International Journal on Recent and Innovation Trends in Computing and Communication
Volume: 5 Issue: 7
ISSN: 2321-8169
121 – 131

Real-Time Vehicle Classification and Counting Via Low-Cost Collaborative Sensing

Ms. Shital Ramrao Tayade
M.Tech. student, Dept of Electronics & Telecommunication Engineering, Government College of Engineering, Jalgaon, Maharashtra, India

*****

1. INTRODUCTION

1.1 Introduction

Vehicle recognition is having the crucial importance in various applications including the traffic monitoring, load monitoring, number plate recognition, vehicle theft prevention, traffic violation detection, management of traffic etc. Some of the parking areas uses the vehicle recognition as the surveillance method to avoid the vehicle theft. Earlier sensors, radar, loop detector used for vehicle recognition but there installation and maintenance cost were high so to overcome these drawbacks the authors use computer vision approach. These applications generally work in video form, as the video is captured in real time and transformed to the sequence of images. These images can be processed to perform the vehicle identification or vehicle class type identification. The basic process model for vehicle detection and classification is as shown in Fig. 1.1

Firstly, the image is captured from the real environment using high resolution cameras in real environment. As the image is captured, it can have number of impurities or the background overlapping, therefore there is requirement to filter and enhance the image. Moreover to improve the image, the image transformation is applied which includes the size level, contrast level, color level adjustments. Now this filtered enhanced image is used as the main featured image for classification process. In next stage, the background is separated from the raw input image. This process of background separation comes under segmentation process. The high level segmentation is applied over the image to identify the vehicle region of interest (ROI) over the image. The vehicle area identification is done using the thresholding method which is identified and the final work is to perform the feature extraction. There are number of related methods to explore the low level features over the image and the same features includes edge detection, feature point identification etc. As the image set is transformed to the feature set, the final stage is to apply the classification algorithm. There are number of classifiers to perform the recognition or classification of image. This classifier includes Linear Discriminant Analysis (LDA) classifier, Bayesian network, SVM, neural network, etc. The same classification model can be applied over the image to perform the recognition and classification of vehicle. Section II will provide a framework of related work carried out by the researchers and authors, in section III, objectives of the research work is presented following the methodology and model for classification of vehicles into three categories in section IV. Section V explores the results and discussion obtained from the work following the concluding remarks in section VI.

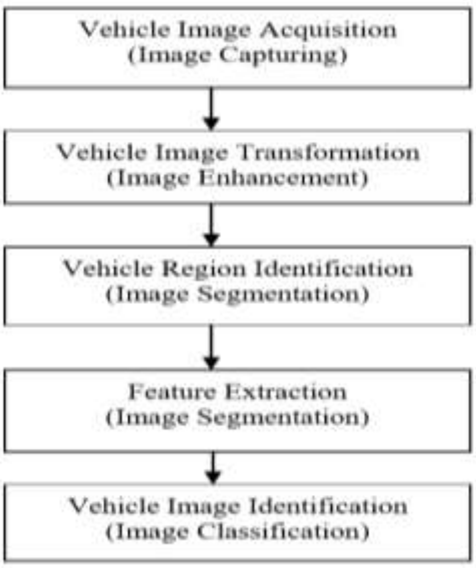


Fig. 1.1 Vehicle Classification

1.3 Objectives Of Present Work

Although authors used various classifiers such as Linear Discriminant Analysis (LDA) classifier, Bayesian network, SVM, neural network etc. to perform the recognition and identification of images but in present investigation authors presented a framework of Gaussian filter improved SVM model for real time vehicle recognition and vehicle class identification which is defined as the three stage model. The objectives of the presented research work will be achieved by using Gaussian filter and SVM approach methodology is as mentioned under:

1. To define a three stage model for vehicle class identification.
2. To apply the Gaussian transformation model to acquire the vehicle image features.
3. To apply SVM model to identify the vehicle class.
4. To improve the accuracy of recognition process.

1.4 Organisation of Report

The project deals with detection of vehicle which classify and count through low cost collaborative sensor. It has six chapters which are as follow. First chapter is Introduction of project, which includes Introduction part, Methodology and Computational Model of present investigation. Second chapter is about Literature survey which includes Introduction having related work to other available survey and Literature summery.

Third chapter is System Development which includes development stages which is related to classification and having Binary Classification, Linear Classifiers, The Perceptron Classifier, Target Classification, Linear SVM and HOG and Vehicle Counting. Chapter four includes advantages, disadvantages and application. Fifth chapter includes Result of Recognized Image, Result of Vehicle Detection, Result of Vehicle Counting. Last chapter includes Conclusion and Improving Performance.

