**Density Based Traffic Control System**

***A Project Report Submitted By***

**Jeevan H 4NM16IS034**

**Ashwin Adiga 4NM16IS017**

**Adarsh B Patgar 4NM16IS002**

**Hemanth M 4NM16IS031**

***Under the Guidance of***

**Mr. Jason Elroy Martis**

**Assistant Professor**

**Mr. Santhosh**

**Assistant Professor**

##### Department of Information Science and Engineering

***In partial fulfillment of the requirements for the award of***

**Bachelor of Engineering in Information Science and Engineering**

**From**

**Visvesvaraya Technological University, Belagavi**



**Nitte - 574 110, Karnataka, India**

April - 2020



**CERTIFICATE**

*Certified that the project work entitled*

**“Density Based Traffic Control System”**

*is a bonafide work carried out by*

**Jeevan H 4NM16IS034**

**Ashwin Adiga 4NM16IS017**

**Adarsh B Patgar 4NM16IS002**

**Hemanth M 4NM16IS031**

*in partial fulfilment of the requirements for the award of*

***Bachelor of Engineering Degree in Information Science and Engineering***

*prescribed by* ***Visvesvaraya Technological University, Belagavi***

*during the year 2019 - 2020.*

*It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

**Signature of the Guide Signature of the HOD Signature of the Principal**

Mr. Jason Elroy Martis. Dr. Karthik Pai B. H. Dr. Niranjan N Chiplunkar

**Semester End Viva-Voce Examination**

Examiner’s Name Signature with Date

1. . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . .

2. . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . .

# ACKNOWLEDGEMENT

It is with great satisfaction and euphoria that we are submitting the project report on **“Density Based Traffic Control System”**. We have completed it as a part of the curriculum of Visvesvaraya Technological University, Belagavi for the award of Bachelor of Engineering in Information Science and Engineering.

We are profoundly indebted to our guides, **Mr. Jason Elroy Martis** and **Mr. Santhosh**, Assistant Professors, Department of Information Science and Engineering for innumerable acts of timely advice, encouragement and we sincerely express our gratitude.

We also thank **Mr. Devidas**, Project Coordinator & Assistant Professor, Department of Information Science and Engineering for his constant encouragement and support extended throughout.

We express our sincere gratitude to **Dr. Karthik Pai B. H**, Head and Professor, Department of Information Science and Engineering for his invaluable support and guidance.

We sincerely thank **Dr. Niranjan N Chiplunkar**, Principal, NMAM Institute of Technology, Nitte, who have always been a great source of inspiration.

We are also thankful to the teaching and non-teaching staff for their constant assistance in developing the project.

Finally, yet importantly, we express our heartfelt thanks to our family and friends for their wishes and encouragement throughout the work.

**Project Associates**

**Jeevan H**

**Ashwin Adiga**

**Adarsh B Patgar**

**Hemanth M**

# ABSTRACT

Automatic traffic monitoring, managing, and surveillance are important for road utilization and management. Traffic parameter evaluation has been an active research area for the development of Intelligent Transportation Systems. It is identified that vision-based camera systems are more adaptable for traffic parameter evaluation. Along with the qualitative description of road congestion, image measurement can provide a quantitative description of traffic status which includes speeds, vehicle counts. Image tracking of moving vehicles can give us a quantitative description of traffic. We have created a road model, which consist of three distinct lanes. We collect traffic images in real time from wireless cameras at an interval of five minutes. Several cameras collectively capture images and transmits them to a central relay station. Pre-processing consists of converting each depiction into grayscale images for performance enhancement. Vehicle reckoning processing consists of blob detection counting. We further use the vehicle count in the detection of traffic density in the specified timeframe. If the traffic crosses a pre-specified threshold, we estimate via route to diminish congestion.

# TABLE OF CONTENTS

[ACKNOWLEDGEMENT i](#_Toc47452474)

[ABSTRACT ii](#_Toc47452475)

[TABLE OF CONTENTS iii](#_Toc47452476)

[LIST OF FIGURES v](#_Toc47452477)

[LIST OF TABLES vi](#_Toc47452478)

[1 INTRODUCTION 1](#_Toc47452479)

[1.1 MOTIVATION 1](#_Toc47452480)

[1.2 OBJECTIVE 2](#_Toc47452481)

[2 LITERATURE SURVEY 3](#_Toc47452482)

[3 SYSTEM REQUIREMENTS SPECIFICATION 8](#_Toc47452483)

[3.1 FUNCTIONAL REQUIREMENTS 8](#_Toc47452484)

[3.1.1 Spyder 8](#_Toc47452485)

[3.1.2 Arduino 8](#_Toc47452486)

[3.2 NON-FUNCTIONAL REQUIREMENTS 8](#_Toc47452487)

[3.2.1 Performance Requirements 8](#_Toc47452488)

[3.2.2 Safety Requirements 9](#_Toc47452489)

[3.3 SOFTWARE QUALITY ATTRIBUTES 9](#_Toc47452490)

[3.3.1 Reliability 9](#_Toc47452491)

[3.3.2 Durability 9](#_Toc47452492)

[3.3.3 Portability 9](#_Toc47452493)

[3.4 INTERFACE REQUIREMENTS 9](#_Toc47452494)

[3.4.1 Hardware Interfaces 9](#_Toc47452495)

[3.5 HARDWARE REQUIREMENTS 10](#_Toc47452496)

[3.6 SOFTWARE REQUIREMENTS 10](#_Toc47452497)

[4 SYSTEM DESIGN 11](#_Toc47452498)

[4.1 BEHAVIOR DIAGRAMS 11](#_Toc47452499)

[4.1.1 Activity Diagram 11](#_Toc47452500)

[4.1.2 Architecture Diagram 12](#_Toc47452501)

[5 HARDWARE DESIGN 13](#_Toc47452502)

[5.1 OVERVIEW OF THE CIRCUIT 13](#_Toc47452503)

[5.1.1 ESP-32 Camera 13](#_Toc47452504)

[6 IMPLEMENTATION 15](#_Toc47452505)

[6.1 OVERVIEW 15](#_Toc47452506)

[6.2 DATA SET 15](#_Toc47452507)

[6.3 SELECTION OF THE BEST TRAFFIC PREDICTION ALGORITHM 16](#_Toc47452508)

[6.3.1 KNN Classification 17](#_Toc47452509)

