Project Report on

**“Detection of Traffic Light Signal”**

Submitted to

**Vishwakarma Institute of Technology, Pune**

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

In partial fulfilment of requirements for

Bachelor of Technology

In

**Electronics and Telecommunication**

**By**

**Arni Dhar**

(GR NO. 151464)

**Atharva Kousadikar**

(GR NO. 151448)

Under the guidance of

**Prof. Vijay Mane**



**Department of Electronics and Telecommunication Engineering**

**Vishwakarma Institute of Technology, Pune - 411037**

**Academic Year: 2018 - 19**

Bansilal Ramnath Agarwal Charitable Trust’s

**Vishwakarma Institute of Technology, Pune – 37**

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

Bansilal Ramnath Agarwal Charitable Trust’s

**Vishwakarma Institute of Technology, Pune – 37**

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)



**Certificate**

This is to certify that the project report titled, “**Detection Of Traffic Signal Light**” submitted by **Arni Dhar** (GR NO. 151464) and **Atharva Kousadikar** (GR NO. 151448) is a record of bonafide work carried out by them under the guidance of College mentor **Prof. Vijay Mane** in partial fulfilment of the requirement for the award of the Degree of Final Year Bachelor of Technology in **Electronic and Telecommunication**.

Signature Signature

**Head of the Department Faculty mentor**

Prof. (Dr.) Shripad Bhatlawande Prof. Vijay Mane

**Acknowledgement**

I wish to express my gratitude to Prof**. (Dr.) R. Jalnekar Director, VIT Pune** for providing the facilities of the Institute and for his encouragement during the course of this work. I also express my deep gratitude to **Prof. (Dr.) Shripad Bhatlawande**, the Head of the Department of Electronics and Telecommunication Engineering, VIT Pune for his guidance and support.

I would like to thank my industry mentors **Mr. M.M Kuber and** the **employees of R&DE(ENGRS), Pune** for providing me with guidance and help on every step of the way during the course of this internship and for imparting me with invaluable knowledge and teaching me the etiquettes of a professional employee.

I would also 1ike to gratefully acknowledge the enthusiastic supervision of my internship guide**, Prof. Medha Wyavhare** for his continuous, valuable guidance, patience, constant care, and kind encouragement throughout the internship work that made me present this internship report in an efficient manner.

Final1y, we wish to thank our family members and friends who have always been very supportive and encouraging.

Student Name: Atharva Kousadikar

**INDEX**

|  |  |  |
| --- | --- | --- |
| Sr No. |  | Title |
| 1 |  | Traffic Signal Controller |
|  | 1.1 | Working of Traffic Signal |
| 2 |  | Image Processing |
|  | 2.1 | Digital Camera |
|  | 2.2 | Image Processing |
| 3 |  | Hough Transformation |
|  | 3.1 | How it works |
|  | 3.2 | Guidelines for use |
| 4 |  | Processing Method |
|  | 4.1 | Flowchart |
| 5 |  | Results |
| 6 |  | Conclusion |
| 7 |  | Future Protocols |

**LIST OF FIGURES**

**Fig No. Figure Name**

Fig 1 Optimised Traffic Signal heading northbound

Fig 2 Non optimised traffic signal heading southbound

Fig 3 North bound and south bound coordination

Fig 4 Bayer’s filter

Fig 5 HSV diagram

Fig 6 Coordination points and straight lines

Fig 7 Parametric description

Fig 8 Flow chart of traffic signal

Fig 9 Detection of green light

Fig 10 Detection of yellow light

Fig 11 Detection of red light

**ABSTRACT**

The project is about an automobile detecting the colour of traffic signals as there are many people not following the road rules, which is leading to death of many individuals. To decrease the amount of accidents, there should be warning given to the driver so that he/she follows the rules. The car will be having a digital camera which will be attached, the camera will be recognizing the traffic signal shown on the road using an image sensor. The camera will then identify the distance between the car and traffic signal and the colour of the signal, after detecting the colour it will show warning to the driver on the dashboard of the car and make a noise for the warning. In our project we are taking pictures of traffic light and detecting the colour of the traffic light and showing a message depending on the colour.

**INTRODUCTION**

As we know, in India there are people who do not follow traffic rules. Numbers show India's traffic junctions as the most dangerous spots, accounting for a large number of fatalities. The annual accident report also highlights the need for better designing of such junctions and proper monitoring of traffic flow. People also blame the government of India for not designing of the traffic signals and road signs. Because of people not following the road rules there are many number of accidents occurring. The latest data show at least 75,200 people lost their lives in crashes at traffic crossings in 2013, which is over 50% of the total deaths on Indian roads. While more than 60,000 such deaths were reported at road crossings, another 15,000 died at road-rail crossings. According to the figures, 56,868 deaths were recorded at uncontrolled junctions having no traffic light or traffic police.

