Density Based Traffic Signal using Image Processing

Ajinkeya Chitrey [1], Viren Baria [1], Dhruv Kadakia [1], Dhanam Parekh [1], Yukti Bandi [2]

1. U.G. student, Electronics and Telecommunication Department, DJSCOE, Vile-Parle (W), Mumbai –400056.
2. Asst. Professor, Electronics and Telecommunication Department, DJSCE, Vile-Parle (W), Mumbai – 400 056.

Abstract— ***Due to the rise within the range of vehicles day by day, traffic congestion and traffic jams area are quite common. To replace the current traffic system which is based on fixed timer allocation to each lane is to develop an intelligent traffic control system which is based on the measurement of traffic density on the road using real time image processing techniques. The theme is management {to regulate or to manage} the traffic by deriving the traffic density on both sides of the road and control the LED's. The density numeration formula works by scrutiny the important time-frame of live video by the reference image and by looking out vehicles solely within the region of interest (i.e. road area). The computed vehicle density will be compared with alternative direction of the traffic so as to regulate the light well. In this method, a camera module is used in each stage of the traffic light in order to take pictures of the roads where traffic is bound to occur. Density of vehicles in these images is calculated using image processing tools in MATLAB and different timings are allocated according to the count along with a green signal for vehicles to pass. In the proposed prototype, the red, yellow and green signals are represented using LED's .***

Keywords— ***Arduino UNO, camera module, image processing, LEDs, MATLAB, servo motor, timer delay, traffic density count .***

I. Introduction

Traffic congestion is a common problem that everyone faces in their lifetime. According to an article in CNBC, a human being spends about a full week i.e. 168 hours at traffic signals each year, which is about 1 or 2 years in his/her lifetime. By the popular proverb 'Time is Money', so the more the time wasted, more is the loss of money. In order to tackle such problems, researchers and innovators have proposed many solutions and ideas of a smart and efficient traffic controlling system.

The system currently used has a fixed timer delay which is given to each traffic lights in different directions. This cycle is followed for the next traffic signal on the 4-way junction. If certain lanes are vacant then, due to equal allocation of timer to each traffic lights the timer given to the vacant lane is more than required. Hence, time is wasted on vacant lanes. If certain lanes are heavily filled then, the timer given is less than required. Hence, time of the commuters waiting on the signal is consumed which causes delay in their usual routine.

Earlier the traffic signals were managed by the traffic police personnel by using hand signs and indicating boards. But manual error can cause major accidents. Thus, to reduce labor and chances of accidents new systems were introduced. One of the currently used models is the timer model. Traffic can be controlled to a great extent by using timers at each phase of the traffic. Another model used is with the help of electronic sensors which detects the presence of vehicles, and produce appropriate signals.

The causes of traffic is dependent on many factors like peak time, special days, season, bad weather, or unexpected events like accidents, special events or constructional activities. Once we get stuck in traffic, we may have to wait for hours to get out of it. We can solve this problem to a great extent by implementing this density based traffic control system using image processing which continuously manages the traffic lights based on traffic. Our project includes interfacing of camera module, image processing system and LED's to Arduino microcontroller. The project can be divided into three phases. Phase 1 includes interfacing the camera module with Arduino microcontroller and storing the image in computer. In phase 2, the image processing is performed on the stored images and density of vehicles in a lane is calculated. Phase 3 includes allocation of timers to each signal based on the density of vehicles in that lane.

We have used CMOS OV7670 camera module to capture grayscale images and they are sent to the computer through Arduino UNO microcontroller. Image processing is performed on the images using MATLAB R2018b. Then, the output signals are shown using LEDs (Red, Yellow, Green) using Arduino UNO.