[6.3.2 SVM Classification 17](#_Toc47452510)

[6.3.3 Random Forest Classification 17](#_Toc47452511)

[6.3.4 Naïve Bayes Classification 18](#_Toc47452512)

[7 RESULTS AND DISCUSSIONS 19](#_Toc47452513)

[8 CONCLUSION AND FUTURE WORK 22](#_Toc47452514)

[8.1 CONCLUSION 22](#_Toc47452515)

[8.2 FUTURE WORK 22](#_Toc47452516)

[REFERENCES 23](#_Toc47452517)

**APPENDIX……………………………………………………………………………………………….25**

# LIST OF FIGURES

[Figure 4.1: Activity diagram for the working of our System 11](file:///F:\FY%20Project\report%20word\REPORT%20DENSITY%20BTCS.docx#_Toc47028090)

[Figure 4.2: Architecture diagram of our System 12](file:///F:\FY%20Project\report%20word\REPORT%20DENSITY%20BTCS.docx#_Toc47028091)

[Figure 5.1: Block Diagram of our model 13](file:///F:\FY%20Project\report%20word\REPORT%20DENSITY%20BTCS.docx#_Toc47028092)

[Figure 5.2: Pin diagram of an ESP32 Camera 14](file:///F:\FY%20Project\report%20word\REPORT%20DENSITY%20BTCS.docx#_Toc47028093)

[Figure 6.1: Selection of best prediction algorithm 16](file:///F:\FY%20Project\report%20word\REPORT%20DENSITY%20BTCS.docx#_Toc47028094)

[Figure 7.1: Circuit Connections of our System. 19](file:///F:\FY%20Project\report%20word\REPORT%20DENSITY%20BTCS.docx#_Toc47028095)

[Figure 7.2: Blob Detection for captured threshold image. 19](file:///F:\FY%20Project\report%20word\REPORT%20DENSITY%20BTCS.docx#_Toc47028096)

[Figure 7.3: Time v/s Count Graph of the day. 20](file:///F:\FY%20Project\report%20word\REPORT%20DENSITY%20BTCS.docx#_Toc47028097)

[Figure 7.4: Confusion Matrix of various classification algorithms applied on dataset. 20](file:///F:\FY%20Project\report%20word\REPORT%20DENSITY%20BTCS.docx#_Toc47028098)

[Figure 7.5: Accuracy Graph of the algorithm applied on the dataset. 21](file:///F:\FY%20Project\report%20word\REPORT%20DENSITY%20BTCS.docx#_Toc47028099)

# LIST OF TABLES

[Table 3.1: Software Requirements 10](#_Toc47028105)

[Table 7.1: Various Classification Algorithms applied on dataset and its Accuracy 21](#_Toc47028106)

# INTRODUCTION

As the population of modern cities is increasing day by day due to which vehicular travel is increasing which leads to the traffic problem. Traffic has been causing many critical problems and challenges in the major and most populated cities. Due to this traffic congestion, there is more wastage of time. The steady increase in the number of automobiles on the road has amplified the importance of managing traffic flow efficiently to optimize utilization of existing road capacity. High fuel cost and environmental concerns also provide important incentives for minimizing traffic delays. Road accident is another main problem in the modern world. If we observe seriously the causes of road accidents, we found that narrow roads and a rapid increase in the means of transport are the main reasons behind the increasing number of road accidents. The system will detect vehicles through images instead of using electronic sensors embedded in the pavement. A camera will be installed It will capture image sequences. Image processing is a better technique to control the state change of the traffic. It shows that it can reduce traffic congestion and avoids the time being wasted by on an empty road. It is also more consistent in detecting vehicle presence because it uses actual traffic images. It visualizes the reality so it functions much better than those systems that rely on the detection of the vehicle’s metal content.

## MOTIVATION

In today’s high-speed life, traffic becomes a serious issue in our day to day activities. It brings down the productivity of the individual and thereby the society as lots of work hour is wasted in the signals. The high volume of vehicles, inadequate infrastructure, and the irrational distribution of the signaling system are the main reasons for this chaotic traffic. It indirectly also adds to the increase in pollution level as engines remain on in most cases, a huge volume of natural resources in forms of petrol and diesel is consumed without any fruitful outcome. Therefore, to get rid of these problems or at least reduce them to a significant level, newer schemes need to be implemented by bringing the density-based automatic traffic control system.

## OBJECTIVE

In our project, the system aims to achieve the following:

* + 1. Differentiates between empty roads and roads with traffic.
    2. Turn on the traffic light to red, if the road is empty.
    3. Turn on the traffic light to red, if the maximum time for the green light has elapsed even if there are still vehicles on the road.

# LITERATURE SURVEY

Considering the existing population travelling has become really a big problem. The increase in vehicles has caused a drastic growth in traffic at every corner of the city. In order to deal with researchers have proposed many solutions.

In this paper, on conduction an exhaustive literature review, we have identified some best techniques for Traffic Controlling System. Which will be working better than the existing timer-based Traffic Controlling System.

**EXISTING SYSTEM**

In their research paper [1], authors have proposed a new approach for controlling traffic system which uses a concept of Internet of Things. As per the proposed methodology, technique uses components like Raspberry Pi, Pi-Camera, RFID, IR sensors. Raspberry Pi is the main component which will control all, it acts like a controller. Density of the traffic is decided with the help of IR sensors and in order to give Green path for emergency vehicles RFID technology is used. Also, RFID is used to trail stolen automobiles. The paper proposes control of system in 2 modes i.e. Automatic mode in which depending upon sensors output the decision is taken. In manual mode we can control the traffic, this is done by authorized person in control room. As a result of this, researchers were successful in providing a solution to congestion, traffic clearance to ambulance and other emergency vehicles and tracking of stolen vehicles is also done. Proposed system proved better accuracy by yielding better results than the traditional methods implemented.

Authors in their article [2], proposed an intelligent traffic controller using real time image processing. In this research article, the image sequences from a camera are analysed using various edge detection and object counting methods to obtain the most efficient technique. Subsequently, the number of vehicles at the intersection is evaluated and traffic is efficiently managed. They have also proposed a real-time emergency vehicle detection system. In case of an emergency that lane is given priority over all the others. Authors have compared between various types of edge detection algorithms for ten images taken from real traffic intersections. Canny Edge detector was found to be the best with accuracy of 93.47 among those compared.