In one of the many surveys conducted in India regarding traffic nuisance and traffic violations, an average Indian commuter actually spends more time behind the wheel in comparison to their counterparts in China, Thailand, the Philippines and Australia. This is because of traffic jams created by traffic violations by commuters. People like violating laws themselves, but later crib when others violate the same laws. Moreover, everyone lives a super-paced life. Everyone hates getting late but in fact, hates travelling in time and the only reason we follow any traffic rule ever is the fear of getting caught by the traffic police.

**1. Traffic Signal Controller**

A standard set of traffic signals consists of:

* a traffic signal controller
* vehicle detector loops and pedestrian push buttons
* traffic signal lanterns; and
* posts, pits and underground electrical cables that connect all the components together.

### **The traffic signal controller**

Housed in a grey metal box on a corner of the intersection, the controller is the 'brain' of the system. It is a computer that processes information received from the detector loops and pedestrian push buttons and changes the signal lanterns in accordance with its programming. Based upon the prevailing demands, the controller determines the length of the green signal for each traffic movement and controls the transition from one combination of green and red signals (known as phase) to the next. It can operate in a 'standalone' manner or be programmed to coordinate with a series of adjacent traffic signals.

### **Vehicle loop detectors and pedestrian push buttons**

Vehicle loop detectors and pedestrian push buttons are the 'eyes' of the system. They are mechanisms motorists and pedestrians use to make the controllers 'see' them and change the signal to give them right of way. Vehicle loop detectors are loops of wire buried in the road leading up to the stop line at the intersection. When a vehicle is passing over the loop the magnetic field (inductance) of the loop changes. The controller detects that a vehicle is waiting to proceed through the intersection. Likewise, when the pedestrian push button is pressed the controller knows that a pedestrian is waiting to cross.

**1.1 Working of Traffic Signal**

The following example shows the coordination of three made up intersections, Smith Street, Jack Street and Bill Street. The left-hand side of the diagram shows the actual intersections on the street. The right-hand side of the diagram shows what is known as the space-time diagram. The x-axis (horizontal) represents time and the y-axis (vertical) represents distance along the roadway. At each intersection, going across the page is the green and red signals (A phases and B phases) that the drivers would see over time.

As shown in Figure 16 below, if the car leaves Smith Street at point A it would get to Jack Street (point B) as the signal turns green. Likewise, the signal would turn green as the car reaches point C at Bill Street. In this case the timing of the signals is really good for the car heading north as the offsets between the start of Smith St's green and the other intersections allows for good northbound coordination.

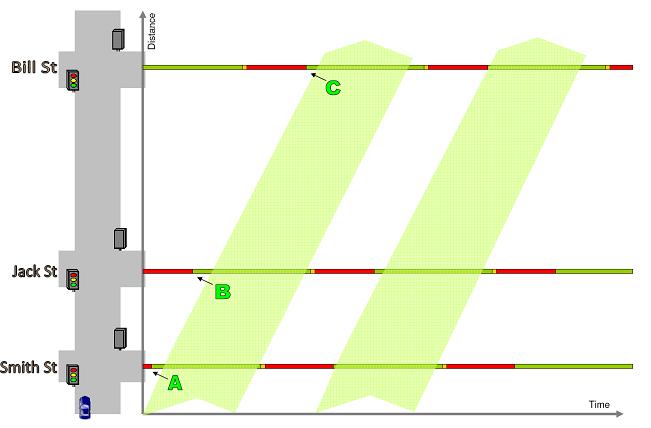


Fig 1 -Optimised traffic signals heading northbound

Now using the exact same settings (offsets, cycle and phase lengths) let's look at a car heading southbound. In Figure 17 below, if the car leaves Bill Street (Point D) at the start of the green it will get to Jack Street (point E) mid-way through the green and hit the red at Smith Street (point F).

