2018 Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV) and
2018 2nd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)

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

Md. Fahim Chowdhury, Md. Ryad Ahmed Biplob, Jia Uddin
Dept. of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
{md.fahim693, rb147258}@gmail.com, jia.uddin@bracu.ac.bd

***Abstract*—In recent times, traffic jam has become a common problem in the major cities all over the world. 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. A video sample was collected and then Mixture of Gaussian algorithm was used for background subtraction method and for foreground detection to keep the count of the cars in each lane. The vehicles are detected by their line of centroid. A movement in centroid confirms a vehicle. The traffic lights at the intersections will change dynamically according to the conditions of traffic that will be detected from the video feeds. In between two intersections, there will be multiple cameras installed to count the number of vehicles entering and leaving each intersection. Furthermore, we restrict the vehicles to take right turns in the intersections. To validate our work, density measurement algorithm, images of live feeds and logics for traffic control are shown in this paper.**

Keywords—Dynamic traffic control; Real-time video processing; Background subtraction; Foreground detection;Mixture of Gaussian

I. INTRODUCTION

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. It has been observed in many cities that traffic lights are not working or are not being followed by the drivers. To fight such situations traffic polices are placed and the traffic congestions are handled by them manually. In congested cities, it is very tough to handle huge traffic by a traffic police.

With flourishing growth of technologies, researchers are coming up with many solutions to fight traffic congestion. A major requirement to come up with a solution for traffic ease 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[1].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.

Considering traffic jam of some under developed cities, 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.

Rest of the paper is organized as follows. Background study is in Chapter II, proposed methodology is Chapter III, Chapter IV includes experimental results and finally conclusion is Chapter V.

II. BACKGROUND ANALYSIS

A. Literature Review

Since traffic congestion is a growing problem, numbers of method for detecting vehicles have been proposed. In paper [2], 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 [3], 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 [4] 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 [5, 6] 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 [7]. 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 [8,9], in which the centroid points from previous frame are subtracted from points from current frame.

B. *Mixture of Gaussian (MoG) Algorithm*

Background subtraction 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

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

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

According to [10], the algorithm is placed in multiple categories, from history-based algorithm to adaptive learning algorithm. We chose adaptive learning algorithm over history-based algorithm [11]. 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, for our system. 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.

III. METHODOLOGY

A. *Traffic Detection*

1. Camera Placement

Traffic in specific lanes will be measured based on camera placements on the entries and exits of those lanes. Each camera will have their own sub system, counting vehicles entering and exiting the lane. It is assumed that vehicles entering and leaving the lane through various premises are identical and thus have no effect on the number of cars in lane. These data will then be supplied to the central hub for evaluation of traffic lights.

No. of cars in lane = Cars Entering the lane – Cars Leaving the lane (2)

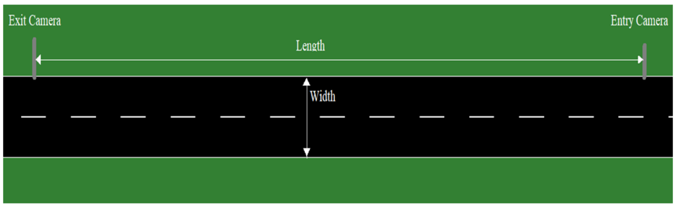

Figure 1. Lane Dimension

The calculation is done based on the dimension of lanes. Number of cars in a lane is calculated and the density is measured by dividing it with lane area.

$$Density\ of\ Lane = \frac{Number\ of\ cars\ in\ lane}{Length\ of\ lane * Width\ of\ lane} \quad (3)$$

Considering all the calculations above, the camera is placed in such a way that the camera gets to record all the vehicles within the lane. Figure 3illustrates the situation.

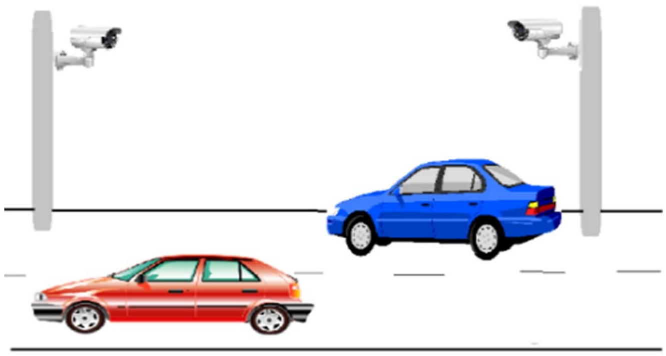

Figure 2. Camera Placement

2. Background Subtraction and Object Detection

Video input from the camera provides initial cache of frames for background subtraction. We are using Mixture of Gaussian (MoG) implementation of BS. This approach reduces memory usage and makes it possible to be implemented on less powerful hardware. Based on the background model, we will identify the foreground of the video input and thus identify the moving objects in the video. Contouring is then used to identify objects in the foreground image provided by the BS algorithm. The centroids of the objects are then identified and used from successive frames to form the path taken by the object. An exit region is marked and based on the position of the centroid the number of vehicles are counted.