In an article [3], authors have presented a technique that has been implemented by using the MATLAB software and it aims to prevent heavy traffic congestion. This research does not actually measure the number of vehicles present on the road, but measures the area covered by vehicles on the road. The implementation consists of following steps. In the first step, the image is received through camera in RGB format. this is needed to be converted in grayscale. Thus, it is processed in MATLAB in order to convert it into grayscale. The green strip area is in white colour and background is in black in this grayscale image. So, it is required to measure green area of the Image. Green pixel area of the received image is measured. By using this information, the set points are set in the code. It will help to specify the three conditions to control the traffic i.e. low, medium and high. Thus, according to pixel count a signature code is generated in MATLAB which is to be sent to controller to allocate timer. A Microcontroller based hardware unit receives this code which was generated in MATLAB through serial port. Therefore, depending upon the signature code received on microcontroller it allocates the time on traffic timer. The above paper eliminates the drawback of earlier techniques has time being wasted on green light on the empty road. Thus, this paper illustrates that image processing is the best way to control traffic when it comes to real time feedback. The key feature of this paper is that it removes the need of hardware sensors such as infrared sensors and RFID tags.

In their research article [4]. Authors have proposed a system for controlling the traffic light by image processing. In this research article, authors have detected vehicles through images instead of using electronic sensors embedded in the pavement. A camera is installed on the traffic light. It will capture image sequences. The image sequence will then be analysed using digital image processing for vehicle detection, and according to traffic conditions on the road, traffic light is controlled. In this research article, authors have concentrated on image enhancement and edge detection techniques useful for the vehicle detection & density computation. The proposed system encounters many limitations i.e. timing is not based on number of vehicles which has the draw backs like Heavy traffic jams, Green Light for an empty road No traffic, but the pedestrians still need to wait. The result achieved is good in every cases. This particular technique overcomes the limitations of the earlier (in use) techniques used for controlling the traffic. Upon comparison of various edge detection algorithms, it was inferred that Canny Edge Detector technique is the most efficient technique.

In their article [5], authors have proposed a novel real-time traffic control system which can easily keep traffic in control using image processing techniques. This system is implemented in MATLAB with an objective to reduce the traffic based on vehicle density. As per the proposed system, four main steps are considered for the system: image acquisition, RGB to grayscale transformation, image enhancement and morphological operations. A camera is installed and used to capture video of the highway. The video is recorded continuously in consecutive frames and each frame is compared to the initial captured image. The total number of cars present in the video is found out using image processing algorithms. If the total number of car count exceeds a predefined threshold, traffic is considered as heavy. As a result of this, researchers were successful in implementing efficient density-based traffic control system which provides a good traffic control mechanism without time wastage. The proposed system outperforms the existing system in terms of accuracy and simplicity.

In their research paper [6], researchers have proposed a novel unified method of automated object detection for urban surveillance systems. In this research article, authors have focused on a design of low-complexity automated object detection algorithms in urban surveillance system. This proposed method will determine and pick out the highest energy frequency areas of the images from the camera, that is, either to pick the vehicle license plates or the vehicles out from the images. This method can not only help to detect object vehicles rapidly and accurately, but also can be used to reduce the big data needed to be stored in urban surveillance systems. The results were amazing with the use of simple and effective filter to get the result. According to authors they are the first to design this kind of filter to detect the vehicle or license plate objects. Initially original RGB image is converted into grey scale image and the size of the image file can be reduced to a lower level. By investigate the characteristics of the object, they found that, within the object boundaries, the energy frequency is high, and the energy frequency curves down sharply outside the object boundaries. After that they designed a new two-dimensional filter to figure out the horizontal and perpendicular frequency energy curves. By researching the filtered image, they find that the noise comes from the thin lines of the filter. The results show that the filter can automatically detect the highest energy frequency areas out from the images, which makes our proposed algorithm a simple and effective method to automatically detect vehicle objects for IoT and smart cities applications.

In their research article, [7] authors presented real-time traffic light control using image processing. The proposed system detects vehicles through images instead of using electronic sensors embedded in the pavement. A camera is installed along with the traffic light, which captures image sequences. The image sequence then subjected to digital image processing for vehicle detection, and according to traffic conditions on the road traffic light can be controlled. The major steps involved here are Image acquisition, RGB to greyscale conversion, Image enhancement, and Image matching using edge detection. Initially, an image of the empty road is captured which is considered as a reference image. RGB to greyscale conversion is applied to the reference image. Next gamma correction is done to enhance the sample image. Edge detection to this sample image is done with the help of a Prewitt edge detection operator. Then the real-time image of the traffic on the road is captured using the camera. The same procedures as above are applied to the real-time image. Now both the real-time image and reference image are compared to each other. The traffic lights are controlled based on the percentage of matching of reference and real-time image. Thus, the research proved that image processing is a better technique to control the state change of the traffic light and also, it can reduce the traffic congestion and avoid the time getting wasted by a green light on an empty road. It visualizes the reality so, it functions much better than those systems that rely on the detection of the vehicle's metal content.

In an article [8], authors in their research work has proposed a system for the control of traffic signals based on the priority outcome using an object counting methods and detection of emergency vehicles simultaneously. Proposed system aims to control traffic signals with the help of surveillance camera present at the junction points. The frames obtained from camera through continuous video processing used to calculate the density. The image from the camera is used to calculate the number of vehicles in each lane. According to the number of vehicles present at each lane, the time for respective green signal is given which varies time to time. Authors have successfully made the prototype for real time image processing for smart automation of traffic signal system for density calculation and emergency vehicle detection. The results obtained with this system are much encouraging and the system does help the present traffic control system to be more efficient.

In their research article [9], authors have identified the problems existing in the earlier in automatic traffic control use of timer had a drawback that the time is being wasted by green light on the empty. This technique avoids this problem. In this particular method, traffic density of lanes is calculated using image processing which is done using the images of lanes that are captured using digital camera. After that reference image and the image to be matched is continuously captured using a camera that is installed at the junction. The image undergoes various image pre-processing and Edge detection of pre-processed images is carried out using canny edge detection technique. At last, the output images of canny edge detection technique are matched using pixel by pixel method. According to this matching percentage the decision is taken for the dynamic allocation of time to different lanes. The proposed technique overcomes all the limitations of the earlier techniques used for controlling the traffic and it also removes the need for extra hardware such as sound sensors.

