIFICATION AND RECOGNITION USING SPEEDED-UP ROBUST FEATURES

Janak Trivedi[1], Mandalapu Sarada Devi[2], Brijesh Solanki[3]

[1]*Faculty of Electronics & Communication Engineering Department, Government Engineering College Bhavnagar, Affiliated to Gujarat Technological University, A.I.C.T.E., India,*
e-mail; trivedi_janak2611@yahoo.com
[2]*Faculty of Electronics & Communication Engineering Department, Ahmedabad Institute of Technology, Affiliated to Gujarat Technological University, A.I.C.T.E., India.*
[3]*Faculty of Electrical Engineering Department, Government Engineering College Bhavnagar, Affiliated to Gujarat Technological University, A.I.C.T.E., India.*

SUMMARY

Vehicle classification is a crucial task owing to vehicles' diverse and intricate features, such as edges, colors, shadows, corners, and textures. The accurate classification of vehicles enables their detection and identification on roads and facilitates the development of an electronic toll-collection system for smart cities. Furthermore, vehicle classification is useful for traffic signal control strategy. However, achieving accurate vehicle classification poses significant challenges due to the limited processing time for real-time applications, image resolution, illumination variations in the video, and other interferences. This study proposes a method for automated automobile detection, recognition, and classification using statistics derived from approximately 11,000 images. We employ SURF-based detection and different classifiers to categorize vehicles into three groups.

The Traffic Management System (TMS) is crucial for studying mobility and smart cities. Our study achieves a high automobile classification rate of 91% with the medium Gaussian Support Vector Machine (SVM) classifier. The paper's main objective is to analyze five object classifiers for vehicle recognition: Decision Tree, Discriminant Analysis, SVM, K-Nearest Neighbor Classifier (KNN), and Ensemble Classifier. In the discussion section, we present the limitations of our work and provide insights into future research directions.

Keywords: *Classifier, feature extraction, Support Vector Machine (SVM), Speeded Up Robust Features (SURF), k-mean.*

INTRODUCTION

With the development of hardware and the reduction in manufacturing costs, there has been a significant increase in surveillance devices, and high-resolution video cameras have been increasingly employed in such systems. Consequently, many video sources generate large volumes of information that require analysis and interpretation, but this amount of information needs to be reduced for human operators to handle. Thus, researchers utilize technology such as the Intelligent Transportation System (ITS). Automatic extraction of vehicles from surveillance videos using artificial intelligence is a prominent research topic. Vehicle classification using traditional methods poses two main problems: the need for human supervision to classify vehicles and the availability of variations in different vehicle models. Computer vision-based traffic management is more intelligent, faster, and more efficient than human operators. Artificial Intelligence (AI) plays a significant role in ITS using computer vision. One of the most exciting aspects of AI is its potential to revolutionize the computer industry and any industry that affects our lives. Deep learning and machine learning are part of AI, and they offer a new direction with high accuracy for an automatic vehicle recognition system in computer-vision techniques. Deep learning works similarly to the human brain in the proposed system, and its layer-by-layer structure extracts useful information for the given dataset.

Recent progress in artificial intelligence (AI) has significantly impacted the research related to vehicle detection, identification, and classification. Vehicle classification has applications in multiple domains, including recognizing emergency vehicles in automated toll-collection systems. Janak et al. have presented a comprehensive review of the utilization of computer vision-based Intelligent Traffic Control systems in Smart Cities [1]. Vehicle classification can be implemented through online, offline, and real-time applications. Automated vehicle classification can be specifically advantageous in toll plazas, where it can rapidly identify the vehicle type and automatically collect the toll charge. Furthermore, vehicle classification and recognition can assist in developing an adaptive traffic control system and identifying emergency vehicles. These advancements in vehicle classification and recognition using AI have the potential to pave the way for the development of more efficient and intelligent transportation systems, thus making our cities smarter and safer.

The deep learning algorithm represents a recent trend in object detection and recognition in artificial intelligence systems utilizing computer vision. In deep learning, tiny feature extraction plays a crucial role in identifying the type or class of a vehicle from selected images. Feature extraction is crucial in acquiring helpful information about the detected features. Feature matching is essential in applications such as motion tracking, image retrieval, object recognition, object classification, and human activity recognition. The limitation of the changing pixel values due to light intensity can be overcome by selecting the most appropriate features. These features include edges, corners, blobs, gradient values, magnitude, and phase.

Various classifiers are used to analyze images selected from various web search tools. Feature extraction, a general term for creating combinations of variables to solve classification problems while adequately representing the information, is a critical step. Many machine learning practitioners believe that properly optimized feature extraction is key to successful model construction. Features are pieces of information obtained from an image. Feature extraction is achieved through various image characteristics, such as color, shape, corners, position, and dominant edges. These features are critical for recognition, matching, and detection. The number and size of features play an important role in different real-time applications. Feature detectors such as SHIFT and SURF are popularly used. In our research, we classified vehicles into bikes, cars, and trucks, using k-means clustering algorithms and SURF on a generated dataset of around 11k images.

Various methods and approaches for vehicle detection, recognition, and classification contribute to versatile traffic light control systems at intersections. Vehicle recognition helps determine the number and type of vehicles on specific roads. This system is also useful in identifying emergency vehicles, such as ambulances and traffic management systems (TMS), which can handle difficult situations in real time. The application of deep learning and artificial intelligence (AI) is essential in developing Intelligent Transportation Systems (ITS) in smart cities. These techniques are valuable and trending in the field of ITS research.

Vehicle classification helps in traffic signal control strategy. For example, high-emissions vehicles stopped less frequently, so recognizing these vehicles and clearing the traffic jams beneficial in terms of air quality. Vehicle classification helps optimize infrastructures and increase the return on toll gates by traffic flows. Vehicle classification can classify *(A) According to purpose*: (1) Passenger vehicles: like car, bus, taxi, etc. (2) Goods vehicles: like truck, tempo, container, etc. (3) Special purpose vehicles: like ambulance, fire brigade, etc. *(B) According to load carrying capacity:* (1)

Light Motor Vehicles (LMV): like cars, jeep, etc. (2) Medium Motor Vehicles (MMV): like tempos, pick up vans, etc. (3) Heavy-duty Motor Vehicles (HMV): tractor, truck, container, etc. *(C) According to number of wheels:* (1) Two wheelers: like motor-cycle, scooters, etc. (2) Three wheelers: like auto rickshaw (3) Four wheelers: like cars, jeep, mini vans, etc. (4) Six wheelers: like truck, bus, etc. *(D) According to fuel used:* (1) Petrol vehicles: like motor-cycle, cars, etc. (2) Diesel vehicles: like cars, trucks, buses, etc. (3) Gasoline vehicles: like LPG vehicles, CNG vehicles, etc. (4) Electric vehicles: like Yo bikes, cars, etc. (5) Hybrid vehicles: like vans, cars, etc. *(E) According to drive of vehicles:* (1) Front wheel drive vehicles: like cars (2) Rear wheel drive vehicles: like truck, buses, etc. (3) Four-wheel drive vehicles: like military vehicles. This article has classified bike, car, and truck-vehicle accordingly to their purpose category using computer vision technique.

