An Automated
Traffic Signal Management System

Supervisor: Dr. Jia Uddin

Partha Narayan Chowdhury 13301062

Tonmoy Chandra Ray 13301023

Department of Computer Science and Engineering,

BRAC University

Submitted on: 18th April 2017

DECLARATION

We, hereby declare that this thesis 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	Partha Narayan Chowdhury
	Tonmoy Chandra Ray

ACKNOWLEDGEMENTS

We are extremely grateful to Dr. Jia Uddin, our thesis supervisor, for his assistance and guidance throughout the entire journey of ours project. His support and motivation always kept us moving forward with our work and finally obtaining our expected goals.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Today, traffic congestion has turned out to be a very significant issue due to the increasing number of vehicles and poor traffic management in our busy overcrowded cities. In order to maintain the continuous piling of vehicles at junctions, an efficient and productive approach is required which saves manual human effort, precious time and fuel. In this paper, we propose a model for the detection of vehicles at junctions in real time using image processing which results into an efficient traffic management process. This model uses the value parameter which indicates the image's brightness level, from the HSV format of the image to distinguish between day and night time images. The day and night images are processed using two different methodologies in order to extract the number of vehicles present at each junction. In order to increase the computational speed of our system, we eliminated the traditional edge detection approach and implemented a foreground and background image comparison in order to obtain the number of existing vehicles during the day. Images during the night are processed for extracting the number of headlights present at junctions. Moreover, the circularity of extracted headlights is analyzed for differentiating between the headlights and its reflection on roads. The redundant count of vehicles due to illuminated fog lights is also handled using the image coordinate system. The experimental results of this proposed model show an average accuracy of 94.59% for vehicle detection during both day and night time using our dataset. The results from the image processing of both day and night time images are used as inputs in a signal time optimization algorithm which gives an efficient signal time for vehicles waiting at each road end.

CHAPTER 01

INTRODUCTION

1.1 Motivations

A 40-year-old reported to daily star to be struggling from sore neck and a full bladder after struggling for hours in a CNG run auto rickshaw on a scorching hot day. Another office goer shares the similar everyday hardships he has to overcome to reach his work station from home [15]. These are countless stories of any daily commuters living in urban busy cities. Addressing this issue, we propose a model of handling the traffic management system which can minimize our daily struggles to a certain extent.

The existing models addressing the traffic management system used various methodologies. Sensor based methodology uses tools like RFID active tags, RFID passive tags, RFID readers & infrared detectors [5,6]. Such approach requires the implementation of a lot of complex hardware and consequently we used a more convenient and efficient way by using image processing to implement our proposed model. Works related to image processing uses edge detection technique to detect the number of vehicles present on the road [1,2,3 & 10]. However, this requires a much longer computation time and in order to avoid this drawback we implemented the comparison of foreground and background images using image subtraction to obtain the counting of vehicles.

Some of the existing models mainly works with day time images avoiding the necessity of the model during the night [1-3]. Whereas, our proposed model works for both day and night. The night image processing uses circular object detection and distinguishes the pair headlights and fog lights in order to obtain a more precise count of vehicles.

The two most common application in traffic signal control are use of manual orders by traffic police and a fixed time counter which alters after a constant period of time [8 &9]. In our model, the results from the day and night image processing are manipulated in an algorithm which will provide an optimal time for the vehicles ensuring minimum waiting.

Table 1 – Symbol Table

Symbol	**Definition**
n_v	Number of vehicles
n_h	Number of headlights
n_p	Number of pair headlights
n_f	Number of fog lights
h_i	Current headlight
h_j	Adjacent headlight
bf	(Background –Foreground) image
fb	(Foreground –Background) image
m	Merged image
TH_v	Brightness threshold
$TH_{connectivity}$	Connectivity threshold
TH_{area}	Area threshold
$TH_{circularity}$	Circularity threshold

1.2 Contribution Summary

The summary of the main contributions is as follows:

- Our model uses V, the value (brightness)parameter of HSV image to differentiate between day and night time eliminating the analysis of hue, H in order to obtain a lower computation time.
- Avoiding edge detection technique, we implemented the subtraction of foreground and background images for achieving the count of existing vehicles. This approach reduces the computation duration and make the system more efficient.
- We achieved the distinguishing of single headlight and double headlight vehicles which was not done in many earlier works. Moreover, the identification of pair headlights of same vehicle and fog lights is also obtained in order to avoid faulty count of vehicles and this increases the overall accuracy of our proposed model.
- The algorithm providing the optimized signal time is an improvement from the constant timer technique as it avoids the unnecessary waiting of vehicles at each road ends.

1.3 Thesis Orientation

The rest of the thesis is organized as follows:

- Chapter 02includes the necessary background information regarding the proposed image process based approach.
- Chapter 03 presents the proposed model of our system.
- Chapter 04 demonstrates the experimental results and comparison.
- Chapter 05 concludes the thesis and states the future research directions.

CHAPTER 02

BACKGROUND INFORMATION

2.1 RGB, HSV color space and brightness value

A RGB image, also known as *truecolor* image refers to the three-color components, red, green and blue [16]. These color components are stored as m-by-n-by-3 data array. The RGB format of an image does not identifies the intensity of the image. In order to obtain the intensities of individual pixels the RGB image is converted to HSV which stands for hue, saturation and value [17]. The value component is analyzed against a threshold value to distinguish between day and night time images.