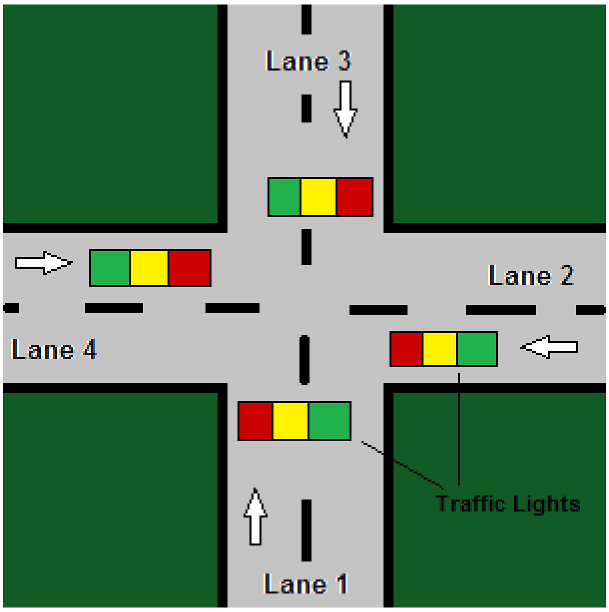

Fig.1. Four way Traffic Signal System

II. Literature Survey

The number of vehicles on the roads has increased to a great extent. This has caused the rise in pollution, wastage of time and resources. Even emergency vehicles like ambulances and fire brigades get stuck on the roads due to traffic congestion. In order to avoid these situations, it has become necessary to design a system which can prevent such traffic jams. The main reason behind the traffic jams is that same priority is given to all the lanes irrespective of the number of vehicles present on the lane.

Infrared sensors can be used to compute the number of vehicles in a lane of a road. Infrared sensor also called as IR sensor is an electronic device used to sense the objects either by emitting or detecting infrared radiation. The range of an ideal IR sensor is up to 150 cm with supply voltage of 4.5-5.5 D.C volts. Here, 4 IR sensors are used for 4-way traffic signal system and hence 4 microcontrollers are used in the circuit. The microcontrollers used are Arduino Nano (ATmega 328). Arduino Nano works with a Mini-B USB cable, rather than the standard one. This system is therefore based on microcontroller and it contains IR transmitters and receivers which are mounted on both sides of single road. Here, the IR sensor gets activated when a vehicle passes or stops in between the sensors. Now the data formed by the IR sensors is transferred to the microcontroller for further processing. The density of vehicles in each lane of a 4-way traffic signal is obtained and based on it; the microcontroller decides the output of the LEDs on each signal. The IR sensors come with certain disadvantages that can be crucial at many situations in real life. The IR sensor works only for short distances and it may also absorb normal light as obstacle. For more accuracy, more number of IR sensors will be required and thus the efficiency of the system will be reduced. Hence, we can replace the IR sensors by a more cost effective and efficient system like image processing. [2]

IOT can also be used in traffic controlling system; here the traffic density of a road i.e. the count of vehicles is done at each side of road by placing a transmitting sensor and a receiving sensor. Here, ultrasonic sensors and ultrasonic receivers are placed opposite to each other. In here sensors are placed at 50 meters distance from one another. The logic here is that as the vehicles crosses the first pair of sensors, a digital signal is produced and accordingly sensor assumes that there is traffic congestion on the road. So based on the data gathered, microcontroller sends the timing signal output by comparing with adjacent road's traffic. As the vehicle crosses the second pair of sensors, Sensor assumes that it contains high traffic density respectively. According to the density of the vehicles the time allocation will be decided i.e. for high density, more time will be allotted and for low density traffic normal time is allotted. The data collected on traffic density is transferred to Raspberry Pi3. The system will store this data with date and time and analyze it later for further uses. This stored data can be uploaded to a Cloud webpage which can be used by the Traffic Police to analyze the flow of vehicles. [1]