In another article, [10] authors performed research on traffic congestion control of vehicles based on edge detection using image processing. This research is mainly used to determine traffic congestion on the roads based on the total number of vehicles in the lane. Initially, the images are captured in the form of video frame inputs. By using a video object detector, the input videos are converted into the number of frames. First, the image of the empty road is captured, which is used as a reference image at a particular site specified in the program. RGB to grey conversion is done on captured images. Later, gamma correction is done on each of the captured grey images to achieve image enhancement. Image matching process carried out by using the Canny edge detection technique to locate the pixels in the image that correspond to the edges. The authors used this technique because it can be implemented on the GPU as a sequence of filters. They calculated the density in terms of percentage by subtracting the foreground and background image from the total area of vehicles. Based on the number of vehicles the green signal is set for the allocated time. They discovered that the accuracy in calculation of time due to the single moving camera depends on the registration position while facing the road every time. The output of the GUI indicated some expected results. It is shown that the green signal is mainly based upon the number of vehicles that crossed the rectangular Gaussian filter box in the road.

Authors in their research paper [11] proposed a traffic light controlling system using image processing for effectively regulating the traffic by taking images of traffic at a junction. Initially, the image is captured by using a webcam placed at the road junction. The camera, controlled by the Arduino microcontroller, rotates in the clockwise direction and stops to take pictures of each lane. The clicked pictures are sent to MATLAB for image processing. Then the priority for each lane is given. Priority is decided as per its traffic density. The lane with the highest relative traffic density is given the highest priority and the lane with the lowest traffic density is given the least priority. The lanes are arranged in the descending order of their priorities. When all the lanes have green signals, the traffic light completes its one cycle of traffic monitoring and congestion control. This process repeats and the duration of the green signal given to each lane keeps adjusting itself after every rotation of the camera. The experimental results show that this technique is cost-efficient and does not require the installation of complex machinery to monitor the traffic. Also, the waiting time for the motorists on the road with a higher density is reduced. Deploying this system will not only save the time consumed in waiting at traffic junctions but will also conserve a lot of resources that are otherwise wasted.

After carrying out an exhaustive literature survey it is evidently found out that there is sufficient gap in the concerning area of research regarding traffic control. We have also seen that the existing traffic control system have low accuracy in predicting traffic. With regard to this scope we proposed a novel traffic detection system which has improved accuracy within real time as well as adopt to it.

# SYSTEM REQUIREMENTS SPECIFICATION

## FUNCTIONAL REQUIREMENTS

Functional requirements specify the software and hardware requirements of the system. We are planning to use the following components for developing a system that can run on the computer.

### Spyder

Spyder is an open-source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates with many prominent packages in the scientific Python stack, including NumPy, SciPy, matplotlib, pandas, and ipython as well as other open-source software. It is created and developed by Pierre Raybaut in 2009. It facilitates the use of Python for scientific and engineering software development. It is available cross-platform through Anaconda, on Windows, on macOS through MacPorts, and on major Linux distributions such as Arch Linux, Debian, Fedora, Gentoo Linux, OpenSUSE, and Ubuntu. Spyder of version 3.3.2 is used in the project.

### Arduino

Arduino is a prototype platform (open-source) based on easy-to-use hardware and software. It consists of a circuit board, which can be programed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board. The Arduino IDE of version 1.8.33 is used in the project.

## NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements are the requirements which specify criteria that can be used to judge the operation of a system, rather than specific behaviors. They include timing constraints, the capability of I/O devices, cost, reliability, and scalability. The non-functional requirements for the proposed system are given below:

### Performance Requirements

The system should perform 24/7 without any interruptions and it should count the vehicles on each lane for every 5 minutes without any delay.

### Safety Requirements

The power should be continuously supplied to the system and we have to maintain the system regularly.

## SOFTWARE QUALITY ATTRIBUTES

Quality has many attributes. Some of the important attributes for this software are:

### Reliability

Reliability defines the system’s consistent performance. Our system will avoid accidents during peak hours due to heavy traffic and it will help us in controlling traffic efficiently rather than using a timer.

### Durability

Durability defines the system’s ability to function in testing times and 24/7 availability. Our system satisfies this functionality by providing pictures throughout the day.

### Portability

Our system is portable as it can be easily carried or moved, because being of a lighter and smaller version.

## INTERFACE REQUIREMENTS

### Hardware Interfaces

* + - * Windows or Linux Operating System.
      * Spyder IDE.

## HARDWARE REQUIREMENTS

* Computer or Laptop.
* ESP32 Wi-Fi – Camera.
* Arduino Uno.
* PCB Board.
* Wires for connections.

## SOFTWARE REQUIREMENTS

It should make sure that the Spyder IDE and the Arduino IDE is installed in either Windows or Linux Operating System environments. Table 3.1 shows the software interfaces used.

|  |  |
| --- | --- |
| **SOFTWARE USED** | **DESCRIPTION** |
| Operating System | Platform independent-Windows, Linux. |
| Spyder IDE | An interactive environment to run the python code |
| Arduino IDE | An environment to code the Arduino microcontroller |

Table 3.1: Software Requirements

# SYSTEM DESIGN

## BEHAVIOR DIAGRAMS

Behavior diagrams emphasize what must happen in the system being modeled. Since behavior diagrams illustrate the behavior of a system, they are used extensively to describe the functionality of software systems.

### Activity Diagram

Activity Diagram describes the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control. Inactivity diagram the object may be real or abstract in either case create a swim lane for each attribute imported object, firstly identify the precondition of initial state and post condition of the final state. Render the transaction that connects these actions and active states and states with sequential flows consider branching, forking, and joining. Figure 4.1 shows the activity diagram of our system.

