Density Based Smart Traffic Control System Using Canny Edge Detection Algorithm for Congregating Traffic Information

Taqi Tahmid[1] and Eklas Hossain[2]

[1]Khulna University of Engineering & Technology, Dept. of EEE, Khulna 9203, Bangladesh

[2]Oregon Tech, Department of Electrical Engineering& Renewable Energy, OR-97601, USA

taqitahmid@gmail.com; eklas.hossain@oit.edu

***Abstract*—As the problem of urban traffic congestion intensifies, there is a pressing need for the introduction of advanced technology and equipment to improve the state-of-the-art of traffic control. The current methods used such as timers or human control are proved to be inferior to alleviate this crisis. In this paper, a system to control the traffic by measuring the real-time vehicle density using canny edge detection with digital image processing is proposed. This imposing traffic control system offers significant improvement in response time, vehicle management, automation, reliability and overall efficiency over the existing systems. Besides that, the complete technique from image acquisition to edge detection and finally green signal allotment using four sample images of different traffic conditions is illustrated with proper schematics and the final results are verified by hardware implementation.**

Keywords—Smart Traffic Control; Density based Traffic Control; Edge Detection; Image Processing in Traffic Control.

I. INTRODUCTION

Traffic congestion is one of the major modern-day crisis in every big city in the world. Recent study of World Bank has shown that average vehicle speed has been reduced from 21 km to 7 km per hour in the last 10 years in Dhaka [1]. Intermetropolitan area studies suggest that traffic congestion reduces regional competitiveness and redistributes economic activity by slowing growth in county gross output or slowing metropolitan area employment growth [2].As more and more vehicles are commissioning in an already congested traffic system, there is an urgent need for a whole new traffic control system using advanced technologies to utilize the already existent infrastructures to its full extent. Since building new roads, flyovers, elevated expressway etc. needs extensive planning, huge capital and lots of time; focus should be directed upon availing existing infrastructures more efficiently and diligently.

Previously different techniques had been proposed, such as infra-red light sensor, induction loop etc. to acquire traffic date which had their fair share of demerits. In recent years, image processing has shown promising outcomes in acquiring real time traffic information using CCTV footage installed along the traffic light. Different approaches have been proposed to glean traffic data. Some of them count total number of pixels [3], some of the work calculate number of vehicles [4-6].These methods have shown promising results in collecting traffic data. However, calculating the number of vehicles may give false results if the intravehicular spacing is very small (two vehicles close to each other may be counted as one) and it may not count rickshaw or auto-rickshaw as vehicles which are the quotidian means of traffic especially in South-Asian countries. And counting number of pixels has disadvantage of counting insubstantial materials as vehicles such as footpath or pedestrians. Some of the work have proposed to allocate time based solely on the density of traffic. But this may be disadvantageous for those who are in lanes that have less frequency of traffic.

Edge detection technique is imperative to extract the required traffic information from the CCTV footage. It can be used to isolate the required information from rest of the image. There are several edge detection techniques available. They have distinct characteristics in terms of noise reduction, detection sensitivity, accuracy etc. Among them, Prewitt [7], canny [8],Sobel [9], Roberts and LOG are most accredited operators. It has been observed that the Canny edge detector depicts higher accuracy in detection of object with higher entropy, PSNR(Peak Signal to Noise Ratio), MSE(Mean Square Error) and execution time compared with Sobel, Roberts, Prewitt, Zero crossing and LOG [10-12].Here is a comparison between distinct edge detection techniques [13].

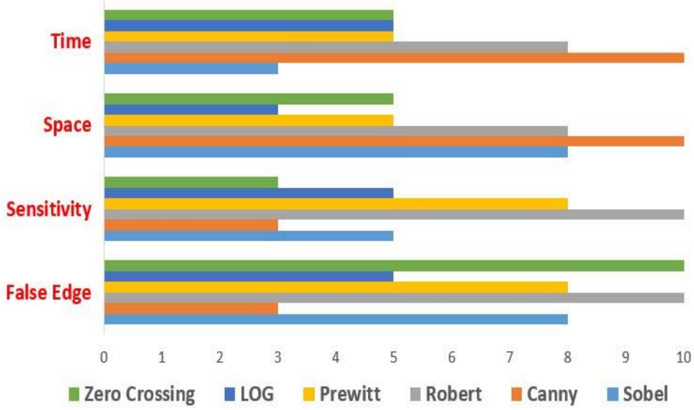


Fig 1: Comparative analysis of distinct edge detection techniques

In this paper, a system in which density of traffic is measured by comparing captured image with real time traffic information against the image of the empty road as reference image is proposed. Here, in figure 1, the block diagram for proposed traffic control technique is illustrated.