The proposed system is to control the traffic based on the density of vehicles on each lane. A surveillance camera is placed at the junction, which is used to calculate the number of vehicles in each lane. According to the number obtained in each lane, the time for green light will be allotted to that particular lane. Basic timer circuit will be followed in case the density of any two lanes is same. In case an emergency vehicle is detected, priority will be given to that particular lane. The hardware used for this system is USB web camera to capture images, PC for image processing, Arduino board for signal prioritization and toy cars for the prototype of a road model. The software module used here is MATLAB for detection of vehicle density. The traffic signal is controlled using MSP430 microcontroller. We first obtain the images of the road using the camera. These images are compared with the reference image which is of an empty road. These images are then cropped to get the region of interest. For the detection of an ambulance, we detect its siren. By using image segmentation, we detect the red and blue colors of the siren. The distance between these colors is checked. Now if this distance is less than or equal to the defined distance and if the centroid lie on the same vehicle, then the vehicle is assumed to be an ambulance. To estimate traffic density, we first detect the vehicles by converting the current and the reference frame to grayscale. For further enhancement, the image is converted to binary and then filtered using Gaussian filter to obtain the vehicles on the road. The vehicles are then counted by bounding of connecting pixels. In this way, traffic can be smartly controlled based upon the density of vehicles and emergency vehicles can be detected. This system is cost-effective as additional hardware such as IR sensors are not required. [3]

III. DESCRIPTION

- Arduino UNO.

Arduino UNO is an open-source microcontroller board based on the MicrochipATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by a USB cable or by an external 9 volt battery, though it accepts voltages between 7 and 20 volts. The ATmega328 on the Arduino Uno comes preprogrammed with a boot loader that allows uploading new code to it without the use of an external hardware programmer.

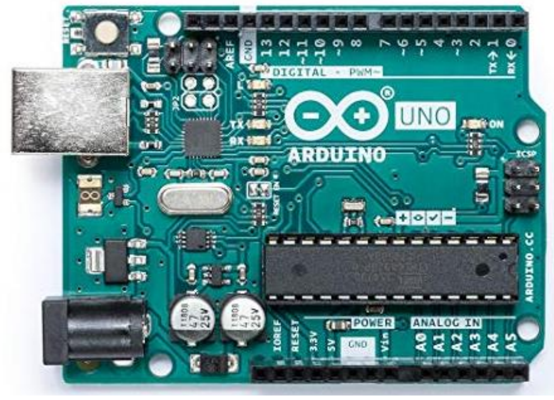

Fig.2.Arduino UNO

- MATLAB 2018b.

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming environment. Furthermore, MATLAB is a modern programming language environment: it has sophisticated data structures, contains built-in editing and debugging tools, and supports object-oriented programming. It has powerful built-in routines that enable a very wide variety of computations. It also has easy to use graphics commands that make the visualization of results immediately available. It is used for image processing and interfacing the result with Arduino board.

- CMOS OV7670 Camera Module.

The OV7670 CAMERACHIPTM is a low voltage CMOS image sensor that provides the full functionality of a single-chip VGA camera and image processor in a small footprint package. The OV7670 provides full-frame, sub-sampled or windowed 8-bit images in a wide range of formats, controlled through the Serial Camera Control Bus (SCCB) interface. This product has an image array capable of operating at up to 30 frames per second (fps) in VGA with complete user control over image quality, formatting and output data transfer. All required image processing functions, including exposure control, gamma, white balance, color saturation, hue control and more, are also programmable through the SCCB interface. In addition, OmniVision CAMERACHIPs use proprietary sensor technology to improve image quality by reducing or eliminating common lighting/electrical sources of image contamination, such as fixed pattern noise (FPN), smearing, blooming, etc., to produce a clean, fully stable color image.

Fig.3. CMOS OV7670 Camera module

- Servo SG90.

Servo motors operates from 4.8V to 6.5V, the higher the voltage higher the torque we can achieve, but most commonly they are operated at +5V. Almost all hobby servo motors can rotate only from 0° to 180° due to their gear arrangement so make sure you project can live with the half circle if no, you can prefer for a 0° to 360° motor or modify the motor to make a full circle. The torque at which the motor operates is the 2.5kg/cm.

