iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** I **Month of publication:** January 2020

DOI: http://doi.org/10.22214/ijraset.2020.1100

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 8 Issue I, Jan 2020- Available at www.ijraset.com

Intelligent Traffic Light Control using Image Processing

Pradnya Maturkar[1], Prateek Joshi[2], Pushkar Paranjpe[3], Akshay Bahe[4], Sumit Manmode[5]
[1]*Professor, Dept. of ETC, Rajiv Gandhi College of Engineering and Research, Nagpur, Maharashtra, India*
[2, 3, 4, 5]*UG Student, Dept. of ETC, Rajiv Gandhi College of Engineering and Research, Nagpur, Maharashtra, India*

Abstract: Today, the major problems that cities face is that of traffic. Empty roads are becoming a luxury nowadays. Heavy traffic eventually means high waiting time at the traffic signal. The traffic light timers at squares are set according to the average amount of traffic flow at these squares. This results in traffic congestion at certain junctions due to sudden changes in traffic density. These sudden changes in traffic densities resulting in the need for a smart system that could efficiently handle traffic congestion based on the density of traffic. This paper discusses some of the existing traffic light control systems, their drawbacks as well as our proposed system where we are trying to avoid the drawbacks of existing systems. The main motto of our project is to provide a smooth flow of traffic and to reduce the time duration for which one has to stand at the traffic signal. We have achieved this by using raspberry pi processors and pi cameras corresponding to the four traffic lanes. The cameras will monitor the density of vehicles of standing at the signal using image processing. In addition to that, we will be providing automated number plate recognition as well as speed monitoring in the night time.
Keywords: Image Processing, Raspberry pi 3, python, RTC Module, Traffic Signals.

I. INTRODUCTION

As the population of modern cities is increasing day by day due to which vehicular travel is increasing which leads to the congestion problem. Traffic congestion has been causing many critical problems and challenges in the major and most populated cities. The increased traffic has to lead to more waiting times and fuel wastages. Due to these congestion problems, people lose time, miss opportunities, and get frustrated.

Fast transportation systems and rapid transit systems are some of the important aspects of economic developments for any nation. Mismanagement and traffic congestion occurs due to long waiting times, loss of fuel and money. Thus it is necessary to have a fast, economical and efficient traffic control system for national development. The traffic monitoring authority needs to find new methods to overcome these issues because the number of vehicles is rapidly increasing day by day Recently to improve traffic flow and safety, intelligent control methods are being implemented. Intelligent traffic control will become an important issue in the future as the number of road users constantly increases, and resources provided by current infrastructures are limited. Traffic congestion may result due to heavy traffic at a junction. But no technique is perfect by itself as the real-time situations are generally continuously changing and the system has to adapt itself to change in the continuously changing circumstances. We tried to provide some traffic management strategy so as to fit in to continuously changing real-time traffic scenarios. In this system, time is assigned to traffic light of a particular lane according to the traffic density on the road.

Limitations of Existing system

A. Heavy Traffic Jams

With the increasing number of vehicles on the road, heavy traffic congestion has substantially increased in major cities. This happened usually at the main junctions commonly in the morning, before office hours and in the evening, after office hours. The main effect of this matter is the increased time-wasting of the people on the road.

B. No Traffic, But Still Need To Wait

At certain junctions, sometimes even if there is no traffic, people have to wait. Because the traffic light remains red for the preset time period, the road users should wait until the light turn to green. This happens because the timer values are already set on the basis of average traffic on that lane.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 8 Issue I, Jan 2020- Available at www.ijraset.com