2.2 Image Masking, Image Subtraction

Masking is the process of allocating a certain background value to an image which is utilized for separating the interest area of the image from its background [18]. Image subtraction is the difference between the matrix which hold the pixel values of two images.

2.3 Image Binarization and Boundary Detection

Image binarization is a method for creating a binary image from a RGB image by substituting all values greater than a certain global threshold with 1s and all others with 0s [19]. Boundary detection is a method in MATLAB to detect the exterior boundaries in a binary image [20]. This method is implemented in our model to detect the boundaries of vehicles in our binary images and finally counting the number of vehicles present.

2.4 Round Object Detection

Round object detection methodology is implemented in our model to detect the circular headlights of vehicles during night time image processing. The first method used in round object detection is *regionprops* which determines the area & centroid of the binary images. The vehicle area perimeter is calculated using the *boundary* value we get from *bwboundaries* method. Finally, roundness of objects is calculated using the *area* from *regionprops* and perimeter from *boundary* using the following equation.

$$roundness = 4 * pi * \frac{area}{perimeter^2} \qquad \textit{equation (I)}$$

CHAPTER 03

PROPOSED MODEL

3.1 Introduction

Figure 3.1represents a block diagram of our proposed model. There are four components of our proposed model. First of all, we detect whether the captured image is of day time or night time. We do this by analyzing the brightness of the images. Then we apply our proposed day time or night time image processing techniques for day time or night time images respectively. The day time image processing techniques involve foreground and background image comparison using image subtraction and object detection methodology. On the other hand, the night time image processing techniques involve single vehicle headlight, pair headlight and fog light detection. After processing the day and night time images, we obtain the density of the connected roads of the junction. Finally, we use the density to calculate an optimal signal time.

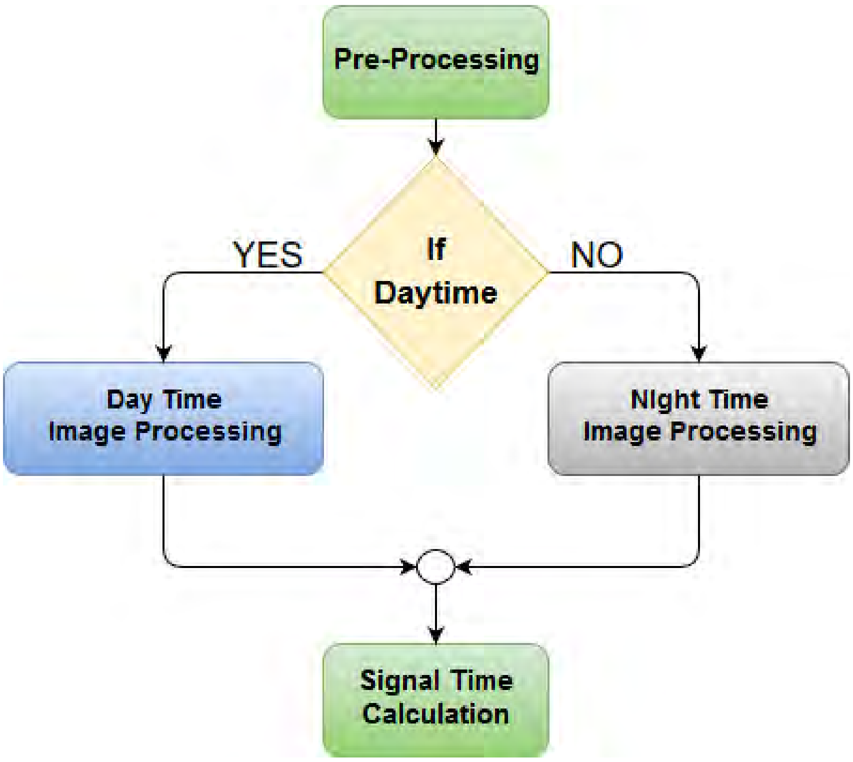


Fig 3.1: Block diagram of proposed model.

3.2 (Pre-processing) Day and Night image detection

Figure 3.2 presents a block diagram of day and night image detection procedure.