2. LITERATURE SURVEY

2.1 Introduction

Heterogeneous sensor networks (HSNs) which can support multiple applications for HSNs with multiple types of sensors are gaining popularity in diverse fields. In order to measure the same physical quantity, different type's sensor is characterized by its own performance and energy efficiency. In most cases, low-quality sensors such as magnetic sensor are energy-efficient, but provide a very limited resolution. On the other hand, high-quality sensors such as camera sensor can provide more accurate characterization of sensed phenomenon at the cost of higher energy usage. Therefore, collaboration between the low-quality sensor and high-quality sensor can achieve tradeoff between accuracy and energy efficiency. In real applications, low-quality sensors can provide a coarse grained characterization of the sensing field or trigger an event. Then, accurate, but power-limited, sensors can be activated with measurements which are used to improve the coarser description or complete complex tasks. Therefore, Accuracy can be traded off with energy efficiency through collaboration among low-quality sensors and high-quality sensors. Normally wireless sensor networks are deeply integrated with physical environments. The uncertainties of various physical environments are ubiquitous. These uncertainties include the dynamics of monitored target, stochastic sensor noises and unpredictable environment changes. But most of existing collaboration methods use fixed parameter configuration and processing strategies, which fail to adapt to various physical uncertainties. Thus the performance of collaboration in HSNs is inevitably undermined by these uncertainties, which pose great challenges to the collaboration between low-quality sensors and high-quality sensors. Next, we give an example to illustrate the problem mentioned above. The collaboration between magnetic sensor and camera sensor is utilized to make classification for targets in this example. Magnetic sensor is used to detect the arrival of targets, and trigger the camera sensor node to classify the targets when they are detected. However, if the distance between the targets and camera sensor node is too far or the targets have not yet entered into the camera's field of vision (FOV), the targets will not be sensed. Only the targets which are in the camera sensor's sensing range can be detected by the camera sensor. If the magnetic sensor nodes detect the target, the target is still outside the camera sensor's sensing range or the sampling frequency of the camera sensor is too high and the sampling time is tooling, the camera sensor node will gather a large amount of useless images if the camera is triggered at this moment. Thus it causes some unnecessary overheads. But the precise time when the target enters into the camera's FOV, the camera sensor's optimal sampling frequency and sampling time can not be obtained directly. Therefore, we can not obtain the camera's active opportunity, sampling frequency and sampling time precisely. In this paper, active opportunity stands for the time when the target enters into the camera's sensing range, when the target is detected by the magnetic sensors in the vicinity of the camera sensor. Thus, in order to obtain the camera sensor's active opportunity, optimal sampling frequency and sampling time precisely, we need to obtain the target state in real time. As the active opportunity arrives, the camera sensor node is triggered with the optimal sampling frequency and sampling time, which can reduce the quantity of the redundant images and maintain the performance at acceptable level. The active opportunity, sampling frequency and sampling time of the high-quality sensors are key issues for energy-efficiency and performance in the process of collaboration between low-quality sensors and high-quality sensors. In this paper, we utilize the collaboration between the magnetic sensor and camera sensor to achieve the trade-off between performance and energy efficiency. We note that the magnetic sensors and camera sensors can be easily extended to other low-quality sensors and high quality sensors. The major challenges in this paper include two aspects: 1) how to design a lightweight filtering algorithm to predict the target's state for the sensor node's limited resources; 2) how to compute the active opportunity, sampling frequency and sampling time of the camera sensor dynamically. For the two challenges, we propose an adaptive collaboration (EasiAC) method, which uses magnetic sensor nodes to predict the state of the target via distributed Bayesian filtering. Then EasiAC tunes the camera sensor's active opportunity, sampling frequency and sampling time dynamically according to the estimated results from the magnetic sensors. Thus EasiAC can decrease the camera's active time, sampling frequency and sampling time compared with the traditional method that when the low-quality sensor detects the target, the camera is triggered immediately and works with the maximum sampling frequency. Finally, because boosting algorithm can combine weak classifiers to form strong classifier to get high performance and low complexity, a boosting based algorithm is proposed to make classification for camera sensors. This paper is the extension of

ISSN: 2321-8169
121 – 131

our previous work. Difference from, we compute optimal sampling time according to target state and propose a boosting based classification method in dynamic environments in this paper. The contributions of this paper are summarized as follows. 1)We provide general principles guiding the prediction of the dynamic target's state precisely in HSNs. The magnetic sensor nodes are used to predict target's state via distributed Bayesian filtering and overcome the limitations of the sensors' resources to get the target's state. 2) We derive active opportunity, sampling frequency and sampling time of the camera sensor node according to the magnetic sensor nodes' estimated results to deal with the problems of the dynamics of the monitored target. They can adapt the changes of the monitored target and achieve trade off between performance and energy-efficiency. 3) We propose a boosting based algorithm named BbTC algorithm. BbTC can achieve high performance and low complexity when it is utilized to make classification for the dynamic target.

2.2 Literature Summery

The work model used LDA classifier for improving the classification rate. Probabilistic model to perform the vehicle prediction. The feature vector analysis with Bayesian network to perform the vehicle class identification. Vehicle detection and tracking under surveillance camera processing. Vehicle video analysis based on the frame averaging and provided the dimension specific analysis to extract the image features. Neural network for classification of vehicles and estimation of traffic on road. Surveillance of the videos and analyzed the key feature analysis for real video processing Classifies the vehicle depending on length using the algorithm developed. Comparing different classification approaches the HDBN provides better recognition rate. Model built with the help of Gaussian filter and SVM which provided better performance than model built with PCA and neural network. Utilized SVM and PCA classifiers for vehicle detection. video based vehicle tracking and classification applied on real time scenario and provided the Gaussian filter, Bayesian and fuzzy SVM for classification and recognition. Video stream processing using Bayesian network approach and defined the enhanced security mechanism to identify the target vehicle.

3. SYSTEM DEVELOPMENT

3.1 Binary Classification

Given training data (xi, yi) for i=1...N, with xi ∈ Rd and yi ∈{−1,1},learn a classifier f(x) such that f(xi)(≥0yi=+1<0 yi=−1 i.e. y if(xi)>0 for a correct classification.

Fig 3.1 shows binary classification, fig. 3.2 shows Linearly separable and fig. 3.3 shows Non-linearly separable.