II. LITERATURE SURVEY

One of the primary objectives of the traffic control system is to provide fluent traffic flow irrespective of the variation in traffic density. When surveyed the existing systems to know more about the drawbacks so that we can overcome them in our system. The first system[1] uses an ultrasonic sensor controlled by the microcontroller. This system places the ultrasonic sensor at all the intersections of the square and after detecting the vehicle it checks for the timer as well as traffic signal status of that lane but, in certain places the ultrasonic sensors may be affected by the weather conditions. The second system[2] uses the lines drawn on each lane at several distances from the center of the intersection. Here a camera detects the number of lines visible to detect the traffic density. The number of lines visible is inversely proportional to the density of the traffic, the system may have an increase in distortion as the lines fade away and in some cities, the system may be affected in rainy seasons. The third system[3] uses the ARM processor to control the cameras and the received pictures are first processed through MATLAB before setting the traffic signals. Use of ARM 7 processor provides High Operation speed Usage of external software can be avoided to build an efficient system, use of MATLAB may require the dedicated desktops at each intersection. The fourth system[4] uses the raspberry pi and the pi-camera located at the center of the intersection to read the traffic densities by using image processing. Raspberry pi is used here over the ARM processor as the graph plotting and image processing can be done using Python and it requires no external dedicated device, as the single pi-camera is rotated the system may affect the speed of operation at certain intersections with low traffic densities.

Our system provides regular traffic flow control as well as the vehicle speed monitoring system. Also, the system will monitor for the traffic rules violation and the number plate of the vehicles violating the traffic rule will be recorded in the database.

III. SYSTEM MODEL AND ASSUMPTIONS

In this system, the four raspberry pi with a camera mounted on it is placed in four corners of the intersection. The cameras are located such that they face the opposite lane from the lane they are located on. One of the four raspberry pis will be the master raspberry pi and it will control every process of the system, the remaining three raspberry pi are slaves which are used only to detect the traffic density by image processing. All the raspberry pi are connected to each other by using the inbuilt wifi module and are interfaced by connecting them on the same wifi router or hotspot terminal. The system will restart itself every day at 9 am and for the first iteration all the four cameras will capture the current traffic densities by using image processing and the count is sent to the master raspberry pi. Then it will set the traffic signals accordingly. When one of the lanes is green, the cameras facing the remaining lanes will monitor both the traffic rules violation as well as current traffic density to make it real-time and if the traffic rules are violated then the number plate will be recorded. For night time the same system will monitor for the over-speeding vehicles.

IV. METHODOLOGY

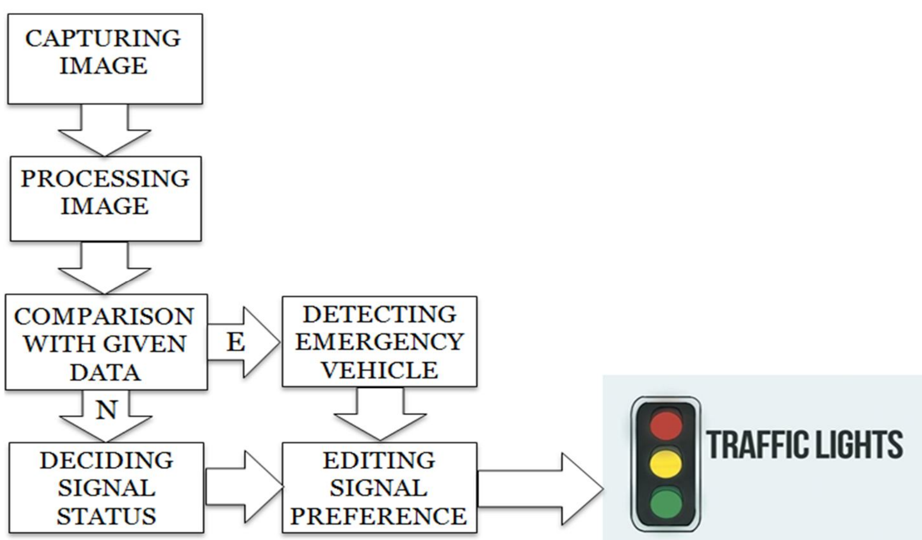


Figure 1: Flow chart of the system controlling the traffic lights

In the Flow chart given above, the comparison with given data refers to comparing the captured image with the custom classifiers made to identify vehicles in the image and deciding the density count accordingly. For image detection, we have used the canny edge detection method.