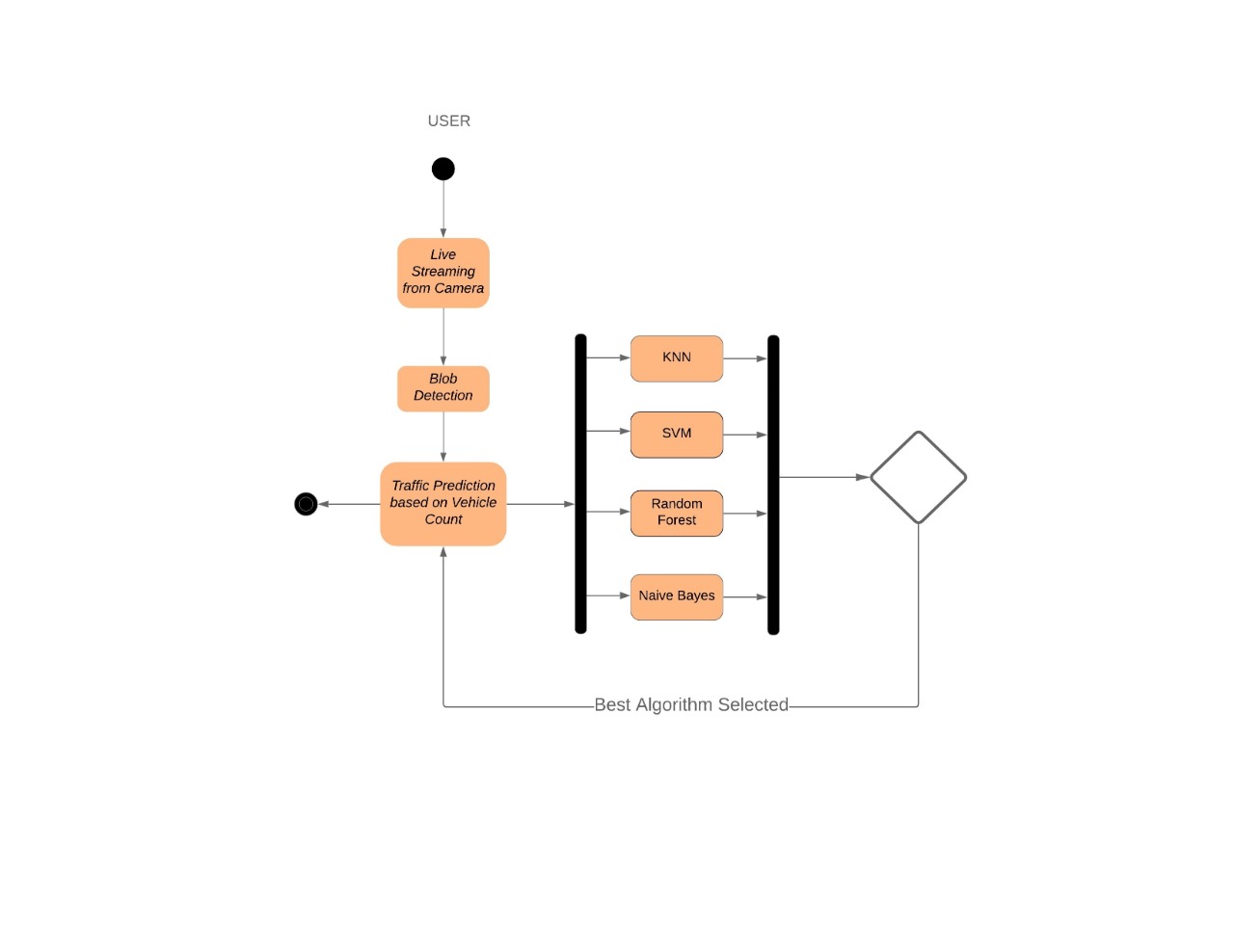


Figure 4.1: Activity diagram for the working of our System

### Architecture Diagram

Architecture diagram can help system designers and developers visualize the high- level, overall structure of their system or application to ensure the system meets their users’ needs. You can also use architecture diagrams to describe patterns that are used throughout the design. Figure 4.2 shows the architecture diagram of our system.

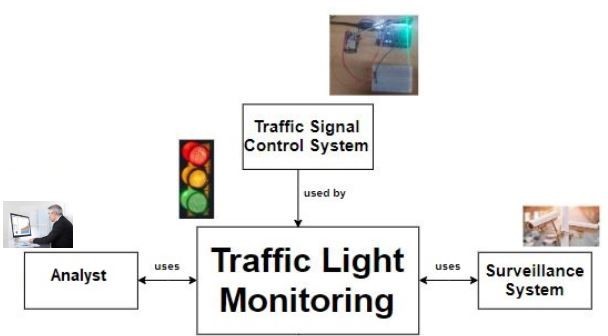


Figure 4.2: Architecture diagram of our System

# HARDWARE DESIGN

## OVERVIEW OF THE CIRCUIT

Figure 5.1: Block Diagram of our model

### ESP-32 Camera

The ESP32-CAM (Figure 5.2) has a very competitive small-sized camera module that can operate independently as a minimum system with a footprint of only 27\*40.5\*4.5mm and a deep sleep current of up to 6mA (see APPENDIX). It can be widely used in various IoT applications. It is suitable for home smart devices, industrial wireless control, wireless monitoring, QR wireless identification, wireless positioning system signals, and other IoT applications. It is an ideal solution for IoT applications. It adopts a DIP package and can be directly inserted into the backplane to realize the rapid production of products, providing customers with high-reliability connection mode, which is convenient for application in various IoT hardware terminals.

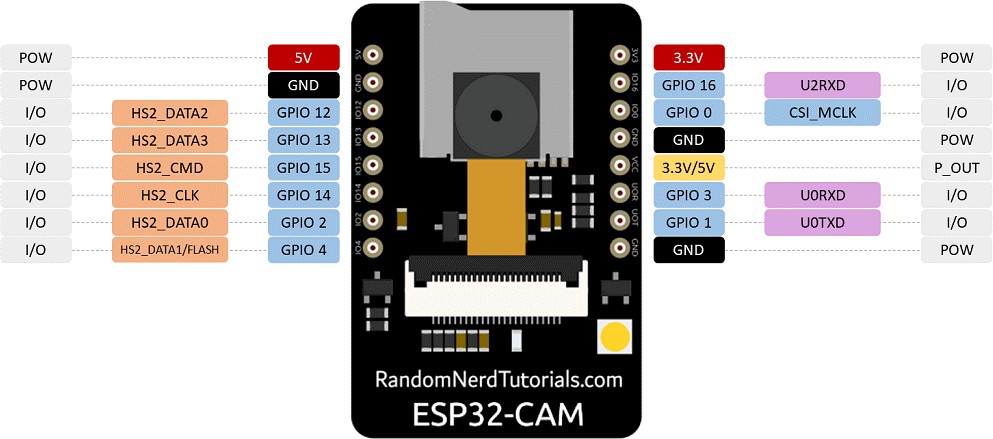


Figure 5.2: Pin diagram of an ESP32 Camera

# IMPLEMENTATION

## OVERVIEW

We have implemented a system using an ESP-32 Camera which continuously takes a snapshot for every 5-minute interval and sends data to a local nearby server. The ESP-32 Camera is placed in such a way that we get a top view of the road on a single lane at a distance of 20 feet. Similarly, many such cameras are placed at various locations to relay the data back to the server. The server continuously runs the selected algorithm and predicts the traffic with certain confident values. We have created our custom dataset by placing the camera at strategic locations in the vicinity of our campus.

