



Congregating of Traffic Information using Density based Smart Traffic Control System

S Pooja Redy¹, T Manoj Kumar Reddy², P Shiva Chandra³, Mohammad Izhaar Ahmed⁴,
Avancha Deepak Kumar⁵

^{2,3,4,5} UG Scholars, Department of CSE, AVN Institute of Engineering and
Technology, Hyderabad, Telangana, India.

¹ Assistant Professor, Department of CSE, AVN Institute of Engineering and Technology,
Hyderabad, Telangana, 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. To compare current traffic we will take one reference image with high traffic and comparison will be done between uploaded image white pixels and reference image white pixels. Using below codewe will allocate time to green signal.

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. 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. 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, some of the work calculate number of vehicles. 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 works 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.

Existing 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, canny, Sobel, 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. Here is a comparison between distinct edge detection techniques.

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

Functional requirements:-

In software engineering, a functional requirement defines a system or its component. It describes the functions a software must perform. A function is nothing but inputs, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform.

Functional software requirements help you to capture the intended behavior of the system. This behavior may be expressed as functions, services or tasks or which system is required to perform.

Non –Functional Requirements:-

A non-functional requirement defines the quality attribute of a software system. They represent a set of standards used to judge the specific operation of a system. Example, how fast does the website load?

A non-functional requirement is essential to ensure the usability and effectiveness of the entire software system. Failing to meet non-functional requirements can result in systems that fail to satisfy user needs.

project and a detailed project overview.

Modules of the Application:

Upload Image Module:

In this module current traffic image will be uploaded to

application and then convert colour image into Gray Scale image format to have pixels values as black and white colour.

Pre-process module:

In this module Gaussian Filter will be applied on uploaded image to convert image into smooth format. After applying filter Canny Edge Detection will be applied on image to get edges from the image. Each vehicle will have white colour pixels and non-vehicle will have black colour pixels.

White Pixel Count Module:

Using this module we will count white pixels from canny image to get complete traffic count

SCREEN CAPTURES

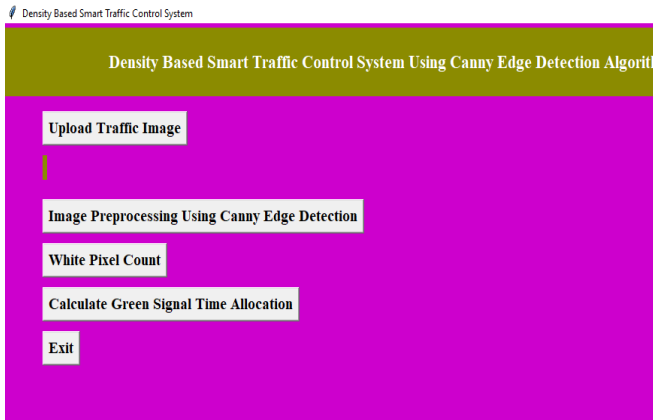
User Login Screen:

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

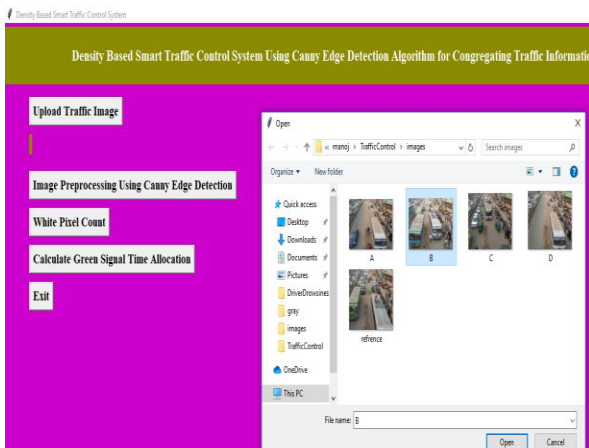


We can upload above 4 images to application to calculate traffic signal time.

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



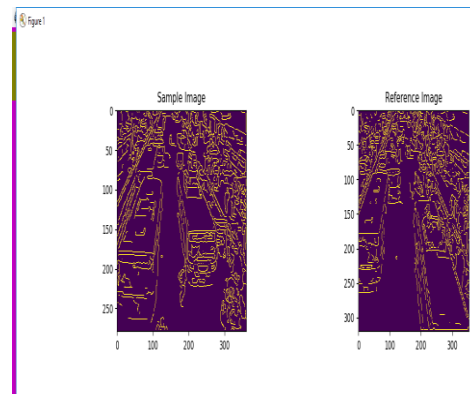
In above screen click on 'Upload Traffic Image' button to upload image.



