ARPN Journal of Engineering and Applied Sciences

AN IMPROVED APPROACH FOR DETECTION AND CLASSIFICATION OF VEHICLES IN VIDEO USING SUPPORT VECTOR MACHINES

Selvanayaki K. S.[1] and Somasundaram Rm[2]
[1]Department of Computer Applications, Easwari Engineering College, Chennai, India
[2]Computer Science and Engineering, SNS College of Engineering, Coimbatore, India
E-Mail: kssindia2004@yahoo.co.in

ABSTRACT

Deployment of effective surveillance and security measures is important in these days. The system must be able to provide access and track movement of different types of vehicles and people entering the secured premises, to avoid any mishap from happening. There are many existing approaches which are used for tracking objects. Edge matching, Divide-and-Conquer search, Gradient matching, Histograms of receptive field responses, Pose clustering, SIFT; SURF etc are some of the approaches applied. All these methods are either Appearance based methods or Feature based methods. They lag in one or the other way when it comes to real time applications. So there has been a need for creating a new system that could combine positive aspects of both the methods and increase the efficiency in tracking objects, when it comes to real life scenario. A novel approach for car detection and classification is presented, to a whole new level, by devising a system that takes the video of a vehicle as input, detects and classifies the vehicle based on its make and model. It takes into consideration four prominent features namely Logo of vehicle, its number plate, colour and shape. The classification is done by a very efficient algorithm called Support vector machines. Experimental results show that our system is a viable approach and achieves good feature extraction and classification rates across a range of videos with vehicles under different conditions.

Keywords: vehicles, classifier, feature detection, edge matching, gradient matching.

INTRODUCTION

The detection and classification of moving objects is an important area of research in video processing. An object is detected either in every frame or when the object first appears in video. A common approach for object detection is to use information in a single frame. However, some object detection methods make use of the temporal information computed from a sequence of frames to reduce the number of false detections. This temporal information is usually in the form of frame differencing, which highlights changing regions in consecutive frames. The temporal characteristics of an object are information between frames with respect to time i.e. the information slowly vary in consecutive frames with respect to time.

As a scientific discipline, image processing is concerned with the theory and technology for building systems that obtain and process information from images or videos. One of the challenges in implementing vehicle recognition and classification systems is to design models that work across a wide variety of users and environments. Image processing is based on features that are captured by the image detection and recognition system from a video or image. Several approaches have been developed in the past for detecting car by attending to different elements like shape of car or logo of car etc. Indeed, there are different static and dynamic features that we use to successfully identify and classify them.

Support Vector Machines are a family of learning algorithms. The goal of a learning algorithm is to use a set of observations S, which we call the training set, to learn a rule which can be used in the future to classify any new object $x \in X$ into a class $y \in \{-1, +1\}$. SVM is one particular algorithm which performs this pattern recognition task, i.e. learning from S a classifier for future objects [9].

An observation is simply an object x together with its class y, which we denote by (x, y). With these notations a series of N observations can be written as follows:

$$S = \{(x1, y1), \ldots, (xN, yN)\} \quad (1)$$

Where for any i = 1,...,N, (xi,yi) is an observation.

SVM are usually among the best algorithms, with efficient results. Being efficient means several things which we can quickly sum up:

- Good generalization performance: Once the SVM is presented with a training set, it is able to learn a rule which can correctly classify any new object quite often.
- Computational efficiency: The algorithm is efficient in terms of speed and complexity; it is equivalent to looking for the minimum of a convex functional, i.e. with no local minima (unlike neural networks).
- Robust in high dimensions: Dealing with large dimensional objects (like images of gene expression data) is usually difficult for learning algorithm, because of the overfitting issue. SVM seem to be more robust than other methods in such cases.

Theoretical results suggest that its efficiency is mainly due to its capacity to find rules which classify objects with high confidence, to prevent them from over fitting [10].

ARPN Journal of Engineering and Applied Sciences

RELATED WORK

The detection and recognition of a car by a system has improved from the time, machines started to become more and more intelligent with the advantage of filling in, correcting, or helping the lack of human abilities and senses, feature extraction techniques have been keenly concentrated and improved on fields of time, complexity and accuracy. These subjects are as old as computer vision and have always been an active research field because of its non-invasive nature. Two main approaches formed the early part of research in feature extraction, first Appearance based methods and second Feature based methods. The Appearance based methods take into consideration the basic appearances like edges, colour, histogram etc for feature extraction, but the Feature based methods like SIFT and SURF take the keypoints into consideration. Similarly there have been significant breakthroughs in the machine learning field, which led to the finding of novel and efficient methods like SVM classifier [4]. There have been active developments in car recognition systems in recent years. And this work presents some of the novel and efficient approaches for feature extraction and classification.