Fig 3.1 binary classification

Linear Separability

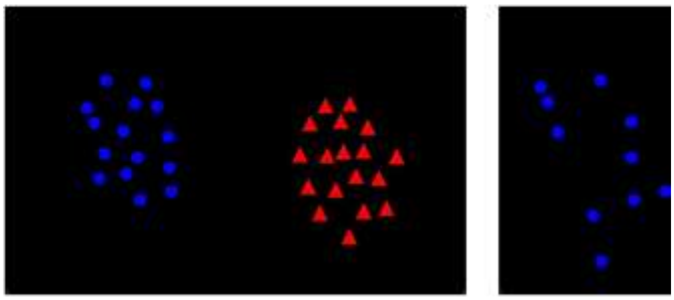


Fig. 3.2 linearly separable

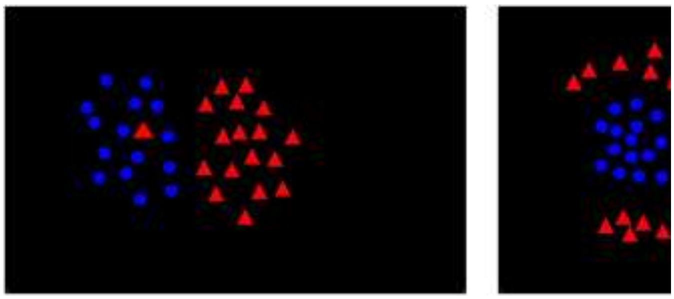


Fig. 3.3 Not linearly separable

3.2 Linear Classifiers

A linear classifier has the form

f(x)=w>x+b

Fig. 3.4 and fig. 3.5 shows Linear Classifier(a) and Linear Classifier(b) respectively.

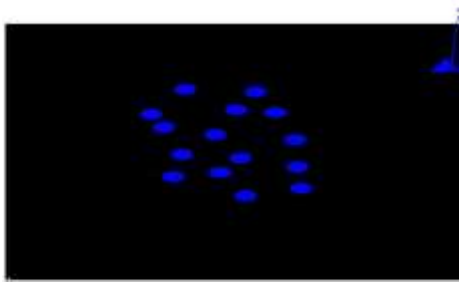


Fig. 3.4 linear classifiers(a)

- in 2D the discriminant is a line
- is the normal to the line, and b the bias
- is known as the weight vector

ISSN: 2321-8169
121 – 131

Linear classifiers

A linear classifier has the form

f(x) = w>x + b

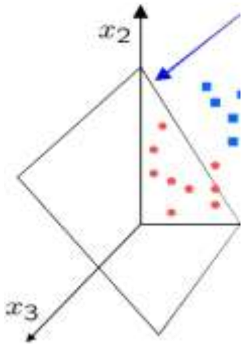


Fig. 3.5 linear classifiers (b)

•in 3D the discriminant is a plane, and in nD it is a hyperplane

For a K-NN classifier it was necessary to `carry' the training data For a linear classifier, the training data is used to learn wand then discarded Only wis needed for classifying new data.

3.3 The Perceptron Classifier

Given linearly separable data xi labelled into two category find a weight vector w such that the discriminant function

f(xi) = w>xi + b separates the categories for i = 1, .., N

• how can we find this separating hyperplane ?

3.3.1 The Perceptron Algorithm

Write classifier as f(xi) = w˜>˜xi + w0 = w>xi

where w = (w˜, w0), xi = (˜xi, 1)

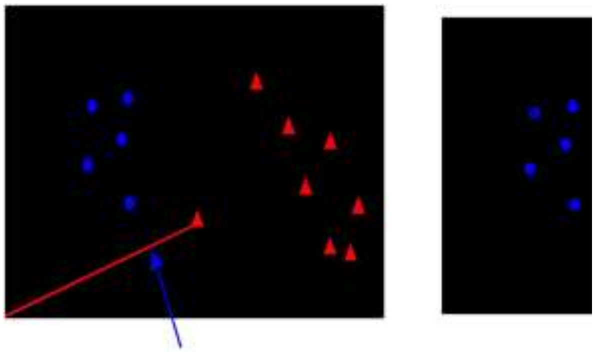
- Initialize w = 0
- Cycle though the data points { xi, yi }

• if xi is misclassified then w ← w + α sign(f(xi • Until all the data is correctly classified

For example in 2D

- Initialize w = 0
- Cycle though the data points { xi, yi }
- if xi is misclassified then w ← w + α sign(f(xi
- Until all the data is correctly classified.

Fig. 3.6 shows Perceptron Algorithm



NB after convergence w = PNi αixi **w ← w**

Fig. 3.6 The Perceptron Algorithm

- if the data is linearly separable, then the algorithm will c
- convergence can be slow …
- separating line close to training data
- we would prefer a larger margin for generalization
- maximum margin solution: most stable under perturbation.

3.4 Target Classification

In this section, we introduce target classification using a boosting algorithm based method and image samples captured by camera sensors. Next, we first discuss feature extraction and feature selection, and then we introduce BbTC algorithm inspired by boosting algorithm.

3.4.1 Feature Extraction and Selection Algorithm:

In order to make classification, we should extract feature from the integrated images captured by camera sensors through some basic image process methods. Note that we take bicycle and car as example for the target to illustrate our scheme in this paper.

3.4.2 Feature Extraction

In this paper, to reduce the computation overhead, we first make mean compression for the original image. Then we make background subtraction, first-order gradient, and thresholds binarization for the compressed image respectively. The processed images of minibus, car and bicycle are shown. To extract the target feature, we obtain the vehicle outline as shown. The features including the length, width, area, perimeters, length/width (LHR), area/perimeter (PAR) of the vehicle are computed by counting the number of pixel in the vehicle outline according to features extraction algorithm. Fig 3.8 shows Image Processing and fig. 3.9 shows Vehicle Outline.

