

DENSITY BASED SMART TRAFFIC CONTROL SYSTEM USING CANNY EDGE DETECTION

**E. LINGAPPA¹, ELAKURTHI ARAVIND YADAV², MANDA GANESH³, JANGILI
SUMAN PATEL⁴, MUDDANGALA NITHISH KUMAR⁵, MOHAMMAD SALMAN⁶**

¹Assistant Professor, Department of CSE-IOT, Malla Reddy College of Engineering Hyderabad,
TS, India.

^{2,3,4,5,6} UG students, Department of CSE-IOT, Malla Reddy College of Engineering Hyderabad,
TS, India.

ABSTRACT:

In this paper author is describing concept to control or automate green traffic signal allotment time based on congestion available at road side using Canny Edge Detection Algorithm. To implement this technique we are uploading current traffic image to the application and application will extract edges from images and if there is more traffic then there will be more number of edges with white colour and if uploaded image contains less traffic then it will have less number of white colour edges. Empty edges will have black colour with value 0. By counting number of non-zeroes white pixels we will have complete idea of available traffic and based on that we will allocate time to green signal. If less traffic is there then green signal time will be less otherwise green signal allocation time will be more.

Keywords: Password, graphical, high security.

1. 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. 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.

2. EXICITING SYSTEM:

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].

To implement this technique we are uploading current traffic image to the application and application will extract edges from images and if there is more traffic then there will be more number of edges with white colour and if uploaded image contains less traffic then it will have less number of white colour edges.

Proposed System :-

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.

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.

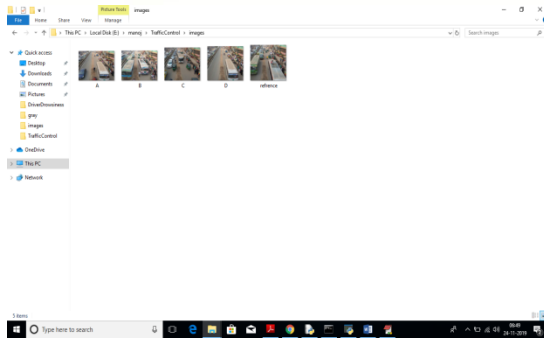
image acquisition, image resizing, RGB to gray conversion and noise reduction is explained in section II. At section III, canny edge detection operation and white point count are depicted. Canny edge detector operator is selected because of its greater overall performance.

Advantages :-

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: $(R+B+C)/3$.

3. METHODOLOGY

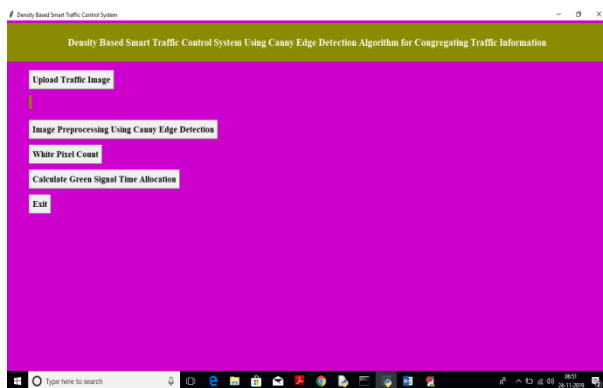
To implement this project we are using 4 input images given in paper and on reference image. Below are the images screen shots saved inside images folder



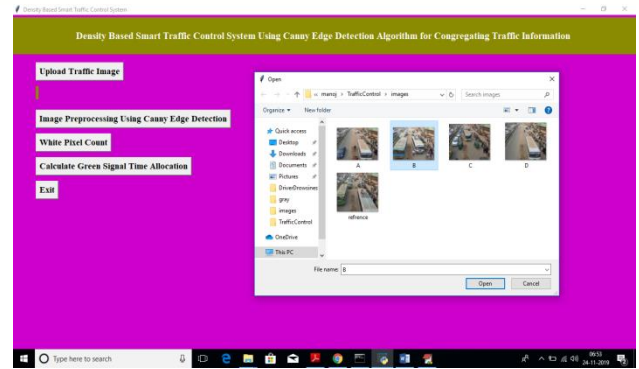
We can upload above 4 images to application to calculate traffic signal time.

Screen shots

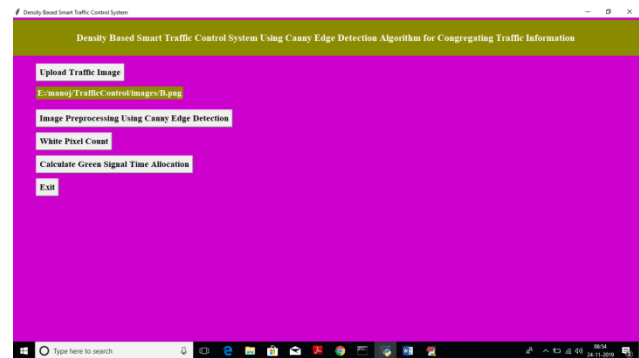
To run this project double click on 'run.bat' file to get below screen



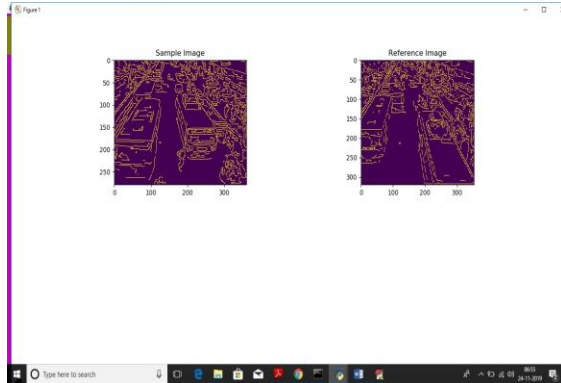
In above screen click on 'Upload Traffic Image' button to upload image.