The active safety system helps in reducing automotive accidents. They are accurate, reliable and efficient in identifying the dangerous conditions. This system makes the vehicle intelligent by avoiding collisions and by detecting lane departures, pedestrians and other vehicles on the road. This system uses a vision based on road vehicle detectors. In this paper a general model for robust active learning based vehicle recognition and tracking is introduced [1].

Using this model the system is implemented and their performance is evaluated for both real world video and vehicle images. The performance metrics by which the system is tested are: robustness, recall, precision and localization. Vehicle detection uses Principal Component Analysis and Independent Component Analysis. This showed a very good result in detecting static images of parked vehicles. Active learning refers to the process where there is some control over the input data. It is more powerful than passive learning. Vehicle recognition using rectangular features and an Adaboost classifier is also discussed. QUAIL was used to perform selective sampling. Thus the system helps in tracking the vehicles on road. It could be using local descriptors [8].

SIFT consists of four major stages: scale-space extrema detection, keypoint localization, orientation assignment and key point descriptor. The first stage used difference-of-Gaussian function to identify potential interest points, which were invariant to scale and orientation. DOG was used instead of Gaussian to improve the computation speed.

$$D(x, y, s) = (G(x, y, ks) - G(x, y, s)) * I(x, y) = L(x, y, ks) - L(x, y, s) \qquad (2)$$

In the keypoint localization step, they rejected the low contrast points and eliminated the edge response. Hessian matrix was used to compute the principal curvatures and eliminate the keypoints that have a ratio between the principal curvatures greater than the ratio. An orientation histogram was formed from the gradient orientations of sample points within a region around the keypoint in order to get an orientation assignment. According to the paper's experiments, the best results were achieved with a 4 x 4 array of histograms with 8 orientation bins in each. So the descriptor of SIFT that was used is 4 x 4 x 8 = 128 dimensions [5].

SURF (Speeded up Robust Feature) is a robust image detector and descriptor, first presented by Herbert Bay *et al.* in 2006, that can be used in computer vision tasks like object recognition or 3D reconstruction. It is partly inspired by the SIFT descriptor. SIFT and SURF algorithms employ slightly different ways of detecting features [2]. SIFT builds an image pyramids, filtering each layer with Gaussians of increasing sigma values and taking the difference. On the other hand, SURF creates a "stack" without 2:1 down sampling for higher levels in the pyramid resulting in images of the same resolution. Due to the use of integral images, SURF filters the stack using a box filter approximation of second-order Gaussian partial derivatives, since integral images allow the computation of rectangular box filters in near constant time [2].

In key point matching step, the nearest neighbor is defined as the keypoint with minimum Euclidean distance for the invariant descriptor vector. Lowe used a more effective measurement that obtained by comparing the distance of the closest neighbor to that second-closest neighbor [2] so the author of this paper decided to choose 0.5 as distance ratio like Lowe did in SIFT. The comparison of SVM vs Artificial Neural Networks is presented in Table-1[11] [12].

ARPN Journal of Engineering and Applied Sciences

Table-1. Comparison of SVM and Neural networks.

<table>
<tr><th>Criteria</th><th>SVM</th><th>Artificial Neural Networks</th></tr>
<tr><td>Methodology</td><td>SVM is a supervised learning model with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. The basic SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the output, making it a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.</td><td>A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases a neural network is an adaptive system changing its structure during a learning phase. An ANN is typically defined by three parameters:
1. The interconnection pattern between different layers of neurons
2. The learning process for updating the weights of the interconnections
3. The activation function that converts a neuron's weighted input to its output activation.</td></tr>
<tr><td>Merits</td><td>1. By introducing the kernel, SVMs gain flexibility in the choice of the form of the threshold separating solvent from insolvent companies, which need not be linear and even need not have the same functional form for all data, since the kernel function is non-parametric and operates locally.
2. SVMs provide a good out-of-sample generalization, if the parameters C and r (in the case of a Gaussian kernel) are appropriately chosen. Thus. by choosing an appropriate generalization grade,
SVMs can be robust, even when the training sample has some bias.
3. SVMs deliver a unique solution, since the optimality problem is convex. This is an advantage compared to Artificial Neural Networks, which have multiple solutions associated with local minima and for this reason may not be robust over different samples.
4. The overall accuracy rate remains stable even when the size of the sample is changed. It is independent of the data set.</td><td>1. Because of their parallel architecture, they can overcome most of the limiting computational difficulties.

2. Since they are trained on example data, they are more adaptable to changes in the input data by allowing the training to continue during the processing of new information.

3. The high degree of connectivity allows neural networks to self-organize, which is important when the structure of the data is not known beforehand.</td></tr>
<tr><td>Limitations</td><td>1. The biggest limitation of SVM lies in the choice of the kernel. The best choice of kernel for a given problem is still a research problem.

