Real Time Traffic Density Measurement Using Computer Vision and Dynamic Traffic Control

Supervisor: Dr. Jia Uddin

Md. Fahim Chowdhury	14101221
Md Ryad Ahmed Biplob	14101182

Department of Computer Science and Engineering,

BRAC University

Submitted on: 17th April, 2018

DECLARATION

We, hereby declare that the thesis titled **"Real time traffic density measurement using computer vision and dynamic traffic control"** is based on the results found by ourselves. Materials of work found by other researcher are mentioned by reference. This Thesis, neither in whole or in part, has been previously submitted for any degree.

Signature of Supervisor	Signature of Author
Dr. Jia Uddin	Md. Fahim Chowdhury
	Md Ryad Ahmed Biplob

ACKNOWLEDGEMENTS

We would like to thank almighty Allah, the most merciful, beneficent and most gracious, who always guided us to the right path and helped us complete our thesis. There are many people we would like to acknowledge, for their support during the journey of preparing for our thesis.

We thank our thesis supervisor, Dr. Jia Uddin for his constant support in completion of our thesis. His guidance and encouragement for work has helped us learn new things as well as gave us extraordinary experiences throughout the work. We thank all the faculty staffs of Computer Science and Engineering department, BRAC University for guiding us in the whole study period at BRAC University.

We express our sincere gratefulness to our beloved parents, for the overwhelming love and care they bestow upon us. Our deep love and thanks are due to our brothers and sisters. Finally, we would like to thank our friends who helped us directly or indirectly to complete our thesis.

ABSTRACT

In recent times, traffic jam has become a common problem in the major cities all over the world. Our capital city Dhaka is no exception. Numbers of people are being victim of traffic jam each day. The main causes of such situation are more cars on the street, poor traffic management and lack of proper infrastructure. In this paper, we propose a dynamic traffic control system by measuring the traffic density at the intersections by real time video feeds and image processing. We used MOG algorithm for background subtraction method and for foreground detection to keep the count of the cars in each lane. The traffic lights at the intersections will change dynamically according to the conditions of traffic that will be detected from the video feeds.

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 General

Traffic congestion causes chaos on the road and it makes it difficult for the commuters to travel. Many big cities are facing the problem. A country's economy can be affected immensely due to traffic jam. There are many reasons behind this problem among which the common reasons are poor traffic management, cars changing lane, unplanned stoppage etc.

In recent days, the traffic condition in Dhaka has become chaotic. Huge population of the city are commuting each day using public or private transports to reach their destination. Unfortunately, the road facilities in Dhaka are not good enough to keep up with the huge transports hitting the road every day. Dhaka's infrastructure does not match the scale of its population. Just 7% of the city is covered by roads. There are 650 major intersections with reportedly 60 traffic lights, many of which don't work [1]. The traffic congestion is at peak during the rush hours. In most of the traffic intersections, traffic polices are seen to fight the situation manually. As a result, a road is given priority over another without knowing that the other roads may have already been congested and need to be cleared.

With flourishing growth of technologies, researchers are coming up with many solutions to fight traffic congestion. Requirement to come up with a solution is to detect and count the number of vehicles on the road. It can be done using loop detector, infrared detector, radar detector or video-based solution [2].Video-based solutions are more advantageous and tend to give more accurate solutions. Our main focus would be detecting vehicles and measuring the density of each side of an intersection using computer vision and develop an algorithm to dynamically control traffic lights.

1.2 Motivation

Considering traffic jam of Dhaka, there is no doubt that the traffic rules and infrastructures are lacking the standard. Even in the era of technology there is hardly any technology implemented to reduce traffic congestion. Most traffic lights at the signals are either not working or are not being followed by the drivers. Traffic lights are changed based on the static time that has been set, not considering which lane/road is more congested and needs to be cleared.

Implementation of computer vision technology can help mitigate the problem. By using computer vision we can detect vehicles, count vehicles and monitor any unwanted situation on roads. One way to detect a real-time moving vehicle is background subtraction method i.e. the difference between the current frame and background model. The static background is identified at first. Then it is removed from the current frame that is being detected to get the moving object. Thus it will help to calculate the density at each lane.

