International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 07 | July 2020 www.irjet.net p-ISSN: 2395-0072

TRAFFIC DENSITY ESTIMATION BY COUNTING VEHICLES USING AGGREGATE CHANNEL FEATURES

Joel Shaji[1], Anisha Mohammed[2]

[1]*Scholar, Dept. of Electronics and Communication Engineering, College of Engineering Kallooppara, Kerala, India*
[2]*Asst. Professor, Dept. of Electronics and Communication Engineering, College of Engineering Kallooppara, Kerala, India*

Abstract - *Vehicle detection methods have become useful for many artificial intelligent systems such as self-driving cars or autonomous vehicles and traffic congestion control methods. The most popular Google car are based on the result of vehicle detection methods from the driver perspective. The vehicle detection methods have huge amount of benefits in the modern world. Vehicle detection methods can be classified on the basis of basic methods and computer vision based methods. The primitive methods include the basic image processing techniques to detect the vehicle from its background. A comparative study along with a new method using aggregate channel feature is used to determine the density estimation of vehicle. The results can used for setting the flow of traffic lights. For this comparative study, MATLAB have been used. Every method from image preprocessing to object detection were utilized through image processing and computer vision*.

Key Words: *ROI, Threshold, ACF, BLOBs, Computer Vision, Morphology*

1. INTRODUCTION

Vehicle congestion is becoming more popular in developed and developing countries as the number of vehicles in the road increases day by day. Proper congestion control systems can be introduced to limit the congestion to an extent, one way of doing it is based on introducing manual systems such as traffic police to guide the moment of vehicles in traffic. There are limitation for such kind of systems as the amount of congestion increases daily. If proper congestion controlling systems based on the latest image processing techniques and computer vision based techniques that rely on the convolutional neural networks are made along to assist in the manual method the traffic congestion control will become more efficient.

The novel method that is introduced here is based on an evolved form of Viola Jones object detection known as (ACF) Aggregate Channel Features .The method is more efficient than the Viola Jones method. The novel method only needs small amount of data set also the amount of false positives will be 1000 fold less than the Viola Jones method for object detection.

The primitive methods as described earlier uses the normal image processing techniques such as threshold, morphology and Edge detection method. The commonly used method for object detection includes the BLOB analysis which identifies the region of interest based on certain parameters. When including BLOB analysis with the primitive methods that were used the vehicles can be detected. The drawbacks of the primitive methods used is that the image for detection should contain the foreground (vehicle) and background that can be well distinguished and these methods cannot be used for high level of traffic congestion.

The modern computer vision based method such as the RCNN or the Fast RCNN methods seems to be more reliable in identifying object in an image. The novel method used can be incorporated with the Fast RCNN method to make the object detection real time and accurate. Further computer vision methods such as the newly introduced method YOLO (You Only Look Once) can be introduced along with the novel method. The YOLO method is a multiple object detection method which helps in identifying one or more objects in the image.

2. LITERATURE SURVEY

The simplest method for vehicle detection was introduced by [1] which describes the basic methods such as threshold and morphology that can be used to detect vehicles from a well distinguished image. It is one of the simplest method for calculating the density of vehicles in an image. The method have less accuracy in distant frame videos. Real time density estimate method [2] which selects the region of interest by background subtraction

for detecting the vehicle from the image. Gaussian mixture model is used for ROI based method, the BLOBs analysis is incorporated along with background subtraction. Gaussian mixture method with the Kalman filter [3] was used for multiple vehicle detection based on the shadow removal technique. Another approach based on the Kalman filter tracking [4] is based on the color feature of the vehicle. For more accuracy in finding the vehicle a method based on Canny edge detection was introduced [5] the paper implements Canny edge detection along with the basic BLOBs analysis method for finding the vehicle. A similar approach was introduced in [6] which uses the same principle of edge detection but uses Gabor filter for the precision in detection. [7] Describes the Linear Quadratic Estimation method which uses mask and morphological operations to detect the vehicle.