2. The speed and size for large training sets is very high.

3. The optimal design for multiclass SVM classifiers is also still a research area.</td><td>1. ANNs often converge on local minima rather than global minima, meaning that they are essentially "missing the big picture" sometimes (or missing the forest for the trees)
2. ANNs often overfit if training goes on too long, meaning that for any given pattern, an ANN might start to consider the noise as part of the pattern.
3. When the size of the data set increases, the overall accuracy decreases. When the size increases to 800 pixels per class, accuracy percentage reduces to 58%, while the accuracy level at 20 pixels per class sample size is 67%.</td></tr>
<tr><td>Applications</td><td>▪ Network areas. For E.g.: classifying the different network application like FTP, HTTP, P2P, etc.,
▪ Text classification
▪ Speech recognition
▪ Image clustering for image compression
▪ Image classification
▪ hand written character/digit recognition problem
▪ recognition of shape and hand gesture in static as well as in dynamic environment</td><td>In industry:
▪ Sales forecasting
▪ Industrial process control
▪ Customer research
▪ Data validation
▪ Risk management
▪ Target marketing.
▪ In Medicine:
▪ Modelling and Diagnosing the Cardiovascular System
▪ Electronic noses
• Instant Physician</td></tr>
</table>

VOL. 10, NO. 10, JUNE 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Table-2. Comparison of performance measures of SVM and Neural networks.

Criteria (for classifying MODIS time series data set)	SVM	Neural networks
Accuracy for small samples (20 pixels per class)	Best (77%)	Good (67%)
Accuracy for large samples (800 pixels per class)	Best (74%)	Good (58%)
Misclassification rate	Less (59%)	High (77%)
Omission errors	Less (33%)	High (38%)
Processing time	High	Low

The Support Vector Machine (SVM) provides a robust, accurate and effective technique for pattern recognition and classification. Although the SVM is essentially a binary classifier, it can be adopted to handle multi-class classification tasks. The conventional way to extent the SVM to multi-class scenarios is to decompose an m-class problem into a series of two class problems, for which either the one-vs.-one (OVO) or one-vs.-all (OVA) approaches are used. In this paper, a practical and systematic approach using a kernelled SVM is proposed and developed such that it can be implemented in embedded hardware within a road-side camera [12]. The metrics are accuracy for small and large samples, misclassification errors, omission errors and processing time [6].

PROPOSED WORK

The classification and counting process is done in two steps as shown in Figure-1.

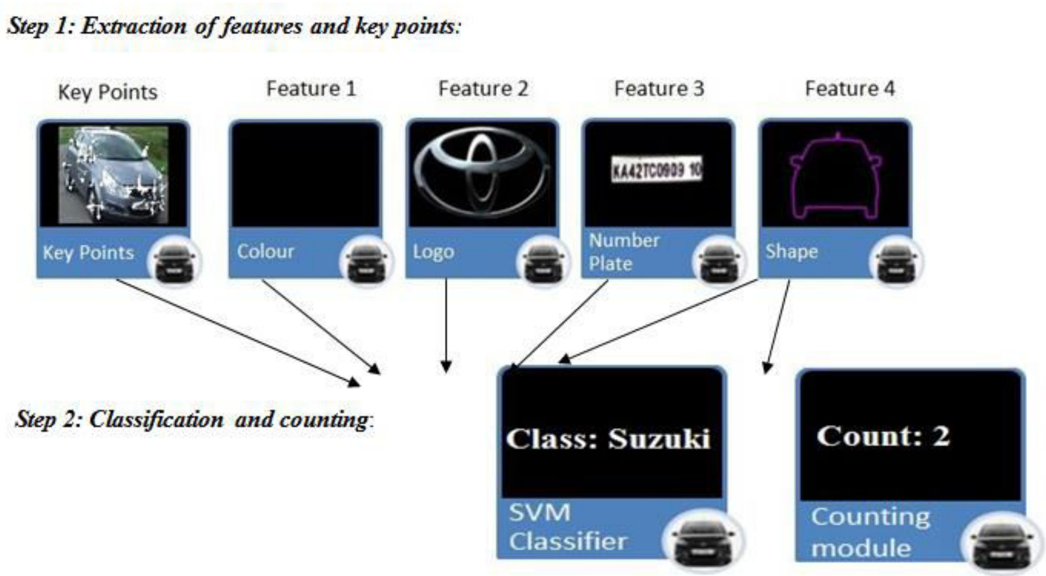

Figure-1. Overall flow diagram.

In the first step all the features and the key points are extracted. Each feature is given a value if it meets the evaluation criteria. For example colour feature is set to be one if the car is black, gray or white. Similarly all other features are extracted and given a specific value if it meets the criteria.

In the second step these features are fed in to SVM classifier which predicts the class to which the car belongs to. It returns a “0” value to indicate that car does not belong to any of the classes considered. The counting module gives the count of cars in the video. Eventually, this system replaces the usual method of manual checking of cars by automatically authenticating it, once the video is fed in. Cars, which is proved to be a more difficult object for detection and recognition due to its varying structure from different perspectives of view of the same car, as well as varying between different makes and models, this system is capable of recognizing and giving a authentication message to the user.