This study investigates using an alternative classifier for a dataset of approximately 11,000 images randomly collected from various web search tools. A comprehensive literature review of various methods is presented in Section 2, followed by a detailed explanation of different feature detection techniques in Section 3, including Scale Invariant Feature Transform (SIFT) and Speeded-Up Robust Feature (SURF) detection. Section 4 provides the details of five primary classifiers and 22 sub-classifiers. The vehicle recognition process's workflow is presented in Section 5, utilizing SURF and k-means algorithms, as depicted in Figure (1). Section 6 presents the results regarding feature selection, performance analysis, and verification and testing on random images, as shown in Figures (4-6). The study concludes with a discussion of limitations, a summary of the findings, and future research directions.

LITERATURE SURVEY

Recently, computer vision techniques using deep learning and artificial intelligence have been extensively employed for developing Intelligent Transportation Systems (ITS). These systems require automatic detection, recognition, and classification of vehicles to enhance the Traffic Management System (TMS) efficiency in a smart city. Francesco et al. [2] have presented the European Process of smart city development using three interconnected processes, which include smart city planning theories, smart city development, and smart city rules and policy. For effective smart city planning, smart TMS is necessary.

The TMS comprises real-time vehicle information, accident detection, vehicle speed measurements, parking information, bus-train route information for users and commuters, and an automatic toll collection system. Antonello et al. [3] have discussed the integration of conventional vehicle reviews (which provide little information) with extensive knowledge from Information and Communication Technology (ICT) in constructing Transport System Models (TSMs).

The effective implementation of Intelligent Transportation Systems (ITS) in a smart city requires automated vehicle detection, recognition, and classification. The classification of vehicles is crucial for a wide range of applications, including vehicle speed detection, autonomous driving, intelligent parking systems, and toll collection. In the literature, various classification approaches have been proposed, such as the nearest neighbor classification approach for multispectral images as demonstrated in Hardin et al. [4], and the use of KNN classification by Ghosh Anil [5]. The creation of feature-based vector trees and the AdaBoost method-based classification with random forest and linear estimation of regression trees are discussed in Breiman [6] and Loh [7]. The decision tree-based application of classification from tree graphs is also explained by Paensuwan et al. [8]. These approaches have been shown to classify different types of vehicles effectively and can be applied to real-time ITS systems in smart cities.

Various vehicle classification methodologies have been proposed in the literature, including length-based or shape-based, information-based, vision-based, movement-based, and distinctive feature extraction-based approaches, as explained by Wen et al. [9]. Kim et al. [10] demonstrated on-street vehicle detection using the Pi-Histogram of Oriented Gradients (HOG) strategy with support vector machines (SVMs). Stocker et al. [11] utilized a supervised learning approach with multilayer perceptron (MLP) feed-forward artificial neural networks (ANNs) for vehicle classification and identification. Arinaldi et al. [12] proposed SVM and Faster Region Convolutional Neural Networks (RCNN) based on vehicle identification and classification using MIT traffic data. Atiya [13] discussed simple pattern recognition-based algorithms, support vector classifiers, and support vector regression with different kernel principles.

In Intelligent Transportation Systems (ITS), automatic vehicle detection, recognition, and classification are essential for developing smart cities. The classification of vehicles is necessary for various purposes, such as vehicle speed detection, autonomous driving vehicles, intelligent parking systems, and toll collection.

Sarikan et al. [14] implemented K-Nearest Neighbors (KNN) and decision tree-based vehicle classification. Gorges et al. [15] clarified the utilization of street profile estimation calculation for grouping two-wheeled vehicles. Machine learning classifiers such as KNN, decision trees, and support vector machines (SVMs) have been employed to develop automatic vehicle classification systems. However, the accuracy rate of SVM and Artificial Neural Network (ANN)-based classification approaches is low, as discussed by Bautista et al. [16]. Convolutional Neural Network (CNN) has shown a higher accuracy rate for vehicle recognition than SVM and ANN.

In recent years, Bag of Features (BOF) has been used to classify Magnetic resonance images. Deepika et al. [17] utilized SVM for BOF-based classification and achieved 93% accuracy. Similarly, Pranata et al. [18] used BOF-based classification with SURF for detecting fractures in medical images. Sykora et al. [19] utilized SIFT and SURF for gesture recognition and SVM for classification. Anca and Ioan [20] discussed palm print characterization and recognition using SURF. Anzid et al. [21] employed SURF and different types of SVM (linear, non-linear, and multiclass) for multimodal image classification. These works demonstrate the potential of machine learning methods and feature extraction techniques for image classification and recognition.

FEATURE DETECTION

Feature detection is a crucial step in many computer vision applications, which involves extracting relevant features from an image for subsequent processing. One common method for feature detection is corner detection, which identifies points of interest in an image by detecting changes in the X and Y gradient values. Corners are useful features because they are invariant to changes in illumination or viewpoint. The Harris corner detector is a widely used corner detection algorithm that identifies corners based on changes in gradient in all directions. It operates in three cases: flat regions where there is no change in gradient, edges where there is a change of gradient value in one direction and corners where there is a change in gradient value in all directions. SIFT and SURF are popular feature detection methods that utilize the Hessian matrix to calculate and identify features.

SIFT Feature calculation

The SIFT (Scale-Invariant Feature Transform) feature calculation process involves selecting a 16 x 16-pixel neighborhood and dividing it into 4 x 4 block neighborhoods. Each neighborhood is rotated and aligned with the previously calculated orientation, and eight orientations are calculated at a 4 x 4 bin array which results in 128-dimensional features which are used for object detection and recognition applications. The unit magnitude can be used to normalize the illumination problem. The SIFT features have several properties, including finding feature points, repeatability of key points, scale-rotation invariance, and robustness to viewpoint changes, as explained by David in [22].

SURF feature calculation

In feature extraction, three important properties are repeatability, matching speed, and descriptiveness. The number and size of features are also critical factors. Feature extraction consists of two steps: feature detection and feature extraction. The SIFT method has been widely used due to its high descriptiveness but comes at a high computational cost. In contrast, the SURF method improves upon the scale-invariant feature detector by using the Hessian matrix to detect the gradient magnitude and orientation to select points of interest. The Hessian matrix is calculated in Equation (1), and the sum of the Haar wavelet can also be used to improve orientation performance. Bay et al. [23] discussed the use of SURF and its advantages over the Difference of Gaussian (DoG) method.

(1)

For multivariate function, the Hessian matrix contains all the second partial derivate of function D. Dxx represents twice the partial derivative for x, Dyy represents twice the partial derivative for y, Dxy represents the partial derivative first with x, then for y Dyx represents the partial derivative first for y than with x.

