SMART TRAFFIC MONITORING SYSTEM PROJECT REPORT

2

INDEX

ABSTRACT

Congestion in traffic is a major problem these days. Despite the fact that it appears to pervade all over, urban cities are the ones most influenced by it. And it's ever increasing nature makes it imperative to know the road traffic density in real time for better signal control and effective traffic management. There can be different causes of congestion in traffic like insufficient capacity, unrestrained demand, large Red Light delays etc. While insufficient capacity and unrestrained demand are somewhere interrelated, the delay of respective light is hard coded and not dependent on traffic. Therefore, the need for simulating and optimizing traffic control to better accommodate this increasing demand arises. In recent years, image processing and surveillance systems have been widely used in traffic management for traveller's information, ramp metering and updates in real time. The traffic density estimation can also be achieved using Image Processing. This project presents the method to use live images feed from the cameras at traffic junctions for real time traffic density calculation using image processing. It also focuses on the algorithm for switching the traffic lights according to vehicle density on road, thereby aiming at reducing the traffic congestion on roads which will help lower the number of accidents. In turn it will provide safe transit to people and reduce fuel consumption and waiting time. It will also provide significant data which will help in future road planning and analysis. In further stages multiple traffic lights can be synchronized with each other with an aim of even less traffic congestion and free flow of traffic. The vehicles are detected by the system through images instead of using electronic sensors embedded in the pavement. A camera will be placed alongside the traffic light. It will capture image sequences. Image processing is a better technique to control the state change of the traffic light. It shows that it can decrease the traffic congestion and avoids the time being wasted by a green light on an empty road. It is also more reliable in estimating vehicle presence because it uses actual traffic images. It visualizes the practicality, so it functions much better than those systems that rely on the detection of the vehicles' metal content.

1 Introduction

Over the last decade, the adoption and use of technologies like Mobility, Cloud and Social Platforms has made it possible for common, middle class users to use small, focused applications for making their life easier and comfortable. Whether it is simply paying your utility bills using mobile banking or getting that favourite movie ticket by just clicking couple of buttons, use of technology has really changed the way we live, play and work. Though we have been referring to Smart Cities and communities for some time now, let us look at how use of Information and data available to us can be used to really create some smart services, which in a true sense provide us with better living. We shall look at a key case, which impacts us almost daily: traffic management. Use of technology and real time analysis can actually lead to a smooth traffic management. The common reason for traffic congestion is due to poor traffic prioritization, where there are such situations some lane has less traffic than the other. Vehicular congestion is increasing at an exponential rate. Let us take the case study of Chandigarh, one of the Union Territories of India. Chandigarh has the largest number of vehicles per capita in India. According to Chandigarh Transport Undertaking, more than 45,000 vehicles were registered this year in Chandigarh making the total count of more than 8 lakhs vehicles on the road. While the number of vehicles are increasing at a fast pace, the infrastructure in the city is not being able to match this growth. Traffic jams during rush hours are becoming a routine affair, especially in the internal sectors where long queues of vehicles can be seen stranded. Therefore, we have tried to address the problem with the help of our project wherein the focus would be to minimize the vehicular congestion. We have achieved this with the help of image processing that can be obtained from surveillance cameras and eventually to deploy a feedback mechanism in the working of the traffic lights where the density of the traffic would also be factored in the decision making process.

2 Purpose

Road transport is one of the primitive modes of transport in many parts of the world today. The number of vehicles using the road is increasing exponentially every day. Due to this reason, traffic congestion in urban areas is becoming unavoidable these days. Inefficient management of traffic causes wastage of invaluable time, pollution, wastage of fuel, cost of transportation and stress to drivers, etc. Our research is on density based traffic control. So, it is very much necessary to design a system to avoid the above casualties thus preventing accidents, collisions, and traffic jams. Connecting Smart Traffic Management System of the city and using the power of analytics is a key to smooth traffic management. Using real time analytics of data from these sources and linking them to some trends, we can manage traffic flow much better.