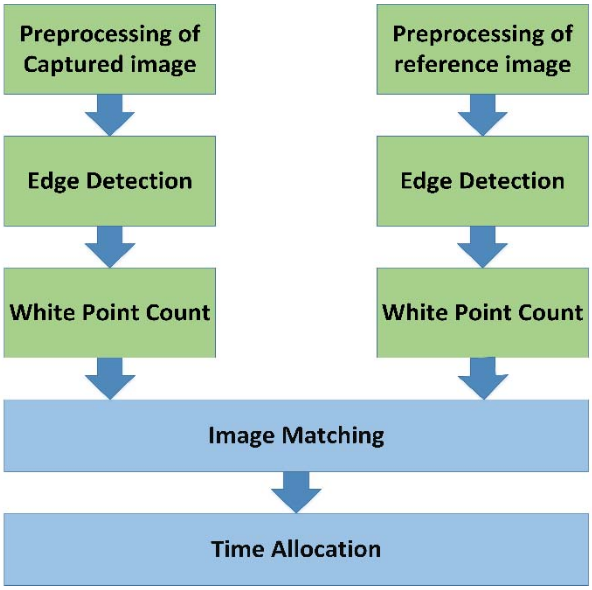


Fig 2: Block diagram of proposed density based smart traffic control system.

Each lane will have a minimum amount of green signal duration allocated. According to the percentage of matching allocated traffic light duration can be controlled. The matching is achieved by comparing the number of white points between two images. The entire image processing before edge detection i.e. image acquisition, image resizing, RGB to gray conversion and noise reduction is explained in section II. At section III, canny edge detection operation and white point count are depicted. Canny edge detector operator is selected because of its greater overall performance. Percentage matching for different sample images and traffic time allocation for them are demonstrated in section IV. The content of this paper completely serves the purpose of demonstrating the limitations of current traffic control techniques and the solution of this limitations with detailed explanation. Image matching by comparing detected edges is a novel approach to identify the vehicular density with propitious accuracy. As far as we know, matching images by comparing detected edges has not been used before for smart traffic control application.

II. IMAGE PREPROCESSING

In this section, image preprocessing is performed to convert the raw images into more accessible form for edge detection purpose. At first, four images of different traffic scenario are selected and image of the empty road is chosen as reference image. All the images are then resized into 400*400 pixel using the following formula for constant spatial resolution and greater computational efficiency.

Original height/original width*new width = new height (1)

Original width/original height*new height = new width (2)

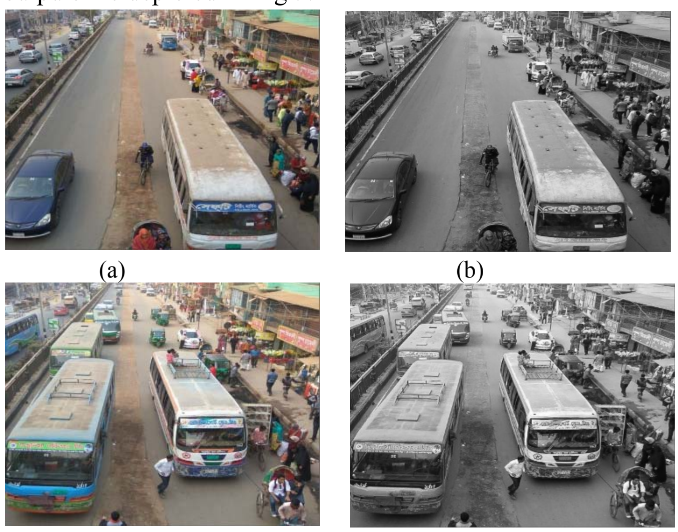
As grayscale image has superior signal to noise ration compared to RGB image, it is advantageous to convert RGB images into grayscale for further processing. When converting an RGB image to grayscale, it is pertinent to consider the RGB values for each pixel and make as output a single value reflecting the brightness of that pixel. One of the approaches is to take the average of the contribution from each channel: (R+B+C)/3. However, since the perceived brightness is often dominated by the green component, a different, more "human-oriented", method is to consider a weighted average:

$$I=0.3R + 0.59G + 0.11B \tag{3}$$

Since Gaussian noise is the preeminent source of noise in digital image, the cardinal purpose of applying Gaussian filter is to eliminate these noises. This step is crucial to impede noises associated with the digital images detected as false edge. The equation for a Gaussian filter kernel of size (2k+1)*(2k+1) is given by,

$$H_{ij} = \frac{1}{2\pi\sigma^2} \mathrm{e}^{\left(-\frac{\left(i-(k+1)\right)^2+\left(j-(k+1)\right)^2}{2\sigma^2}\right)} \tag{4}$$

It is important to understand that the selection of the size of the Gaussian kernel will affect the performance of the detector. The larger the size is, the lower the detector's sensitivity to noise. Additionally, the localization error to detect the edge will slightly increase with the increase of the Gaussian filter kernel size. A 5×5 is an appropriate size for most cases, but this will also vary depending on specific situations. A 5×5 Gaussian filter with sigma=1.4 is used in this paper. Once a suitable kernel has been selected, then the Gaussian smoothing can be performed using standard convolution methods. After image preprocessing the resultant outputs are depicted in Fig 3.



(a) (b) (c) (d)

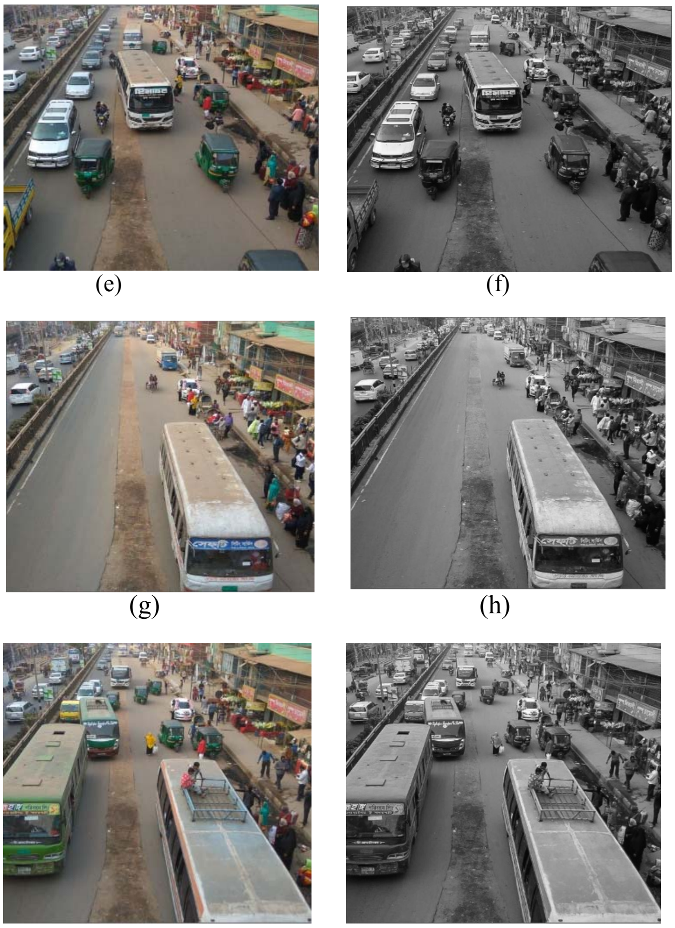


Fig 3: (a) First sample image (image A) with fairly congested traffic (b) Preprocessed output of first image (c) Second sample image (image B) with minor traffic (d) Preprocessed output of second image (e) Third sample image (image C)with moderate traffic (f) Preprocessed output of third image (g) Fourth sample image (image D) with moderate traffic (h) Preprocessed output of fourth image (i) Reference image (j) Preprocessed output of reference image