The machine learning approach enables the automatic classification of objects without requiring manual labeling or feature extraction. Matlab is a powerful tool for classifying different objects in a given dataset. The accuracy, prediction, and output results are then validated. In machine learning, data can be divided into numerical and categorical classes. Numerical data are represented by numbers (floating point or integer), while discrete label groups represent categorical data. Using Matlab, five classifiers are tested, further subdivided into twenty-two classifiers.

DIFFERENT CLASSIFIERS

Decision Tree

A *decision tree* is a hierarchical structure that resembles a flowchart used for classification and prediction. Each node in the tree represents a test for a specific attribute, and each branch represents the result of that test. The terminal nodes of the tree, also known as leaf nodes, contain the class labels. Decision trees are one of the most widely used and powerful tools for classification and prediction. They can handle both categorical and numerical data, and different decision trees can be used based on the number of splits made during the classification process.

(1) Fine Trees: Many leaves for acceptable discrimination.
(2) Medium Trees: Moderate number of leaves for finer discrimination between classes.
(3) Coarse Trees: some leaves to roughly distinguish between classes [24].

Discriminant analysis

Discriminant analysis is a statistical method used to classify objects into two or more groups based on their measured characteristics or features. When the predicted variable is binary, discriminant analysis can be viewed as a regression analysis. Suppose we have a set of objects that belong to two categories, and we measure several quantities that help determine the category of the object, which we assign the values 0 and 1. The discriminant analysis classifier is used to classify numerical data only. Different types of discriminant analysis can be used to create linear or nonlinear boundaries between classes.

(1) Linear Discriminant: This creates linear boundaries between classes.
(2) Quadratic Discriminant: This creates nonlinear class boundaries [24].

SVM

The SVM algorithm seeks to find the one that maximizes the margin between the two classes, i.e., the distance between the hyperplane and the nearest data points from each class. The data points closest to the hyperplane are called support vectors, which play a critical role in determining the hyperplane's location. The SVM algorithm can also handle nonlinearly separable data by transforming the data into a higher-dimensional space using kernel functions. SVMs are popular machine learning classifiers due to their ability to handle high-dimensional data and generalize well to new, unseen data.

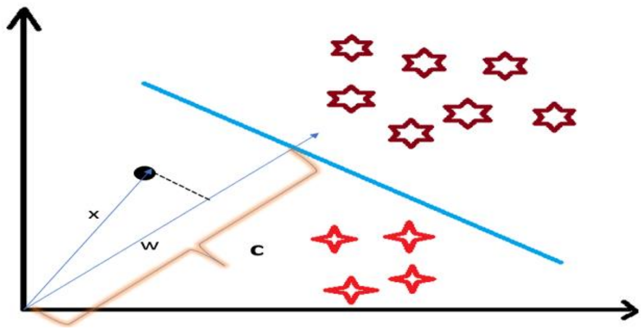

Figure 1 projection of vector x, vector w perpendicular to the hyperplane, and decision boundary c [25]

$$\begin{aligned} \vec{\mathbf{x}} \bullet \vec{\mathbf{w}} &= c \\ \vec{\mathbf{x}} \bullet \vec{\mathbf{w}} &> c \\ \vec{\mathbf{x}} \bullet \vec{\mathbf{w}} &< c \end{aligned} \tag{2}$$

The inner product, which is the projection of one vector onto another, is utilized in the support vector machine (SVM) algorithm. The dot product of the x vector and the w vector is taken, and if the resulting value is greater than a threshold 'c', the point is classified as belonging to the positive samples category. Conversely, if the dot product is less than 'c', the point is classified as belonging to the negative samples category. If the dot product is equal to 'c', the point is on the decision boundary between the two categories. The SVM classifier is commonly used for both categorical and numerical classification tasks.

Different types of SVM depend on their kernel selections. A *kernel function* is a method used to take data as input and transform it into the required format for processing data.
(1) Linear SVM: This makes a linear separation between two classes. Data is classified with the help of hyperplane and straight-line.
(2) Quadratic SVM: This makes a non-linear separation between classes. It cannot be easily separated from a linear line. The quadratic kernel is used to perform this task.
(3) Cubic SVM: This makes a non-linear separation between classes. Use cubic kernels to make non-separable data into separable data.
(4) Fine Gaussian SVM: Gaussian kernels can separate nonlinearly separable data by mapping the input vectors into Hilbert space. Makes finely detailed distinctions between classes, with kernel scale set to sqrt(P)/4.
(5) Medium Gaussian SVM: Medium distinctions, with kernel scale set to sqrt(P).
(6) Coarse Gaussian SVM: Makes coarse distinctions between classes, with kernel scale set to sqrt(P)*4, where P is the number of predictors [24].

Nearest Neighbor Classifier (KNN)

The k-nearest neighbor (KNN) algorithm, also known as ANN or k-NN, is a nonparametric supervised learning classifier that leverages the concept of proximity to make predictions or classifications about the clustering of single data points. KNN classifiers can be used to classify both categorical and numerical data types and typically utilize either Hamming distance calculation for categorical data or Euclidean distance calculation for numerical data. The specific type of KNN algorithm employed depends on factors such as the distinctions between classes and the number of nearest neighbors set.

(1) Fine KNN: Finely detailed distinctions between classes. The number of neighbours is set to 1.
(2) Medium KNN: Medium distinctions between classes. The number of neighbours is set to 10.
(3) Coarse KNN: Coarse distinctions between classes. The number of neighbours is set to 100.
(4) Cosine KNN: Medium distinctions between classes, using a Cosine distance metric. The number of neighbours is set to 10.

(5) Cubic KNN: Medium distinctions between classes, using a cubic distance metric. The number of neighbours is set to 10.
(6) Weighted KNN: Medium distinctions between classes, using a distance weight. The number of neighbours is set to 10 [24].

Ensemble Classifier

Ensemble learning is a technique used to generate a diverse set of base classifiers, from which new classifiers are derived that outperform any individual classifier. The base classifiers may differ in algorithms, hyperparameters, representations, or training sets. Ensemble methods aim to reduce bias and variance in the classification process. The Ensemble Classifier is mainly used to classify categorical data, except for Subspace Discriminant, which is used for numerical data. Depending on their specific characteristics, there are different types of ensemble methods in KNN.

(1) Boosted Trees: The ensemble method is AdaBoost, with Decision Tree learners.
(2) Bagged Trees: The ensemble method is a Random Forest Bag with Decision Tree learners.
(3) Subspace Discriminant: The ensemble method is Subspace, with Discriminant learners.
(4) Subspace KNN: The **e**nsemble method is Subspace, with Nearest Neighbor learners. This kind of operation is suitable for many predictors.
(5) RUSBoosted Trees: The ensemble method is RUSBoost, with Decision Tree learners. This is good for skewed data (with many more observations of 1 class) [24].