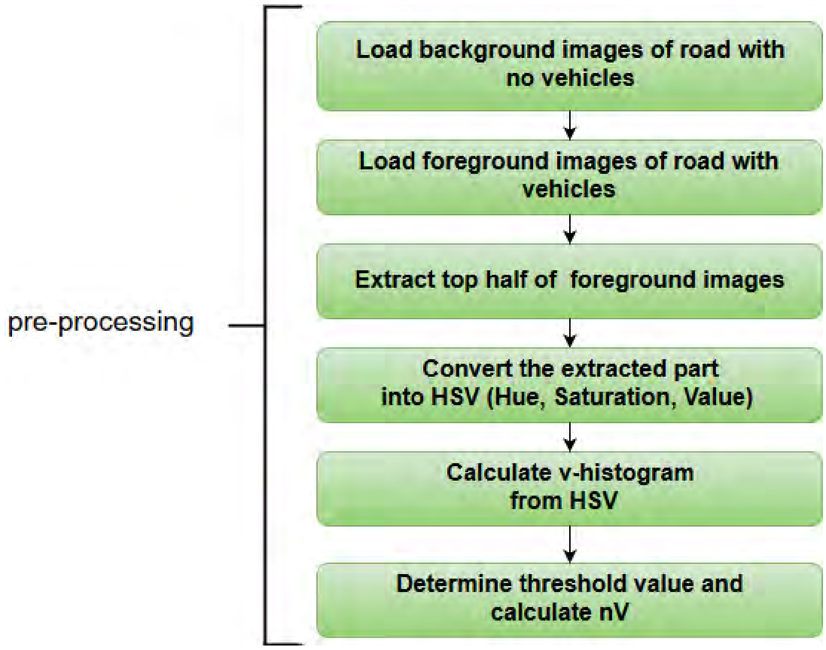


Fig 3.2: Block diagram of pre-processing

At the beginning, we crop out the top half of the captured image since the top half portion provides enough brightness to distinguish between day time and night time image (figure 3.3& 3.4). The captured image is in RGB form. We convert it into HSV form. Then we extract

Fig 3.3: Top half of foreground day image

Fig 3.4: Top half of foreground night image

the v-channel values of the HSV image and compare it with our threshold value of 0.59 [4]. The threshold value is set at 0.59 as a huge portion of pixels have brightness value above 0.59

during the day but below 0.59 in case of night time images. The following histogram figures 3.5 and 3.6 justify out reason.

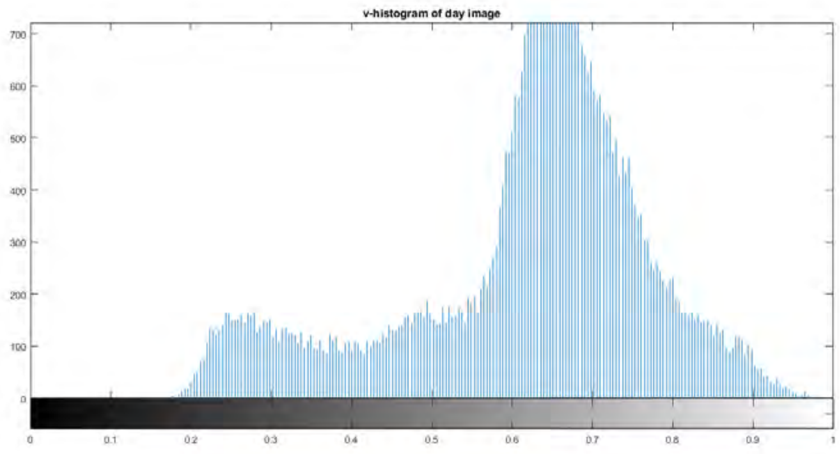


Fig 3.5: Histogram of day image

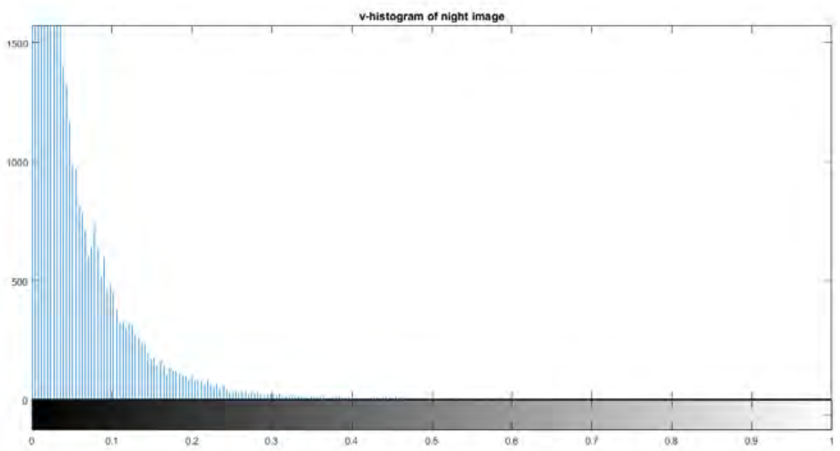


Fig 3.6: Histogram of night image

If the number of total pixels having value higher than our threshold is above 1000, it is a day time image. Otherwise it is a night time image.

$$\{n_V = n_V + 1 \quad if\ pixelValue\ \geq THv \qquad \textit{equation (II)}$$

$$where \qquad THv = night\ brightness\ intensity\ threshold(0.59)$$

$$n_V = number\ of\ vehicle\ pixels$$

3.3 Vehicle Detection in Day Time

Figure 3.7 presents a block diagram of day time vehicle detection procedure. The

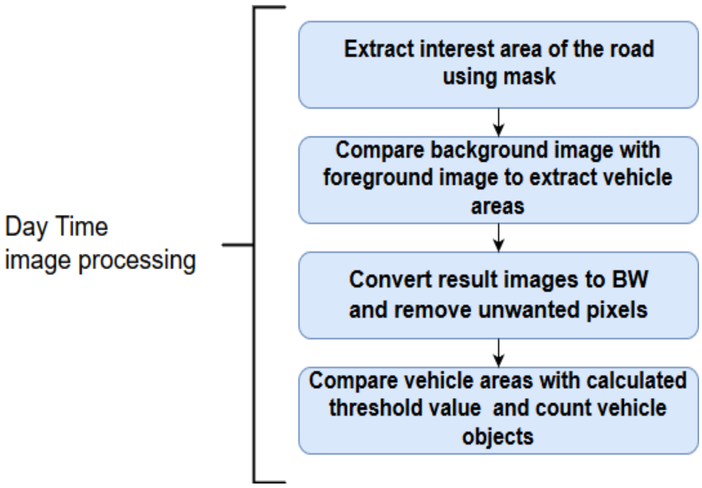


Fig 3.7: Block diagram of day time image processing

images are in RGB format. We implement image processing techniques only on the area where the vehicles are present using mask to reduce computational time. We convert the RGB image to *grayscale* to prepare it for interest area extraction using mask. We loop through the image and copy only the interest area pixels. Figure 3.8& 3.9 are the result images.