RESULTS AND DISCUSSIONS

The basic training of the Haar cascade classifier is done in Ubuntu 11.10

- Intel Pentium processor G630 (2.70GHz).
- 2GB DDR3, SDRAM at 1333MHz.

The training will give xml file which can detect the logo portion in the frame. Since we need to detect 3 different logos, we train 3 Haar Cascade Classifiers to detect the logos [16].

VOL. 10, NO. 10, JUNE 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Figure-2. Outputs of Haar Cascade classifier detection.

The output of Haar Cascade classifier is shown in Figure-2.

As mentioned earlier training Haar Cascade Classifiers need Positive and Negative Images. The classifier gives efficient result when it is trained with more number of Positive Images and Negative Images. Depending on the number of images given for training, the training time also vary. The more the number of training data, the higher is the time to train the Classifier. And the classifier would be more efficient if the number of negative images is more than the number of positive images. Below are some observations:

Table-3. Observations for Haar Cascade classifier.

Logo	Positive images used	Negative images used	Results obtained for image	Results obtained for video	Time taken for training
Suzuki (1)	10	Zero	Accepted only the trained images.	Ineffective in video.	10 min
Suzuki (2)	200	50	Worked well with other data sets of Suzuki cars.	Moderately effective detection with some false positives.	60 min
Hyundai (1)	50	50	Good detection, with logo in favorable orientations.	Same kinds of results were found in video also.	40 min
Hyundai (2)	220	100	Worked well with other data sets of Hyundai also.	Effective with less number of false positives.	90 min
Toyota	1000	500	Worked better compared to other Haar Cascade Classifiers.	Better detection even for different orientations of logo.	2 ½ days

SIFT matches almost similar images; but it cannot essentially match a logo image from a whole car. This can be solved by extracting the logo region alone using the Haar Cascade Classifier and using that image for matching with the template image.

ARPN Journal of Engineering and Applied Sciences

Figure-3. SIFT matching.

An observation of the performance of the whole system involving logo detection by Haar cascade classifier and SIFT matching in a video.

Table-4. SIFT Matching metrics for different brands of vehicles.

	Hyundai	Toyota	Suzuki
Total frames	35	32	51
No. of frames logo should be detected	27	25	40
No. of frames correctly detected by algorithm	13	11	18
Average SIFT matching with template	2	3	2
Average time taken for SIFT matching	49	45	72

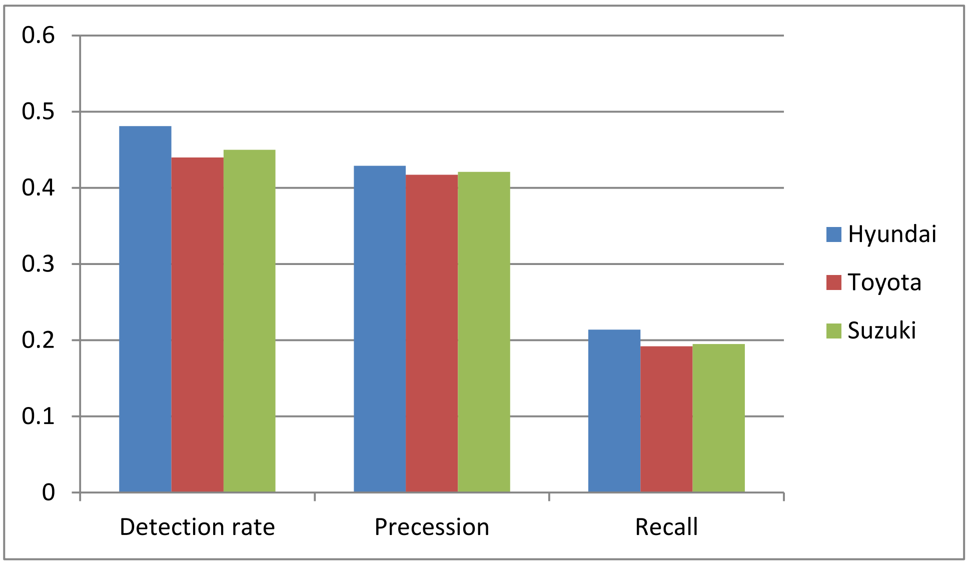

Figure-4. Overall system performances.

It is known that if the number of Positive Images and Negative Images are less, the training will be considerably fast but the classifier will have high False Rejection Rate - rejects the proper logo and False Acceptance Rate - detects a lot of false positives, which are not to be detected. But even if the count of the input images is high, there is a chance that it can detect a false object. Below is an observation of the false positives:

VOL. 10, NO. 10, JUNE 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Table-5. Result of Misclassifications due to false positives.