III. EDGE DETECTION AND WHITE POINT COUNT

Edge detection is used to identify distinct types of shapes. In this paper, it is used for isolating different shapes of vehicles from rest of the image. After comparing different edge detectors, canny edge detector is found suitable for this experiment. At first, images are smoothed by applying Gaussian filter to reduce unwanted texture and details,

$$g(x,y) = G\sigma(x,y) * I(x,y) \tag{5}$$

where,

$$G_\sigma(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{m^2+n^2}{2\sigma^2}\right)} \tag{6}$$

Then the intensity gradient is computed using Sobel gradient operator,

$$M(x,y) = \sqrt{g_m^2(x,y) + g_n^2(x,y)} \tag{7}$$

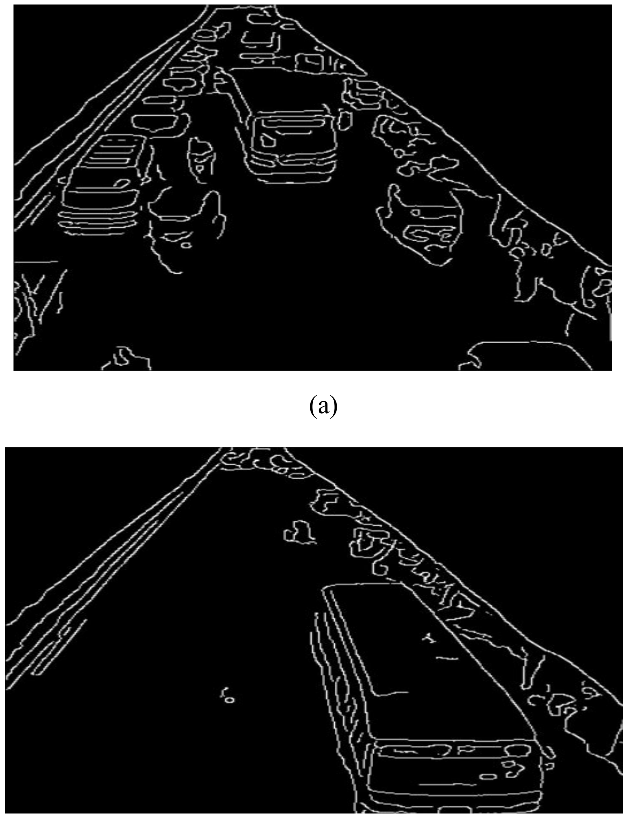
and,

$$\theta(x,y) = \tan^{-1} g_n(x,y) \big/ g_m(x,y) \tag{8}$$

Threshold M

$$M_T(x,y) = \begin{cases} M(x,y) & if\ M(x,y) > T \\ 0 & otherwise \end{cases} \tag{9}$$

Where T is so chosen that all edge elements are kept while most of the noise is suppressed. After that, non-maxima suppression is applied by checking whether each non-zero $M_T(x,y)$ is greater than its two neighbors along the gradient direction θ(x,y). If so, $M_T(x,y)$is kept unchanged, otherwise, set to 0. The resultant output examples after edge detection are delineated in Fig 4.



(a)

(b)

Fig 4: Edge detected output of (a) image A and (b) reference image

Other images also share the same resemblance. After edge detection, the resultant images are binary image with only black and white pixels. Binary image is originally a two-dimensional matrix of values 0 and 1. The value '0' denotes black color while value '1' denotes white color. The white pixels essentially represent the detected edges. So, images with different traffic conditions will have different white point counts. Since the reference image has least number of vehicles, it would have the least number of white pixels among these five images. Therefore, this image is used as a unit image to measure the traffic density. Total number of white pixels are calculated for each individual image intended for matching purpose.