WORKFLOW

5.1.1 Detailed discussion of the dataset creation

The first step is to collect images of bikes, cars, and trucks from various sources, resulting in a dataset of approximately 11,000 images, with 3.5 thousand bike images, 5.8 thousand car images, and 1.8 thousand truck images. These images were resized to a fixed resolution of 640 (width) x 480 (height) to ensure consistency, although this may impact the accuracy of the results. The dataset was split into training and testing sets, with 70% of images used for training and 30% used for testing from each vehicle category.

Operation on input images

We use the bag of visual words approach with k-means clustering for feature extraction, which involves iteratively clustering the feature descriptors extracted from representative images of each vehicle category into k mutually exclusive clusters. The resulting clusters are compact and separated by similar properties, and the center of each cluster represents a visual word or function. Feature descriptors can be extracted based on a feature detector or defined raster.The Speeded Up Robust Features (SURF) detector is widely used to extract features invariant to scale and rotation. These features are extracted from the representative images of each category using keypoint extraction, creating feature vectors, and clustering of features through the bag of visual words approach using k-means clustering. Based on these feature descriptors, a histogram of the frequency of visual words is then created for each image. These histograms are then used to train an image category classifier.

After the classifiers are trained, they must evaluate their performance using appropriate metrics. The confusion matrix is a commonly used tool to evaluate the performance of classification models. The confusion matrix summarizes the number of true positive (TP), false positive (FP), true negative (TN), and false negative (FN) predictions made by the model. From this, we can calculate various parameters such as True Positive Rates (TPR), False Negative Rates (FNR), Positive Predictive Values (PPV), and False Discovery Rates (FDR). These metrics can help assess the classifier's accuracy, precision, and recall. The workflow for evaluating the classifiers is typically shown in a figure, such as a Figure (2), which provides an overview of the entire process from training to evaluation.

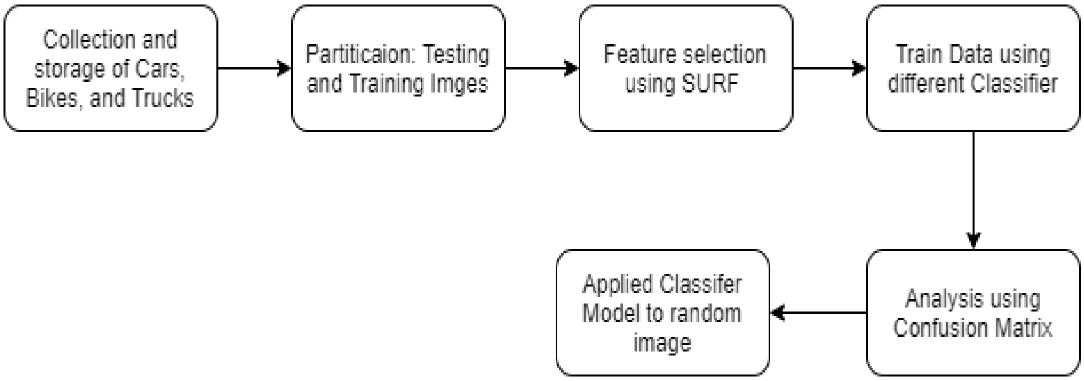

Figure 2 Workflow

RESULTS

In this study, the training dataset was used to validate various classifiers for categorizing vehicles into one of three categories: bike, car, and truck. A dataset of size 4800 x 451 was used for this purpose. A total of 22 different classifiers were evaluated using metrics such as True Positive Rates (TPR), False Negative Rates (FNR), Positive Predictive Values (PPV), and False Discovery Rates (FDR) with the help of a confusion matrix. The results for all the decision tree classifiers are shown in Figure 3(a).

To elaborate, Figure 3(a) demonstrates the accuracy of decision tree classifiers, where the fine tree classifier outperforms the medium and coarse trees. Additionally, Figure 3(b) shows support vector machine classifiers' accuracy and prediction speed, where a maximum accuracy of approximately 91% is achieved. The accuracy of a classifier is determined by its ability to accurately classify true positives (Tp), true negatives (Tn), false positives (Fp), and false negatives, which are taken into account during the calculation of classification accuracy.

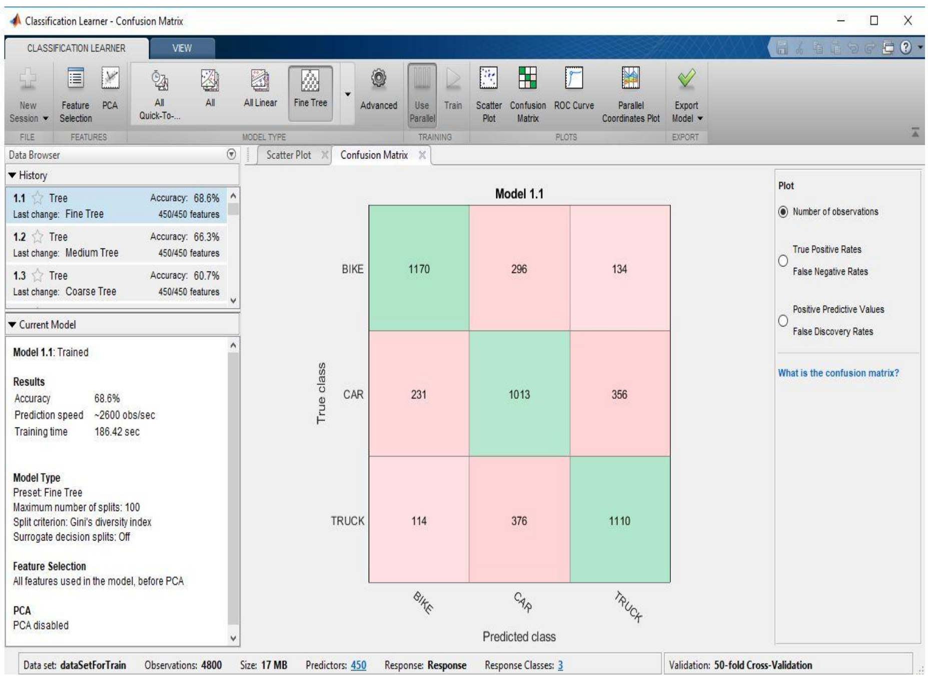

Figure 3 (a) Confusion matrix for decision tree classifiers

The number of accurate guesses divided by the number of predictions is the categorization accuracy as per equation (2) [17].

$$\text{Accuracy} = (Tp + Tn) / ((Tp + Tn + Fp + Fn) \qquad (2)$$

In this study, we present our findings, categorized into three distinct areas: feature selection, performance analysis and comparison, and verification and testing on random images. Feature selection was conducted to identify the most relevant and informative features contributing significantly to the classification task.

Performance analysis and comparison were performed to evaluate the effectiveness of the proposed method against state-of-the-art approaches. Lastly, we conducted verification and testing on a set of random images to validate the generalizability and robustness of the proposed method. Our results provide valuable insights into the efficacy of the proposed approach and demonstrate its potential for real-world applications.