The computer vision based method [8] which used the Viola Jones algorithm for object detection. The method can also be described as cascade object detector it has more accuracy in finding the vehicles when compared to the previous methods described. The drawback of this method is that the aspect ratio of object that are similar are only identified by the detector. The aggregate channel feature method [9] similar to the cascade object detector was introduced for detecting the face but unlike the cascade object detector it uses the sliding window approach and has more accuracy in finding multiple objects in an image. Fast RCNN method [10] uses the region based convolutional network for the detection of objects, the method is an evolved form of the Viola Jones method incorporating the neural network.

3. IMAGE ACQUISITION

Since real time video is used for the processing the images are obtained from video frames converted in specific time period. 24-30 frames are there in a second of video. The frames which are close to each other are almost similar therefore we need to remove the frames similar to each other. For removing the similar video frames, the frames are obtained in an interval of $5 \times i\,th$ frame per second.

$$No\ of\ frames\ selected = \sum_{i=1}^{n} 5i \quad (1)$$

If the frame rate of the video is 24 frames per second then n = 8 and if it is 30 frames per second then n = 10. For the proposed project 360 frames were obtained in a minute. For videos with more length the number of images obtained will increase. The next phase is image preprocessing for training the detector the images should be in specified dimensions and this is done using MATLAB toolbox.

4. IMAGE PREPROCESSING

For training, testing and validating, the images obtained are shuffled into training data, test data and validating data. For the proposed system three videos with different backgrounds were combined and were processed. The three minute video produced almost 1,080 video frames and were split into three datasets. Each dataset contain 360 frames for testing, training and validating. But most of the images are of a certain time frame of similar objects the number of training images used should be further reduced.

Once images are shuffled the next step is to resize the images to a specified size so that the training time can be reduced. When reducing the size of images it shouldn't be too small, so that it produces more false positives while training. For the proposed method the size of resized images were 320x640x3. This is because we will be dividing the image into 8x8 and 16x16 patches to extract the features. For the extraction of features we will be using the fast pyramid approach based on [11] which uses the features from different channels of color space. We need to convert the RGB color space to CIE XYZ and then using these parameters we can find the CIE LUV color space. If the chromaticity coordinates of an RGB system (x_r, y_r), (x_g, y_g) and (x_b, y_b) and its reference white (X_w, Y_w, Z_w), here is the method to compute the 3 × 3 matrix for converting RGB to XYZ:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = [M] \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2)$$

$$[\mathrm{M}] = \begin{bmatrix} S_r X_r & S_g X_g & S_b X_b \\ S_r Y_r & S_g Y_g & S_b Y_b \\ S_r Z_r & S_g Z_g & S_b Z_b \end{bmatrix} \quad (3)$$

$$X_r = x_r / y_r \quad (4)$$

$$Y_r = 1 \quad (5)$$

$$Z_r = (1 - x_r - y_r)/y_r \quad (6)$$

$$X_g = x_g / y_g \quad (7)$$

$$Y_g = 1 \quad (8)$$

$$Z_g = (1 - x_g - y_g)/y_g \quad (9)$$

$$X_b = x_b / y_b \quad (10)$$

$$Y_b = 1 \quad (11)$$

$$Z_b = (1 - x_b - y_b)/y_b \quad (12)$$

Once the RGB color space is converted to XYZ color space using the above parameters we can find the LUV color space using the following equation:

$$L = \begin{cases} 116\sqrt[3]{y_r} - 16 & if\ y_r > \epsilon \\ \kappa y_r & \text{otherwise} \end{cases} \quad (13)$$

$$u = 13L(u' - u'r) \quad (14)$$

$$v = 13L(v' - v'r) \quad (15)$$

Where,

$$y_r = \frac{Y}{Y_r} \quad (16)$$

$$u' = \frac{4X}{X+15Y+3Z} \quad (17)$$

$$v' = \frac{9Y}{X+15Y+3Z} \quad (18)$$

$$u'r = \frac{4Xr}{Xr+15Yr+3Zr} \quad (19)$$

$$v'r = \frac{9Yr}{Xr+15Yr+3Zr} \quad (20)$$

$$\epsilon = \begin{cases} 0.008856\ Actual\ CIE\ standard \\ \frac{216}{24389}\ Intent\ of\ the\ CIE\ standard \end{cases}$$

$$\kappa = \begin{cases} 903.3\ Actual\ CIE\ Standard \\ \frac{24389}{27}\ Intent\ of\ the\ CIE\ standard \end{cases}$$