IV. PERCENTAGE MATCHING AND TIME ALLOCATION

Since reference image is basically an image of the empty road, the less the similarity between sample image and reference image, the more vehicles are present on the road. As irrelevant edges detected in sample images such as edges of footpaths, islands etc. are also existent in reference image, their effects are nullified when comparing them with reference image. The percentage of matching is calculated using the following formula,

$$\%match = \frac{Total\ number\ of\ white\ points\ in\ reference\ image}{Total\ number\ of\ white\ points\ in\ sample\ image} * 100 \quad (9)$$

Time allocated to green signal is governed by the percentage of matching of consecutive lanes. The proposed time allocation is based on assumption. Contemporary time allocation may depend on miscellaneous factors; for instance, number of vehicles, traffic condition on neighboring intersections etc. For example, if the vehicle queue is larger than average, it would require longer time to clear. Additionally, it would be pernicious if other vehicles from neighboring intersections exacerbate the traffic congestion of adjacent intersections.

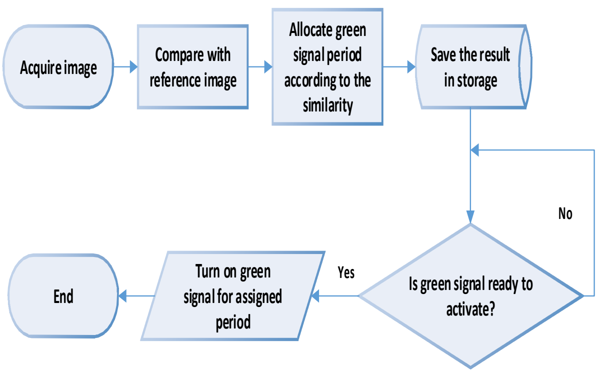
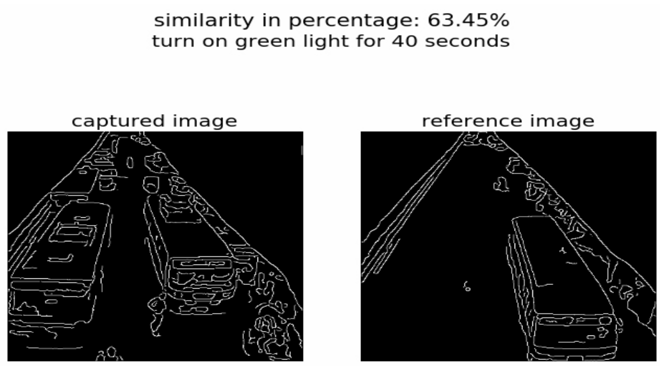


Fig 5: Proposed time allocation algorithm for autonomous traffic control system

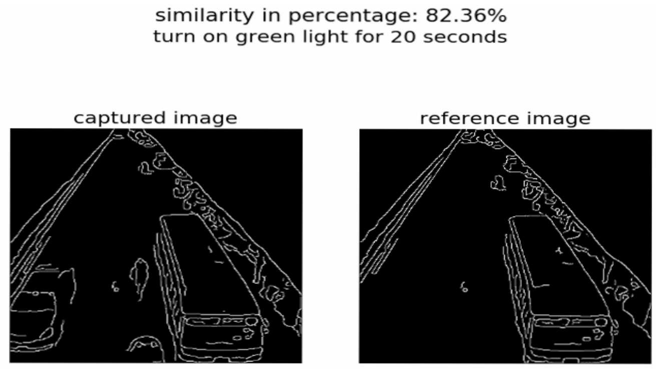
Table 1: Green signal allocation table.

Similarity in Percentage	Allocated Green Signal Time
0%-50%	60 seconds
51%-60%	50 seconds
61%-70%	40 seconds
71%-80%	30 seconds
81%-100%	20 seconds

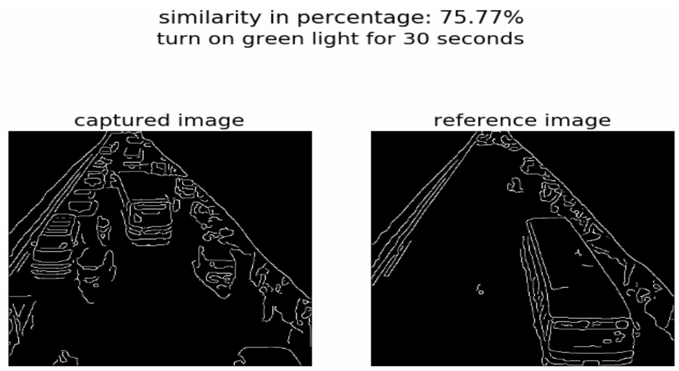
Now implementing the proposed algorithm, four different similarity in percentage and consequently different allotted green signal time are found for four sample images. The similarity in percentage and allocated green signal time are calculated using python programming language. These outcomes are illustrated in Fig 6.