ISSN: 2321-8169
121 – 131

Fig.3.8 Image processing

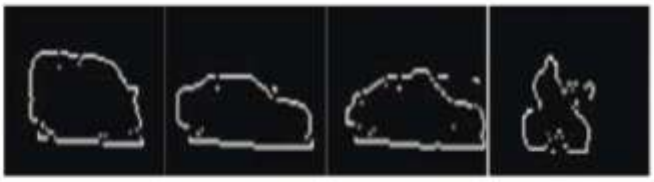


Fig.3.9 Vehicle outline

3.4.3 Feature Selection

We manually collected sample images, including 259 bicycles, 140 cars. According to the analysis of features of the integrated images, we show that LHR and PAR are more helpful features in classification as shown. Therefo re, they are used to classify the vehicles. Fig. 3.10 shows Feature Selection for classification.

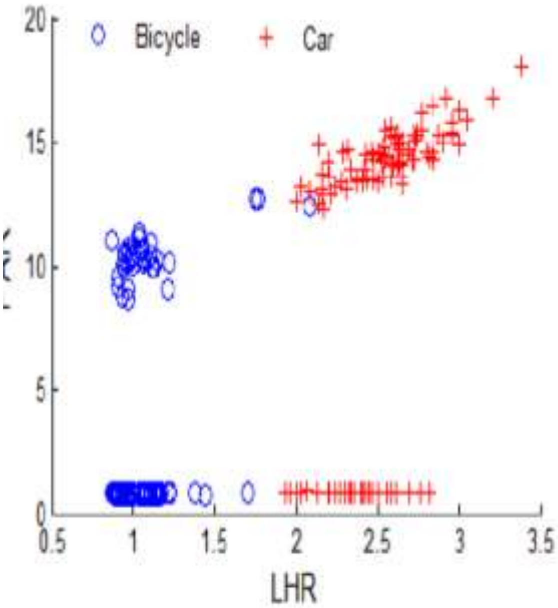


Fig. 3.10 Feature selection

The work carried out in a real time image processing model is defined for vehicle recognition and vehicle class identification. The presented devised model is based on Gaussian filter & SVM approach which is implemented in MATLAB environment and the same is applied on different vehicle datasets to obtain the analytical results. The presented work has defined hybrid model to perform the vehicle recognition and classification.

3.5 LINEAR SVM & HOG

3.5.1 Linear SVM

Linear SVM is a fast machine learning algorithm which is used for solving multiclass classification from large dataset. It get classify between vehicle and non-vehicle.

SVMs is primarily two-class classifier which has been shown to be an attractive and more systematic approach to learning linear or non-linear decision boundaries. Given set of points, which belong to either of two classes, SVM finds the hyper-plane leaving the large set possible fraction of points of the same class on the same side, while maximizing the distance of either class from the hyper-plane. This is equivalent to performing structural risk minimization to achieve good generalization.

$$(x_1, y_1)(x_2, y_2)...(x_l, y_l), \ \ x_i \in R^N, y_i \in \{-1, +1\}$$

Assuming examples from two classes find the optimal hyper-plane implies solving a constrained optimization problem using quadratic programming. The optimization criteria is the width of the margin between these classes. The discriminate hyper-plane is defined as:

$$f(x) = \sum_{i=1}^{l} y_i a_i k(x, x_i) + b$$

Where, k(x,xi) is a kernel function and the sign of f(x) indicates the membership of x. Constructing the optimal hyper-plane is equivalent to find all the nonzero ai data point. Any data point xi corresponding to a nonzero ai is a support vector of the optimal hyper-plane. Suitable kernel functions can be expressed as a dot product in some space and it satisfy the Mercer's condition. By using different kernels, SVMs implement a variety of learning machines (e.g., a sigmoidal kernel corresponding to a two-layer sigmoidal neural network while a Gaussian kernel corresponding to a radial basis function (RBF) neural network)

The Gaussian radial basis kernel is given by,

$$k(x, x_i) = \exp\left(-\frac{\| x - x_i \|^2}{2\delta^2}\right)$$

The Gaussian kernel used in this study (i.e., our experiments have shown that the Gaussian kernel outperforms other kernels in the context of our application).

3.5.2 Advanced Feature Extraction Technique

- Viola-Jones (& Ada boost).
- Optical Flow (& Motion analysis).
- Histograms of Orientation Gradients.
- Histograms of Flows.
- Scale Invariant Feature Transforms.

ISSN: 2321-8169
121 – 131

3.5.3 Histograms of Orientation Gradients

A popular features extractor which is known as HOG (histogram of oriented gradient) and it is used to extract the feature vectors of the object (i.e. vehicles). Histogram of Oriented Gradients commonly known as HOG which is one dimensional features vectors extraction. HOG is the type of appearance – based methods on large model base. This detector however used mostly in human and vehicle detection.

Objective: Object Recognition

Basic idea:

Local shape information is described by the distribution of intensity gradients or edge directions even without precise information about the location of the edges themselves.

3.5.4 Algorithm Overview

- Divide taken image into small sub-images: “cells”
- These Cells can be rectangular (R-HOG) or circular (C-HOG)
- Then Accumulate a histogram of edge orientations within that cell
- The combined histogram entries are used as the feature vector describing the object
- Then to provide better illumination invariance (lighting, shadows, etc.) normalize the cells across larger regions incorporating multiple cells: “blocks”.