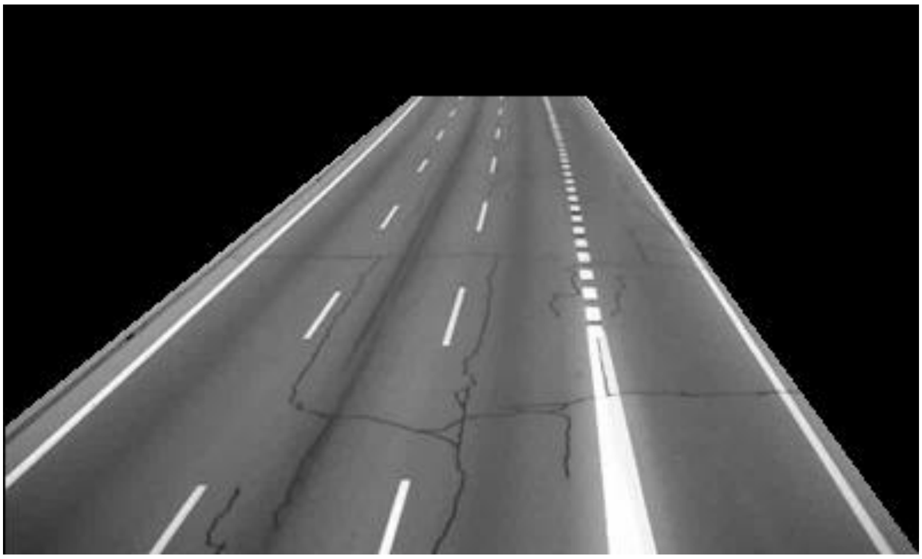


Fig 3.8: Interest area of day time background image

Fig 3.9: Interest area of day time foreground image

Now, we subtract the foreground and background images to extract the vehicles from the images. At first, we subtract the foreground image from the background image and this gives us the low intensity pixels of the vehicles (figure 3.10). Secondly, we subtract the background image from the foreground image which gives us the high intensity part the vehicles (figure 3.11).

Fig 3.10: Background and foreground image subtraction

Fig 3.11: Foreground and background image subtraction

Thirdly, we binarize the two subtraction results with threshold luminance greater than 0.10. Figure 3.12 and 3.13 represent the result of binarization.

Fig 3.12: Binarization of background and foreground image subtraction result

Fig 3.13: Binarization of foreground and background image subtraction result

Fourthly, we merge the two binarized images to get complete binarized vehicles using the computation stated in *equation (III)* (figure 3.14).

$$\begin{cases} m_{ij} = bf_{ij} + fb_{ij} & if\ pixel_{ij} \in interestArea\ and \\ & if\ (bf_{ij} == 0\ and\ fb_{ij} == 0)\ is\ false \\ m_{ij} = 0 & else \end{cases} \qquad \textit{equation (III)}$$

$where$:

$bf_{ij} = subtraction\ result\ of\ foreground\ from\ background$

$fb_{ij} = subtraction\ result\ of\ background\ from\ foreground$

$m_{ij} = new\ merged\ image$

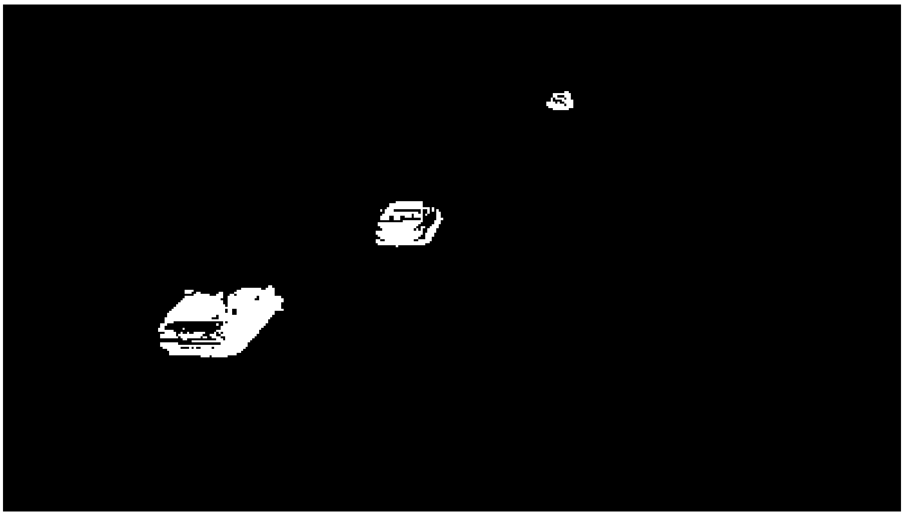


Fig 3.14: Merged binarized image

Finally, we apply the connected component methodology on the merged binarized image to get individual vehicles and compare area with a threshold vehicle area value to count the vehicles [14]. The process is presented in the following *equation (IV).*

$$\text{Vehicle count} = \begin{cases} n_V = n_V + 1 \;\; if\ connectivity\ \geq TH_{connectivity} \\ \qquad\qquad and\ if\ area \geq TH_{area} \end{cases} \qquad \text{equation (IV)}$$

$where$: $\quad TH_{connectivyty} = connected\ component\ area\ threshold\ (8)$

$TH_{area} = threshold\ area\ of\ a\ vehicle\ (37)$

$n_V = number\ of\ vehicle$

3.4 Night time image processing

Figure 3.15presents a block diagram of night time vehicle detection procedure.