3 Outcomes

There are several drawbacks of earlier methods - Wastage of time by lighting green signal even when road is empty. Image processing removes such problem. Slight difficult to implement in real time because the accuracy of time calculation depends on relative position of camera. This project provides a solution to reduce traffic congestion on roads overriding the older system of hard coded lights which cause unwanted delays. Reducing congestion and waiting time will lessen the number of accidents and also reduces fuel consumption which in turn will help in controlling the air pollution. This will also provide data for future road design and construction or where improvements are required and which are urgent like which junction has higher waiting times.

4 Literature Survey

An excessive amount of research has been conducted to mitigate the problem of vehicular congestion.

[4] Khekare, G.S., Sakhare A.V. proposed the development of VANETs (Vehicular Ad Hoc Networks), which are the quintessential of the new types of networks emerging in the wireless technologies. The salient features of VANETs are to provide communication between vehicles themselves and between vehicles and road side units. VANET also plays an important role in concepts such as smart cities. The paper is based on a framework of a smart city that will transmit information about traffic conditions and will go a long way in aiding drivers to take spontaneous and smart decisions to prevent themselves from vehicular congestion which will ultimately help in reducing the overall congestion.

[5] Badura S., Lieskovsky A. presented a new model for intelligent traffic systems which will encapsulate the features of surveillance via the cameras present on the junction and with the help of data delivery systems let the users access that data. Image Analysis and foreground/background modeling schemes would be the important elements of Surveillance and data transmission over a mobile Ad-hoc network will comprise the data delivery part of the entire system. Various experiments have been conducted in the project and they exhibit great potential in terms of efficiency and real time execution.

[7] Salama A.S., Saleh B.K. and Eassa M.M. provide a design of an integrated intelligent system for management and controlling traffic lights with the help of Photoelectric Sensors. The installation of the sensors is a very important criterion in this system because the traffic management department has to monitor cars moving at a specific traffic and then to transfer this data to traffic control cabinet which can then control the traffic lights according to the sensor's readings by employing an algorithm based on the relative weight of each road. With the calculation of the relative weight of each road, the system will then open the traffic for that road which is more crowded and give it a longer time as compared to the other less congested roads. The real time decision making ability of the system stands out very saliently. Moreover, the system can also be programmed for emergency scenarios such as passing of presidents, ministries, ambulance vehicles and fire-trucks that require virtually zero congestion through an active RFID based technology. As a result the system will guarantee the fluency of traffic for such emergency cases or for the main vital streets and paths that require the fluent traffic all the time, without affecting the fluency of traffic generally at normal streets according to the time of the day and the traffic density. Also the proposed system can be tuned to run automatically without any human intervention or can be tuned to allow human intervention at certain circumstances. [8] Haimeng Zhao, Xifeng Zheng, Weiya Liu presented a design of intelligent traffic control system based on DSP and Nios II. Their model of intelligent traffic control system deploys dual-CPU combined with the logic control of FPGA (Field Programmable Gate Array) which involves functions like cross-phase adjustment, exchanging and establishing related information and live human-computer interaction. In order to achieve vehicular congestion, it is different from the conventional traffic signal controller in way that it works mostly at the mode of timing and multiple phases according to the user demands dynamically. Both the hardware and software system are realised in the paper. The system proposed by Sakhare, Khekare [4] suffers from a limitation that to

implement VANET the appropriate hardware has to be installed on every vehicle which can be comparatively difficult to install in a two-wheeler. Moreover, the entire framework is user dependent as the overall traffic congestion will depend on the decisions made by the user. The model designed by Salama [7] requires the deployment of photoelectric sensors and by Zhao [8] requires logic control with the help of FPGA. Both these systems demand constant maintenance both in monetary terms and system analysis. All the more, they are comparatively more prone to damage due to the rugged external conditions in which are deployed. The method proposed by us overcomes the limitations of Khekare [4] as it is implemented on a four-way junction and has no relation to every automobile that crosses it apart from its vehicle density and as the only hardware employed in our research are the surveillance cameras on the four-way junctions therefore no need of constant maintenance and less prone to failure as is the case with Salama [7] and Zhao [8].

5 Proposed System

We propose a technique that can be used for traffic control using image processing. Traffic density of lanes is calculated using image processing which is done of images of lanes that are captured using digital camera. According to the traffic densities on all roads, our model will allocate smartly the time period of green light for each road. We have chosen image processing for calculation of traffic density as cameras are very much cheaper than other devises such as sensors.