1.3 Thesis Orientation

The rest of the thesis is organized as follows:

- Chapter 2 describes the related works that were done before
- Chapter 3 describes the traffic detection algorithm and dynamic traffic control algorithm
- Chapter 4 is about the result
- Chapter 5 concludes the thesis and describes any work to be done in future

CHAPTER 2

LITERATURE REVIEW

In this chapter we discuss the approaches that were taken before and are relevant to our work.

2.1 Vehicle Detection

There are numbers of method for detecting vehicle. In paper [3], vehicles were detected based on rear lamp and license plate with dedicated traffic surveillance camera. Image frames were extracted from a video footage and the car parts are localized to model the vehicle. The parts were then combined using Markov Random Field (MRF). Then using Kalman filters, the vehicles were detected.

In paper [4], a new type of object detection technique was used known as hybrid image template. It is the combination of multiple image patches which include features like texture, color and flatness. The templates were formed by training the system with the patches of an image. Then a three staged SUM-MAX procedure was applied to successfully detect vehicles with local deformations in location and orientation.

A methodology was proposed in paper [5] that uses Raspberry pi and usb camera to capture traffic scene. The Raspberry pi is set with static IP address and is remotely accessible within a private network to access information about traffic from remote places. The pi camera and usb camera capture videos in RGB format. To segment the vehicle colors more absolutely, videos were converted to HSV format from RGB. After segmenting the color, white blobs were found with noise included. Chain code method [6, 7] gives better localization of vehicle region when the input type is binary image. To remove the noise from images, the objects are defined in polygon shape, called convex hull [8]. Multiple Kalman filters were used to calculate centroids for tracking each blobs. A technique called Data association was used to allocate the centroids to the Kalman filters. A cost matrix is built by the centroids [9, 10], in which the centroid points from previous frame are subtracted from points from current frame.

2.2 SVM Algorithm

Support vector machine (SVM) are supervised learning models with associated learning algorithms that analyzed data and recognize patterns, used for classification and regression analysis. In paper [11], objects are identified in three major phases. First phase was the image analysis phase. Images were captured from roadside camera for analysis. At first the images were captured in RGB format using the function imread() in MATLAB. Later, it is converted to binary image. Then the noise of the image was reduced using innoise() function in MATLAB. The actual detection part occurs in the second phase. In this phase, all the steps occurred in first phase were combined. Then by using the MATLAB functions bwareaopens() and bwconncomp() final number of objects are counted. In third phase, the number of objects detected are compared with threshold value to identify the status of traffic density. In this case SVM was used for accurate high or low density measurement by plotting new data comparing it with old datasets.

2.3 Mixture of Gaussian

BS is a widely used moving object detection method which separates the foreground object from current frame with respect to a reference frame. The reference frame is the background model, which is considered as the previous frame

$$| I(x, y, t) - I(x, y, t-1) | > Th \tag{1}$$

The threshold value is set manually. The difference between the absolute value of the frames has to be greater than the threshold.

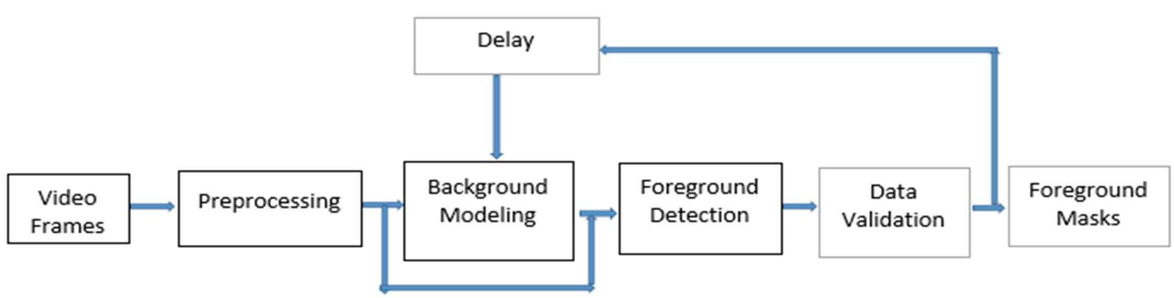


Fig 2.1: Background subtraction for obtaining foreground mask

According to [12], the algorithm is placed in multiple categories, from history-based algorithm to adaptive learning algorithm. We chose adaptive learning algorithm over history-based algorithm [13]. Adaptive learning algorithm requires just a background model and updates the model based on current frame. Thus it requires less memory bandwidth to operate. Conversely, history-based algorithm stores all the video frames and repeatedly access the frame history, thus requiring huge memory bandwidth.