3.5.5 Block Geometries

R-HOG

- Rectangular arrangement of cells
- E.g. 6x6 cells

C-HOG

- Circular arrangement of cells.

3.5.6 Why HOG?

- Capture edge or gradient structure which is very characteristic of local shape
- It is relatively invariant to local geometric and photometric transformations.
- Within the cell rotations and translations do not affect the HOG values.
- The Illumination invariance achieved through normalization.
- The spatial and orientation sampling densities can be tuned for different applications.
- For human detection Dalal and Triggs coarse spatial sampling and fine orientation sampling works best.
- For the hand gesture recognition Fernandez-Llorca and Lacey finer spatial sampling and orientation sampling is required.

3.5.7 Algorithm For Object Detection Using HOG And Linear SVM

1. Collect k numbers of vehicles that usually seen and j numbers of non-vehicles which is the background like roads, trees, lamp post, sign board that is seen on the current frame and some non-vehicle object.
2. Let k be the positive samples and j be the negative samples. In practice j < k, for more accuracy.
3. Resize it into 64*64
4. Extract HOG descriptor for both j and k.
5. Train with linear SVM on the j and k and save it into xml or yml format (say l.xml).
6. Take a frame from the video camera.
7. Perform HOG detect MultiScale parameters.
8. Apply sliding window technique; the size of the sliding window is to be fixed and equal with 64*64.
9. Extract HOG features from the window and apply linear SVM and compare with l.xml. If it is equal apply the window with the bounding box and goto step 14, if not apply step 10.
10. Skip the window and apply step 9 to the next window till it completed all the slides.
11. After completing all the windows slides apply image pyramid (optional) [applying image pyramid will decrease the speed but better accuracy result when the object is greater than the sliding window size].The image pyramid scale should be set accordingly. The smaller the image pyramid Scale the decrease the speed. It is largely depend upon the object on the scene from the camera. Hence, depend largely on the placement of the camera.
12. Apply step 8 to 11 till it is equal to the image pyramid scale.
13. Applying step 11 form overlapping of bounding box. And can be corrected by using nonmaxima suppression.
14. Set the bounding box to be a vehicle.
15. End and proceed to next frame.

ISSN: 2321-8169
121 – 131

Fig. 3.11 shows Extracting HOG features, training with Linear SVM and storing into database.

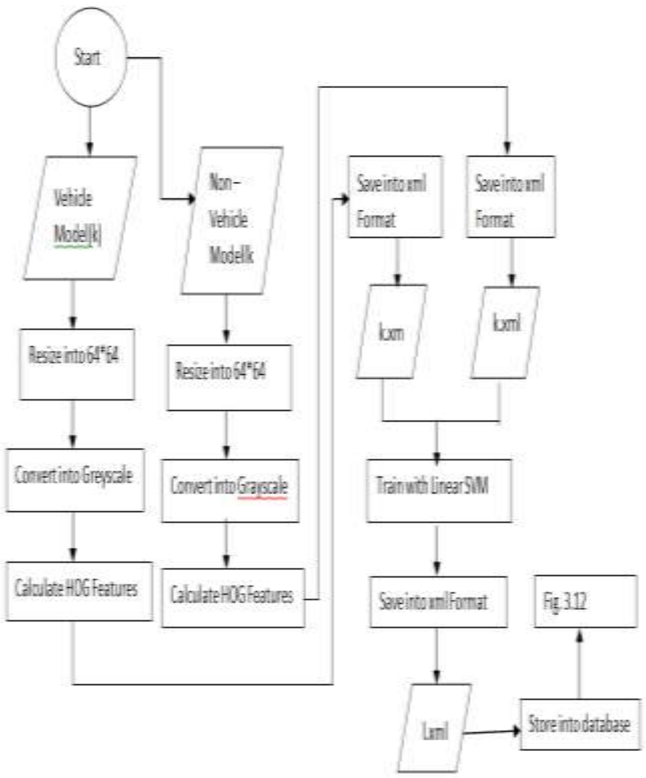


Fig. 3.11 Flowchart of Object Detection Using HOG and Linear SVM.

Fig. 3.12 shows Vehicle Detection On An Image Frame From The Video.

Fig. 3.12 Performing Vehicle detection on an image frame from the video.

3.6 Vehicle Counting

3.6.1 Edge Detection

Edge detection refers to the process of identifying and locating sharp discontinuities in an image. The discontinuities are abrupt changes in pixel intensity which characterize boundaries of objects in a scene. There are number of edge detection methods like Prewett, Sobel, canny, Robert. Out of this Canny Edge detection gives the better result. Therefore we use Canny Edge Detection Method.

3.6.2 Canny Edge Detection Algorithm

Step 1:-

In order to implement the canny edge detector algorithm, a series of steps must be followed. The first step is to filter out any noise in the original image before trying to locate and detect any edges. And because the Gaussian filter can be computed using a simple mask, it is used exclusively in the Canny algorithm.

Step 2:-

After smoothing the image and eliminating the noise, the next step is to find the edge strength by taking the gradient of the image.

Step 3:-

The direction of the edge is computed using the gradient in the x and y directions.

Step 4:-

Once the edge direction is known, the next step is to relate the edge direction to a direction that can be traced in an image.

Step 5:-

After the edge directions are known, non-maximum suppression now has to be applied. Non-maximum suppression is used to trace along the edge in the edge direction and suppress any pixel value (sets it equal to 0) that is not considered to be an edge. This will give a thin line in the output image.

Step 6:-

Finally, hysteresis is used as a means of eliminating streaking. Streaking is the breaking up of an edge contour caused by the operator output fluctuating above and below the threshold.