6 Conceptual Model

Introduction to Image Processing

Image Processing is a technique to enhance raw images received from cameras/sensors placed on space probes, aircrafts and satellites or pictures taken in normal day-today life for various applications. An Image is rectangular graphical object. Image processing involves issues related to image representation, compression techniques and various complex operations, which can be carried out on the image data. The operations that come under image processing are image enhancement operations such as sharpening, blurring, brightening, edge enhancement etc. Image processing is any form of signal processing for which the input is an image, such as photographs or frames of video; the output of image processing can be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Image processing usually refers to digital image processing, but optical and analog image processing are also possible.

Image Acquisition

Generally an image is a two-dimensional function f(x,y)(here x and y are plane coordinates).The amplitude of image at any point say f is called intensity of the image. It is also called the grey level of image at that point. We need to convert these x and y values to finite discrete values to form a digital image. The input image is a fundus taken from stare data base and drive data base. The image of the retina is taken for processing and to check the condition of the person. We need to convert the analog image to digital image to process it through digital computer. Each digital image composed of a finite elements and each finite element is called a pixel.

Formation of Image

We have some conditions for forming an image f(x,y) as values of image are proportional to energy radiated by a physical source. So f(x,y) must be nonzero and finite.
i.e. $0 < f(x,y) < \infty$.

Image Pre-Processing

- **Image Resizing/Scaling**

  Image scaling occurs in all digital photos at some stage It happens anytime you resize your image from one pixel grid to another. Image resizing is necessary when you need to increase or decrease the total number of pixels. Even if the same image resize is performed, the result can vary significantly depending on the algorithm.

- **RGB to GRAY Conversion**

  Humans perceive color through wavelength-sensitive sensory cells called cones. There are three different varieties of cones, each has a different sensitivity to electromagnetic radiation (light) of different wavelength. One cone is mainly sensitive to green light, one to red light, and one to blue light. By emitting a restricted combination of these three colors (red, green and blue), and hence stimulate the three types of cones at will, we are able to generate almost any detectable color. This is the reason behind why color images are often stored as three separate image matrices; one storing the amount of red (R) in each pixel, one the amount of green (G) and one the amount of blue (B). We call such color images as stored in an RGB format. In grayscale images, however, we do not differentiate how much we emit of different colors, we emit the same amount in every channel. We will be able to differentiate the total amount of emitted light for each pixel; little light gives dark pixels and much light is perceived as bright pixels. When converting an RGB image to grayscale, we have 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, e.g.: 0.3R + 0.59G + 0.11B.

- **Image Enhancement**

Image enhancement is the process of adjusting digital images so that the results are more suitable for display or further analysis. For example, we can eliminate noise, which will make it easier to identify the key characteristics. In poor contrast images, the adjacent characters merge during binarization. We have to reduce the spread of the characters before applying a threshold to the word image. Hence, we introduce "**Power-Law Transformation**" which increases the contrast of the characters and helps in better segmentation. The basic form of power-law transformation is

$$s = cr^{\gamma},$$

where r and s are the input and output intensities, respectively; c and γ are positive constants. A variety of devices used for image capture, printing, and display respond according to a power law. By convention, the exponent in the power-law equation is referred to as gamma. Hence, the process used to correct these power-law response phenomena is called gamma correction. Gamma correction is important, if displaying an image accurately on a computer screen is of concern. In our experimentation, γ is varied in the range of 1 to 5. If c is not equal to ’1’, then the dynamic range of the pixel values will be significantly affected by scaling. Thus, to avoid another stage of rescaling after power-law transformation, we fix the value of c = 1. With $\gamma = 1$, if the power-law transformed image is passed through binarization, there will be no change in the result compared to simple binarization. When $\gamma > 1$, there will be a change in the histogram plot, since there is an increase of samples in the bins towards the gray value of zero. Gamma correction is important if displaying an image accurately on computer screen is of concern.