1) *Image Acquisition:* Generally an image is a two-dimensional function f(x,y) (where x and y are plane coordinates).The amplitude of the 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 finite elements and each finite element is called a pixel.
2) *Image Resizing/Scaling:* Image scaling is used here to provide a constant number of pixels for each image. The resolution of the image received may be different for an instance so, to make the algorithm quite simpler and to ease the processes that are to be performed on the received image.
3) *RGB to GRAY Conversion:* 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 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. In simple terms, it is used to reduce the complexity in the image and ease the edge detection as far as possible because it is simpler to scan a single matrix than three different matrices.
4) *Image Enhancement:* Image enhancement is the process of adjusting digital images so that the results are more suitable for display or further analysis. Here we are blurring the image using the bilateral filter because for edge detection, we only need edge values and thus removing details while preserving edges reduces the noise formed by unwanted false contours.
5) *Edge Detection:* Edge detection is the name for a set of mathematical methods that 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. There are different edge detection techniques but, we have used the canny edge detection. Canny edge detection in python is done by giving the threshold values to the defined function cv2.canny and selecting the image on which it is to be executed.
6) *Image Matching:* We have used a totally different approach to image matching. Comparing a reference image with the real-time image pixel by pixel Some of the procedures mentioned above are used for automatic number-plate detection and some are used to find out the densities of the lane. The first three processes are used with the custom classifiers to detect the count of the vehicle whereas the edge detection is performed on the image received from the camera for ANPR detection.

V. REQUIREMENTS

A. *Hardware Requirements*
1) Raspberry Pi 3 (Model B+)
2) Pi Camera
3) RTC Module
4) Laptop

B. *Software Requirements*
1) Python 3.7 or 2.7
2) OpenCV

C. *Software Implementation*
1) *Python:* Python is a powerful modern computer programming language. It bears some similarities to Fortran, one of the earliest programming languages, but it is much more powerful than Fortran. Python allows you to use variables without declaring them (i.e., it determines types implicitly), and it relies on indentation as a control structure. Think of game programming, rendering graphics, GUI interfaces, web frameworks, and scientific computing. Many (but not all) of the things you can do in c can be done in python. Python is generally slower at computations than C, but its ease of use makes Python an ideal language for prototyping programs and designing applications that aren't computationally intensive.
2) *Open CV*[7]*:* Open CV Stands for Open computer vision it is a source library of functions. is released under a BSD license and hence it's free for both academic and commercial use. It has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform. Adopted all around the world, Usage ranges from interactive art to mines inspection, stitching maps on the web or through advanced robotics.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 8 Issue I, Jan 2020- Available at www.ijraset.com

a) *Convert Color:* cv2.cvtColor function is used to convert the color of the image from color to grayscale. This can also be done by adding parameter while reading the image, using this option image is read in grayscale instead of its original format.
b) *The Image Read:* cv2.imread is used to read the image from the given directory here we have to select the image to be read as well as its optional to select the format (original or grayscale) while reading the image.
c) *Image Show:* cv2.imshow is used to display the image; here the new name for the displayed image as well as the image to be displayed has to be selected.
d) *Bilateral Filter:* cv2.bilateralFilter is used to reduce the noise from the grayscale image. Here the images whose edges are to be detected as well as various threshold values are selected.
e) *Canny Edge Detection*[5]*:* The OpenCV library has an inbuilt function for the Canny edge detection named cv2.canny and here we need to provide the threshold values which are used to decide and draw contours on the edge detected image.
f) *Pytesseract:* The pytesseract is an application that is used for the object to text conversion. This software is used through the program by installing it in our Raspberry Pi and calling the app by giving the app location in the program.
g) *YOLO V3*[6]*:* We have used YOLO (You Only Look Once) model of version 3 to customize the classifiers as the system may vary depending on the local vehicles, and the design of the number–plate of that local area. For that purpose we have used the trainer GUIs for training the customized classifiers.