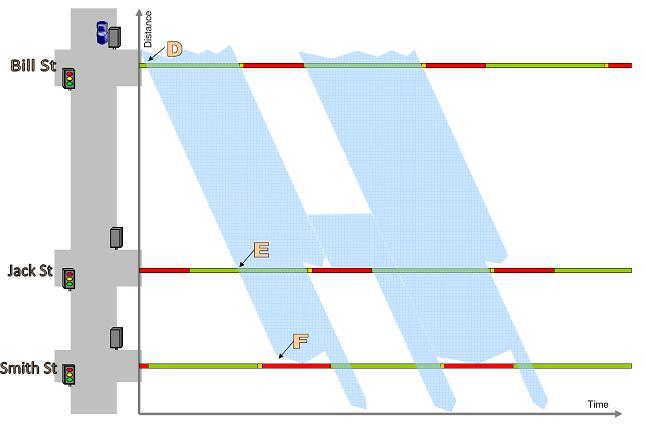


Fig 2 Non optimised traffic signals heading southbound

The same coordination offsets that provides great coordination in the northbound direction resulted in poor coordination for the southbound direction. This is shown in Figure 18 below.

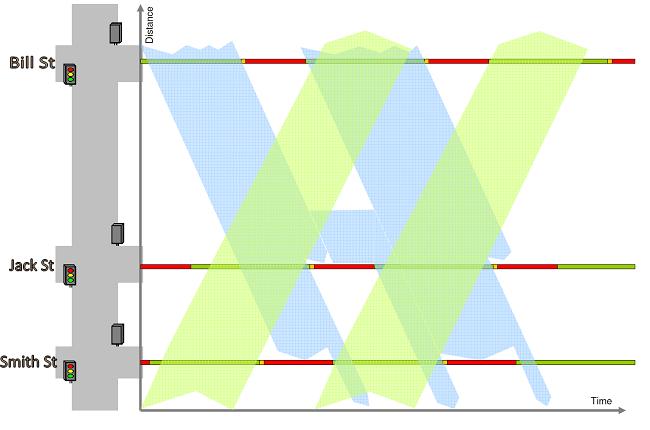


Fig 3 North bound and south bound coordination

The above example shows that it is distance between the intersections (and speed limit of the road) that determines the possible coordination. When working out the coordination for the traffic signals we take into account the volume of traffic heading in each direction and try to provide favourable coordination for the direction with the greater volume. Coordination of signalised intersections is achieved through providing green band progression to vehicle platoons travelling along a corridor. In coordinating traffic signals, we assume that a vehicle's movement along a corridor is known and can be anticipated. Knowing when a vehicle is going to pass an intersection allows for a green light to be pre-programmed for that vehicle's direction of travel. In an ideal world, once a vehicle has joined a platoon of vehicles it should not stop at a red light as long as it stays travelling with the platoon

**2. Image Processing**

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Usually Image Processing system includes treating images as two-dimensional signals while applying already set signal processing methods to them. It is among rapidly growing technologies today, with its applications in various aspects of a business. Image Processing forms core research area within engineering and computer science disciplines too.

Image processing basically includes the following three steps.

1. Importing the image with optical scanner or by digital photography.
2. Analyzing and manipulating the image which includes data compression and image enhancement and spotting patterns that are not to human eyes like satellite photographs.
3. Output is the last stage in which result can be altered image or report that is based on image analysis.

**Purpose of Image processing**

The purpose of image processing is divided into 5 groups. They are:

1.      Visualization - Observe the objects that are not visible.

2.      Image sharpening and restoration - To create a better image.

3.      Image retrieval - Seek for the image of interest.

4.      Measurement of pattern – Measures various objects in an image.

5.      Image Recognition – Distinguish the objects in an image.

**2.1 Digital Camera**

Digital cameras have an optical system typically using a lens with a variable diaphragm to focus light onto an image pickup device. The diaphragm and shutter admit the correct amount of light to the imager, just as with film but the image pickup device is electronic rather than chemical.

Digital cameras use image sensor. An image sensor or imaging sensor is a sensor that detects and conveys information used to make an image. It does so by converting the variable attenuation of light waves (as they pass through reflect objects) into signals, small bursts of current that convey the information. The waves can be light or other electromagnetic radiation. Image sensors are used in electronic imaging devices of both analog and digital types, which include digital cameras, camera modules, medical imaging, night vision equipment such as thermal imaging devices, radar, and others.

The image sensor converts the light information via the lens into the electrical signals, which are further stored in a device. The technology used in image sensor is Charged Coupled Device (CCD). A CCD is an array of capacitors that are sensitive to light. As particles of light (photons) strike the capacitors, they generate electrons. This creates an overall charge that can then be read as an indication of light intensity. CCD don’t read the colour of the light they read the intensity of the light. The basic colours, which are primarily favourable, are red, green and blue. To identify the colours in the camera we use the colour lattice named as Bayes filter which is similar to three-coloured chessboard. The image sensor converts the information and saves the information in the computer, where the image processing is done.