Module	Suzuki	Hyundai	Toyota
Logo detection	100 (Positive Images), 150 (Negative Images) False positives are high but logo detection is acceptable.	1000 (Positive Images), 2000(Negative Images) False positives are less and logo detection is proper.	1500 (Positive Images), 2400 (Negative Images) False positives are more but logo detection is considerably good.
Observations	84 frames 23 false positives	50 frames 18 false positives	150 frames 59 false positives

When the extracted contours are matched with the contours extracted from templates, the percentage error in matching is calculated with each template and decided if the extracted contours come from a car or not.

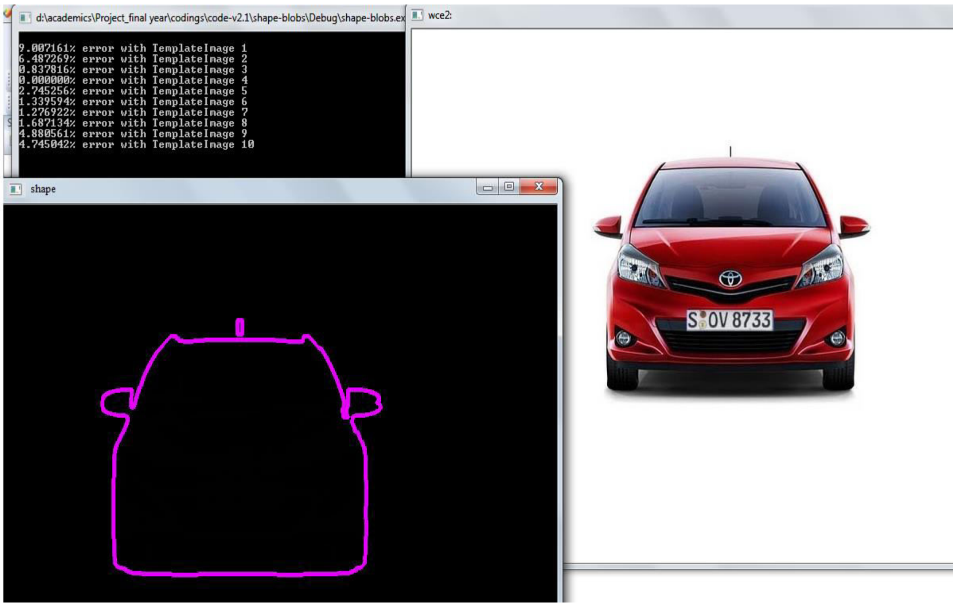

Figure-5. Contour matching.

This method gives good results only under certain conditions like:

1) Less background and more prominent foreground image containing car. Since background objects hinders finding and exact contours that define the shape of the car.
2) Appropriate illumination such that car is visible, as too much lighting will cause some edges invisible.

So the proposed method employs Blob detection for detection of the largest interest region (car) and then finds the contours for the blob that is detected. This will reduce the noise that intrudes during the contour extraction, which in turn reflects in contour matching also.

In this work, the contours are found after applying the canny edge detection method. Canny is considered to be the most effective method in reducing the background and giving the proper edges and contours in an image.

So it can be inferred that Canny is more advantageous than other edge detection methods. So even in edge detection, the proposed work is efficient than its counterparts.

Step-1: Preparing proper dataset with all combinations of feature values.

Step-2: Generating the input training file (train.txt) by giving the dataset as input for the integrated system, which will find the feature values and store in the training file. The training file contains the class labels along with the feature values of each and every train entity given.

Example of the train file format:

Class_label Feature1: Feature1_value Feature2: Feature2_value…

Example: +1 1:0.56 2: 0.89 3:0.67 4:0.00 …

VOL. 10, NO. 10, JUNE 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Step-3: Generating the model file which is a consolidated repository of the trained data and the rule which is going to be used to classify the test data. This will depend on the type of kernel function and the type of SVM which is going to be used. These parameters can be changed by specifying during initializing the SVM as shown in Figure-6.