5. METHODOLOGY

Now we have all the sources to make a vehicle detector. The next step is to extract all the features from the channels found previously. Once the feature pyramid is formed using the feature descriptor using the multiple channels which were formed from RGB color space we need to train the detector using these features. There are many methods that can be used as a feature measure like HOG, Haar, SIFT and SURF, but the method that we are using for this detector is HOG descriptor. Since we need to detect the vehicles which have similar shape and size from the image the HOG descriptor is best suited to extract features based on structure and shape. As in normal cases we need to find the directional gradient G_x and G_y. Then the angle $\emptyset$ or orientation using the Pythagoras theorem using the Histogram of Oriented Gradient method. As the name suggests we will make a histogram from the orientation bins and directional gradients.

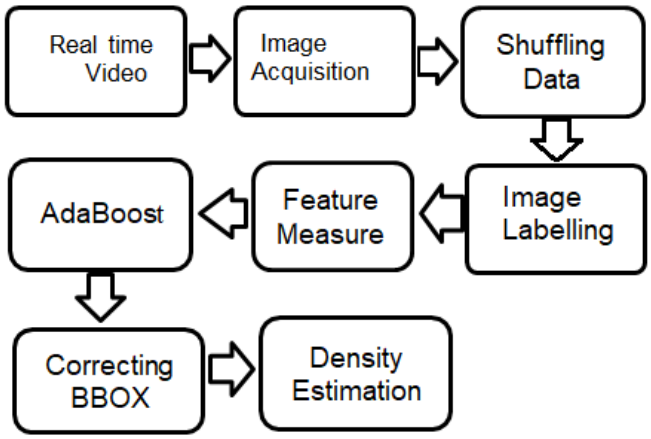

Fig -1 Block Diagram of proposed method.

Detecting objects of different scales are difficult so as proposed by [11] The ACFs have properties of extracting unique features from the image. For extract the features the input image will be created as a multiple resolution pyramid. If an input image is given we compute several channels $C = \Omega(I)$, it sums every patches of pixels in image C, and the algorithm smooth the resulting lower resolution channels. The features obtained will be simple extracted pixel based features as described in the previous section. [12] Describes different channels from which features can be extracted that are useful for vehicle detection the same channels are used here for vehicle detection. As described the histogram of oriented gradients is included with it. Normalized gradient magnitude, histogram of oriented gradients (6 channels) and LUV color channels are the channels used.

5.1 Adaptive Boosting

For vehicle detection AdaBoost algorithm is used to create a strong classifier. The Adaptive Boosting consist of stages were each stage is a group of weak learners. Each of these stages are trained using a technique called boosting. Boosting helps in creating a highly accurate classifier by taking a weighted sum of decisions made by the weak learners. The steps for the generalized adaptive boosting algorithm is as follows:

Step 1: Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where y_i = -1, 1 for negative and positive examples.

Step 2: Initialize weights w_i = 1/2M, 1/2L for y_i = -1, 1 respectively, where M and L are the number of negatives and positives respectively.

Step 3: For t = 1 to T

1) Normalize the weights, so that $\sum_{i=1}^{n} w_i = 1$
2) Choose the classifier, h_j, with the lowest error ε_j: $\varepsilon_j = \sum i\ w_i\ |h_j(x_i) - y_i\ |$
3) Update the weights for each example:

$$w_i = \begin{cases} w_i\beta_t \ example\ x_i\ is\ classified\ correctly \\ w_i\ otherwise \end{cases}$$

Where $\beta_t = \varepsilon_t/(1 - \varepsilon_t)$ and $\alpha_t = -\log \beta_t$