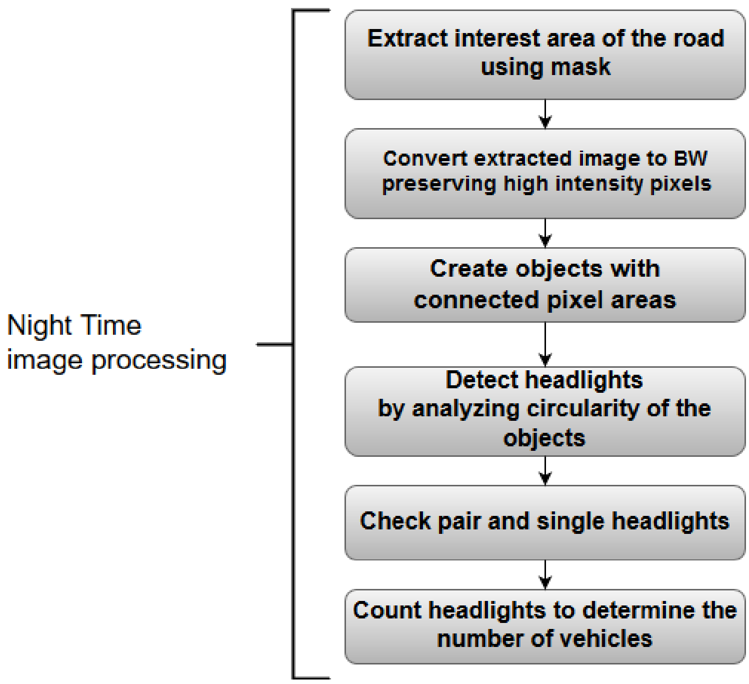


Fig 3.15: Block diagram of night time image processing

During night time image processing, we again extract the interest road area and detect vehicle headlights (figure: 3.16). Then we binarize the interest area with high luminance level of 0.9 to extract out only the high intensity pixel (figure 3.17).

Fig 3.16: Interest area of night time image

Fig 3.17: Extraction of high intensity pixels

Afterwards, we filter the image to remove any pixel with area less than our calculated headlight area size threshold. Later on, we analyze the circularity of each remaining areas to distinguish actual headlights and reflections. Then we apply object counting methodology to count the remaining objects [14].

$$\{n_H = n_H + 1 \quad if\ areaCircularity\ \geq TH_{circularity}$$ *equation (V).*

$$where: \qquad TH_{circularity} = areaCircularity\ with\ no\ holes (0.80)$$ *equation (V)*

$$n_H = number\ of\ headlights$$

Finally, we have the pair headlights from cars, vans, trucks and single headlights from bikes, three-wheelers and reflections of the lights on the road. We detect the pairs by analyzing the area sizes, x co-ordinates and y co-ordinates of each headlight. Headlights of same vehicle will have almost same area size. Moreover, they will have almost same height from the ground which can be detected by y-co-ordinates. Again, they will be situated in left and right sides of the car so their x-co-ordinate difference can be used to detect if they belong to the same vehicle (figure: 3.18). The following calculation are done to detect pair headlights.

$$\text{Pair count} = \begin{cases} n_P = n_P + 1 \; if \; H_i \; is \; horizontally \; adjacent \; to \; H_j \; and \\ if \; (|H_i(y) - H_j(y)| \; \leq 2) \; and \\ if \; (|H_i(x) - H_j(x)| \; \leq 35) \; and \\ if \; (|H_i(area) - H_j(area)| \; \leq 20) \end{cases} \qquad \text{equation (VI)}$$

$where$:

$H_i = curent\ headlight$

$H_j = adjacent\ headlight$

$H_i(x) = x\ co-ordinate\ of\ current\ headlight$

$H_i(y) = y\ co-ordinate\ of\ current\ headlight$

$H_j(x) = x\ co-ordinate\ of\ adjacent\ headlight$

$H_i(y) = y\ co-ordinate\ of adjacent\ headlight$

$n_P = number\ of\ pair\ headlight$

The fog lights are situated right below each pair headlights. In this case, the x-co-ordinate values of a headlight and a fog light situated below it are almost same. On the other hand, the y-co-ordinate difference value is within a threshold limit (figure: 3.17). The following calculation are done to detect fog lights.

$$\text{Fog light pair} = \begin{cases} nF = nF + 1 \;\; if \; H_i \; is \; vertically \; adjacent \; to \; H_j \; and \\ if \; (|H_i(x) - H_j(x)| \; \leq 2) \; and \\ if \; (|H_i(y) - H_j(y)| \; \leq 15) \end{cases} \qquad \text{equation (VII)}$$