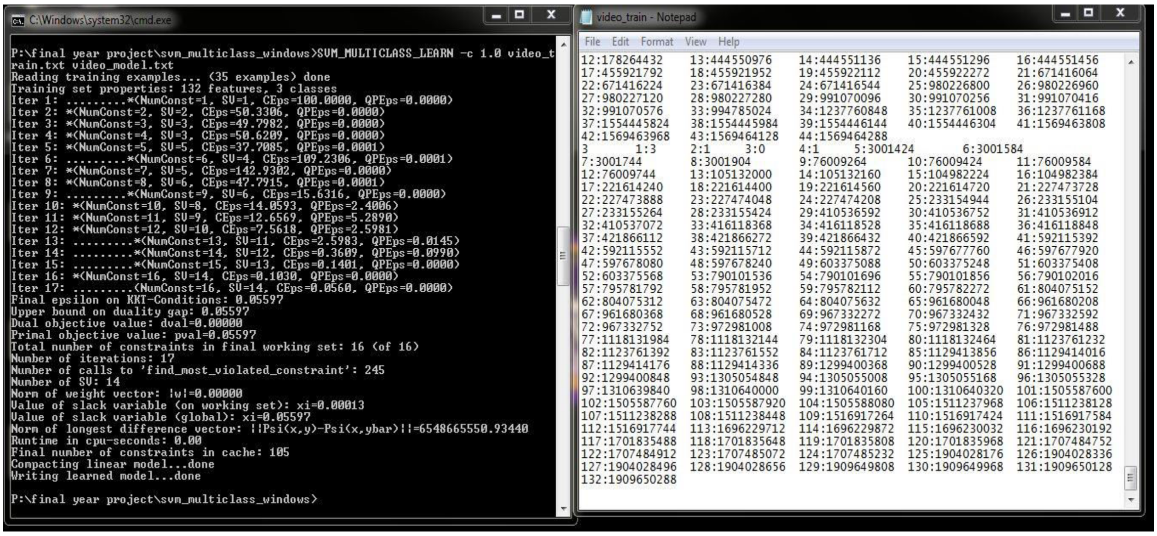

Figure-6. Generating model file.

From the above figure we can see that train.txt (training data) is given as input for generating the model file - train.txt.model. The 1s and 0s in the first column of the train file denote the classes. The Rest denote the feature values.

Step-4: Generating the testing file (text.txt) by giving the test video to the integrated system, which will analyse the video as per the proposed models and produces the feature values. The test file is also of the same format as the input training file. But the only difference between the training and testing file is that the test file will not have a class label or some random number is given for maintaining a uniform format.

So the Feature vector is formed with the following rules:

Color: 1 if color = black/Gray/white and 0 otherwise

Number plate: 1 if it is detected and authenticated and 0 otherwise

Logo: 3 for Suzuki, 2 for Toyota and 1 for Hyundai

Shape: 1 if shape matches with at least 4 templates in every frame and 0 otherwise

The feature set for training is shown in Table-6.

Table-6. Feature set for training.

Dataset	Color	Number plate	Logo	Shape template
Black Hyundai cars	1	1	1	1
Red Suzuki Car	0	1	3	1
Yellow Tata truck	0	0	0	0
Gray Toyota car (without number plate)	1	0	1	1

Step-5: This is final step of classification, where the feature vectors of the test videos are given as a test file to the model file and the class of each and every feature vector is predicted using the rule framed during the training period. This stage will produce an output text file which contains the class labels of the corresponding feature vectors as shown in Figure-7.

ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

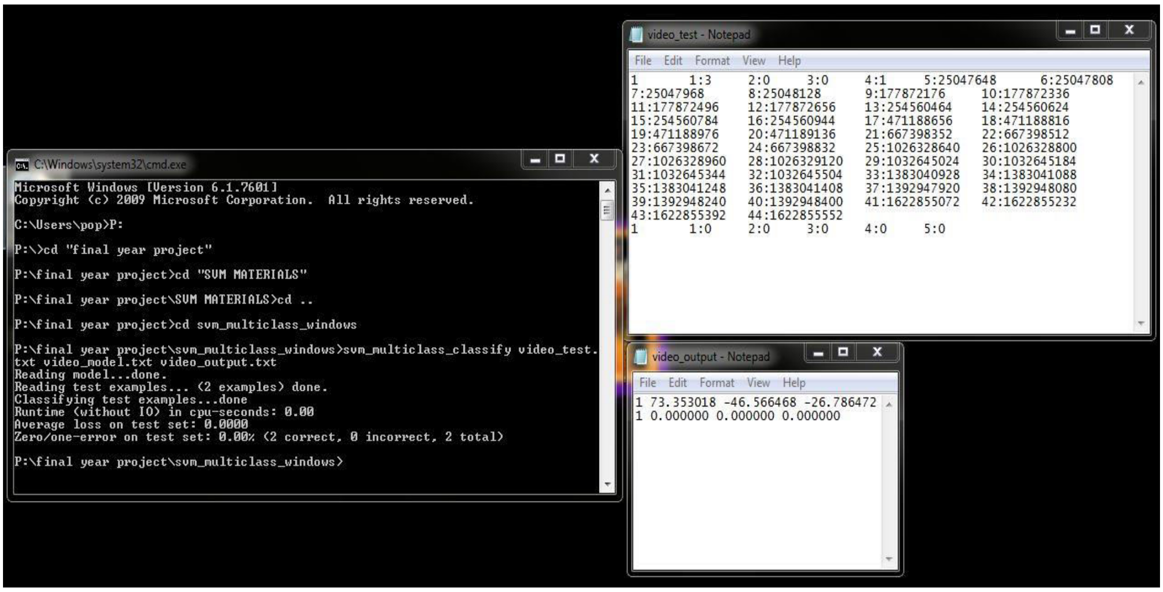

Figure-7. Prediction of Class label.