4) The final strong classifier is:

$$h(x) = \begin{cases} 1 \ \sum_{t=1:T} \propto_t h_t \geq \lambda \sum_{t=1:T} \propto_t \ where\ \lambda = 1/2 \\ 0\ otherwise \end{cases}$$

Pseudo code for the Adaboost algorithm is given above. It can be defined for the proposed method. First we need to define the learning data N with images that contain vehicle as M and the images that does not contain vehicle as L each item x_i of $(x_1, y_1) \ldots\ldots (x_n, y_n) \in y_{i\{-1,1\}}$ where y_i is the class to which x_i is defined. The next step is to initialize sample weights to create the first weak learners. Third step is the training stage were we need to find the error rate between predicted values and the previously available values. The error rate can be defined as the sum of learning data values which were mistakenly classified multiplied with the weight values. The weights needs to be updated until we get a weak classifier with minimum error rate. The next step is to create a strong classifier as the sum of weak classifiers. Using a sliding window approach the detector is made to find whether the vehicle is found in the region or not. A bounding box is made around the region of interest if the vehicle is found,

5.2 Correcting Bounding Box

Since our proposed method is for calculating the number of vehicles in an image and use the result for calculating the density we need bounding boxes over the region of interest that was found by the detector. Now there will be multiple number of bounding boxes over the region of interest which will be over lapped with each other. We need to remove these overlapped bounding boxes and only need a single bounding box over the region of interest. The overlapped boxes are found using the criteria:

$$\min overlapping = \frac{area(A \cap B)}{\min(area(A), area(B)}$$

(21)

After the minimum overlapping is found the bounding boxes with the least confidence can be removed from the region of interest. The algorithm check for overlapping and if found the next step is to check for the score or the confidence score of the bounding box. If the score of box is higher than the score of the other then the box with highest score will be maintained and the other will be removed. The process continues to check for the next minimum overlap until there will be no more overlaps present.

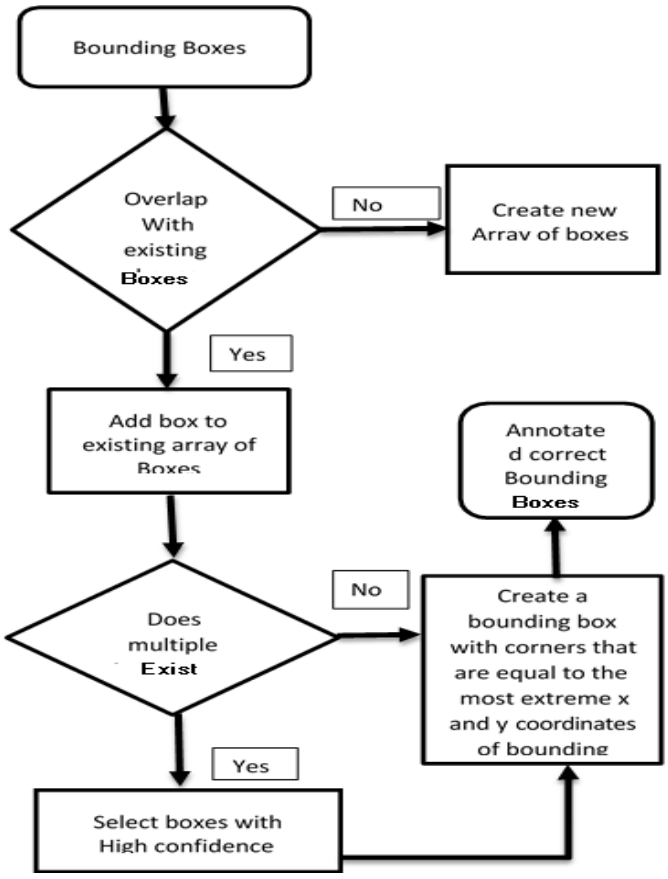

Fig -2 Block Diagram for correcting bounding box.

The bounding boxes will be selected according to the above mentioned block diagram.

4.3 Density Estimation

Once the bounding boxes are created the density of the vehicles need to be calculated this is done by the counting of bounding boxes and dividing it with the maximum density of vehicle that could occur.

$$Density\ of\ vehicles = \frac{vi}{\max(vi)}$$

(22) Were, Vi is the total number of bounding boxes with ROI that can occur. (Maximum number of vehicles that can accommodate in the area).