$where$:

$H_i = curent\ headlight$

$H_j = adjacent\ headlight$

$H_i(x) = x\ co-ordinate\ of\ current\ headlight$

$$H_i(y) = y\ co - ordinate\ of\ current\ headlight$$

$$H_j(x) = x\ co - ordinate\ of\ adjacent\ headlight$$

$$H_i(y) = y\ co - ordinate\ of adjacent\ headlight$$

$$nF = number\ of\ fog\ pair\ headlight$$

Fig 3.18: Pair headlight and fog light detection

3.5 Signal Time Calculation

After the day and night time image processing we have the density of the connected road of the junction. Now we compare the density to set a signal time. This signal time can range from 30 seconds to 180 seconds. The road with higher density will get green signal. After every 60 seconds the system will again check the density even if the green signal time is not finished in the background. If the latest density indicates comparatively high density on the opposite road, the green signal will be swapped to the opposite road otherwise the current road with the green signal will use the remaining signal time. If any of the road is empty, the system will set 30 seconds of green signal time to the opposite road and recheck the density after 30 seconds. This way we ensure minimum waiting time in a junction.

CHAPTER 04

EXPERIMENTAL ANALYSIS

4.1 Introduction

Our proposed model is entirely based on the MATLAB environment where all the image processing, calculation and result simulation is carried out. Our whole system is analyzed and tested in a personal computer (PC) with the configuration Intel(R) Core i5-3210 CPU @ 2.50 GHz, 8GB RAM, running Windows 10.

4.2 Day vs Night Result Analysis

As shown in the table 2, the comparison of *hue* and *value* count for both day and night is clearly observable. The *value* parameter indicates a percentage difference of above 90% between day and night images while the *hue* count difference ranges from about 10% to 45% which shows a vast fluctuation and inconsistent change. Consequently, the difference in *value* alone is taken into consideration in order to distinguish between day and night time images.

Table 2 – Day Night Brightness Comparison Chart

	Day	Night	Difference	Difference Percentage
nH	20162	13640	6522	32.23%
	22631	16515	6116	27.02%
	24593	13714	10879	44.23%
	24435	21299	3136	12.83%
nV	14553	250	14303	98.28%
	11897	314	11583	97.36%
	33441	413	33028	98.76%
	33729	371	33358	98.90%

4.3 Day Time Image Processing Result Analysis

The following table demonstrates the accuracy for vehicle detection on roads. Two different datasets were used for this result analysis [13]. The model achieves an average accuracy of 95.08% for detecting the vehicles compared to 93.4% of the edge detection methodology used by the one of Chandrasekhar et.al [9]. The errors in false counting is contributed by the existence of vehicles at a far distance which are demonstrated in the following figure 4.1 & 4.2

Table 3 – Day Time Vehicle Detection Result Chart

	Actual Vehicle Count	Detected Vehicle Count
Dataset 1	5	5
	6	6
	7	7
	6	5
	5	5
Dataset 2	8	7
	10	8
	10	10
	9	9
	6	6

Fig 4.1: Heavy congested day image

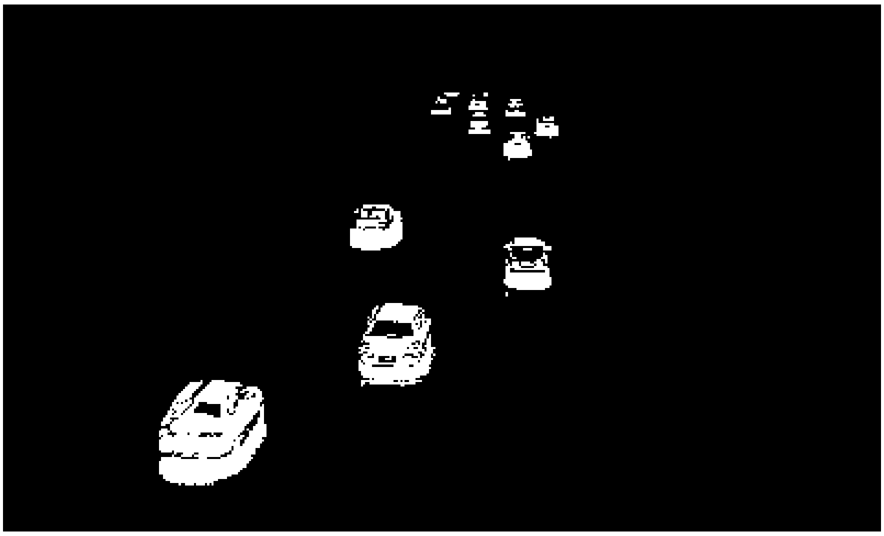


Fig 4.2: Binarized heavy congested image

4.4 NightTime Image Processing Result Analysis

The following chart describes the percentage accuracy of detected vehicles during the night. This model achieves an accuracy of 94.1% tested on two sets of images containing light and heavy traffic. Perera et. al. obtained an accuracy of 90.5% in their respective research model [7]. The reflection of high intensity headlights and existing vehicles at a far distance limited our accuracy of vehicle detection to 94.1% (Figure: 4.3- 4.5).