Fig.4. Tower Pro Micro Servo SG90

- 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 gray level of image at that point. We need to convert these x and y values to finite discrete values to form a digital image. Each digital image composed of a finite elements and each finite element is called a pixel.

- Image Resizing / Scaling.

Image scaling occurs in all digital photos at some stage whether this be in Bayer demos icing or in photo enlargement. 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. Images are resized because of number of reasons but one of them is very important in our project. Every camera has its resolution, so when a system is designed for some camera specifications it will not run correctly for any other camera depending on specification similarities. So it is necessary to make the resolution constant for the application and hence perform image resizing.

- 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 $s = cr^{\wedge} \gamma$, 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.

- 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. We use Canny Edge Detection method. The Canny edge detection algorithm consists of the following basic steps:

1. Smooth the input image with Gaussian filter.
2. Compute the gradient magnitude and angle images.
3. Apply non maxima suppression to the gradient magnitude image.

4. 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 matrix in memory and the real time image is also converted in the desired matrix. 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

$$\%\ \text{Match} = \frac{No.\ of\ pixels\ matched\ sucessfully}{Total\ no.\ of\ pixels}$$

IV. METHODOLOGY

1. We have a reference image and the image to be matched is continuously captured using a camera that is installed at the junction.
2. The images are pre-processed in two steps:
   - Images are resized to 300x300 pixels.
   - Then the above rescaled images are converted from RGB to gray.
3. Edge detection of pre-processed images is carried out using canny edge detection technique.
4. The output images of previous step are matched using pixel to pixel matching technique.
5. After matching the timing allocation is done depending on the percentage of matching as-
   - If matching is between 0 to 30% - green light is on for 90 seconds.
   - If matching is between 30 to 50% - green light is on for 60 seconds.
   - If matching is between 50 to 70% - green light is on for 30 seconds.
   - If matching is between 70 to 90% - green light is on for 20 seconds.
   - If matching is between 90 to 100% - red light is on for 90 seconds.