Once the density is calculated we can use the data given in the table below to set the traffic light flow. The table given (Table 1) is only an assumption that the data can be used to create further algorithms. The given data only shows how the timing of the green signal light can be controlled according to the corresponding density estimates during peak hours and normal hours of traffic.

Density (%)	Time(Sec)	
	Peak Hour	Non-Peak Hour
0-15	60 s RED	50 s RED
15-25	30 s GREEN	20 s GREEN
25-45	60 s GREEN	50 s GREEN
45-60	70 s GREEN	60 s GREEN
60-75	80 s GREEN	70 s GREEN
75-85	90 s GREEN	
>85	120 s GREEN	

Table -1 Setting of traffic light on density.

6. RESULTS AND DISCUSSION

To validate the proposed method, experimental images were obtained from video camera in real-world road conditions at various times. The images obtained for the experiment were 1080 × 920 pixels image. The experiment was done on a PC with Windows 10 OS (3.3 GHz Dual Core, 4 GB RAM, Single CPU) and MATLAB 2018 a. To reduce the processing time, the size of the input image was reduced to 480 × 720 pixels by the bilinear interpolation method. In the proposed method, the number of positive vehicle images used for training of the ACF detector was 360 and non-vehicle images were selected for the remaining area that excluded the vehicle regions. The average size of the training vehicle regions for the vehicle detection was 31 × 34, the repetition period (T) of the ACF detector algorithm is set to 4, the number of training samples in the non-vehicle (negative) for each learning step was 4 (S: 4), and the maximum number of weak classifiers was set to 744.

Fig -4 Sample images used for training.

Before going to the results of the proposed method, we need to discuss why this method was proposed on other simplest methods which are available at present. The first method that was implemented was using the region of interest based method which highlights the vehicle from its background using the difference between the reference image and the target image as shown in figure (3). Then uses bounding boxes to calculate the number of vehicles.

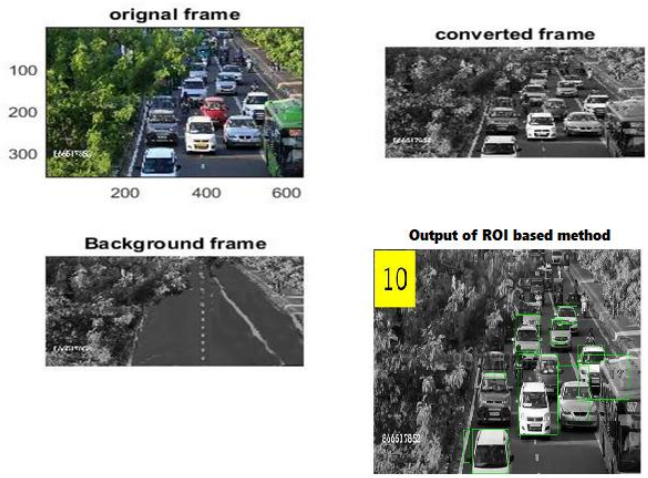

Fig -3 Result of using first method.

The result was not even close to the actual number of vehicles in the image. The bounding boxes will consider the closely placed objects as a single object. Therefore the result will not be accurate. This method can be used for implementing simple density estimation tasks were the background and foreground images can be well distinguished.

The next method that was studied was based on basic image processing techniques like the morphological

operations and threshold as in figure (5). The result was calculated using BLOBs analysis. Like the previous method the accuracy is not much, since the background and foreground images are not well distinguished.

The third method was based on Canny edge detection and the image filling method to create a structure and then using the BLOBs analysis the number of vehicle is calculated as in figure (7). The density is estimated as the total number of vehicles divided by the area of the image. When compared to the previous two methods the edge detection method gave better result in detecting the vehicles.

Now the cascade detector uses the same principle of finding the gradient as feature like the edge detector. A cascade vehicle detector was implemented to find its performance on a single class 'car' as shown in figure (8).