Edge detection of all four types was performed on Figure 3.13 and the results are shown in Figure 3.14. This figure 3.14 shows that Canny yielded the best results.. Canny yielded the best results. This was expected as Canny edge detection accounts for regions in an image. Canny yields thin lines for its edges by using non-maximal suppression.

- Visual Comparison of various edge detection Algorithms

ISSN: 2321-8169
121 – 131

Fig 3.13 Image used for edge detection analysis

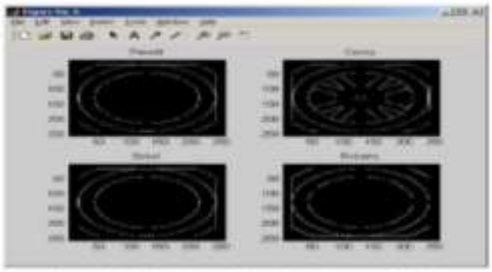


Fig 3.14 Results of Edge Detection

3.6.3 Background Subtraction:

Background subtraction is a technique in the fields of image processing and computer vision wherein an image's foreground is extracted for further processing (object recognition etc.). Generally an image's regions of interest are objects (humans, cars, text etc.) in its foreground. After the stage of image pre-processing (which may include image denoising etc.) object localisation is required which may make use of this technique. Background subtraction is a widely used approach for detecting moving objects in videos from static cameras. The rationale in the approach is that of detecting the moving objects from the difference between the current frame and a reference frame, often called “background image”, or “background model”. Background subtraction is mostly done if the image in question is a part of a video stream. The background is assumed to be the frame at time t. This difference image would only show some intensity for the pixel locations which have changed in the two frames. Though we have seemingly removed the background, this approach will only work for cases where all foreground pixels are moving and all background pixels are static. Background subtraction is a popular technique to segment out the interested objects in a frame. This technique involves subtracting an image that contains the object, with the previous background image that has no foreground objects of interest. The area of the image plane where there is a significant difference within these images indicates the pixel location of the moving objects. These objects, which are represented by groups of pixel, are then separated from the background image by using threshold technique. Figure.3.15 shows the flowchart for vehicle counting.

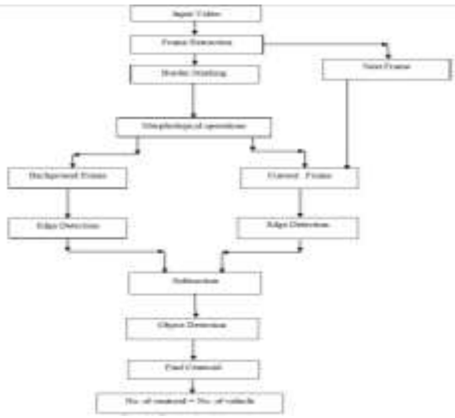


Fig. 3.15 Flowchart of Vehicle Counting

3.6.4 Centroid:

The centroid or geometric centre of a two-dimensional region is, informally, the point at which a cardboard cut-out of the region could be perfectly balanced on the tip of a pencil (assuming uniform density and a uniform gravitational field). Formally, the centroid of a plane figure or two-dimensional shape is the arithmetic mean ("average") position of all the points in the shape. The definition extends to any object in n-dimensional space: its centroid is the mean position of all the points in all of the coordinate direction.

3.6.5 Threshold Technique:

Threshold means a region marking a boundary. It is also defined as the line determining the limits of an area. Thresholding techniques are often used to segment images. The threshold image in general is defined as,

$$g(x,y) = \begin{cases} 1, & if\ f(x,y) > T \\ 0, & if\ f(x,y) \geq T \end{cases}$$

equation...3.1

3.6.6 Threshold Selection:

The key parameter in the thresholding is the choice of threshold value. Several different methods for choosing a threshold exist. User can manually choose a threshold value or a thresholding algorithm can compute a value automatically, which is known as automatic thresholding. In our work we are using the Thresholding technique on the area of the vehicle.

4. TECHNICAL ASPECTS

4.1 Advantages

It has Emergency vehicle notification systems.
It has Automatic road enforcement.
It has Variable speed limits.
It has Collision avoidance systems.
It has Dynamic traffic light sequence.

4.2 Disadvantage

There are some errors occurred during counting of vehicles.
Particular matlab software is required to run this system.

4.3 Applications

Vehicle identification or classification is one of the application which is come under real time image processing. Vehicle recognition have the significance in various applications including the traffic monitoring, load monitoring, number plate recognition, vehicle theft prevention, traffic violation detection, management of traffic etc.

The increasing traffic volume over the last decades gives high challenges on today's traffic research and planning. Detection, Counting and classification of vehicles in a video has become a potential area of research due to its numerous applications to video-based intelligent transportation systems. For most traffic surveillance systems, major stages are used to estimate desired traffic parameters, i.e., vehicle detection, Counting, tracking, and classification. Each year, motor vehicle crashes account for about thousands deaths, more than million injuries. Counting vehicles over a period of time on a busy intersection will help the concerned authority to efficiently control the duration of traffic signal on road thus reducing the level of traffic congestion during rush hours. It helps to minimize the possibilities of fraudulent activities in toll collection. It is necessary to provide better traffic system to reduce the accidents. So the main Goal of my paper is to provide better traffic surveillance.