## DATA SET

The data set consist of 5000 rows with respective columns. The first column contain Time and the second column contain vehicles counts. In the Time attribute we have taken only hours and minutes.

This data is similar to the data set found in <https://www.kaggle.com/coplin/traffic/data?>

## SELECTION OF THE BEST TRAFFIC PREDICTION ALGORITHM

We have implemented our traffic prediction mode by selecting 4 classification algorithms and training them from the dataset mentioned above. The prediction score is compared and the best score is selected as a traffic prediction algorithm. Figure 6.1 shows the selection of the best traffic prediction algorithm.

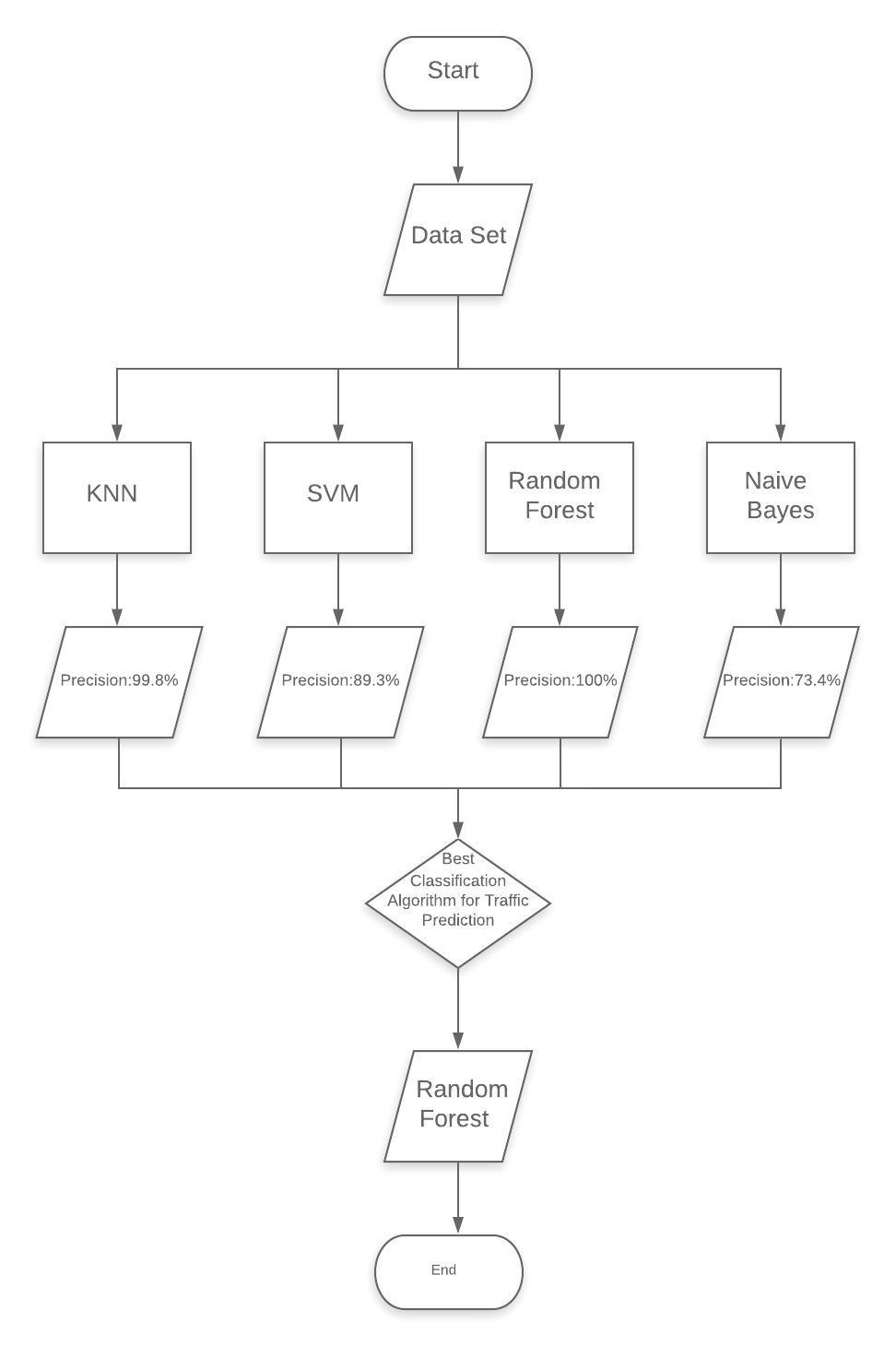


Figure 6.1: Selection of best prediction algorithm

### KNN Classification

from sklearn.neighbors import KNeighborsClassifier

classifier = KNeighborsClassifier(n\_neighbors=5)

classifier. fit (X\_train,y\_train)

KNeighborsClassifier(algorithm='auto', leaf\_size=30, metric='minkowski',metric\_params=None, n\_jobs=None, n\_neighbors=5, p=2,weights='uniform')

y\_pred = classifier.predict(X\_test)

KNN classification is done by discovering K instances that are near to the unseen instance. In the above code snippet, we took the nearest neighbor value as 5 and Minkowski as a distance metric and predicted the output.

### SVM Classification

from sklearn.svm import SVC

svm\_model = SVC(kernel='rbf')

svm\_model.fit(X\_train,y\_train)

SVC(C=1.0, cache\_size=200, class\_weight=None, coef0=0.0,

decision\_function\_shape='ovr', degree=3, gamma='auto\_deprecated',

kernel='rbf', max\_iter=-1, probability=False, random\_state=None,

shrinking=True, tol=0.001, verbose=False)

svm\_preds = svm\_model.predict(X\_test)

SVM classification classifies the data into two different classes over a hyper-plane. In the above code snippet, we have used RBF as kernel which gives better prediction and accuracy.