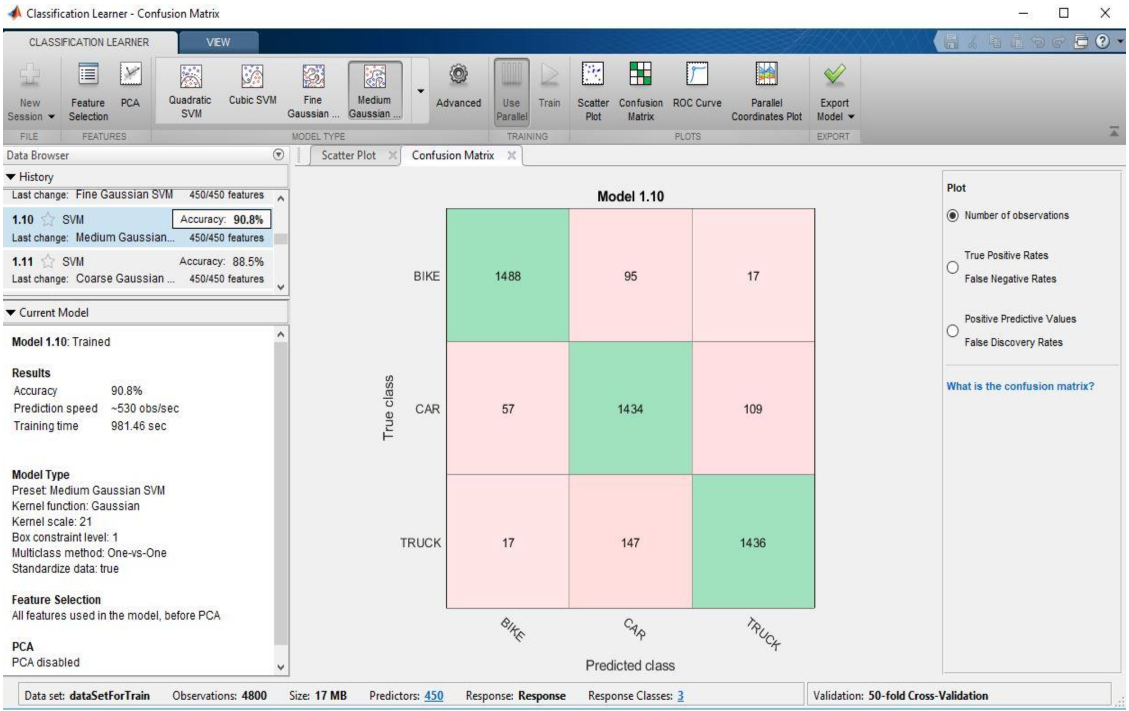

Figure 3 (b) Confusion matrix for SVM classifiers

Feature selection

This study investigates the effect of different feature selection methods on vehicle classification. We selected different images from each class of vehicles and extracted the number of features for each category. Specifically, we tested the k-means clustering method using around 250 words to classify bikes, cars, and trucks. We used the Speeded-Up Robust Features (SURF) algorithm to extract features, which were represented using the bag of visual words approach. Key points and descriptors were extracted from the images to identify features invariant to changes in image size, orientation, and compression. The number of features extracted from each vehicle category is presented in Annexure-I.

We compared the performance of the k-means clustering method using 250, 350, and 450 visual words for the same dataset of bikes, cars, and trucks. The results of our experiments, including the number of iterations and required time for each method, are summarized in Table 1.

Performance analysis and comparison

Multiple classifiers, including Support Vector Machine (SVM), K-Nearest Neighbors (K-NN), Decision Trees, Discriminant Analysis, and Ensemble classifiers using MATLAB, were evaluated for their performance in vehicle classification. We used the features extracted from the images of bikes, cars, and trucks, which were classified into different categories. The accuracy of each classifier was measured using cross-validation on the dataset.

The results of our experiments are summarized in Table 2, which shows that the medium Gaussian SVM classifier achieved the highest accuracy of nearly 91% for vehicle identification. These findings demonstrate the effectiveness of the proposed approach for vehicle classification and suggest that a medium Gaussian SVM classifier may be a suitable choice for similar applications.

Table 1. Number of iterations and the computational time for different images

Sr .No.	Number of Test Images	Bag of Visual words	Number of Iterations	Time Required (S)
1	100	250	25	84.755746
		350	17	76.971196
		450	20	80.409061
2	200	250	26	178.666412
		350	17	157.776405
		450	28	196.360222
3	300	250	22	256.148293
		350	18	237.015765
		450	30	341.758059
4	400	250	18	365.215308
		350	25	359.875082
		450	30	393.088333
5	500	250	28	523.370598
		350	29	497.012188
		450	18	442.437712
6	600	250	27	573.092221
		350	24	621.244559
		450	37	697.177003
7	700	250	75	1164.067380
		350	50	908.966623
		450	35	855.142737
8	800	250	22	859.710340
		350	20	843.597179
		450	19	826.786462
9	900	250	38	1083.519846
		350	22	943.515072
		450	19	1016.507187
10	1000	250	18	1024.145876
		350	27	1075.037355
		450	23	1130.071606

Annexure II presents the results of vehicle classification for bikes (B), cars (C), and trucks (T) in terms of true positive rate (TPR), false positive rate (FPR), positive predictive value (PPV), and false discovery rate (FDR) as percentages. We used different decision tree classifiers to identify the three types of vehicles, and the confusion matrix in the first row was used to assess the classification performance.

The diagonal elements in the confusion matrix indicate the correct identification of the selected vehicles. These results provide valuable insights into the effectiveness of the proposed approach for vehicle classification and can be used to optimize the system's performance in real-world applications

Table 2. Accuracy of different classifiers.

Sr. No.	Classifier Name		accuracy (%)	prediction speed (Obs/Sec)
1	Decision Trees	Fine Trees	68.6	2600
2		Medium Trees	66.3	2500
3		Coarse Tress	60.7	3100
4	Discriminant Analysis	Linear Discriminant	88.1	2400
5		Quadratic Discriminant	80.4	2000
6	Support Vector Machines (SVM)	Linear SVM	89	2100
7		Quadratic SVM	90	790
8		Cubic SVM	90.3	660
9		Fine Gaussian SVM	43.7	250
10		Medium Gaussian SVM	90.8	530
11		Coarse Gaussian SVM	88.5	550
12	K-Nearest Neighbor Classifier (KNN)	Fine KNN	77.7	220
13		Medium KNN	80.9	260
14		Coarse KNN	81.1	260
15		Cosine KNN	81.7	240
16		Cubic KNN	79.8	10
17		Weighted KNN	82.3	270
18	Ensemble Classifier	Boosted Trees	78.1	1300
19		Bagged Trees	83.3	1100
20		Subspace Discriminant	88.4	630
21		Subspace KNN	82.6	20
22		RUSBoosted Trees	65.6	1500

Testing on random images

We assessed the performance of different classifiers for predicting the type of vehicle in a randomly selected image. Specifically, we used the medium Gaussian SVM classifier to test the system on a set of random images, and the results are presented in Figures (4) and Figure (5). These figures demonstrate the effectiveness of the proposed approach for vehicle classification and illustrate the classification results for a range of images. The results suggest that the medium Gaussian SVM classifier may be a suitable choice for similar applications, and provide important insights into the system's performance under realistic conditions.