We selected Mixture of Gaussian (MoG), which is based on adaptive learning algorithm. To model a pixel's background, MoG uses K Gaussian components each with weight, mean and deviation. To track changes in the new frame, the Gaussian components of both frames are compared with each other. The pixel's Gaussian components are updated based on the learning factor from new pixels. If none of the Gaussian components match the new pixel value, then the pixel is considered as a foreground.

CHAPTER 3

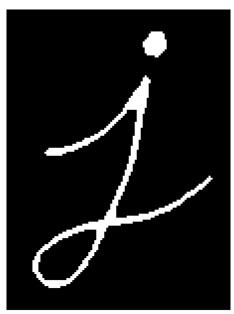
TRAFFIC DETECTION AND DYNAMIC TRAFFIC CONTROL

3.1 Background Detection

We collect a cache of frames (500 frames in total) from the input video and use it to train a background model using openCV built in function cv2.createBackgroundSubtractorMOG2(). It is based on an adaptive learning algorithm. It uses a method to model each background pixel by a mixture of an appropriate number of Gaussian distributions per pixel. The weights of the mixture represent the amount of time each individual pixel stays on the screen. The pixels that stay on the screen longer are more probable in being part of the background.

Original Image

Erosion



Dilation

Opening

Closing

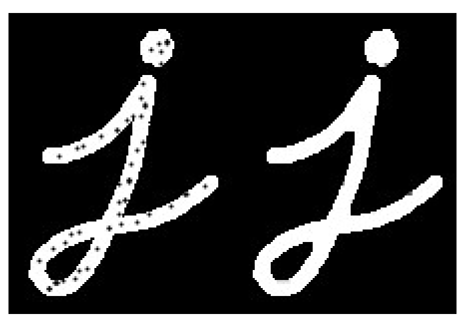


Fig 3.1: Filtering Example

With the creation of a background model we identify the foreground on every next frame from the video input. The foreground is still filled with noise so use filters on it to reduce noise make the foreground more useful for future use. The filters include:

Fig 3.2: Foreground Mask of a processed Frame

- **Closing:** Responsible for removing small holes or black points in the foreground object mask. It mainly involves **Dilation** followed by **Erosion.**
- **Opening:** Opposite of **Closing.** Used for removing noise in the foreground mask.
- **Erosion:** It decreases the thickness of foreground image, which results in removing white noise.
- **Dilation: Erosion** decreases the size of the image foreground image and this can cause the image to be broken down into several separate sections. Dilation joins the close parts together and also thickens the image.

3.2 Object Detection Using Contouring

Contouring is used to detect objects in the foreground image as shown in Fig 3.2. The centroids of the objects identified are marked and tracked. In Fig 3.3, it can be seen that the centroid makes up a path taken by the object between adjacent frames. This is used to track the object exiting an exit region which is marked. If one of the centroids of an object lies in the exit region while the others outside, the vehicle is counted. The vehicle is no longer tracked and this continues throughout all future frames.