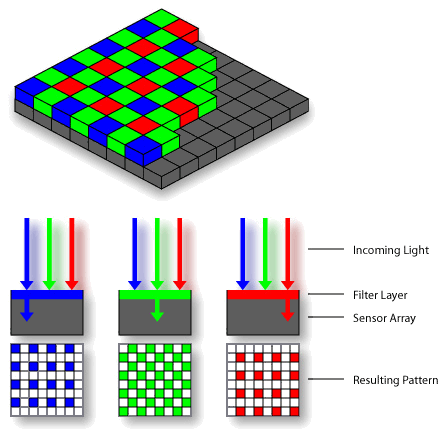


Fig 4 Bayer’s Filter

**2.2** **Image Processing**

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Usually Image Processing system includes treating images as two-dimensional signals while applying already set signal processing methods to them.

Image processing basically includes the following three steps:

1. Pre-processing- Importing the image with optical scanner or by digital photography.
2. Detection- Analysing and manipulating the image which includes data compression and image enhancement and spotting patterns that are not to human eyes like satellite photographs.
3. Recognition- Output is the last stage in which result can be altered image or report that is based on image analysis.

**Pre-Processing**

Colour vision can be processed using RGB space value or HSV space value.RGB space value describes colours with red, green and blue present. HSV colour space describes Hue Saturation Value. The HSV model describes colours similarly to how the human eye tends to perceive color. RGB defines color in terms of a combination of primary colors. In situations where color description plays an integral role, the HSV color model is often preferred over the RGB model.'Hue' represents the color, 'Saturation' represents the amount to which that respective color is mixed with white and 'Value' represents the amount to which that respective color is mixed with black (Gray level).In RGB, we cannot separate color information from luminance. HSV or Hue Saturation Value is used to separate image luminance from color information.

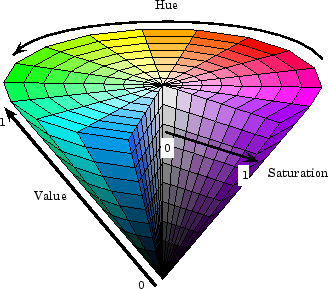


Fig 5 HSV diagram

**Detection and Recognition**

In computer vision, image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super-pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyse. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics. The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image (see edge detection). Each of the pixels in a region are similar with respect to some characteristic or computed property, such as colour, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s).

**3.** **Hough Transforms**

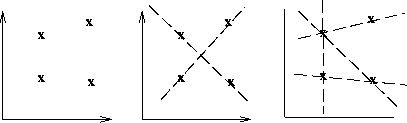
The Hough transform is a technique which can be used to isolate features of a particular shape within an image. Because it requires that the desired features be specified in some parametric form, the *classical* Hough transform is most commonly used for the detection of regular curves such as lines, circles, ellipses, *etc.* A *generalized* Hough transform can be employed in applications where a simple analytic description of a feature(s) is not possible. Due to the computational complexity of the generalized Hough algorithm, we restrict the main focus of this discussion to the classical Hough transform. Despite its domain restrictions, the classical Hough transform (hereafter referred to without the *classical* prefix) retains many applications, as most manufactured parts (and many anatomical parts investigated in medical imagery) contain feature boundaries which can be described by regular curves. The main advantage of the Hough transform technique is that it is tolerant of gaps in feature boundary descriptions and is relatively unaffected by image noise.



* 1. **How It Works**

The Hough technique is particularly useful for computing a global description of a feature(s) (where the number of solution classes need not be known *a priori*), given (possibly noisy) local measurements. The motivating idea behind the Hough technique for line detection is that each input measurement (*e.g.* coordinate point) indicates its contribution to a globally consistent solution (*e.g.* the physical line which gave rise to that image point).

As a simple example, consider the common problem of fitting a set of line segments to a set of discrete image points (*e.g.* pixel locations output from an edge detector). Figure 6 shows some possible solutions to this problem. Here the lack of *a priori* knowledge about the number of desired line segments (and the ambiguity about what constitutes a line segment) render this problem under-constrained.

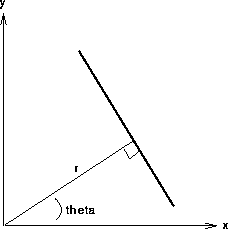


**Figure 6** **a)** Coordinate points. **b)** and **c)** Possible straight-line fittings.

We can analytically describe a line segment in a number of forms. However, a convenient equation for describing a set of lines uses *parametric* or *normal* notion:



where  is the length of a normal from the origin to this line and  is the orientation of  with respect to the X-axis. (See Figure 2.) For any point  on this line,  and  are constant.