- **Edge Detection**

Edge detection is the name for a set of mathematical methods which aim at identifying points in a digital image at which the image brightness changes sharply or, more technically, has discontinuities or noise. The points at which image brightness alters sharply are typically organized into a set of curved line segments termed edges. Edge detection is a basic tool in image processing, machine vision and computer envisage, particularly in the areas of feature reveal and feature extraction.

Following are list of various edge-detection methods:-

Sobel Edge Detection Technique

Perwitt Edge Detection

Roberts Edge Detection Technique

Zerocross Threshold Edge Detection Technique

Canny Edge Detection Technique

In our project we use “**Canny Edge Detection Technique**” because of its various advantages over other edge detection techniques.

- **Canny Edge Detection**

The Canny Edge Detector is one of the most commonly used image processing tools detecting edges in a very robust manner. It is a multi-step process, which can be implemented on the GPU as a sequence of filters. Canny edge detection technique is based on three basic objectives.

1) **Low error rate:**

   All edges should be found, and there should be no spurious responses. That is, the edges must be as close as possible to the true edges.

2) **Edge point should be well localized:**

The edges located must be as close as possible to the true edges. That is, the distance between a point marked as an edge by the detector and the center of the true edge should be minimum.

3) **Single edge point response:**

The detector should return only one point for each true edge point. That is, the number of local maxima around the true edge should be minimum. This means that the detector should not identify multiple edge pixels where only a single edge point exists.

The Canny edge detection algorithm consist of the following basic steps;

i. Smooth the input image with Gaussian filter.

ii. Compute the gradient magnitude and angle images.

iii. Apply non-maxima suppression to the gradient magnitude image.

iv. Use double thresholding and connectivity analysis to detect and link edges.

- **Image Matching**

Recognition techniques based on matching represent each class by a prototype pattern vector. An unknown pattern is assigned to the class to which is closest in terms of predefined metric. The simplest approach is the minimum distance classifier, which, as its name implies, computes the (Euclidean) distance between the unknown and each of the prototype vectors. It chooses the smallest distance to make decision. There is another approach based on correlation, which can be formulated directly in terms of images and is quite intuitive. We have used a totally different approach for image matching. Comparing a reference image with the real time image pixel by pixel. Though there are some disadvantages related to pixel based matching but it is one of the best techniques for the algorithm which is used in the project for decision making. Real image is stored in matric in memory and the real time image is also converted in the desired matric. For images to be same their pixel values in matrix must be same. This is the simplest fact used in pixel matching. If there is any mismatch in pixel value it adds on to the counter used to calculate number of pixel mismatches. Finally, percentage of matching is expressed as

$$\%match = \frac{\text{No.of pixels matched sucessfully}}{\text{total no.of pixels}}$$

Introduction to Cloud Computing

A *cloud* refers to a distinct IT environment that is designed for the purpose of remotely provisioning scalable and measured IT resources. The term originated as a metaphor for the Internet which is, in essence, a network of networks providing remote access to a set of decentralized IT resources. Prior to cloud computing becoming its own formalized IT industry segment, the symbol of a cloud was commonly used to represent the Internet in a variety of specifications and mainstream documentation of Web-based architectures.

The cloud provides three basic services: **IaaS** (Infrastructure as a Service), **PaaS** (Platform as a Service), **SaaS** (Software as a Service). The proposed model makes use of all fore-mentioned cloud services to work effectively, efficiently, reliably and quickly.

7 Hardware Architecture

The proposed model is constructed as follows:

We have a Raspberry Pi that is connected to 4 sets of LEDs that represent the traffic lights. The captured images and the reference images are fed manually to the Raspberry Pi currently but we have in mind an automated way to do this via a CCTV camera.

1. **Raspberry Pi**

   In this model, we have made use of the Raspberry Pi model 3. The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. The original model became far more popular than anticipated, selling outside of its target market for uses such as robotics. Peripherals (including keyboards, mice and cases) are not included with the Raspberry Pi. The Raspberry is the primary controller of the entire system. It makes use of a python service that runs on startup automatically to control the lights. It takes the captured image and compares it to the reference image. Depending on the percentage match with the reference image, the Pi determines which signal gets first priority for a green signal. If the traffic is higher on a particular road compared to the other roads at the junction, it gets first priority and the longest green signal. Additionally, the Pi transmits data to the cloud for analytics purposes. The data is sent in a JSON format which includes the following- the location of the signal, the percentage matches of each image and the time when the images were captured. The Pi also receives data from the cloud occasionally about the statistics of the particular junction.