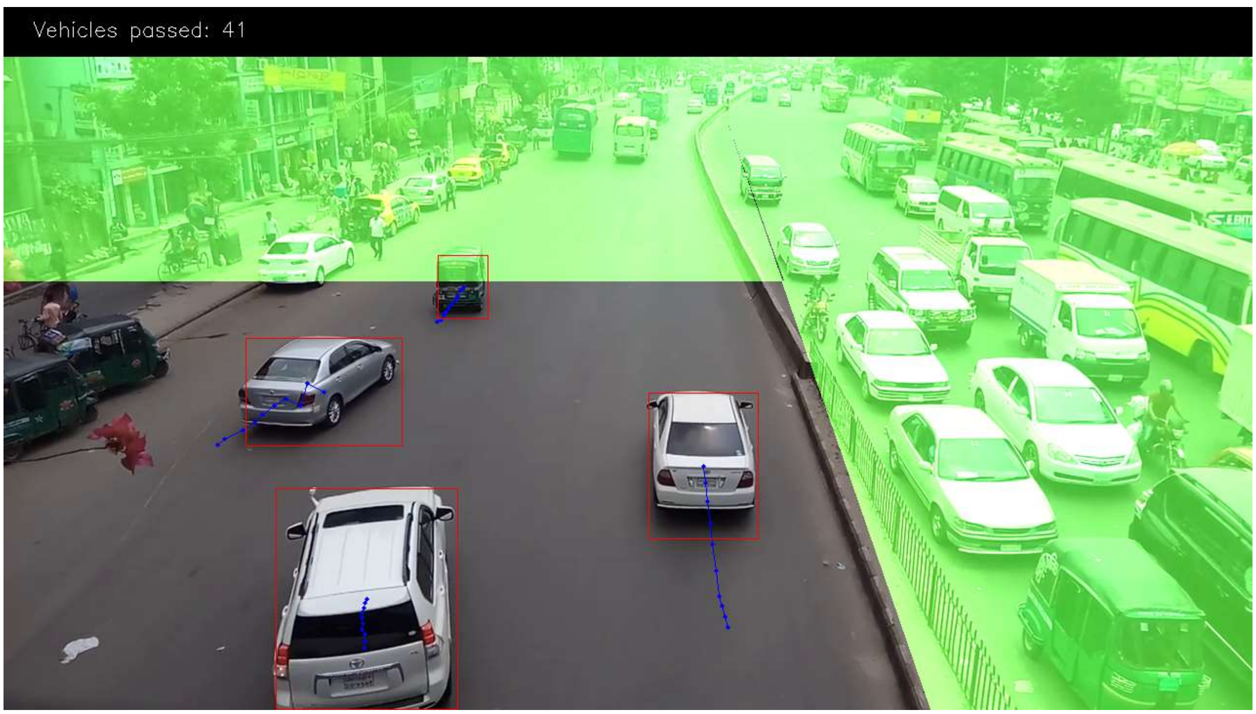


Fig 3.3: Resulted Frame with Object boundary after detection

Light also plays an important role in detection. Proper lighting allows the system to generate a proper foreground image. We are using contouring to find objects in the frame of a binary image (black and white foreground mask) and lack of proper lighting can result in missing objects in the binary image.

Fig 3.4: Foreground mask poorly generated due to lack of light

This can be seen in the results found of a night traffic provided to the system (Fig 3.4). An easy solution is the use of infrared cameras, which makes the system functional without any change to the algorithm.

3.3 Density Measurement from Vehicle Counting

The system that we are proposing involves the counting of cars entering and leaving a specific lane. This involves the placement of two individual computer vision systems, as mentioned in the previous chapter. They will be responsible for counting the cars entering a specified region and this will be calibrated manually since all lanes are not of the same dimensions. This data will be used to measure the number of cars present in a lane at any given time.

Number of cars in lane = Cars Entering the lane – Cars Leaving the lane (2)

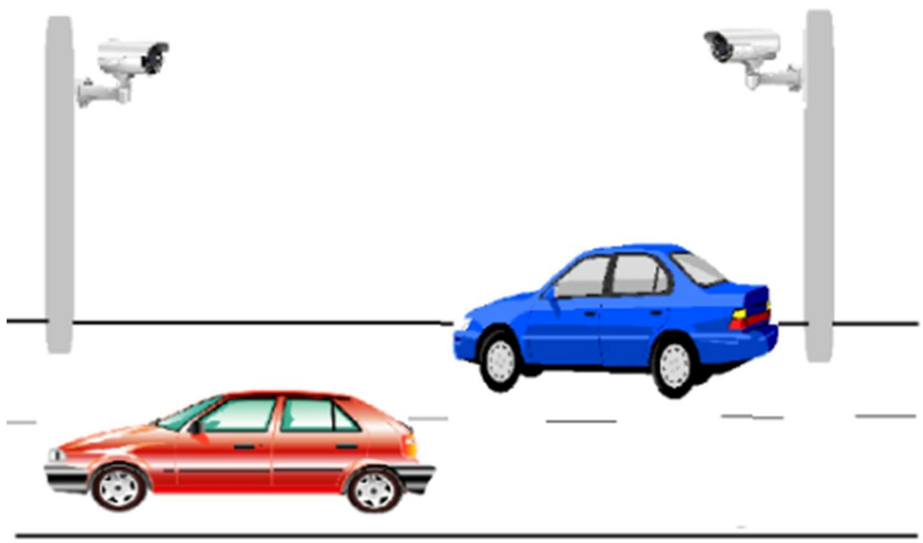


Fig 3.5: Camera Placement

We have assumed that the number of vehicles, entering or leaving different premises between the two ends of a lane, are approximately the same and doesn't affect the number of cars present in the lane at any given time significantly. Furthermore, the system requires manual entry of the lanes dimensions to the nearest metre (m), which will then be used to measure the density of the lane for comparison and giving priority of one lane over another at an intersection.