**Figure 7** Parametric description of a straight line.

In an image analysis context, the coordinates of the point(s) of edge segments (*i.e.*  ) in the image are known and therefore serve as constants in the parametric line equation, while  and  are the unknown variables we seek. If we plot the possible  values defined by each , points in cartesian image space map to curves (*i.e.* sinusoids) in the polar Hough parameter space. This *point-to-curve* transformation is the Hough transformation for straight lines. When viewed in Hough parameter space, points which are collinear in the cartesian image space become readily apparent as they yield curves which intersect at a common  point.

The transform is implemented by quantizing the Hough parameter space into finite intervals or *accumulator cells*. As the algorithm runs, each  is transformed into a discretized  curve and the accumulator cells which lie along this curve are incremented. Resulting peaks in the accumulator array represent strong evidence that a corresponding straight line exists in the image.

We can use this same procedure to detect other features with analytical descriptions. For instance, in the case of *circles*, the parametric equation is



where  and  are the coordinates of the center of the circle and  is the radius. In this case, the computational complexity of the algorithm begins to increase as we now have three coordinates in the parameter space and a 3-D accumulator. (In general, the computation and the size of the accumulator array increase polynomials with the number of parameters. Thus, the basic Hough technique described here is only practical for simple curves.)

**3.2Guidelines for Use**

The Hough transform can be used to identify the parameter(s) of a curve which best fits a set of given edge points. This edge description is commonly obtained from a feature detecting operator such as the Roberts Crocoite or Canny edge detector and may be noisy, *i.e.* it may contain multiple edge fragments corresponding to a single whole feature. Furthermore, as the output of an edge detector defines only *where* features are in an image, the work of the Hough transform is to determine both *what* the features are (*i.e.* to detect the feature(s) for which it has a parametric (or other) description) and *how many* of them exist in the image.

In order to illustrate the Hough, transform in detail, we begin with the simple image of two occluding rectangles,

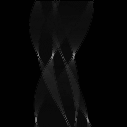


The Canny Edge Detector can produce a set of boundary descriptions for this part, as shown in

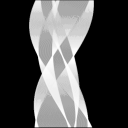


Here we see the overall boundaries in the image, but this result tells us nothing about the identity (and quantity) of feature(s) within this boundary description. In this case, we can use the Hough (line detecting) transform to detect the eight separate straight lines segments of this image and thereby identify the true geometric structure of the subject.

If we use these edge/boundary points as input to the Hough transform, a curve is generated in polar  space for each edge point in cartesian space. The accumulator array, when viewed as an intensity image, looks like



Histogram equalizing the image allows us to see the patterns of information contained in the low intensity pixel values, as shown in



Note that, although  and  are notionally polar coordinates, the accumulator space is plotted rectangularly with  as the abscissa and  as the ordinate. Note that the accumulator space wraps around at the vertical edge of the image such that, in fact, there are only 8 real peaks.

Curves generated by collinear points in the gradient image intersect in peaks  in the Hough transform space. These intersection points characterize the straight-line segments of the original image. There are a number of methods which one might employ to extract these bright points, or *local maxima*, from the accumulator array. For example, a simple method involves thresholding and then applying some thinning to the isolated clusters of bright spots in the accumulator array image. Here we use a *relative threshold* to extract the unique  points corresponding to each of the straight-line edges in the original image. (In other words, we take only those local maxima in the accumulator array whose values are equal to or greater than some fixed percentage of the global maximum value.)

Mapping back from Hough transform space (*i.e.* *de-Houghing*) into cartesian space yields a set of line descriptions of the image subject. By overlaying this image on an inverted version of the original, we can confirm the result that the Hough transform found the 8 true sides of the two rectangles and thus revealed the underlying geometry of the occluded scene



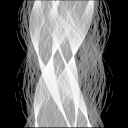
Note that the accuracy of alignment of detected and original image lines, which is obviously not perfect in this simple example, is determined by the quantization of the accumulator array. (Also note that many of the image edges have several detected lines. This arises from having several nearby Hough-space peaks with similar line parameter values. Techniques exist for controlling this effect, but were not used here to illustrate the output of the standard Hough transform.)