The Pi is connected in the following manner to the traffic lights-

Traffic Light 1: R- Pin 7, Y- Pin 5, G- Pin 3

Traffic Light 2: R- Pin 15, Y- Pin 13, G- Pin 11

Traffic Light 3: R- Pin 21, Y- Pin 20, G- Pin 16

Traffic Light 4: R- Pin 7, Y- Pin 8, G- Pin 25

Ground: Pin 6

8 Software Architecture

The proposed model makes use of following software components:

1. **OpenCV**

   OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

   The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. The library is used extensively in companies, research groups and by governmental bodies.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV's deployed uses span the range from stitching street view images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, C, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

2. ThingSpeak Cloud

ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize and analyze live data streams in the cloud. ThingSpeak provides instant visualizations of data posted by your devices to ThingSpeak. With the ability to execute MATLAB® code in

ThingSpeak you can perform online analysis and processing of the data as it comes in. Some of the key capabilities of ThingSpeak include the ability to:

Easily configure devices to send data to ThingSpeak using popular IoT protocols.

Visualize your sensor data in real-time.

Aggregate data on-demand from third-party sources.

Use the power of MATLAB to make sense of your IoT data.

Run your IoT analytics automatically based on schedules or events.

Prototype and build IoT systems without setting up servers or developing web software.

Automatically act on your data and communicate using third-party services like Twilio® or Twitter®.

The following services are provided by the ThingSpeak cloud which makes it super suitable for my idea:

- **Send sensor data privately to the cloud.**

There are sensors all around—in our homes, smart phones, automobiles, city infrastructure, and industrial equipment. Sensors detect and measure information on all sorts of things like temperature, humidity, and pressure. And they communicate that data in some form, such as a numerical value or electrical signal.

- **Analyze and visualize your data with MATLAB.**

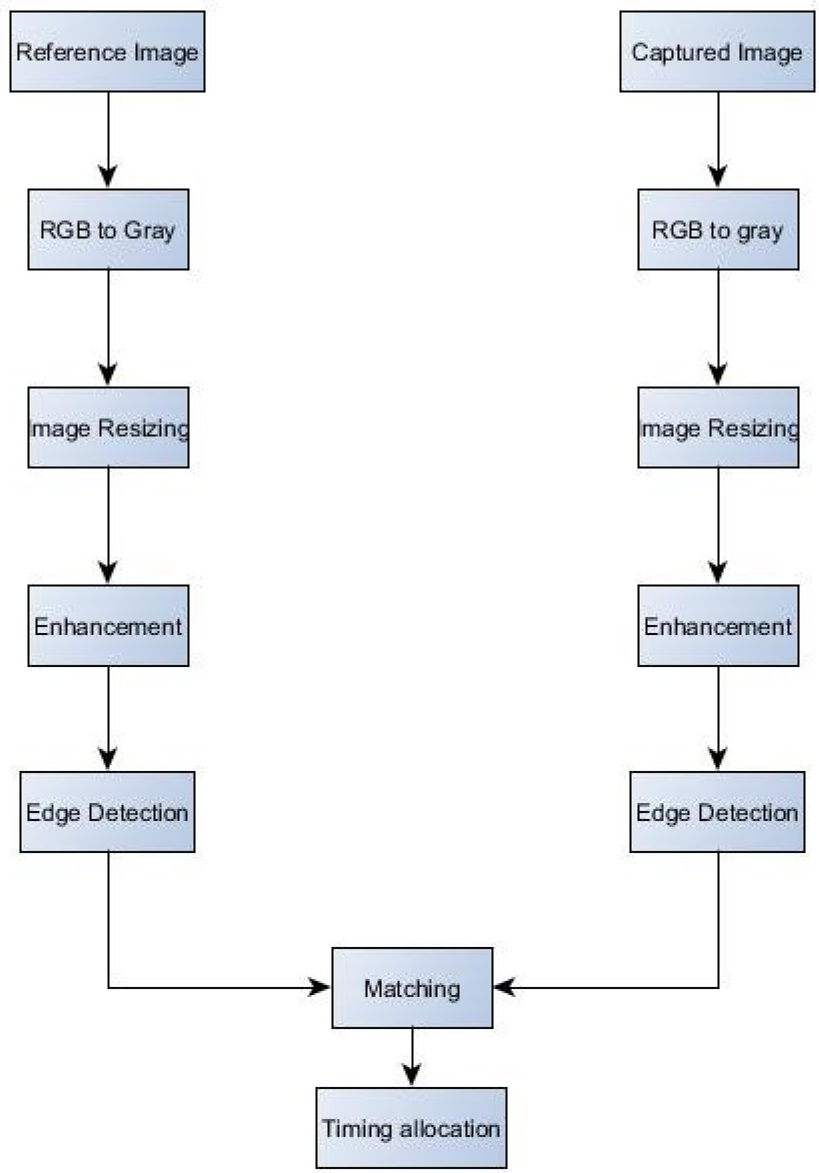
Storing data in the cloud provides easy access to your data. Using online analytical tools, you can explore and visualize data. You can discover relationships, patterns, and trends in data. You can calculate new data. And you can visualize it in plots, charts, and gauges.