$$\boldsymbol{DensityofLane = \frac{Numberofcarsinlane}{Lengthoflane * Widthoflane}} \tag{3}$$

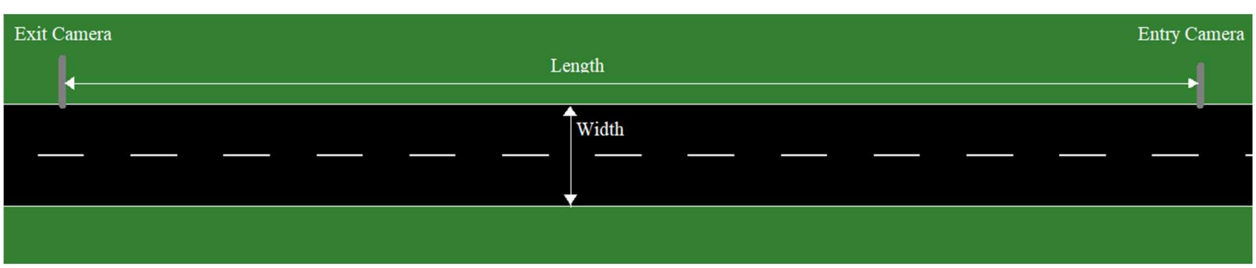


Fig 3.6: Lane Dimensions

The vehicle counting data will be collected by the computer vision systems and will be shared to a hub present at the traffic intersections. Each hub will collect data for lanes containing traffic coming towards the intersections. This hub will be responsible to make decisions for the intersection's traffic lights, measuring traffic density based on given lane dimensions and also communicate with the adjacent intersections. The details about the proposed model will be discussed in the following section.

3.4 Logic for Traffic Control

Currently we have worked on a four-way intersection, since it is one of the most common intersections one can find. As mentioned the intersections will have a central hub making decisions in controlling the lights. Before making any decisions certain rules are set as traffic rules.

Given rules:

- Left turning is always open
- No right turns are allowed at the intersection except for U-turns