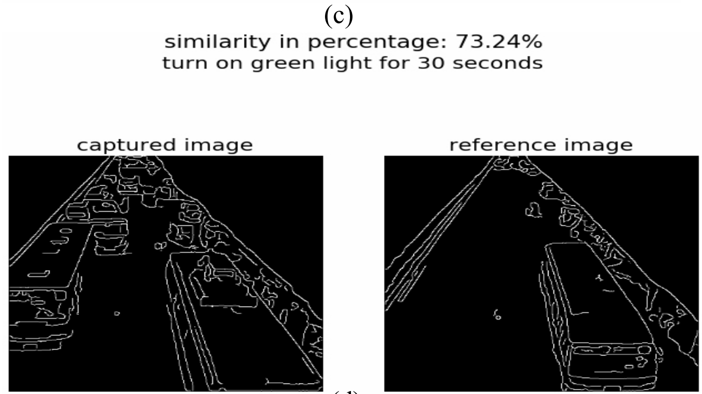
(a)



(b)



(c)



(d)

Fig 6: (a) Similarity between first sample image and reference image is found 63.45% (b) Similarity between second sample image and reference image is found 82.36% (c) Similarity between third sample image and reference image is found 75.77% (d) Similarity between fourth sample image and reference image is found 73.24%

All this collected information are composed in the following table.

Table 2: Allocated time derived from percentage matching

Designation	**Similarity in Percentage**	**Allocated Green Signal Period**
Image A	63.45%	40 seconds
Image B	82.36%	20 seconds
Image C	75.77%	30 seconds
Image D	73.24%	30 seconds

V. HARDWARE IMPLEMENTATION

To verify the above autonomous traffic control technique, a four-way traffic intersection is modeled. Each sample image is used to represent the density of each traffic lane. This four-way intersection is rendered by four arrays of LEDs with each array encompassing a red and a green LED. Python programming language is used for image processing and Arduino development board is used to control the LEDs.

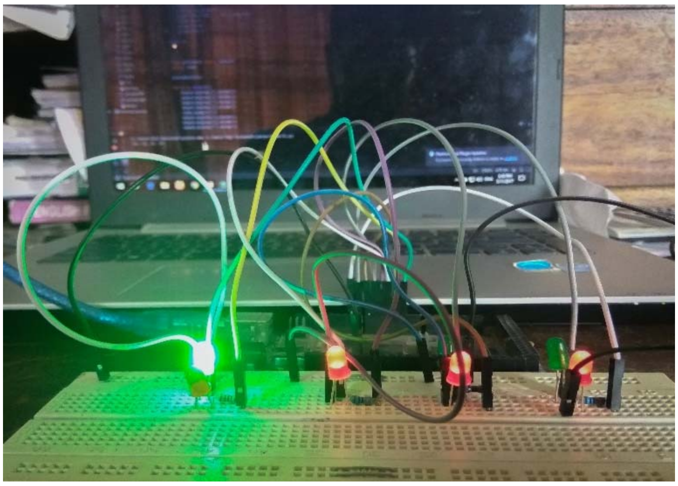


Fig 7: Hardware implementation of a four way traffic intersection

The inputs of these LEDs are connected to the digital I/O pins of the Arduino. All of the pins have common ground connected to the ground of Arduino. The first array of red and green LEDs is considered as lane A. Second, third and fourth arrays of LEDs are considered as lane B, lane C and lane D respectively. These LEDs are controlled by the output pins of the Arduino, which are controlled by the time, allocated to each consecutive lane. At first, the green signal for lane A is activated for 40 seconds. Throughout this time, red signals of other lanes are turned on concurrently. Similarly, after lane A, green signals of second, third and fourth lanes are sequentially activated for 20 seconds, 30 seconds and 30 seconds respectively. Whenever the green signal of one lane is activated, red signals of other lanes are also activated simultaneously.