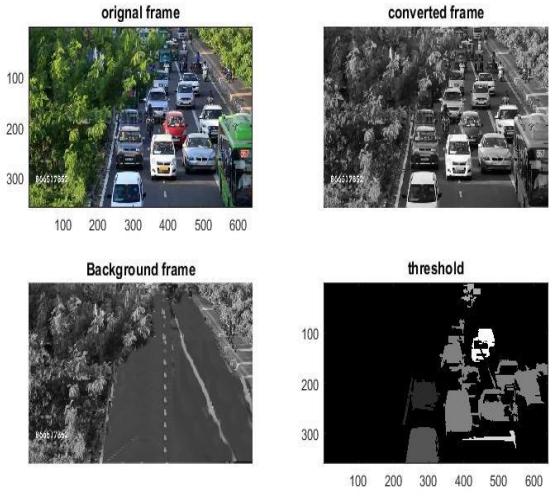

Fig -5 Result of method 2.

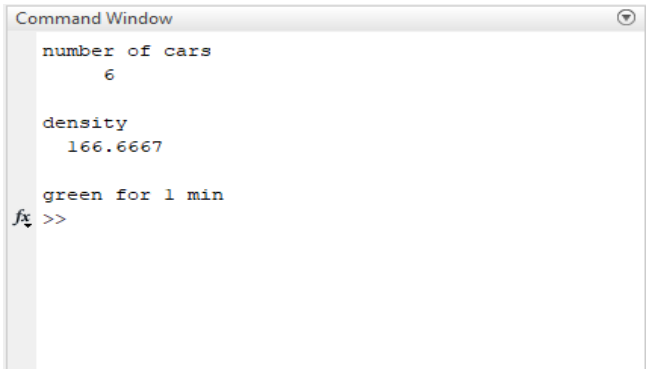

Fig -6 Density estimation result of method 2.

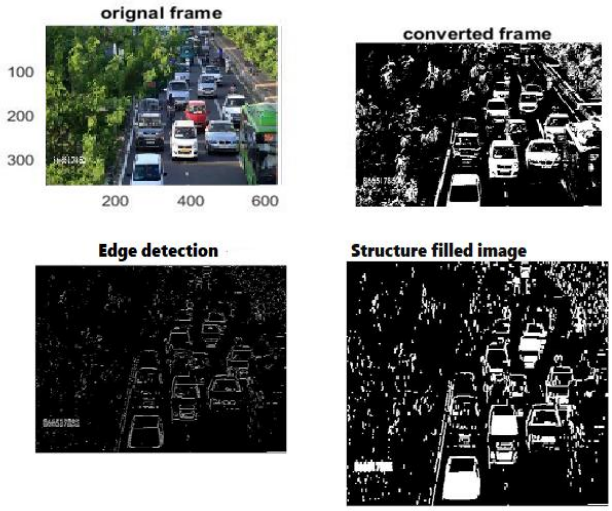

Fig -7 Result of method 3.

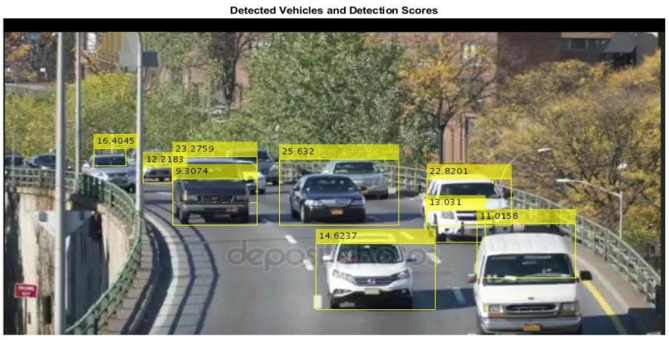

Fig -8 Result of Cascade Detector.

Finally the result for the proposed method is shown in figure. Three classes of vehicles class 1 'car', class 2 'Auto R' and class 3 as 'Bikes' were detected using the ACF based detector. For this there ACF detectors were formed for each class of vehicle and a single input image was given to the three detectors which gives results of three classes in the image.