### Random Forest Classification

from sklearn.ensemble import RandomForestClassifier

clf = RandomForestClassifier(n\_jobs=2, random\_state=0)

clf.fit(X\_train,y\_train)

RandomForestClassifier(bootstrap=True, class\_weight=None, criterion='gini',max\_depth=None, max\_features='auto', max\_leaf\_nodes=None, min\_impurity\_decrease=0.0, min\_impurity\_split=None, min\_samples\_leaf=1, min\_samples\_split=2, min\_weight\_fraction\_leaf=0.0, n\_estimators=10, n\_jobs=2, oob\_score=False, random\_state=0, verbose=0, warm\_start=False)

randomforest\_preds = clf.predict(X\_test)

Random Forest Classification is done by constructing multiple decision trees at training time and outputting the class that is the mode of classes of individual trees. It produces great results most of the time without hyperparameter tuning.

### Naïve Bayes Classification

from sklearn.naive\_bayes import GaussianNB

NB\_model = GaussianNB()

NB\_model.fit(X\_train,y\_train)

GaussianNB(priors=None, var\_smoothing=1e-09)

naivebayes\_preds = NB\_model.predict(X\_test)

Naïve Bayes Classification is done by applying Bayes theorem with strong independence assumptions between the features. It is mainly a probabilistic classifier. Here we used Gaussian Naïve Bayes as a Naïve Bayes model for prediction.

# RESULTS AND DISCUSSIONS

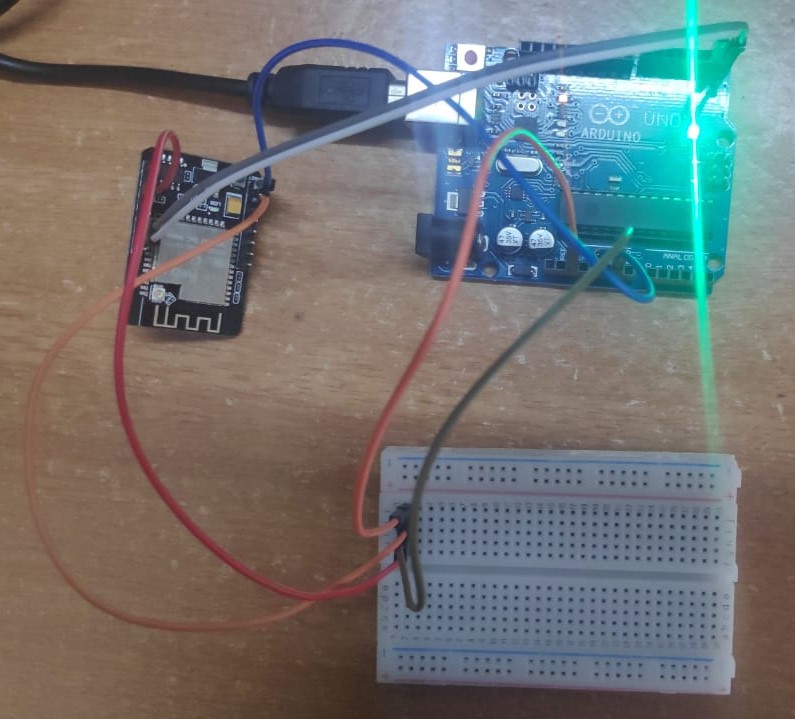


Figure 7.1: Circuit Connections of our System.

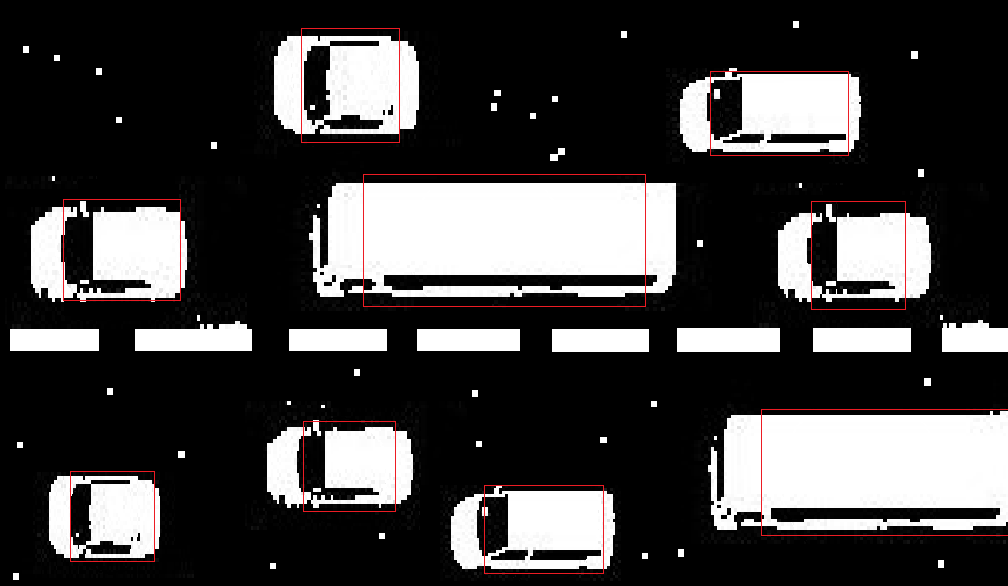


Figure 7.2: Blob Detection for captured threshold image.

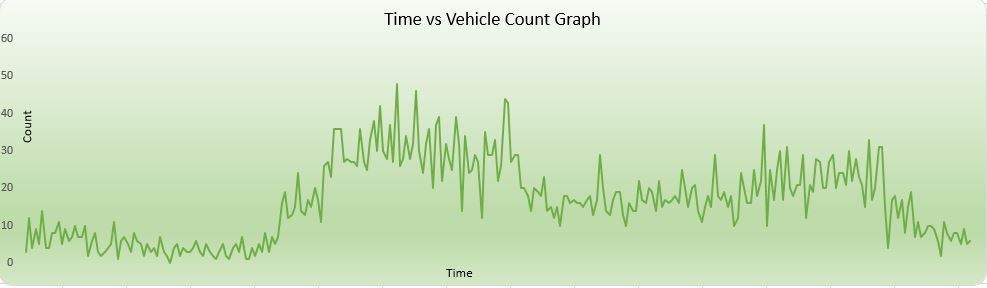


Figure 7.3: Time v/s Count Graph of the day.