Table 4 – Night Time Vehicle Detection Result Chart

	Actual Vehicle Count	Detected Vehicle Count
Dataset 1	2	2
	5	4
	6	6
	11	9
	8	8
Dataset 2	5	4
	6	6
	2	2
	7	7
	7	7

Fig 4.3: Heavy congested night image

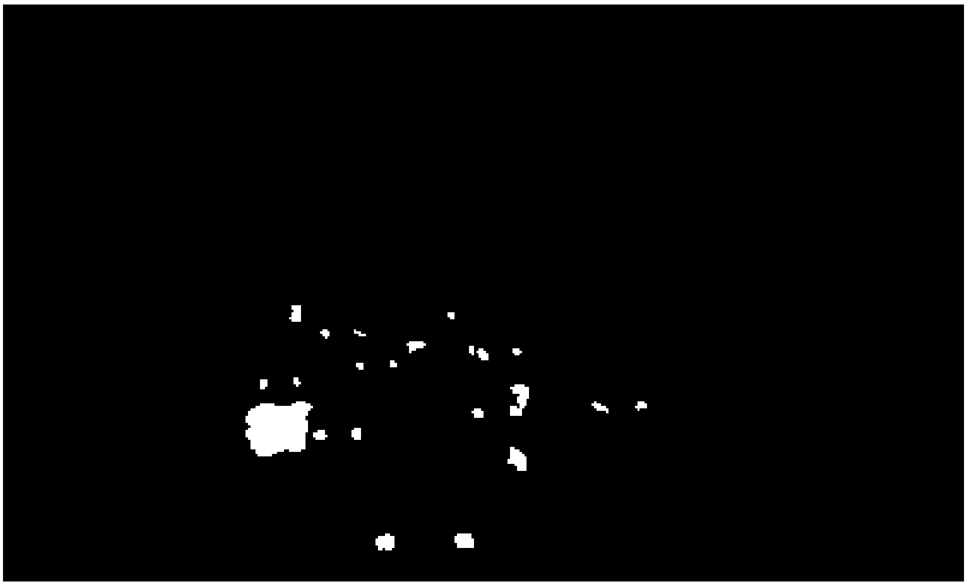


Fig 4.4: Headlights and light reflection of congested night image

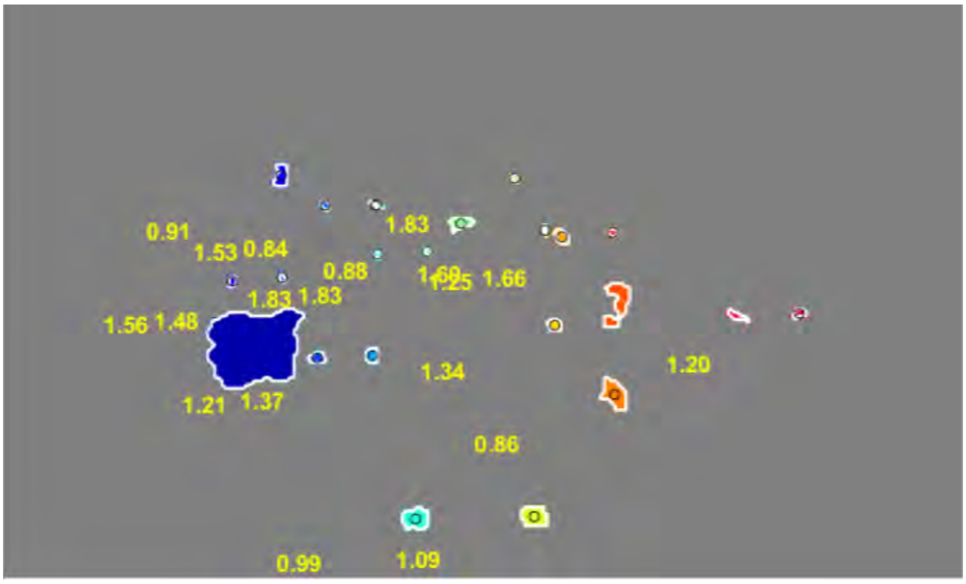
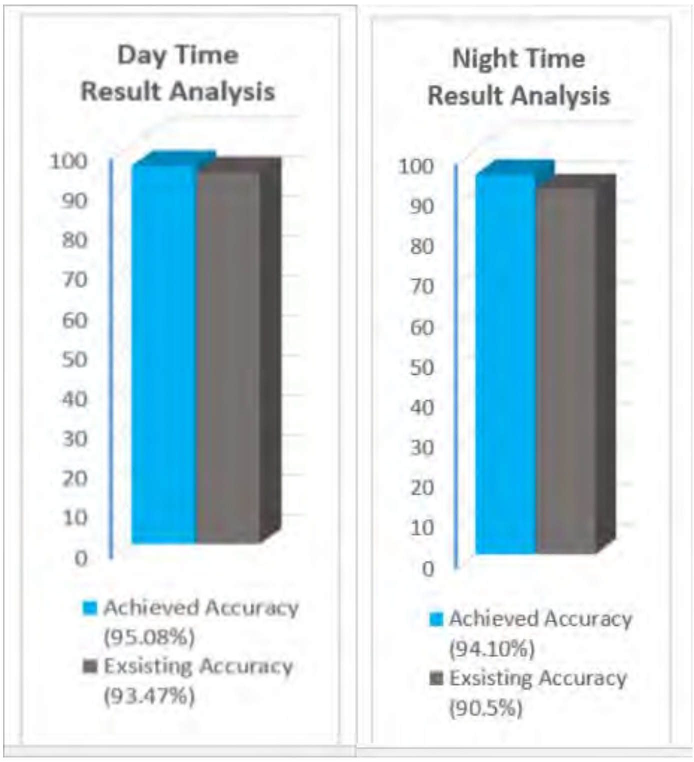