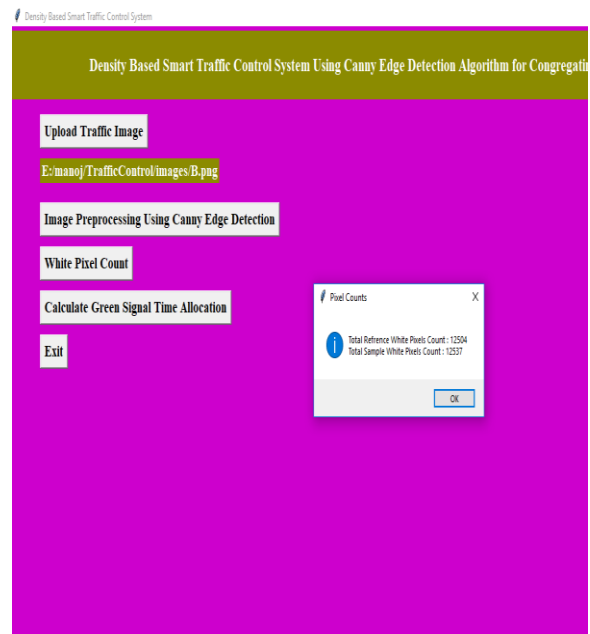
In above screen I am uploading image B and now click on 'Open' button to load image



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

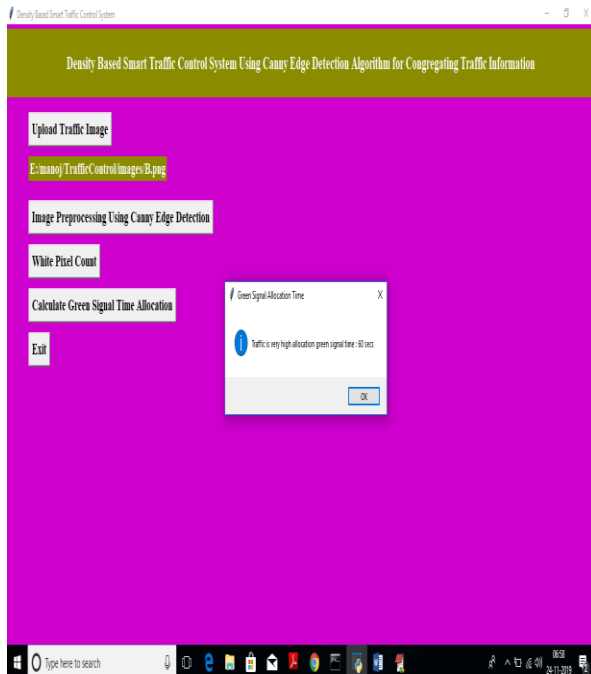


In above screen left side image is the uploaded image and right side is the 'Reference Image', Now close this above screen and click on 'White Pixel count' button to get white pixels from both images

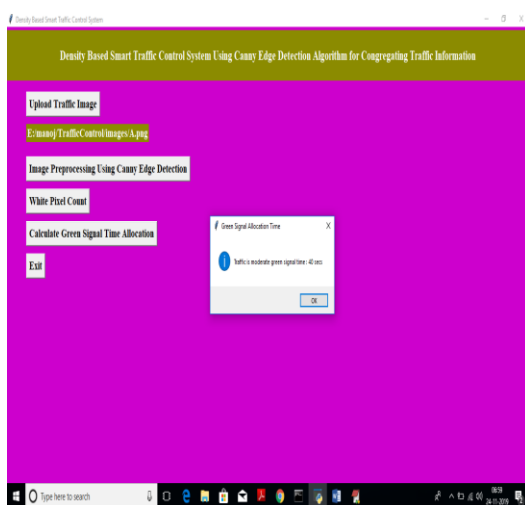


In above screen dialog box we can see total white pixels found in both sample

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

Results

The current android application is developed using Xml, Java, SQL

with Firebase connectivity. It can be used by every individual who are in a need of fulfilling their household services.

At the time of submission of my application was capable of doing the following:

- Displaying the home screen with different fragments.
- Authentication of user by using login screen using Firebase.
- Home screen to display based on user or service provider.
- After successful login of user, they can choose the service and book a slot of their particular service provider from the displayed list.
- Add, update, view, delete the user details.
- After successful login of service provider, they can view all the bookings that are booked by the users and can attend them one by one.
- Service provider can also set his preferences to not available, if he's too busy or many users had already booked him.
- Service provider has the ability to change their particular radius of location for servicing.
- He can set up to 10 km radius.
- Logout and end the session.

Challenges

- Understanding the connections of SQLite Database is a tricky part and confusing when dealing with



multiple tables within a database.

- Making exact orientation API design levels was a difficult task as there are many types of devices like desktop, tablet, mobile with varying screen size and resolutions.
- Implementing synchronization with Firebase was a challenging task.
- Learning different technologies and frameworks with little guidance.

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.

BIBLIOGRAPHY

Code snippets for any errors

<http://stackoverflow.com/>

Android Development Guide

<https://www.udemy.com/android>

Xml and Layout Guide

<https://www.androidhive.com/>

Connecting to Firebase Docs

<https://firebase.google.com>

Software Testing

http://en.wikipedia.org/wiki/Software_testing

Manual Testing

http://en.wikipedia.org/wiki/Manual_testing

Performance Testing

http://en.wikipedia.org/wiki/Software_performance_testing