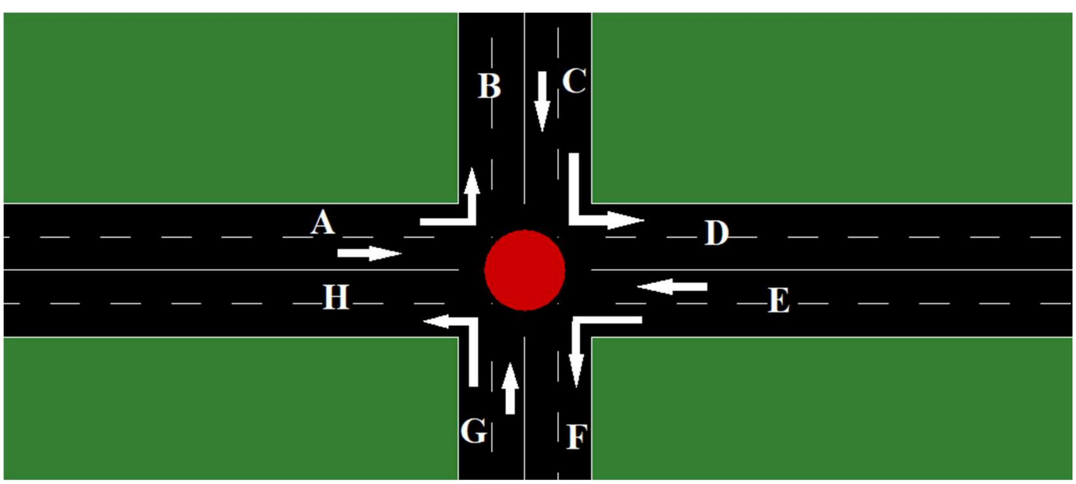


Fig 3.7: Intersection proposal diagram

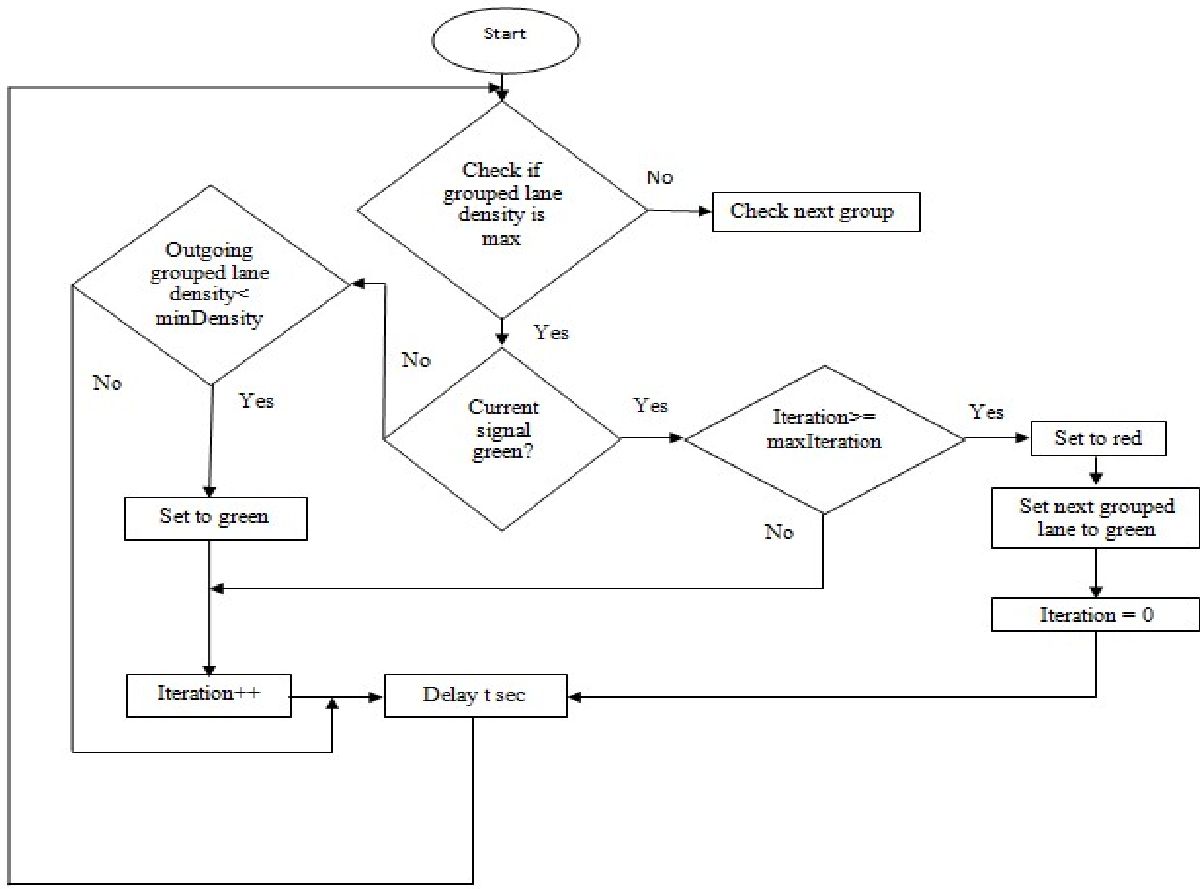


Fig 3.8: Algorithm for dynamic traffic control

The traffic lights will change such that only opposing lanes are green at the same time based on their combined traffic density. The lanes (A,E) and (G,C) are grouped together. Their combined densities are used in making decisions for which grouped lane will be open. The densities will be compared at every 30 sec interval (green time) and there will be a 5 sec delay between light change (yellow time). At every check the grouped lanes with highest traffic density will be given green time whereas the other group will go into red time. A grouped lane can be given successive green times based on traffic density but only max of three successive checks. On the fourth the check the remaining grouped lane will be given priority. This way certain lanes are not given total control of intersections. Furthermore, before a grouped lane is given green time the hub checks with the neighboring hubs whether the lanes in front are free or not. For example, in case of A and E, lanes D and H will be checked for low traffic density. This will be determined through the maximum threshold value of vehicle count, manually set into each hub for the respective lanes.