5. RESULT AND DISCUSSION

5.1 Result

The work in a real time image processing model is defined for vehicle recognition and vehicle class identification. The presented model is based on Gaussian filter & SVM approach which is implemented in MATLAB environment and the same is applied on different vehicle datasets to obtain the analytical results. The presented work defined hybrid model to perform the vehicle recognition and classification. First, the used vehicle dataset is collected by using camera in video form which is in 3gp format and then converted to image database in jpg format which is using VLC media player or other software packages for it. Then the image pre-processing is applied to improve image features or to obtain normalize image which is extracted using thresholding technique. Accordingly the Gaussian adaptive edge analysis model is applied and based on the feature adaption model, the feature set is obtained over the training set. The training set is made by using training SVM classifier. After applying the pre-processing, extraction of ROI and obtaining feature set over test image, the test image is applied to SVM classifier for recognition and classification. The mapping of input image over the dataset to get the recognized image is done by SVM based distance analysis method.

The explained steps are applied on the vehicle dataset and the accuracy is calculated as shown in Table 5.1 to Table 5.4. Table 5.1 which shows the recognition of the dataset for the defined testing set and Fig. 5.1 shows the recognition results of vehicle obtained from presented model when image taken from dataset I and testing set is tested against the training set. This process of testing is repeated for test images and the results in Table 5.1 shows that the 90% recognition rate is obtained in the result. Similarly Table 5.2 shows the recognition obtained for vehicle recognition and class identification for non dataset image. The result gives approximately 86% recognition rate which is obtained from the present work. Table 5.3 shows that the recognition obtained for vehicle recognition and class identification for non dataset image. The result gives about 87.5% recognition obtained from the work.

Fig.5.1 shows Recognized Image has taken from Dataset 1, Fig. 5.2 Recognition Rate of Analysis of Vehicle Detection Classification, Fig. 5.3 shows the result of vehicle detection, Fig. 5.4 shows the result of vehicle counting.

5.2 Result Of Recognized Image

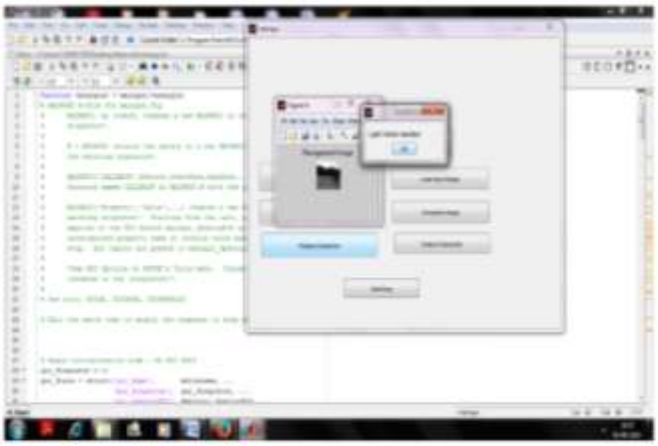


Fig.5.1 Recognized Image taken from Dataset 1

Property	Value
Dataset Size	35
Format	JPG
Resolution	1280x720
Type	Color
Captured	Real Time on Road
Size (Training)	25
Size(Testing)	10
Dataset Images	Yes
Number of Classes	3
Recognition(Successful)	9
Recognition(Unsuccessful)	1
Recognition rate(Successful)	90%
Recognition Rate (Failure)	10%
Property	Value
Dataset Size	35
Format	JPG
Resolution	1280x720
Type	Color
Captured	Real Time on Road

Table 5.1.Dataset I

Dataset Image

Property	Value
Dataset Size	30
Format	JPG
Resolution	1280x720
Type	Color
Captured	Real Time on Road
Size (Training)	35
Size(Testing)	30
Dataset Images	No
Number of Classes	3
Recognition(Successful)	26
Recognition(Unsuccessful)	4
Recognition rate(Successful)	86.66%
Recognition Rate (Failure)	13.34%

Table 5.2. Dataset II

Dataset I- Recognition of Dataset for Defined Testing Set
Dataset II- Vehicle Class Identification for Non Dataset Image

ISSN: 2321-8169
121 – 131

Property	Value
Dataset Size	50
Format	JPG
Resolution	1280x720
Type	Color
Captured	Real Time on Road
Size (Training)	50
Size(Testing)	40
Dataset Images	No
Number of Classes	3
Recognition(Successful)	35
Recognition(Unsuccessful)	5
Recognition rate(Successful)	87.5%
Recognition Rate (Failure)	12.5%

Table 5.3. Dataset III

Dataset III- Recognition for Vehicle Class Identification for Non Dataset Image.

Table 5.4 shows cumulative performance analysis of model for vehicle classification into light, medium and heavy vehicle for dataset I, dataset II and dataset III. Table IV clearly stated that for dataset I author obtain 1 false match out of these test images taken from dataset, for Dataset II author obtained 4 false match out of 30 test images taken from non–dataset image and from dataset III 5 false match out of 40 test images taken from non dataset image by providing accuracy of 90%, 86.6% and 87% respectively.

Input Images	Size (Training)	Size (Testing)	Correct match	False Match	Accuracy
Dataset I (35)	25	10	9	1	90%
Dataset II (50)	35	30	26	4	86.6%
Dataset I (50)	50	40	35	5	87.5%

Table 5.4 Performance Analysis of Devised System

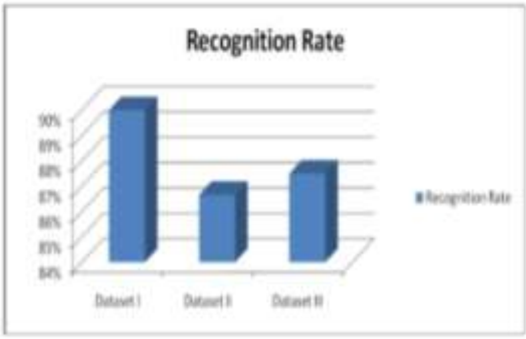


Fig. 5.2 Recognition Rate

Analysis of Vehicle Detection Classification

Fig. 5.2 shows the recognition rate obtained for different vehicle datasets is presented in Table 5.1 to Table 5.3. The first bar is showing the recognition rate obtained. All the images are dataset images and provided the 90% recognition rate. The non–dataset images set provided the result of 86.66% and 87.5% respectively which shows that the investigated work has provided the significant results.