The above Figure shows the text file (example1) with the feature values extracted from test video. When this text file is given as input for the prediction, an output text file is generated (out) with the output class label corresponding to the feature vector.

In the above figure, example1.txt has the first column as a random number since the class is not yet known. The output file will show the output class label. The model file is used for the prediction purpose and classification of class is done.

Step-6: The output class is read and according to the class that is present, the car is assigned the label.

Class 0 - Not authentic
Class 1 - Hyundai
Class 2 - Toyota
Class 3 - Suzuki

For counting the vehicles, Feature extraction is to be done to ensure that only the desired objects are retained in the binary images. The images may sometimes contain unwanted noises even after thresholding. These are to be eliminated before further processing can be done as shown in Figure-8.

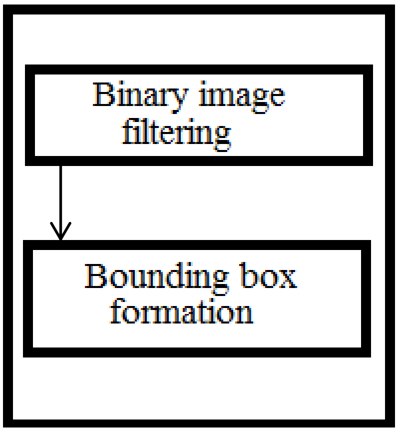

Figure-8. Flow diagram of feature extraction phase.

This is achieved with an area based filtering method. All the objects below a particular area are filtered off. The constant area value is determined such that all the cars have an area value above the constant value. The function used is,

$$THi\,(x, y) = \begin{cases} 0, & if\ area < A \\ 255, & otherwise \end{cases} \tag{3}$$

Here *THi (x, y)* is the thresholded image *area* is the computed area of the object and *A* is the determined fixed area value. After filtering each car is bounded in a filled bounding box to overcome the discontinuities after thresholding. Vehicle tracking and counting is performed with the help of virtual detector. The virtual detector is set such that all the cars pass through it. Once a car enters the detector, its centroid is computed. In order to track the cars, Euclidean distance between the centroid in consecutive frames is calculated.
The equation used is

$$\text{Distance} = ((x_2 - x_1)^2 + (y_2 - y_1)^2)^{1/2} \quad (4)$$

Here (x_1, y_1) and (x_2, y_2) represent the centroid in the current frame and the previous frame respectively. A comparison is done with the calculated distance and a fixed constant distance value. The count is incremented if the calculated distance is less than the constant value.

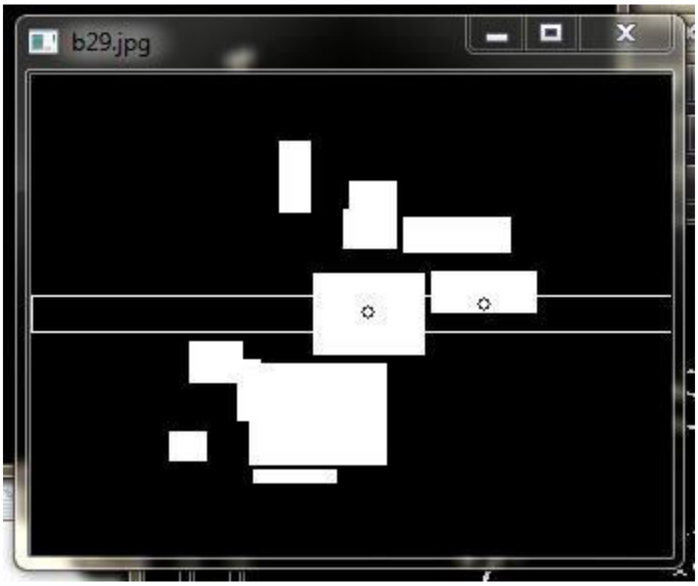

Figure-9. Centroid calculation.

As mentioned above, the system counts the number of cars in the video based on the distance between the centroid of the outline shape of the car in each consecutive frame. Here the threshold value for distance is taken as 75, so whenever distance is greater than 75, count increases by 1. This is shown in Figure-10.