Figure 4. True detection of bike and car

Figure 5. True & false detection of the truck

DISCUSSION

The images of bikes, cars, and trucks were classified and trained using Speeded Up Robust Features (SURF) and k-means clustering algorithms, with different visual feature word sizes of 250, 350, and 450 in MATLAB (as shown in Annexure-I). The most significant features were selected from each vehicle class, and the total number of features was calculated using MATLAB. While some features remained unchanged regardless of the visual feature word size selection, others varied with the change in visual feature word size. This study compared five object classifiers for the three vehicle categories: bike, car, and truck, using the identified features. The results demonstrate the effectiveness of the proposed approach for vehicle classification and provide valuable insights into the system's performance. These findings can be used to optimize the system for real-world applications, and the approach can be extended to other types of objects beyond vehicles.

After identifying the features of the objects, we applied twenty-two classifiers to classify different types of vehicles. These classifiers comprised sub-classifiers from five leading classification algorithms: Decision Tree (three sub-classifiers), Discriminant Analysis (two sub-classifiers), Support Vector Machines (six sub-classifiers), K-Nearest Neighbors (six sub-classifiers), and Ensemble Classifiers (five sub-classifiers). The vehicle dataset was used for training and classification into three categories: bikes, cars, and trucks. The medium Gaussian SVM classifier method of SVM demonstrated superior accuracy compared to the other twenty-one classifiers. Therefore, we recommend using the medium Gaussian SVM classifier for accurate vehicle recognition.

Variations in image resolution can impact the accuracy of inbuilt architecture and network results. To mitigate this issue, we created a custom dataset and evaluated it using a Convolutional Neural Network (CNN), as described by Trivedi et al. [26]. However, our current machine learning approach provided better accuracy for vehicle classification of bikes, cars, and trucks than the CNN approach, particularly for smaller datasets. Our machine learning approach yielded high accuracy, with the bike classification achieving a 99% prediction accuracy and the highest confidence values. In contrast, the truck classification demonstrated a lower highest confidence value than bikes and cars, with a confidence value of 85%.

Next, we applied various classifiers with our machine-learning approach to the same vehicle dataset used in this study. Prior work by Janak et al. [27] has discussed vehicle counting using morphology in different real-time traffic videos. However, the method was less accurate for congested traffic, resulting in a false count of on-road vehicles. To overcome this limitation, we focused on classifying three passenger vehicles using a machine-learning approach for static images only.

The present study has certain limitations. First, the results may vary for different iterations of trained datasets with varying environmental conditions. Additionally, all the images were resized to achieve better evaluation, which may require additional computation to implement these classifiers in real time directly. It may be necessary to increase the number of iterations and the size of the image dataset to increase the vehicle recognition process. It is worth noting that the Math Works website classifies the speed measurements as fast, medium, and slow for 0.01 s, 1 s, and 100 s, respectively, while *memory usage* is defined as small, medium, and large for data sizes of 1 MB, 4 MB, and 100 MB, respectively.

The classifiers' memory usage and prediction speed differ based on the selected algorithm. For the ensemble classifier, the speed and memory usage can range from medium to fast for SVM, high to low, or vice versa. For tree classifiers, memory usage is small, and speed is fast. The discriminant analysis classifier can have small or large memory usage, and the speed can be fast or slow depending on the selected method. KNN has medium memory usage and speed. It is worth noting that the prepared dataset differs from deep convolutional architecture networks, such as AlexNet, VGGNet, and ResNet. This study relies on hand-crafted features like SIFT and SURF, whereas CNN can learn features from data (images) and derive scores from the output that may impact the generalizability of the study's findings.

CONCLUSION AND FUTURE SCOPE

In this study, the accuracy and prediction speed of twenty-two different classifiers were evaluated using a machine learning approach for vehicle recognition of three categories - bikes, cars, and trucks. The medium Gaussian SVM classifier was more accurate than the other twenty-one classifiers. The study also compared the performance of different classifiers on image datasets of varying sizes and iteration times, as presented in Table 1 and 2. Additionally, the study compared the machine-learning approach with the deep-learning approach, as discussed in earlier work using CNN [26], for the same image dataset. The accuracy of the deep-learning approach was measured using a confidence value, while the machine-learning approach measured accuracy using TPR, FPR, PPV, and FDR values. The twenty-two classifiers were used to identify bikes, cars, and trucks.

The study found that SVM and KNN classifiers have True Positive Rates of almost 90%, while the Positive Predictive Values in SVM classifiers are above 93%. The False Positive Rates are almost below 10% in SVM classifiers, and False Discovery Rates are below 8% in SVM classifiers. Among the classifiers, the medium Gaussian SVM classifier provides the best accuracy of around 91%. Furthermore, the medium Gaussian SVM classifier was verified and tested on random images, as shown in Figures (4) - (5), which present the predicted and actual objects.

In summary, this study shows that the machine learning approach is effective for vehicle classification, especially when working with smaller amounts of data. The results of this approach outperform those of a deep learning approach, specifically a CNN, for the same dataset. However, the machine learning approach requires structured data and label information, whereas CNN can work with unstructured data and learn from it. In addition, the machine learning approach requires more computational time when working with larger amounts of data, while CNN requires more data to achieve accurate results. Therefore, the approach choice should be based on the specific data and the problem at hand.

The present study demonstrates the potential of machine learning approaches for accurately classifying different types of vehicles from static images. The future scope of this work is to extend the proposed classifier module for real-time applications that classify various objects. In real-time scenarios, classifying different objects requires the simultaneous extraction of feature information and image localization. The proposed work can be further extended to enable the classification of different objects with real-time scenarios using available surveillance systems. For this purpose, a more extensive dataset of various environmental conditions and sizes can be prepared to train the proposed models for accurate and robust object recognition in real time, which can lead to the development of intelligent surveillance systems capable of real-time object recognition with applications in various fields, including security, transportation, and healthcare.

In addition to the limitations mentioned, it should also be noted that the proposed work uses hand-crafted features like SIFT and SURF, which may not be as robust as the features learned by deep convolutional neural networks. Future work could investigate the use of deep learning methods to extract features directly from the image data.