- **Trigger a reaction.**

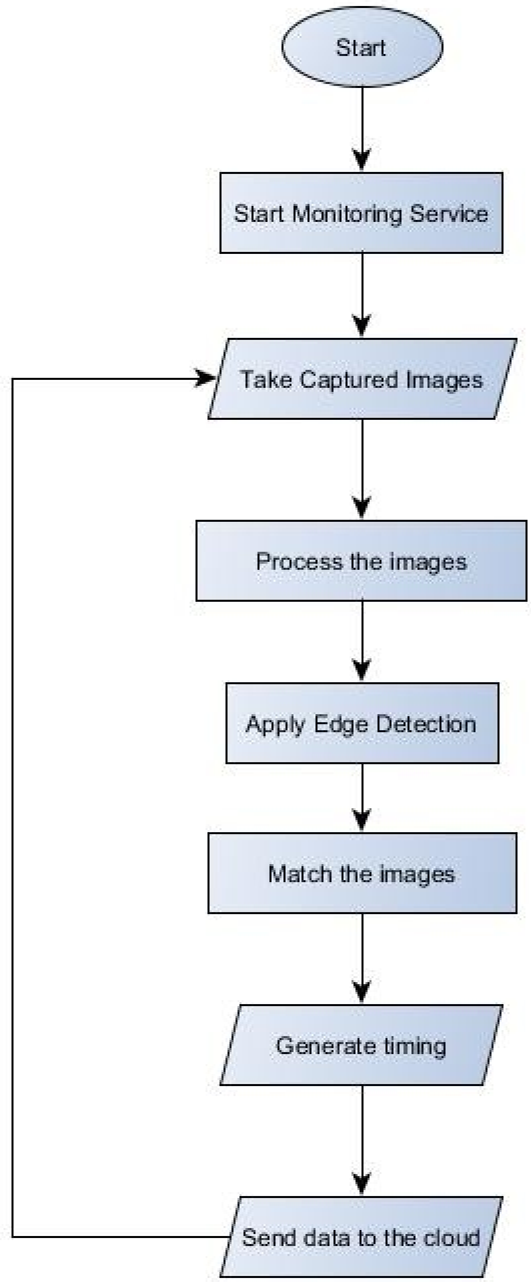
Acting on data could be something as simple receiving a tweet when the temperature you are measuring goes above 70° F. Or you could set up a more intricate action such as turning on a motor when the water level in your water tank drops below a specified limit. You can even remotely control devices, such as battery-operated door locks, using the TalkBack app.