CHAPTER 4

RESULT

We used a video sample to check if our system could detect the vehicles properly and give good accuracy. The following observations were made

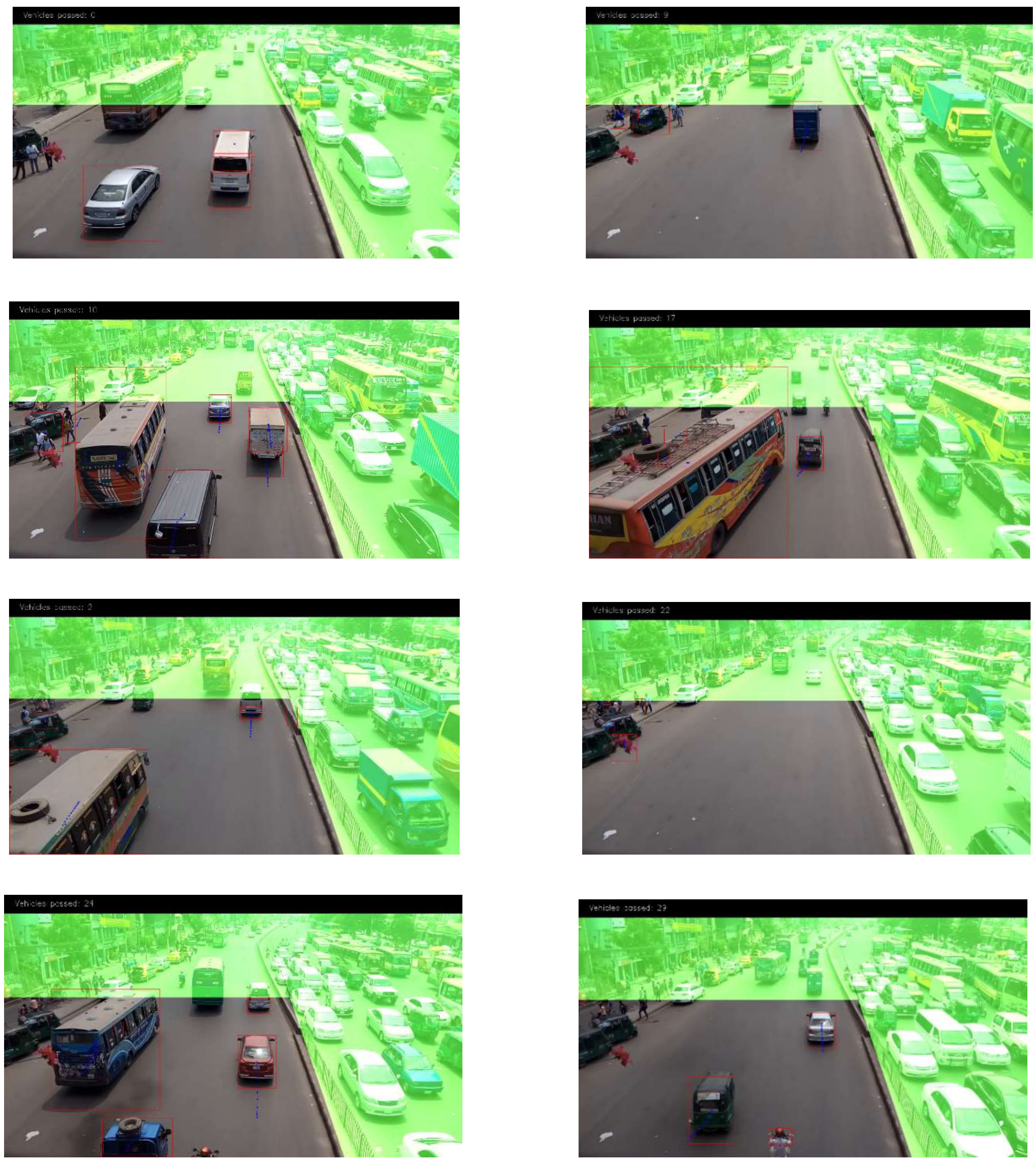


Fig 4.1: Vehicles detected from frame 0 – frame1000

Table I: Vehicles Detected per 100 Frames

Frame number	**Actual number of vehicles passed**	**Number of vehicles detected by the system**
0-100	6	6
101-200	3	3
201-300	1	1
301-400	7	6
401-500	5	5
501-600	1	1
601-700	2	2
701-800	5	5
801-900	6	5
901-1000	5	5