Fig 4.5: Circularity of headlights and light reflections



(a) (b)

Fig 4.7: Day time accuracy(a) Night time accuracy(b)

4.5 Dataset

In order to evaluate the accuracy of our proposed model, we used three types of datasets [12, 13]. Two of the datasets were used for testing out the vehicle count during day time while the other one was used for night time vehicle detection.

CHAPTER 05

CONCLUSIONS AND FUTURE WORKS

5.1 Concluding Remarks

This proposed model is an attempt to detect the number of vehicles on road in real time and accordingly build an algorithm to provide an optimal traffic duration ensuring minimum waiting time for vehicles. Based on the experimental results, we obtained an accuracy of 95.08% for detecting vehicles during the day over the previously achieved 93.47% accuracy by implementing foreground & background image subtraction, image-binarization & object counting methodology. This increase in accuracy for our tested dataset was achieved by avoiding the traditional edge detection methodology. Moreover, this also contributed in a much faster overall computing process.

The results for night time exhibit an accuracy of 94.10% over the conventional 90.5%. The improvement was made possible by considering faulty fog lights and pair headlights detection into our system.

Accuracy and perfection is expected to be increased even more by applying new and improved methodologies over the ones used in this model in the future.

5.2 Future Works

Integration of emergency vehicle detection- The feature of integrating emergency vehicle detection in our system can be added to make the model more effective by prioritizing the passage for emergency vehicles. This can be obtained by converting RGB image to HSV color space and determining the red intensity of emergency light by comparing it against an analyzed threshold value.

REFERENCES

[1] Choudekar, P., Banerjee, S., and Muju, M.“Implementation of Image Processing in Real Time Traffic Light Control”, IEEE, pp. 96, 2011.

[2] Hasan, M., Saha, G., Hoque, A., and Majumder, B.“Smart Traffic Control System with Application of Image Processing Techniques”, 3rd International Conference on Informatics, Electronics & Vision, 2014.

[3] Rahishet, Sahoo, A., Indore, A., Vaibhavdeshmukh and Pushpa.“Intelligent Traffic Light Control Using Image Processing”, Proceedings of 21st IRF International Conference, pp. 2, 2015

[4] Taha, M., Zayed, H., Nazmy, T., and Khalifa, M. “Day/Night Detector Vehicle Tracking in Traffic Monitoring Systems”, International Journal of Computer, Electrical, Automation, Control and Information Engineering. Vol. 10, No. 1, pp. 101, 2016.

[5] Singh, H., Kumar, K., and Kaur, H.“Intelligent Traffic Lights Based on RFID”, International Journal of Computing & Business Research, 2012.

[6] Albagul, A., Hrairi, M., Wahyudi, Hidayathullah, M.“Design and Development of Sensor Based Traffic Light System”, American Journal of Applied Sciences, pp. 1745,2006.

[7] Perera, S., and Sonnadara, U.“Night Time Detection of vehicles”,Proceedings of the Technical Sessions Institute of Physics-Sri Lanka,pp. 34-38, 2010.

[8] Walad, K., & Shetty, J.“Traffic Light Control System Using Image Processing”, International Journal of Innovative Research in Computer and Communication Engineering, Vol.2, Special Issue 5, pp. 289, 2014.

[9] Chandrasekhar, M., Saikrishna, C., Chakradhar, B., Phaneendra, P. & Sasanka, C. “Traffic Control Using Digital Image Processing” , International Journal of Advanced Electrical and Electronics Engineering, Vol-2, Issue-5, pp. 96 &100, 2013.

[10] Maheshwari, P., Suneja, D., Singh, P.“Traffic Image Process System”, International Journal of Computing Applications, Vol. 118, No. 23, pp. 17 & 18, 2015.

[11] Kanamaliga, S., Vasuki, S., Kanimozhidevil, A., Priyadharshni, S., and Rajeswari, S.“Tracking and Counting the Vehicle in Night Scenes”, International Journal of Information Science and Techniques, Vol. 4, No. 3, pp. 166-170, 2014.

[12] https://www.videezy.com/transportation/188-free-cars-driving-at-night-stock-video-in-hd

[13] http://agamenon.tsc.uah.es/Personales/rlopez/data/rtm/

[14] https://www.mathworks.com/help/images/examples/identifying-round-objects.html

[15] http://www.thedailystar.net/backpage/traffic-terrible-1201981

[16] https://www.mathworks.com/help/matlab/creating_plots/image-types.html

[17] https://www.mathworks.com/help/matlab/ref/hsv.html?searchHighlight=HSV&s_tid=do_srchtitle

[18] http://www.xinapse.com/Manual/masking.html

[19]https://www.mathworks.com/help/images/ref/imbinarize.html?searchHighlight=image%20binarization&s_tid=doc_srchtitle

[20] https://www.mathworks.com/help/images/ref/bwboundaries.html