9 Operational Flow Charts

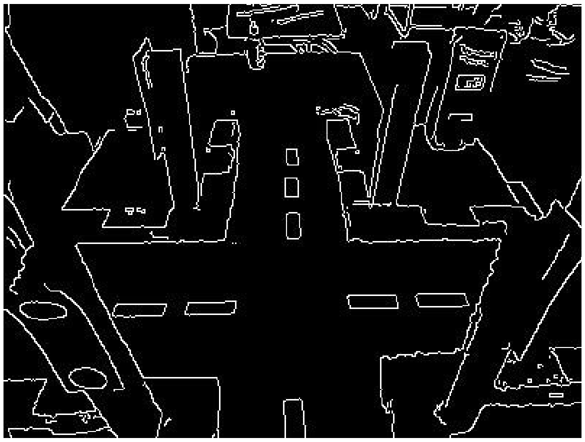
Flow Chart for the Image Processing-



Flowchart for Raspberry Pi Operation-



The edge detected image for the first reference image-

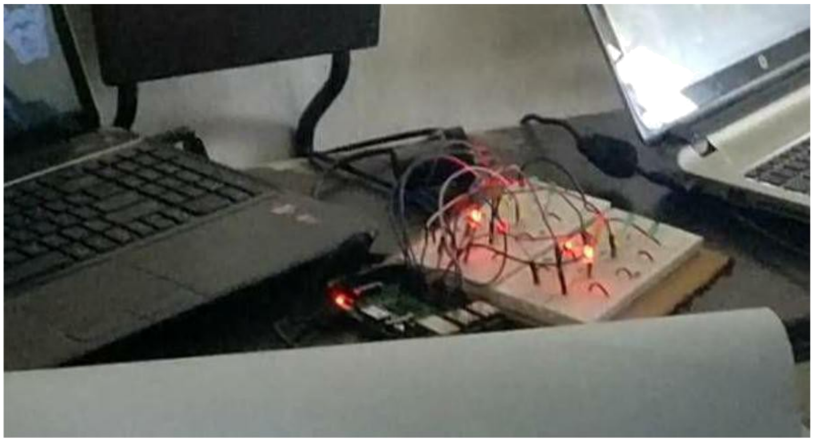


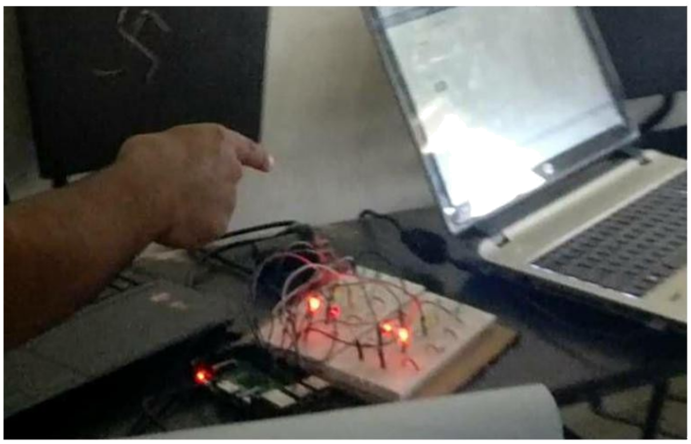
The match percentages for the samples were-

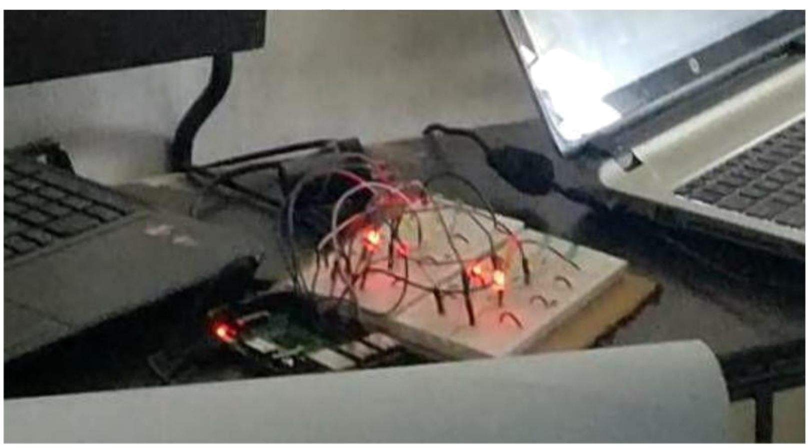
```
76.44%

83.42%

56.54%

95.66%
```

According to the above matches, signal 3 was given first priority followed by 1, followed by 2 and then 4.