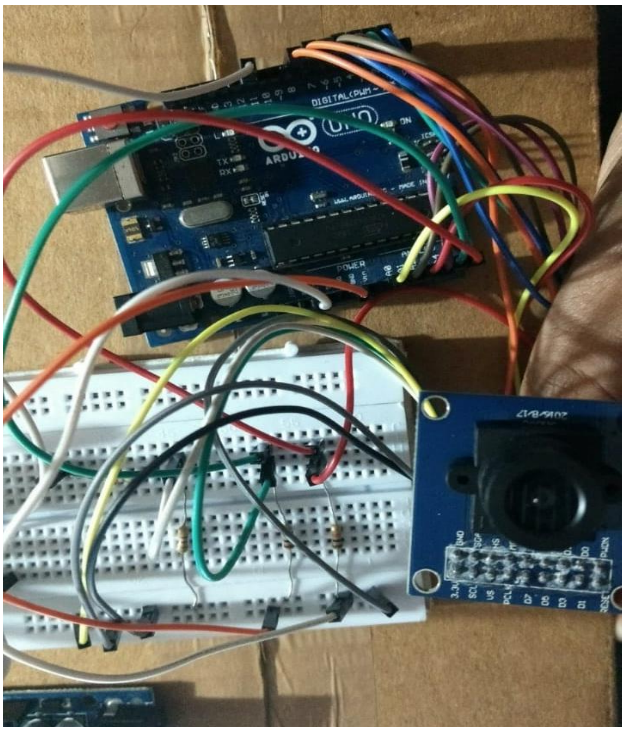

Fig.5. Interfacing OV7670 Camera with Arduino UNO

V. BLOCK DIAGRAM/CIRCUIT DIAGRAM

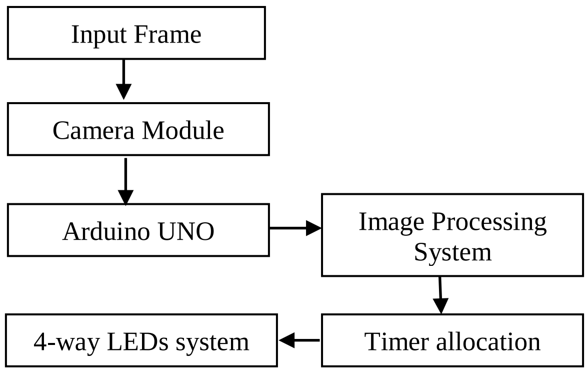

Fig.6. Process flow of the project

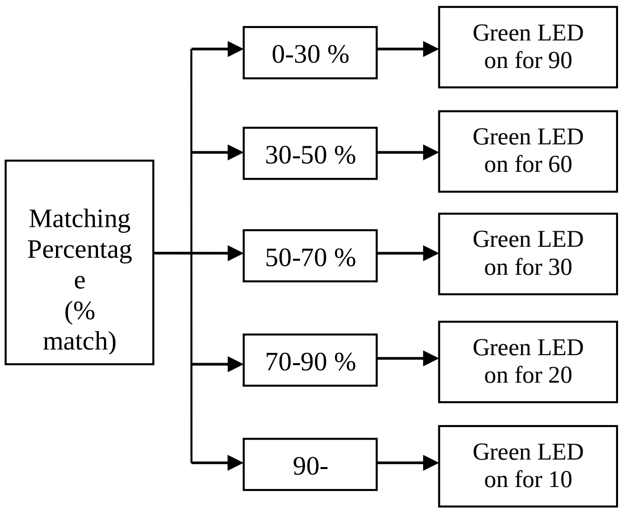

Fig.7. Timer Allocation based on density of vehicles in that lane

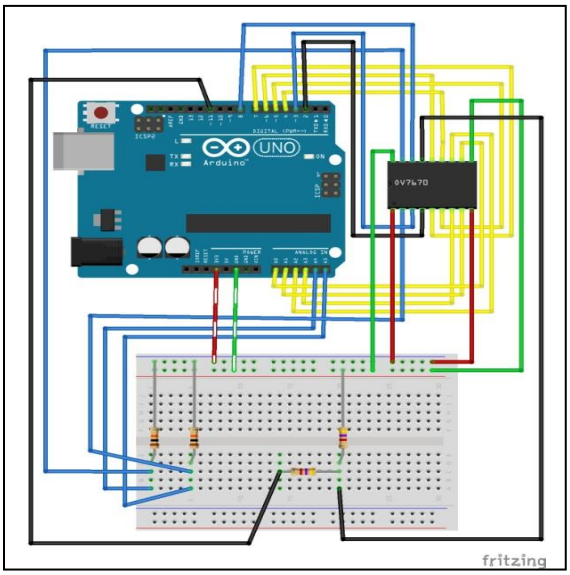

Fig.8. Interfacing camera module with Arduino UNO

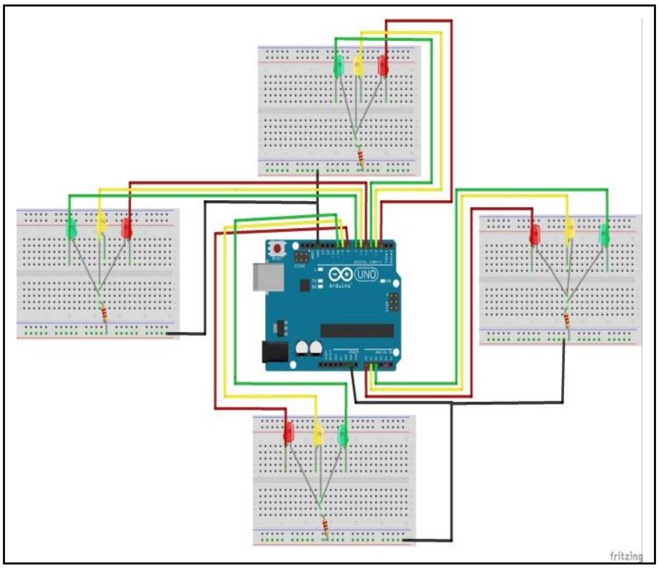

Fig.9. LED connection for 4-way traffic signal

VI. RESULTS

Fig.10. Reference image (Left) and Traffic image (Right)

The percentage match of the above traffic image with the reference image is 14%.