VI. CONCLUSION

In this paper, a smart traffic control system availing image processing as an instrument for measuring the density has been proposed. Besides explaining the limitations of current near obsolete traffic control system, the advantages of proposed traffic control system have been demonstrated. For this purpose, four sample images of different traffic scenario have been attained. Upon completion of edge detection, the similarity between sample images with the reference image has been calculated. Using this similarity, time allocation has been carried out for each individual image in accordance with the time allocation algorithm. In addition, similarity in percentage and time allocation have been illustrated for each of the four sample images using Python programming language. Besides presenting the schematics for the proposed smart traffic control system, all the necessary results have been verified by hardware implementation.

REFERENCES

[1] Siddique, A. (2017). *WB: Dhaka's average traffic speed 7kmph*. [online] Dhaka Tribune. Available at: http://www.dhakatribune.com/bangladesh/dhaka/2017/07/19/dhaka-average-traffic-speed-7-kmph/ [Accessed 25 Jul. 2017].

[2] M. Sweet, "Traffic Congestion's Economic Impacts: Evidence from US Metropolitan Regions," *Urban Studies*, vol. 51, no. 10, pp. 2088–2110, Oct. 2013.

[3] Md. Munir Hasan, Gobinda Saha, Aminul Hoque and Md. Badruddoja Majumder, "Smart Traffic Control System with Application of Image Processing Techniques,"in *3rd International Conference on Informatics, Electronics & Vision*, Dhaka, May 2014.

[4] Vismay Pandit, Jinesh Doshi, Dhruv Mehta, Ashay Mhatre and Abhilash Janardhan, "Smart Traffic Control System Using Image Processing,"*International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, Vol. 3, Issue 1, January – February 2014.

[5] Pallavi Choudekar, Sayanti Banerjee and M. K. Muju, "Implementation of image processing in real time traffic light control," in*3rd International Conference on Electronics Computer Technology*, April, 2011.

[6] T. Semertzidis, K. Dimitropoulos, A. Koutsia and N. Grammalidis, "Video sensor network for real-time traffic monitoring and surveillance,"*The Institution of Engineering and Technology*, Volume 4, Issue 2, June 2010, pp. 103 – 112.

[7] Lei Yang, Dewei Zhao and Xiaoyu Wu, "An improved Prewitt algorithm for edge detection based on noised image,"in *4th International Congress on Image and Signal Processing*, Shanghai, China, October 2011.

[8] Weibin Rong, Zhanjing Li and Wei Zhang, "An improved canny edge detection algorithm,"in *IEEE International Conference on Mechatronics and Automation*, August 2014.

[9] Wenshuo Gao, Xiaoguang Zhang and Lei Yang, "An improved Sobel edge detection," in*3rd International Conference on Computer Science and Information Technology*, July 2010.

[10] G.T. Shrivakshan and Dr.C. Chandrasekar, "A Comparison of various Edge Detection Techniques used in Image Processing,"*International Journal of Computer Science Issues*, Vol. 9, Issue 5, No 1, September 2012.

[11] Pinaki Pratim Acharjya, Ritaban Das and Dibyendu Ghoshal, "Study and Comparison of Different Edge Detectors for Image Segmentation,"*Global Journal of Computer Science and Technology Graphics& Vision*, Volume 12 Issue 13 Version 1.0 Year 2012.

[12] Jiaqiao He, Liya Hou and Weiyi Zhang, "A Kind of Fast Image Edge Detection Algorithm Based on Dynamic Threshold Value,"*Sensors & Transducers*, Vol. 24, Special Issue, August 2013, pp. 179-183.

[13] S.K. Katiyar and P.V. Arun, "Comparative analysis of common edge detection techniques in context of object extraction,"*IEEE Transactions on Geoscience and Remote Sensing,* Vol.50 no.11b.