The following are some pictures of the system in action-







11 Pros and Cons

Pros:

1. **Installation Cost of our system:** It is very less comparatively because we require the live feed which is easily accessible from the surveillance cameras present at each traffic junction. Most

of major cities and the traffic junction where traffic remains high have cameras already installed.

2. **Use of Data analytics:** Use of technology in Traffic management is a known thing. However, it is the use of data from different sources in real-time and processing information to take immediate decisions that is the key to a successful traffic management in our cities. It is the need of the hour to leverage enormous amount of data around us and create a more meaningful and smooth living for us. So, a lot of hardware cost is cutoff, at max some cameras would require to be repositioned or adjusted.
3. **Saving of Fuel:** Our model reduces the wastage of fuel while sticking in traffic jams by reducing the amount of traffic congestion and jams. Our model smartly tackles traffic and thus saves fuel which is need of the hour.
4. **Maintenance Cost:** of our system is virtually negligible as our system does not include any additional hardware components as compared to the other traffic monitoring systems which employ pressure mats which normally suffer the problem of wear and tear due to their placement on roads where they are subjected to immense pressure constantly.
5. **Distance among Vehicles:** One great feature is that traffic always remains at a distance. If we look at the density of vehicles at all lanes in dynamic algorithm mode and compare it to hard coded, we could see a drastic difference which would reduce congestion by almost 4 to 5 times and all vehicles can move spaciously.

Cons:

1. **After Sunset or Low Light Conditions:** Here the system doesn't work up to the expectations due to lower light conditions, in that case we could switch over system to hard coded during night time. Else we could install night vision cameras to keep the dynamic system working as it works during daytime
2. **No Detection of Emergency Vehicles:** The proposed model does not distinguish normal vehicles with some emergency vehicles like ambulance, fire brigades, etc. So, there is no priority for these vehicles.

12 Conclusion and Future Enhancements

Our model provides a solution to reduce traffic congestion on roads overriding the older system of hard coded lights which cause unwanted delays. Reducing congestion and waiting time will lessen the number of accidents and also reduces fuel consumption which in turn will help in controlling the air pollution. Moreover, the purview of our project can be augmented for Coordination Control which places traffic signals on a coordinated system so that drivers encounter long strings of green lights. This will also provide data for future road design and construction or where improvements are required and which are urgent like which junction has higher waiting times.

In future some enhancements can be done on the project. One such improvement can be overcoming of cons (1). We can use infrared cameras and accordingly apply some suitable image processing methods which in turn will enable the system to control traffic effectively in low lights and night time also.

Other enhancement is resolving of cons (2). We can apply some suitable image processing techniques and enable the system to detect the emergency vehicles like ambulances and fire brigades to let them pass as soon as possible i.e. providing priority to such vehicles which will save a lot of lives and property.

References

[1] http://opencv.org/about.html

[2] https://thingspeak.com/pages/learn_more

[3] https://www.slideshare.net/louiseantonio58/image-processing-based-intelligent-traffic-control-systemmatlab-gui

[4] Khekare, G.S.; Sakhare, A.V., "A smart city framework for intelligent traffic system using VANET," Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 2013 International Multi-Conference on, vol., no., pp.302,305, 22-23 March 2013

[5] Badura, S.; Lieskovsky, A., "Intelligent Traffic System: Cooperation of MANET and Image Processing," Integrated Intelligent Computing (ICIIC), 2010 First International Conference on, vol., no., pp.119,123, 5-7 Aug. 2010

[6]http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.htm l

[7] Salama, A.S.; Saleh, B.K.; Eassa, M.M., "Intelligent cross road traffic management system (ICRTMS)," Computer Technology and Development (ICCTD), 2010 2nd International Conference on, vol., no., pp.27,31, 2-4 Nov. 2010

[8] Haimeng Zhao; Xifeng Zheng; Weiya Liu, "Intelligent Traffic Control System Based on DSP and Nios II," Informatics in Control, Automation and Robotics, 2009. CAR '09. International Asia Conference on, vol., no., pp.90,94, 1-2 Feb. 2009