Fig.11. Reference image (Left) and Traffic image (Right)

The percentage match of the above traffic image with the reference image is 80 %.

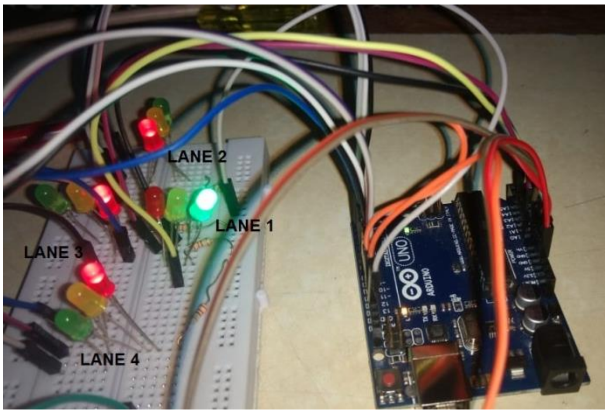

Fig.12. Four-way Traffic Signal Simulation using Arduino UNO

VII. CONCLUSION

An efficient density based traffic control system is simulated and implemented which provides a good traffic control mechanism without time wastage. It is also a much better way of detecting the presence of vehicles on the road since it makes use of image data. So it surely operates much better than systems which rely on the metal content of the vehicles to detect their presence. Image processing techniques overcome the limitations of the all the traditional methods of traffic control. It eliminates the need for extra hardware and sensors. The use of multiple cameras will help to analyze and control traffic in a particular region. The proposed system performs well in terms of accuracy and simplicity.

VIII. FUTURE SCOPE

In the coming future this project will have real industry applications as the image processing and density count mechanisms can help the city pre-plan the traffic management methodologies by studying the traffic statistics on particular routes and how this system can further be integrated with Artificial Intelligence systems so that the signals ahead can plan their signaling response based on traffic at the previous signal junctions for smoother traffic flow. Also help reduce manual labor especially in countries like India and also help the health-care vehicles like ambulances or even fire brigade vehicles to respond in time to casualties irrespective of the road traffic conditions.

IX. ACKNOWLEGEMENT

We would like to acknowledge our sincere thanks to the faculty members of Dwarkadas J. Sanghvi College of Engineering for their valuable guidance throughout the project.

Special thanks to Prof. Yukti Bandi for guiding and reviewing the project as well as providing feedback at every stage of the project.

X. REFERENCES

[1] V. Parthasarathi, M. Surya, B. Akshay, K. Murali Siva, and S. K. Vasudevan, "Smart Control of Traffic Signal System using Image Processing," Indian Journal of Science and Technology, vol. 8, no. 16, Jul. 2015.

[2] Sahu, S., Paul, D. and Senthilmurugan, S., 2018. DENSITY BASED TRAFFIC SIGNAL CONTROL USING ARDUINO AND IR SENSORS.

[3] Ozkurt, C. and Camci, F., 2009. Automatic traffic density estimation and vehicle classification for traffic surveillance systems using neural networks. Mathematical and Computational Applications, 14(3), pp.187-196.

[4] H.-A.-W. Nafisa Anjum, "Traffic Density Measurement using Image Processing: An SVM approach," International Journal of Innovative Research in Computer and Communication Engineering, vol. 03, no. 06, pp. 4979–4986, Jun. 2015.

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

[6] http://www.instructables.com/id/OV7670-Arduino-Camera-Sensor-Module-Framecapture-T/

[7] https://www.mathworks.com/videos/install-the-matlab-and-simulink-support-packages-for-arduino-106497.html