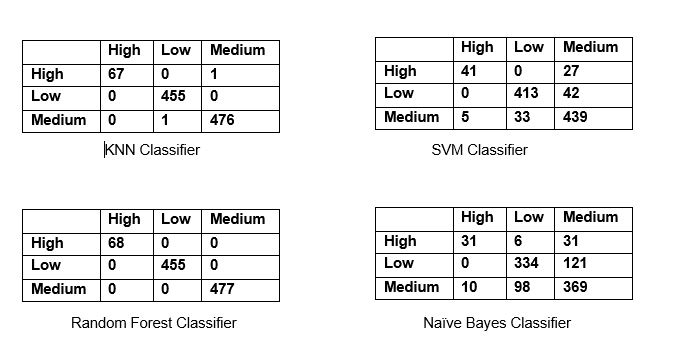


Figure 7.4: Confusion Matrix of various classification algorithms applied on dataset.

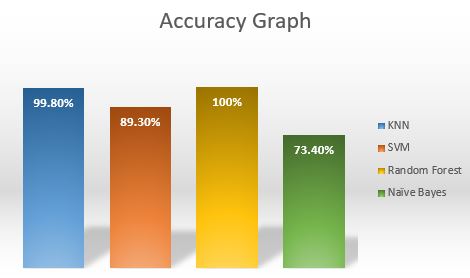


Figure 7.5: Accuracy Graph of the algorithm applied on the dataset.

|  |  |
| --- | --- |
| **Classification Algorithm** | **Accuracy** |
| K-Nearest Neighbors | 99.8% |
| Support Vector Machine | 89.3% |
| Random Forest | 100% |
| Naïve Bayes | 73.4% |

Table 7.1: Various Classification Algorithms applied on dataset and its Accuracy

We found out that from above Bar Graph (Figure 7.5) and Table (Table 7.1), Random Forest Classification gives the highest prediction with 100 % ability. Hence, we can use Random Forest Classification for traffic prediction.

# CONCLUSION AND FUTURE WORK

## CONCLUSION

In this project, we have successfully attained the objective of implementing an efficient system for traffic prediction. Earlier in automatic traffic control, the use of timer had a drawback that the time is being wasted by the green light on the empty road. In the manual controlling system, we need more man power. As we have poor strength of traffic police, we cannot control traffic manually in all area of a city or town. The proposed system overcomes these drawbacks. The camera that we used helps us to stream the vehicles on roads and by performing blob detection, we can count the number of vehicles on the road, then by applying various classification algorithms on that count, we can predict the traffic efficiently. It is also more consistent in detecting vehicle presence because it uses actual traffic images. It visualizes the reality so it functions much better. Upon comparison of various classification algorithms, we infer that Random Forest Classification gives the highest accuracy of 100% and can be considered as the most efficient algorithm for traffic prediction. Overall, our proposed system outperforms the existing system in terms of efficiency.

## FUTURE WORK

We can further implement this system to classify images even in low light condition by using any gamma correction technique or any other low light correction technique. We have also not taken weather conditions into account which may affect the image quality when it becomes foggy or in heavy rains.

# REFERENCES

* + 1. Harshini Vijetha H, Dr. Nataraj K R “*IOT Based Intelligent Traffic Control System*”, International Journal for Research in Applied Science Engineering Technology (IJRASET), ISSN: 2321-9653, Volume 5 Issue V, May 2017.
    2. Vikramaditya Dangi, Amol Parab, Kshitij Pawar & S.S Rathod, “*Image Processing Based Intelligent Traffic Controller”*, Undergraduate Academic Research Journal (UARJ), ISSN: 2278 – 1129, Volume-1, Issue-1, 2012.
    3. Omkar Ramdas Gaikwad, Anil Vishwasrao, Prof. Kanchan Pujari, Tejas Talathi, “*Image Processing Based Traffic Light Control”*, International Journal of Science, Engineering and Technology Research (IJSETR) Volume 3, Issue 4, April 2014.
    4. N. J. Ferrier, S. M. Rowe, A. Blake, “*Real-time traffic monitoring”*, Proceedings of the Second IEEE Workshop on Applications of Computer Vision, pp.81 -88, 1994.
    5. Uthara E Prakash, Athira Thankappan, Arun A. Balakrishnan, “*Density Based Traffic Control System Using Image Processing*”, International conference on Emerging Trends and Innovations in Engineering and Technological Research (ICETIETR2k18) ISBN No 978-1-5386-5743-0.
    6. Ling Hu and Qiang Ni, “*IoT-Driven Automated Object Detection Algorithm for Urban Surveillance Systems in Smart Cities*”, DOI 10.1109/JIOT.2017.2705560, IEEE Internet of Things Journal.
    7. Ms. Pallavi Choudekar, Ms.Sayanti Banerjee, M K Muju, “*Real Time Traffic Light Control Using Image Processing*”, Indian Journal of Computer Science and Engineering (IJCSE), ISSN : 0976-5166 Vol. 2 No. 1.
    8. V. Parthasarathi, M Surya, B Akshay, K Murali Siva and Shriram K Vasudevan,

“*Smart Control of Traffic Signal System using Image Processing”*, Indian Journal of Science and Technology, Vol 8(16), DOI: 10.17485/ijst/2015/v8i16/64622, July 2015. ISSN: 0974-5645.

* + 1. Chirag Thakkar, Rajesh Patil, “*Smart Traffic Control System Based on Image Processing”*, International Journal for Research in Applied Science & Engineering Technology (IJRASET), ISSN: 2321-9653, Volume 5 Issue VIII, July 2017.
    2. J. Vijayaraj, Dr. D. Loganathan, “*Traffic Congestion Control of Vehicle Based on Edge Detection Using Image Processing*”, International Journal of Pure and Applied Mathematics Volume 119 No. 14 2018, 1407-1418 ISSN: 1314-3395.
    3. Rahishet, Aparajita Sahoo, Aparna Indore, Vaibhavdeshmukh, Pushpa U S, “*In- telligent Traffic Light Control Using Image Processing*”, Proceedings of 21st IRF International Conference, 8th March 2015, Pune, India, ISBN: 978-93-82702-75- 7.