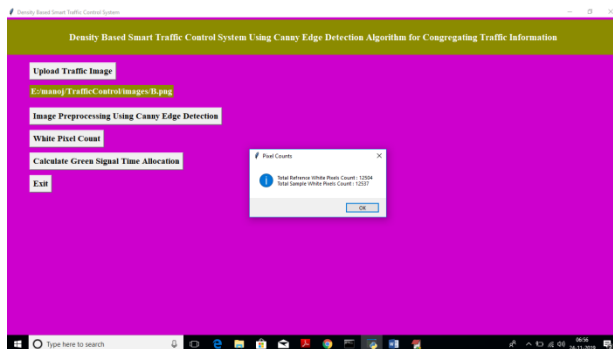
In above screen I am uploading image B and now click on 'Open' button to load image



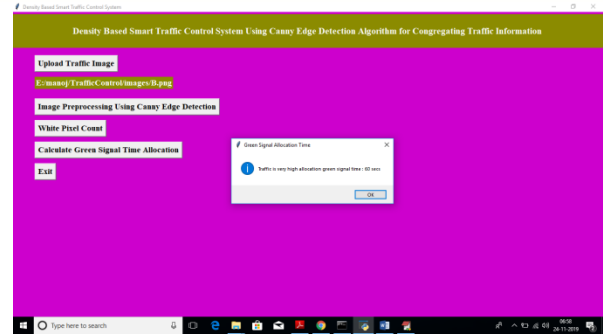
In above screen we got message as input image loaded. Now click on 'Image Pre-processing Using Canny Edge Detection' button to apply Gaussian filter and to get canny edges, after clicking button wait for few seconds till you get below screen with edges



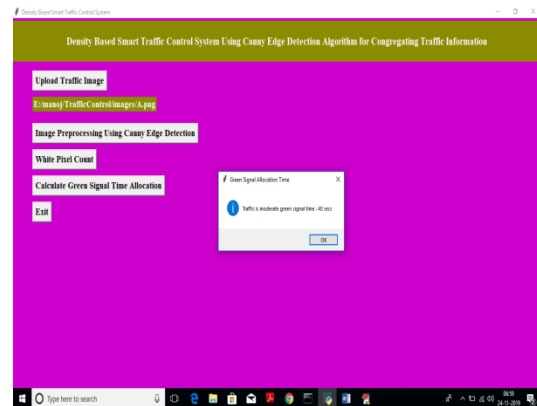
In above screen left side image is the uploaded image and right side is the 'Reference Image', Now close this above screen and click on 'White Pixel count' button to get white pixels from both images



In above screen dialog box we can see total white pixels found in both sample and reference image. Now click on 'Calculate Green Signal Time Allocation' button to get signal time



For that uploaded image we got message as it contains high traffic and signal time must be 60 seconds. Similarly you can upload any image and get output. Below is the output for image A



Above time for image A

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.

Scope for future work

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] William Stallings and Lawrie Brown. Computer Security: Principle and Practices. Pearson Education, 2008.
- [2] Susan Wiedenbeck, Jim Waters, Jean-Camille Birget, Alex Brodskiy, and Nasir Memon. Passpoints: design and longitudinal evaluation of a graphical password system. International Journal of Human-Computer Studies, 63:102–127, July 2005.
- [3] Robert Morris and Ken Thompson. Password security: a case history. Communications of the ACM, 22:594–597, November 1979.
- [4] Daniel V. Klein. Foiling the Cracker: A Survey of, and Improvements to, Password Security. In Proceedings of the

- 2nd USENIX UNIX Security Workshop, 1990.
- [5] Eugene H. Spafford. Observing reusable password choices. In Proceedings of the 3rd Security Symposium. Usenix, pages 299–312, 1992.
- [6] Sigmund N. Porter. A password extension for improved human factors. *Computers & Security*, 1(1):54 – 56, 1982.
- [7] Xiaoyuan Suo, Ying Zhu, and G. Scott Owen. Graphical passwords: A survey. In *Proceedings of Annual Computer Security Applications Conference*, pages 463–472, 2005.
- [8] Antonella De Angeli, Lynne Coventry, Graham Johnson, and Karen Renaud. Is a picture really worth a thousand words? exploring the feasibility of graphical authentication systems. *International Journal of Human-Computer Studies*, 63:128–152, July 2005.
- [9] Real User Corporation. The science behind passfaces, June 2004.
- [10] G. E. Blonder. Graphical password. U.S. Patent 5559961, Lucent Technologies, Inc. (Murray Hill, NJ), August 1995.