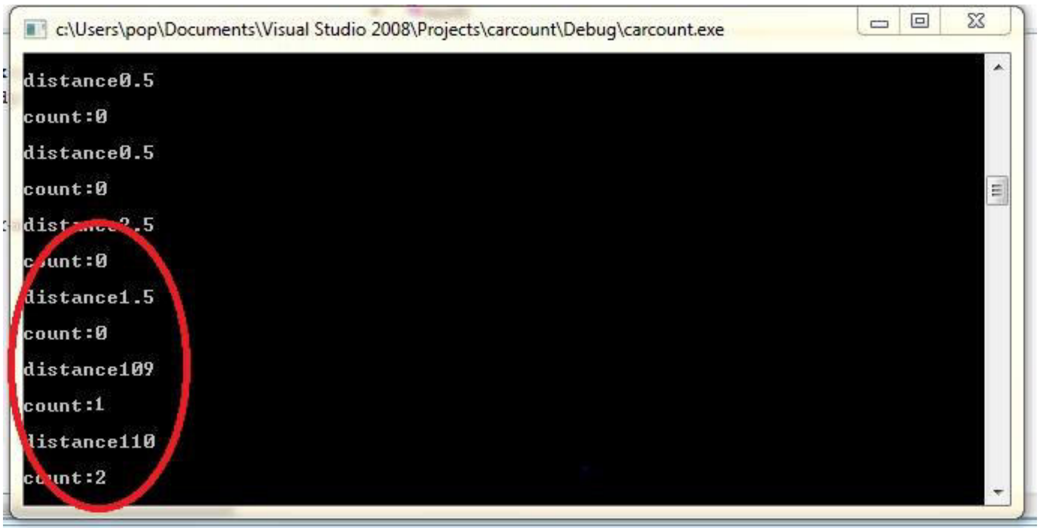

Figure-10. Count of vehicles whenever the distance is greater than threshold.

CONCLUSIONS

The proposed system is a vehicle authentication and detection system, which prevents unauthorized vehicles from entering private areas [7]. Unlike the existing system, which takes only one feature (number plate) for authentication, this system considers the key points as well as four different features namely logo, colour, number plate and shape. This system uses well trained xml files to detect and localize car and logo region in the car. Number plate is detected using blobs extraction techniques. Shape of the vehicle is realized using blobs and contours. Colour of the car is detected by considering centre pixel and seven neighbouring pixels of the car region. These features are fed into classifier which classifies whether a car is authenticated or not. Thus authentication is done without any human intervention. Since the system takes into consideration the key points and four features for detection, the efficiency of the classification system is further enhanced.

As of now the system is only capable of detecting cars of three different manufacturers i.e. Suzuki, Hyundai and Toyota. As part of the future work the system design needs to be modified such that vehicles of any make can be authenticated.

Table-7. Comparison of existing and proposed method.

Feature Used	Existing method	Proposed method
Logo	Haar Cascade Classifier (Image and video)	SIFT + Haar (Image and video)
Number Plate	Blob Detection (Image)	Blob Detection + Optical Character Recognition (Image and video)
Color	HSV Color Detection (Image)	Haar + RGB color Model (Image and video)
Shape	Contour Matching (Image)	Blob Detection + Contour Matching (Image and video)

REFERENCES

[1] Sayanan Sivaraman and Mohan Manubhai Trivedi, 2010, "A General Active-Learning Framework for On-Road Vehicle Recognition and Tracking", IEEE transactions on intelligent transportation systems, vol. 11, no. 2, pp. 267-276.

[2] Luo Juan, Oubong Gwun, 2008, “A Comparison of SIFT, PCA-SIFT and SURF”, International Journal of Image Processing (IJIP).

[3] Humayun Karim Sulehria, Ye Zhang,2007, “Vehicle Logo Recognition Using Mathematical Morphology” , 6th WSEAS Int. Conference on telecommunications and informatics, Dallas, Texas, USA, pp.95-98, March 22-24.

[4] Apostolos P. Psyllos, Christos-Nikolaos, E. Anagnostopoulos, 2010, “Vehicle Logo Recognition Using a SIFT-Based Enhanced Matching Scheme”, IEEE Transactions on intelligent transportation systems, pp. 323-328.

[5] Milos Stojmenovic, 2005, “Real time object detection in images based on an AdaBoost machine learning approach and a small training set”, A thesis submitted to the Faculty of Graduate Studies and Research, Ottawa-Carleton Institute for Computer Science.

[6] Yang Shao, Ross S.Lunetta, 2010, “Comparison of support vector machine, neural network, and CART algorithms for the land cover classification using limited training data points”, The U.S. Environmental Protection Agency.

[7] Krystian Mikolajczyk and Cordelia Schmid, 2005, "A performance evaluation of local descriptors", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.27, no.10, pp. 1615-1630.

[8] V. Vapnik, 1995,” The Nature of Statistical Learning Theory”, Springer, N.Y., ISBN 0-387-94559-8.

[9] Burges C., 1998, “A tutorial on support vector machines for pattern recognition”, In “Data Mining and Knowledge Discovery”. Kluwer Academic Publishers, Boston, (Volume 2).

[10] V. Vapnik, S. Golowich, and A. Smola, 1997, “Support vector method for function approximation, regression estimation, and signal processing”, Advances in Neural Information Processing Systems 9, pages 281-287.

[11] Nello Cristianini and John Shawe-Taylor, 2000, “An Introduction to Support Vector Machines and Other Kernel-based Learning Methods”, Cambridge University Press.