In Table I, we show the number of vehicles detected from a sample video stream. The video contained both moving and stationary vehicles. Only those vehicles were counted which entered the exit region showed in Fig. 3.3. The system gave an accuracy rate of 95.12%.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

We divided our thesis work into two parts. At first, we have successfully detected moving vehicles from a video input. The results gave 95.12% accuracy rate. We did it using adaptive learning background subtraction method (MoG). We discussed our work step by step in this paper. We briefly described the functions which are necessary for real-time computer vision. However, due to inadequate datasets of the variant vehicles in Bangladesh, we could not approach the SVM method which could yield more accurate result. We managed to collect dataset of other countries which didn't work on the video sample we used, due to different model of cars. The next part was about dynamic traffic control. Number of cars in specific region will calculated by the difference of entering and leaving vehicles. Then we calculate the density at any intersection point by dividing the number of cars in that lane by the length and width of the lane. We set a logic for controlling the traffic signals by setting some rules.

While working in the detection part, we noticed detecting vehicles at night was not very accurate. Our system can detect object when there is sufficient amount of light. We are looking forward to fix the problem in future. No priority was given to emergency vehicles. We plan to provide separate priority for emergency vehicles in future. In order to get better result, our future work includes, collecting datasets and models of all type of vehicles. Once they are collected we can train the model to give better detection capability.

REFERENCES

[1] Michael Hobbes, “Welcome to the Traffic Capital of the World” [Online]. Available: https://newrepublic.com/article/118416/what-dhaka-bangladesh-traffic-capital-world-can-teach-us. [Accessed Jul. 3, 2014]

[2] D. Li, B. Liang and W. Zhang, "Real-time moving vehicle detection, tracking, and counting system implemented with OpenCV," 2014 4th IEEE International Conference on Information Science and Technology, Shenzhen, 2014, pp. 631-634.

[3] Bin Tian; Ye Li; Bo Li; Ding Wen, Rear-view vehicle detection and tracking by combining multiple parts for complex urbansurveillance, in: IEEE Transactions on Intelligent Transportation Systems, vol.15, no.2, pp. 597–606 (April 2014).

[4] Y. Li, B. Li, B. Tian, F. Zhu, G. Xiong and K. Wang, "Vehicle detection based on the deformable hybrid image template," Proceedings of 2013 IEEE International Conference on Vehicular Electronics and Safety, Dongguan, 2013, pp. 114-118.

[5] M. Anandhalli, V.P. Baligar, A novel approach in real-time vehicle detection and tracking using Raspberry Pi, Alexandria Eng. J. (2017), http://dx.doi.org/10.1016/j.aej.2017.06.008

[6] Michael B. Dillencourt, HannanSamet, MarkkuTamminen, A general approach to connected-component labeling for arbitrary image representations, J. ACM. 39 (2) (1992) 253.

[7] H. Samet, M. Tamminen, Efficient component labeling of images of arbitrary dimension represented by linear bintrees, IEEE Trans. Pattern Anal. Mach. Intell. (1988).

[8] A.M. Andrew, Another efficient algorithm for convex hulls in two dimensions, Inform. Proc. Lett. 9 (5) (1979) 216–219, http://dx.doi.org/10.1016/0020-0190(79)90072-3.

[9] W. Harold Kuhn, The Hungarian Method for the assignment problem, Naval Res. Log. Quart. 2 (1955) 83–97.

[10] R.E. Burkard, M. Dell'Amico, S. Martello, Assignment Problems (Revised reprint). SIAM, Philadelphia, PA, 2012.

[11] AnjumHosne-Al-Walid, Nafisa. (2015). Traffic Density Measurement using Image Processing: An SVM approach. International Journal of Innovative Research in Computer and Communication Engineering. 03. 4979-4986. 10.15680/ijircce.2015.0306002.

[12] H. Tabkhi, R. Bushey and G. Schirner, "Algorithm and architecture co-design of Mixture of Gaussian (MoG) background subtraction for embedded vision," *2013 Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, 2013, pp. 1815-1820.

[13] A. M. Elgammal, D. Harwood, and L. S. Davis, "Non-parametric model for background subtraction," in *Proceedings of the 6th EuropeanConference on Computer Vision-Part II*, 2000, pp. 751–767.