Note also that the lines generated by the Hough transform are infinite in length. If we wish to identify the actual line segments which generated the transform parameters, further image analysis is required in order to see which portions of these infinitely long lines actually have points on them.

To illustrate the Hough technique's robustness to noise, the Canny edge description has been corrupted by 1% salt and pepper noise



before Hough transforming it. The result, plotted in Hough space, is



De-Houghing this result (and overlaying it on the original) yields

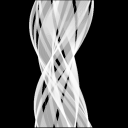


(As in the above case, the relative threshold is 40%.)

The sensitivity of the Hough transform to gaps in the feature boundary can be investigated by transforming the image



, which has been edited using a paint program. The Hough representation is



and the de-Houghed image (using a relative threshold of 40%) is



In this case, because the accumulator space did not receive as many entries as in previous examples, only 7 peaks were found, but these are all structurally relevant lines.

We will now show some examples with natural imagery. In the first case, we have a city scene where the buildings are obstructed in fog,



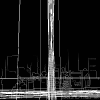
If we want to find the true edges of the buildings, an edge detector (*e.g.* [Canny](http://homepages.inf.ed.ac.uk/rbf/HIPR2/canny.htm)) cannot recover this information very well, as shown in



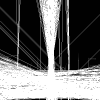
However, the Hough transform can detect some of the straight lines representing building edges within the obstructed region. The histogram equalized accumulator space representation of the original image is shown in



If we set the relative threshold to 70%, we get the following de-Houghed image



Only a few of the long edges are detected here, and there is a lot of duplication where many lines or edge fragments are nearly colinear. Applying a more generous relative threshold, *i.e.* 50%, yields



yields more of the expected lines, but at the expense of many spurious lines arising from the many colinear edge fragments.

Our final example comes from a remote sensing application. Here we would like to detect the streets in the image



of a reasonably rectangular city sector. We can edge detect the image using the Canny edge detector as shown in



However, street information is not available as output of the edge detector alone. The image



shows that the Hough line detector is able to recover some of this information. Because the contrast in the original image is poor, a limited set of features (*i.e.* streets) is identified.

**4. Processing Method**

In this project we are having some images of different traffic lights around the world. We are using OpenCV for detecting the colour in the images. Our main aim of this project is to detect the colour of traffic signals.

The first step is detecting the colours in the image. We have taken various samples of traffic signals images. Each image is first converted from RGB (Red Green Blue) colour space which is the default colour space to HSV(Hue Saturation Value) colour space. There are nearly 150 colour conversion methods in OpenCV but we choose the simple one that is RGB to HSV. We have specified a particular range of red, yellow and green. We can choose any range of the colours as per our application.

Second step is making technique. The mask is basically creating some specific region of the image following certain rules. Here we are creating mask of three colours that is red, yellow and green.

Third we use the concept of Hough transformations. This concept is used to isolate particular features from the image for our information. We are using this concept to circle out the colours we are detecting from the traffic light. In this we are making a circle around the traffic lights, where the colour red, yellow or green is getting detected.

Fourth we are detecting the traffic light. After the first three steps, our colour of traffic signal is having a circle bound which detects the colour and displays which colour it is shown. With the colour, it also shows the message. When the colour of the traffic light is red it displays “Stop!”, when there is green light it displays “Go ahead!” and when it is yellow light it displays “Slow Down!”.

**4.1** **Flow chart**

Image

Green detected

“Go Ahead”

YELLOW detected

“Slow Down!”

RED detected

“Stop!”

Traffic Light Detection

Hough Circle

Masking

Conversion from RGB to HSV

Fig 8 Flowchart of traffic light detection

**5.** **Results**

****

**Fig 9** **Detecting green light**

****

Fig 10 Detecting yellow light



Fig 11 Detecting Red light

**6.** **Conclusions**

In our project we have detected the traffic signals. We will be detecting the colour of traffic signal if it is red, yellow or green. We have used OpenCV from which we have images of traffic signal and we are detecting the colour of the signal and displaying the message.

**7.** **Future Protocol**

As globalization is increasing, technology is also advancing. With enhancement of technology, there will be a world, which is fully automated. Each object will be having some technology that will be sending out some signals that will help in receiving the properties of that object.

In future, the traffic signals will be automated. These infrastructures will be having a transceiver which will be sending signals. These signals will be sending information about the infrastructure. As the cars will be electrical, they will be having a transceiver too which will be getting the information about the infrastructure.

Traffic light will be sending signals to the car about their information whether the traffic light is red, green or yellow. In the car the information will be displayed on the head unit, that will be warning the driver about the traffic signal.