The offered procedure in the study is a machine learning-based vehicle recognition and classification approach, which has several advantages and limitations. One of the significant advantages is that it can achieve high accuracy in vehicle recognition with smaller amounts of data compared to deep learning methods like CNN. Additionally, it provides a more structured approach to data processing, requiring labeled information to classify objects accurately. However, it also has limitations, including a longer computational time when working with larger amounts of data and using hand-crafted features that may not be as robust as deep learning-based features.

The suggested approach has practical applicability in various traffic flow situations, including continuous traffic flow, parking lots, toll facilities, interchanges and intersections, and identifying congestion on urban roads. In a continuous traffic flow situation, the proposed method can monitor traffic and recognize different types of vehicles, such as cars, trucks, and bikes, for better traffic management. In parking lots, the proposed method can help identify different vehicles and prevent unauthorized parking while in toll facilities, it can be integrated into the automatic toll collection system for better efficiency and accuracy.

The presented solution can be used for traffic control and monitoring at interchanges and intersections to prevent accidents and traffic jams. Additionally, it can identify urban road congestion and adjust traffic signals and routes accordingly for better traffic management. Overall, the proposed method can contribute significantly to developing intelligent surveillance systems capable of real-time object recognition, with practical applications in various fields, including transportation and security.

Furthermore, while the proposed work focuses on vehicle classification, it may generalize poorly to other object categories. Thus, future work could explore the applicability of this approach to other object recognition tasks. Additionally, the proposed work assumes that images are captured from a stationary camera, which may not be the case in real-world scenarios. Future work could investigate this approach's use in moving camera scenarios.

Finally, the proposed work has potential practical applications in the field of transportation, particularly in developing more efficient and accurate toll-collection systems. Further research could explore integrating this approach into such systems to improve their functionality and reduce traffic congestion.

(Received February 2023, accepted February 2023)

REFERENCES

[1] Trivedi, J., Devi, M.S. and Dave, D. (2017), ” Review Paper on Intelligent Traffic Control system using Computer Vision for Smart City”, Int. Jour. of Sci. & Engi. Res., Volume 8, Issue 6, pp.14-17.

[2] Russo, F., Rindone, C., Panuccio Panuccio P. (2016)," European plans for the smart city: from theories and rules to logistics test case, European Planning Studies," 24:9, 1709-1726, DOI:10.1080/09654313.2016.1182120

[3] Croce, A., Musolino, G., Rindone, C., and Vitetta A. (2019), “Transport System Models and Big Data: Zoning and Graph Building with Traditional Surveys, FCD and GIS”, Int. J. Geo-Inf., 8(4), 187, DOI:10.3390/IJGI8040187

[4] Hardin, P. and N. Thomson, C. (1992), "Fast Nearest neighbor Classification Methods for Multispectral Imagery", Professional Geographer 44 (2): 191–202, DOI:10.1111/J.0033-0124.1992.00191.X

[5] Ghosh, A. K. (2007), "On Nearest Neighbor Classification Using Adaptive Choice of K" Journal of Computational and Graphical Statistics 16 (2): 482–502, DOI:10.1198/106186007X208380

[6] Breiman, L. (2001), “Random Forests”, Machine Learning 45, 5–32,

[7] DOI: https://doi.org/10.1023/A:1010933404324

[8] Loh, W. (2002), "Regression Trees with Unbiased Variable Selection and Interaction Detection", Statistica Sinica 12, 361-386.

[9] Paensuwan, N., Yokoyama, A., C. Verma, S., and Nakachi Y. (2012) "Application of Decision Tree Classification to the Probabilistic TTC Evaluation" Journal of International Council on Electrical Engineering 1(3):323–30, DOI:10.5370/JICEE.2011.1.3.323

[10] Wen, X., Shao, L., Xue, Y., Fang, W. (2015), "A Rapid Learning Algorithm for Vehicle Classification" Information Sciences 295: 395–406, DOI:10.1016/j.ins.2014.10.040

[11] Kim, J. Jeonghyun Baek, J. and Kim, E. (2015), "A Novel On-Road Vehicle Detection Method Using ΠHOG" IEEE Transactions on Intelligent Transportation Systems 16 (6): 3414–29, DOI:10.1109/TITS.2015.2465296

[12] Stocker, M., Silvonen, P., Rönkkö, M., and Kolehmainen, M. (2016), "Detection and Classification of Vehicles by Measurement of Road-Pavement Vibration and by Means of Supervised Machine Learning" Journal of Intelligent Transportation Systems: Technology, Planning, and Operations 20 (2): 125–37, DOI:10.1080/15472450.2015.1004063

[13] Arinaldi, A., Arya Pradana, J., and Arventa Gurusinga, A. (2018), “Detection and classification of vehicles for traffic video analytics”, Procedia Computer Science 144:259-68, DOI:10.1016/j.procs.2018.10.527

[14] Atiya, A. (2005), “Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond”, IEEE Transactions on Neural Networks, 16 (3), DOI:10.1109/tnn.2005.848998

[15] Sarikan, S., Ozbayoglu, A., and Zilci, O. (2017), "Automated Vehicle Classification with Image Processing and Computational Intelligence" Procedia Computer Science 114:515–22, DOI:10.1016/J.PROCS.2017.09.022

[16] Gorges, C., Öztürk, K., Liebich R. (2018), "Road Classification for Two-Wheeled Vehicles" Vehicle System Dynamics 56 (8): 1289–1314, DOI:10.1080/00423114.2017.1413197

[17] Bautista, M. C., Dy, A. C., Manalac, I. M., Orbe, A. R. and Cordel, M. (2016), "Convolutional neural network for vehicle detection in low-resolution traffic videos," IEEE Region 10 Symposium, pp. 277-281, DOI:10.1109/TENCONSPRING.2016.7519418

[18] Bansal, D., Khanna, K., Chhikara, R., Kumar Dua, R. and Malhotra, R. (2020), “Classification of Magnetic Resonance Images using Bag of Features for Detecting Dementia”, Procedia Computer Science 167, pp. 131–137, DOI:10.1016/j.procs.2020.03.190

[19] Pranata, Y. D., Wang, K.C., Wang, J.C., Idram, I., Lai, J.Y., Liu, J.W., and Hsieh, I.H. (2019), “Deep learning and SURF for automated classification and detection of calcaneus fractures in CT images,” Computer Methods and Programs in Biomedicine 171, pp.27–37, DOI: https://doi.org/10.1016/j.cmpb.2019.02.006

[20] Sykora, P., Kamencay, P., and Hudec, R. (2014), “Comparison of SIFT and SURF Methods for Use on Hand Gesture Recognition Based on Depth Map”, AASRI Conference on Circuits and Signal Processing (CSP 2014), pp- 19 – 24, DOI:10.1016/j.aasri.2014.09.005

[21] Ignat, A. and Pavaloi, I. (2021), “Keypoint Selection Algorithm for Palmprint Recognition with SURF”, 25th Inter. Conf. on Knowledge-Based and Intelligent Information & Engineering Systems, Procedia Computer Science 192, 270–280, DOI: 10.1016/j.procs.2021.08.028

[22] Anzid, H., Le Goic, G., Mammass, D., Bekkari, A. and Mansouri, A. (2019), “Multimodal Images Classification using Dense SURF, Spectral Information and Support Vector Machine”, Second International Conference on Intelligent Computing in Data Sciences, Procedia Computer Science 148, pp-107–115, DOI:10.1016/j.procs.2019.01.014

[23] LoweDavid, G. (2004), “Distinctive Image Features from Scale-Invariant Keypoints”, International Journal of Computer Vision 60 (2): 91–110.