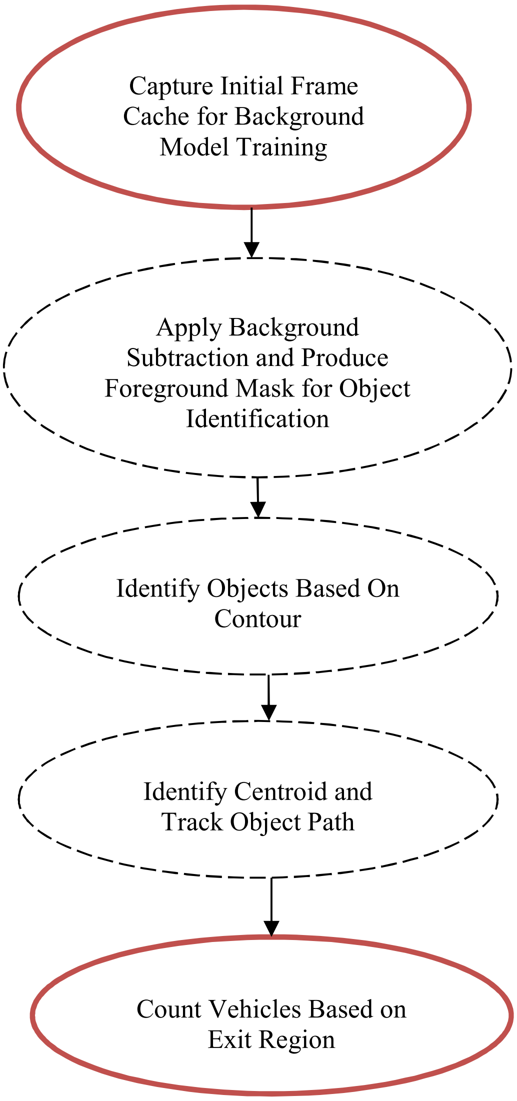

Figure 3. Vehicle Counting Algorithm

Figure 3 illustrates the background subtraction process and the collection of foreground. Vehicles are counted after applying filters and finding centroid and object path.

B. 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

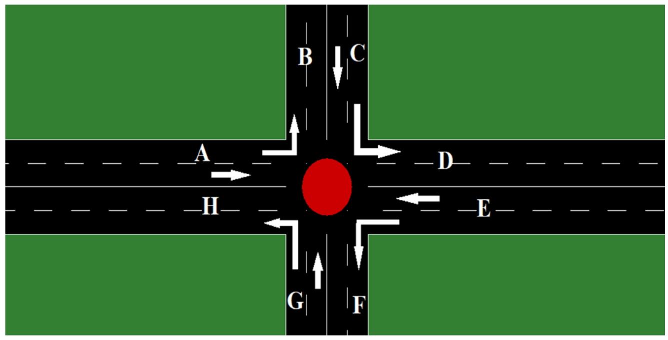

Figure 4. Intersection Proposal Diagram

The traffic lights will change such that only opposing lanes are green at the same time based on their combined traffic density. In Fig. 4 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.

IV. RESULTS

We recorded a video to check whether our model could give a precise count of vehicles. In Figure 5, the red boundaries confirm vehicles. The blue dots are the centroid of that moving object. The green portion in Figure 5 is the exit region. The vehicle is counted once it enters the exit region.

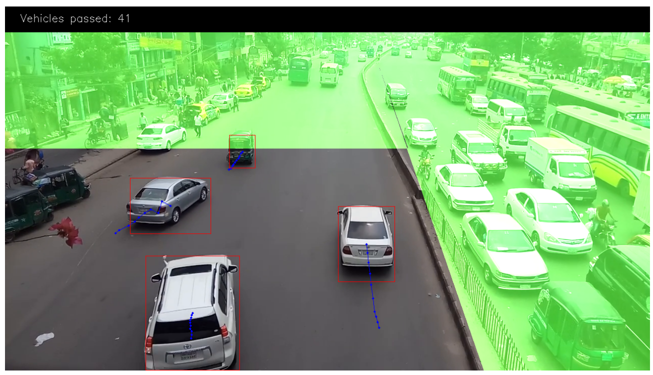

Figure 5. Vehicle Detection

Figure 6. Foreground Mask of a Processed Frame

The foreground mask is obtained after background subtraction. It is illustrated in Figure 6.

Table I: Vehicles Detected per 100 Frames

Frame number	Vehicles found	Vehicles Present
0-100	6	6
101-200	3	3
201-300	1	1
301-400	6	7
401-500	5	5
501-600	1	1
601-700	2	2
701-800	5	5
801-900	5	6
901-1000	5	5

Finally 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 Figure 5. The count was very accurate.

V. CONCLUSION AND FUTURE WORKS

This paper presented a model for density measurement for traffic management. The proposed model allows counting the number of cars entering or leaving a lane based on the exit region. In the experimental evaluation, the proposed system works best on low dense traffic and during cases where the traffic does not cross paths. In the future we plan on trying other detection methods that involve machine learning and increase the accuracy of detection of vehicles.

REFERENCES

[1] 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.

[2] B. Tian, Y. Li, B. Li, D. Wen, "Rear-view vehicle detection and tracking by combining multiple parts for complex urbansurveillance," IEEE Transactions on Intelligent Transportation Systems, vol.15, no.2, pp. 597–606, April 2014.

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

[4] M. Anandhalli, V.P. Baligar, "A novel approach in real-time vehicle detection and tracking using Raspberry Pi," Alexandria Eng. J. (2017), pp. 1-11.

[5] M.B. Dillencourt, H. Samet, M. Tamminen, "A general approach to connected-component labeling for arbitrary image representations," J. ACM. vol. 39, no. 2,pp. 253-280, 1992.

[6] H. Samet, M. Tamminen, "Efficient component labeling of images of arbitrary dimension represented by linear bintrees," IEEE Trans. Pattern Anal. Mach. Intell. vol.10, no.4, pp. 579-586, 1988.

[7] A.M. Andrew, "Another efficient algorithm for convex hulls in two dimensions," Inform. Proc. Lett. vol.9, no.5, pp.216–219, 1979

[8] W. Harold Kuhn, "The Hungarian Method for the assignment problem," Naval Res. Log. Quart. vol.2, pp.83–97, 1955

[9] 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, 2013

[10] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in Proceedings of the 6th EuropeanConference on Computer Vision vol.1843, pp. 751–767, 2000