5.3 Result Of Vehicle Detection

Fig. 5.3 Vehicle Detection

5.4 Result Of Vehicle Counting

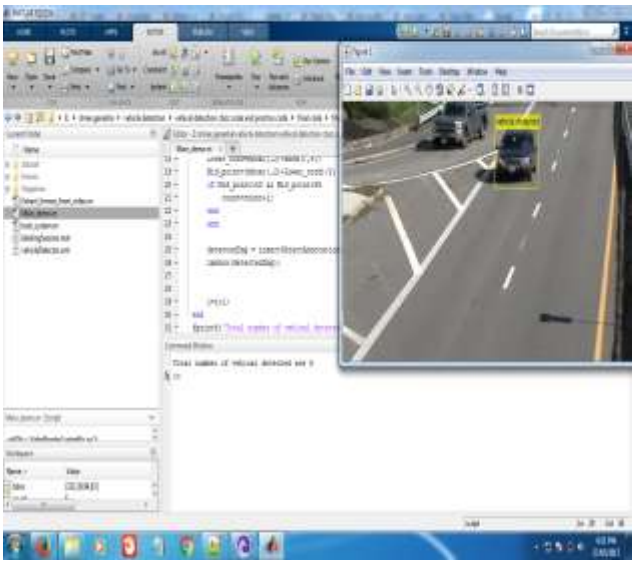


Fig. 5.4 Vehicle Counting

6. CONCLUSION

6.1 Conclusion

Real-time image processing or object processing have its significance in different areas such as traffic monitoring, currency identification, vehicle identification, hand gesture recognition, face recognition, detection of tumor etc. Vehicle Recognition drawn considerable interest and attention from several researchers and authors in past. Vehicle detection and recognition are the application of image processing but still due to large variation in vehicle shape, size, color, a reliable recognition by machine is still a challenge. In present investigation, a Gaussian filter based SVM model applied for vehicle identification and vehicle class identification. The presented work applied on real time vehicle images collected from real time source. The capturing from real time has been done in form of video files which were later on transformed to image set. According to presented model, firstly the input training set and test image has transformed to feature image set. This transformation has been done using segmented Gaussian filter and accordingly the featured edge point is obtained over the dataset. As the feature set obtained, the SVM based distance analysis model has been applied to perform the recognition and classification of vehicle. The investigated model based on Gaussian filter and SVM approaches has been implemented in MATLAB environment to perform the recognition and classification. The results show that the presented work model has provided the significant high recognition rate over 90% for different dataset and non dataset real time vehicle images.

6.2 Improving Performances

Improving performances in speed and accuracy are the main problem for real time application like video surveillance. HOG like feature extraction generally need to be trade-off between accuracy and speed. But in this technology, there should be more emphases on speed rather than accuracy, since accuracy can be improved while on training the sample, and the placement of the camera. The speed will increased by using sunlight or LIBSVM instead of Linear SVM. The GPU versions are now released on every latest version of Open CV to compute HOG features by installing NVIDIA graphic CUDA. The performances can be increased by using a Cascade of HOG by applying Ada Boosting algorithm which will increase up to 70X speedup. The speed and accuracy can be increased by handling with different scales and transferring computation from test time to training time. Another method to increase the speed performances is to avoid constructing image pyramid without resign their performances, thus enabling HOG Sliding window techniques runs in parallel. This approach, however decrease the accuracy rate.

6.3 Future Work

The focus on to implement the controller using DSP as it can avoid heavy investment in industrial control computer while obtaining improved computational power and optimized system structure. The hardware implementation would enable the project to be used in real-time practical conditions. In addition, I propose a system to identify the vehicles as they passes, by giving preference to emergency vehicles and assisting in surveillance on a large scale.

REFERENCES

[1] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2):121–167, 1998.

[2] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[3] S. Gupte, O. Masoud, R. Martin, and N. Papanikolopoulos. Detection and classification of vehicles. IEEE Transactions on Intelligent Transportation Systems, 3(1):37–47, March 2002.

[4] S. S. Keerthi and C.-J. Lin. Asymptotic behaviors of support vector machines with gaussian kernel. Neural Computation, 15(7):1667–1689, 2003.

[5] M. Kirby and L. Sirovich. Application of the karhunenloeve procedure for the characterization of human faces. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(1):103–108, 1990.

[6] K. Robert, N. Gheissari, and R. Hartley. Video Processing for Traffic Surveillance, 2006. Technical Report at the National ICT of Australia, http://www.nicta.com.au/.

[7] M. Turkand A. Pentland. Eigen faces for recognition. Journal of Cognitive Neuroscience, 3(1):71–86, 1991.

[8] S. Zehang, G. Bebis, and R. Miller. On-road vehicle detection using evolutionary gabor filter optimization. IEEE Transactions on Intelligent Transportation Systems, 6(2):125–137, 2005.

[9] C. Zhang, X. Chen, and W. B. Chen. A pca-based vehicle classification framework. IEEE International Conference on Data Engineering, pages 17–17, 2006.