[24] Bay, H., Tuytelaars, T., and Gool L. (2006), “SURF: Speeded Up Robust Features.” Computer Vision, ECCV 404–17, DOI:10.1007/11744023_32

[25] https://in.mathworks.com/help/stats/choose-a-classifier.html.

[26] https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/

[27] Trivedi, J., Devi, M.S. and Dave, D. (2021), “Vehicle classification using the convolution neural network approach,” Scientific Journal of Silesian University of Technology. Series Transport, 112, 201-209. ISSN: 0209-3324. DOI: https://doi.org/10.20858/sjsutst.2021.112.7.16.

[28] Trivedi, J., Devi, M.S. and Dave, D. (2018), “Vehicle Counting Module Design in Small Scale for Traffic Management in Smart City”, IEEE, 3rd Int. Conf. for Convergence in Tech. (I2CT), DOI: 10.1109/I2CT.2018.8529506.

ATTACHMENTS

Annexure-I Number of features for different vehicles.

Sr . No.	Number of test images	Image class /Extracted Number features			
		Bikes A	Cars B	Trucks C	Total Number of Features
1	100	95783	67389	77560	161733
2	200	201081	140479	157553	337149
3	300	294684	204318	232975	490362
4	400	399111	273086	305896	655407
5	500	506155	344331	388061	826395
6	600	603543	420279	465669	1008669
7	700	696414	489869	549068	1175685
8	800	789421	552722	637420	1326534
9	900	886088	622349	724289	1493637
10	1000	986129	688459	826715	1652301

Annexure-II Vehicle classification using different classifiers.

	Decision Trees														
	Fine Trees					Medium Trees					Coarse Trees				
	B	C	T	TPR (%)	FPR (%)	B	C	T	TPR (%)	FPR (%)	B	C	T	TPR (%)	FPR (%)
B	1170	296	134	73	27	1153	336	111	72	28	966	520	114	60	40
C	231	1013	356	63	37	241	1023	336	64	36	222	940	438	59	41
T	114	376	1110	69	31	98	498	1004	63	37	114	479	1007	63	37
PPV (%)	77	60	69			77	55	69			74	48	65		
FDR (%)	23	40	31			23	45	31			26	52	35		

	Discriminant Analysis									
	Linear Discriminant					Quadratic Discriminant				
	B	C	T	TPR (%)	FPR (%)	B	C	T	TPR (%)	FPR (%)
B	1466	109	25	92	8	1340	242	18	84	16
C	78	1353	169	85	15	95	1450	55	91	9
T	20	168	1412	88	12	65	466	1069	67	33
PPV (%)	94	83	88			89	67	94		
FDR (%)	6	17	12			11	33	6		

	Support Vector Machines (SVM)														
	Linear SVM					Quadratic SVM					Cubic SVM				
	B	C	T	TPR (%)	FPR (%)	B	C	T	TPR (%)	FPR (%)	B	C	T	TPR (%)	FPR (%)
B	1471	109	20	92	8	1489	96	15	93	7	1496	85	19	94	6
C	77	1376	147	86	14	75	1382	143	86	14	75	1386	139	87	13
T	22	151	1427	89	11	15	135	1450	91	9	14	136	1450	91	9
PPV (%)	94	84	90			94	86	90			94	86	90		
FDR (%)	6	16	10			6	14	10			6	14	10		
	Fine Gaussian SVM					Medium Gaussian SVM					Coarse Gaussian SVM				
	B	C	T	TPR (%)	FPR (%)	B	C	T	TPR (%)	FPR (%)	B	C	T	TPR (%)	FPR (%)
B	209	1391	0	13	87	1488	95	17	93	7	1467	114	19	92	8
C	0	1600	0	100	0	57	1434	109	90	10	79	1376	145	86	14
T	0	1313	287	18	82	17	147	1436	90	10	33	163	1404	88	12
PPV (%)	100	37	100			95	86	92			93	83	90		
FDR (%)	0	63	0			5	14	8			7	17	10		

	Nearest Neighbor Classifier (NN)														
	Fine KNN					Medium KNN					Coarse KNN				
	B	C	T	TPR (%)	FPR (%)	B	C	T	TPR (%)	FPR (%)	B	C	T	TPR (%)	FPR (%)
B	1431	105	64	89	11	1480	80	40	93	7	1457	106	37	91	9
C	212	986	402	62	38	200	1123	277	70	30	182	1218	200	76	24
T	76	210	1314	82	18	87	232	1281	80	20	37	285	1218	76	24
PPV (%)	83	76	74			84	78	80			84	76	84		
FDR(%)	17	24	26			16	22	20			16	24	16		
	Cosine KNN					Cubic KNN					Weighted KNN				
	B	C	T	TPR (%)	FPR (%)	B	C	T	TPR (%)	FPR (%)	B	C	T	TPR (%)	FPR (%)
B	1517	47	36	95	5	1501	56	43	94	6	1483	75	42	93	7
C	252	1054	294	66	34	282	1001	317	63	37	179	1108	313	69	31
T	111	137	1352	85	15	118	156	1326	83	17	70	173	1357	85	15

PPV (%)	81	85	80			79	83	79			86	82	79		
FDR (%)	19	15	20			21	17	21			14	18	21		
	Ensemble Classifier														
	Boosted Trees					**Bagged Trees**					**Subspace Discriminant**				
	B	C	T	TPR (%)	FPR (%)	B	C	T	TPR (%)	FPR (%)	B	C	T	TPR (%)	FPR (%)
B	1340	203	57	84	16	1456	125	19	91	9	1478	100	22	92	8
C	178	1176	246	74	26	176	1232	192	77	23	81	1364	155	85	15
T	74	291	1235	77	23	74	217	1309	82	18	22	179	1399	87	13
PPV (%)	84	70	80			85	78	86			93	83	89		
FDR (%)	16	30	20			15	22	14			7	17	11		
	Subspace KNN					**RUSBoosted Trees**									
	B	C	T	TPR (%)	FPR (%)	B	C	T	TPR (%)	FPR (%)					
B	1516	49	35	95	5	1147	351	102	72	28					
C	221	1052	327	66	34	277	967	336	62	38					
T	75	130	1395	87	13	102	484	1014	63	37					
PPV (%)	84	85	79			75	54	70							
FDR (%)	16	15	21			25	46	30							