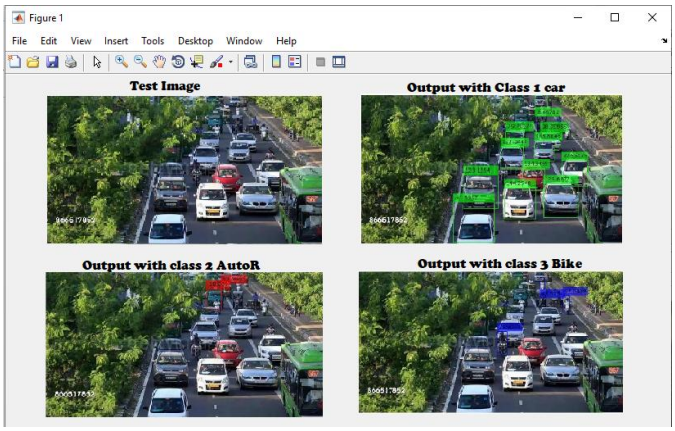

Fig -9 Result of the proposed method.

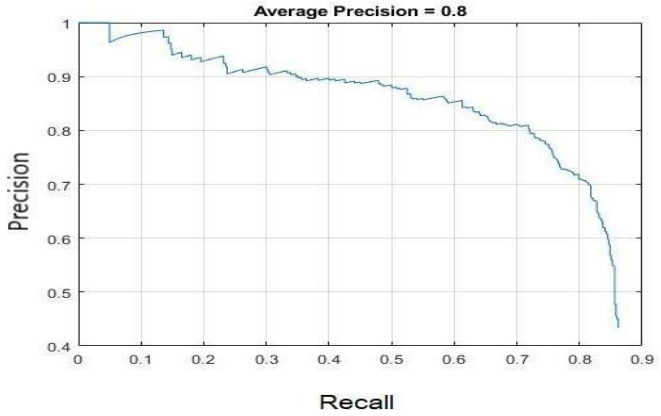

Fig -10 Average precision calculated using precision recall curve.

Table -2 Result of first two test images on proposed method

Test Image/Detected Numbers	No of vehicle class 1 Car	No of Vehicle class 2 Auto R	No of vehicle class 3 Bike	Total Number of vehicles
Test Image 1	12	2	4	18
Detected number of vehicle	11	2	3	16
Test Image 2	8	4	5	17
Detected number of vehicle	8	3	4	15

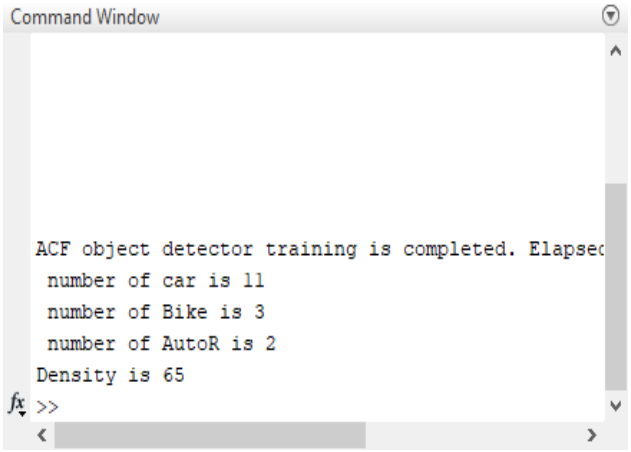

Fig -11 Density estimation result of proposed method

The accuracy of the proposed method was calculated using the average precision from the precision, recall curve. Where precision is the rate of accurate detection of the vehicle and recall is the rate of deviation in the detection area.

The average precision was calculated to be 87%. This was calculated from the average of the three detectors that were used to detect the three classes of vehicle. Accuracy shows how often the vehicle detector provides correct results.

In case of the cascade object detector, the detector was set to a training size of [32 , 38] and almost all the positive samples (825– 859) were used for training. The number of negative samples were set between the limits (1000 – 1500). The false alarm rate was set as 0.1, the sampling factor to 2 and the learning cycle to six. From the experimental result if the learning cycle is increased, the result will be much better but for this we need to feed the trainer with more images.

Table- 3 Average precision calculated at different time

Parameter\Measures	Average Precision	Error rate (αt)	Training Time
T = 2, S = 2	0.7847	0.4345	283.57
T = 2, S = 4	0.5877	0.6366	432.75
T = 4, S = 2	0.8593	0.4637	592.87
T = 4, S = 4	0.8767	0.3017	680.17

7. CONCLUSIONS

Real-time vehicle detector and vehicle density estimator is proposed. The system uses an approach based on aggregate channel features. The proposed method estimates the class of vehicle from the video camera installed at different traffic junctions. In the proposed method, the ACF-based AdaBoost algorithm was used to detect the vehicle region and estimate the density of the vehicles. In the experiments on various road environments, the accuracy of the vehicle detection was estimated to be 87.5%, and the time required for the processing was 0.76 s per frame.

Also a comparison between the cascade detector and the proposed method was validated the proposed method gives an accuracy of 4 percent and also different classes of vehicles can be detected using a single detector. The resultant counting of vehicles were used to estimate the density of the vehicles during peak and non- peak traffic times. Hence the time for the traffic signals can be implemented using the method, also this detection method

can be used with the RCNN (Region based convolutional neural network) to detect and track vehicles for A.I self-driving vehicles.

The drawback of this method is that the number of datasets that can be used for training is less, since the frames are selected from the videos for training. In the experimental road environment, there was a problem of misdetection when the driving vehicles overlapped with the shadows of trees, traffic signs, streetlights or were represented in a color similar to the background. Also the problem of detecting different class of object separately can be solved using further research. Therefore, future research will focus on improving the accuracy of vehicle detection and reducing the processing time

REFERENCES

[1] Singh, Abhijeet, Abhijeet Kumar, and R. H. Goudar. "Online traffic density estimation and vehicle classification management system." Indian Journal of Science and Technology 7.4 (2014): 508.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[2] Abbas, Naeem, Muhammad Tayyab, and M. Tahir Qadri. "Real time traffic density count using image processing." International Journal of Computer Applications 83.9 (2013): 16-19.

[3] Jin, Sheng, et al. "Short-term traffic safety forecasting using Gaussian mixture model and Kalman filter." Journal of Zhejiang University SCIENCE A 14.4 (2013): 231-243.

[4] Xie, Lei, et al. "Real-time vehicles tracking based on Kalman filter in a video-based ITS." Proceedings. 2005 International Conference on Communications, Circuits and Systems, 2005.. Vol. 2. IEEE, 2005.

[5] Balu, Shibin, and C. Priyadharsini. "Smart Traffic Congestion Control System." 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC). IEEE, 2019.

[6] Sun, Zehang, George Bebis, and Ronald Miller. "On-road vehicle detection using evolutionary Gabor filter optimization." IEEE Transactions on Intelligent Transportation Systems 6.2 (2005): 125-137.

[7] Chauhan, Naresh Singh, et al. "Vehicle detection, tracking and counting using linear quadratic estimation technique." 2018 2nd International Conference on Inventive Systems and Control (ICISC). IEEE, 2018.

[8] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001. Vol. 1. IEEE, 2001.

[9] Alionte, Elena, and Corneliu Lazar. "A practical implementation of face detection by using Matlab cascade object detector." 2015 19th International Conference on System Theory, Control and Computing (ICSTCC). IEEE, 2015.

[10] Girshick, Ross, et al. "Region-based convolutional networks for accurate object detection and segmentation." IEEE transactions on pattern analysis and machine intelligence 38.1 (2015): 142-158.

[11] Dollár, Piotr, et al. "Fast feature pyramids for object detection." IEEE transactions on pattern analysis and machine intelligence 36.8 (2014): 1532-1545.

[12] Arunmozhi, Ashwin, and Jungme Park. "Comparison of HOG, LBP and Haar-like features for on-road vehicle detection." 2018 IEEE International Conference on Electro/Information Technology (EIT). IEEE, 2018.