VI. RESULTS

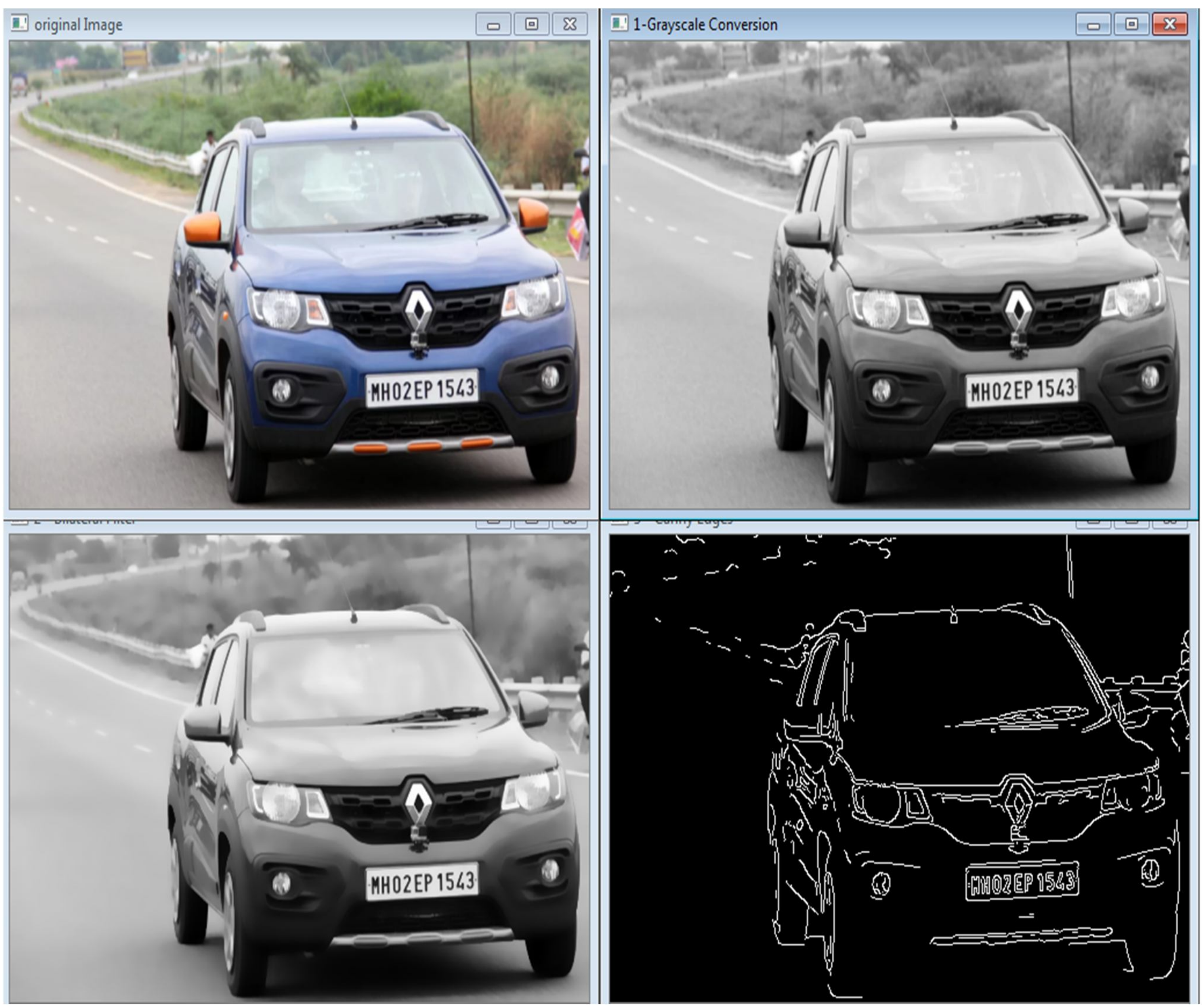


Figure 2: Automatic Number Plate Recognition system acquiring the number plate

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 8 Issue I, Jan 2020- Available at www.ijraset.com

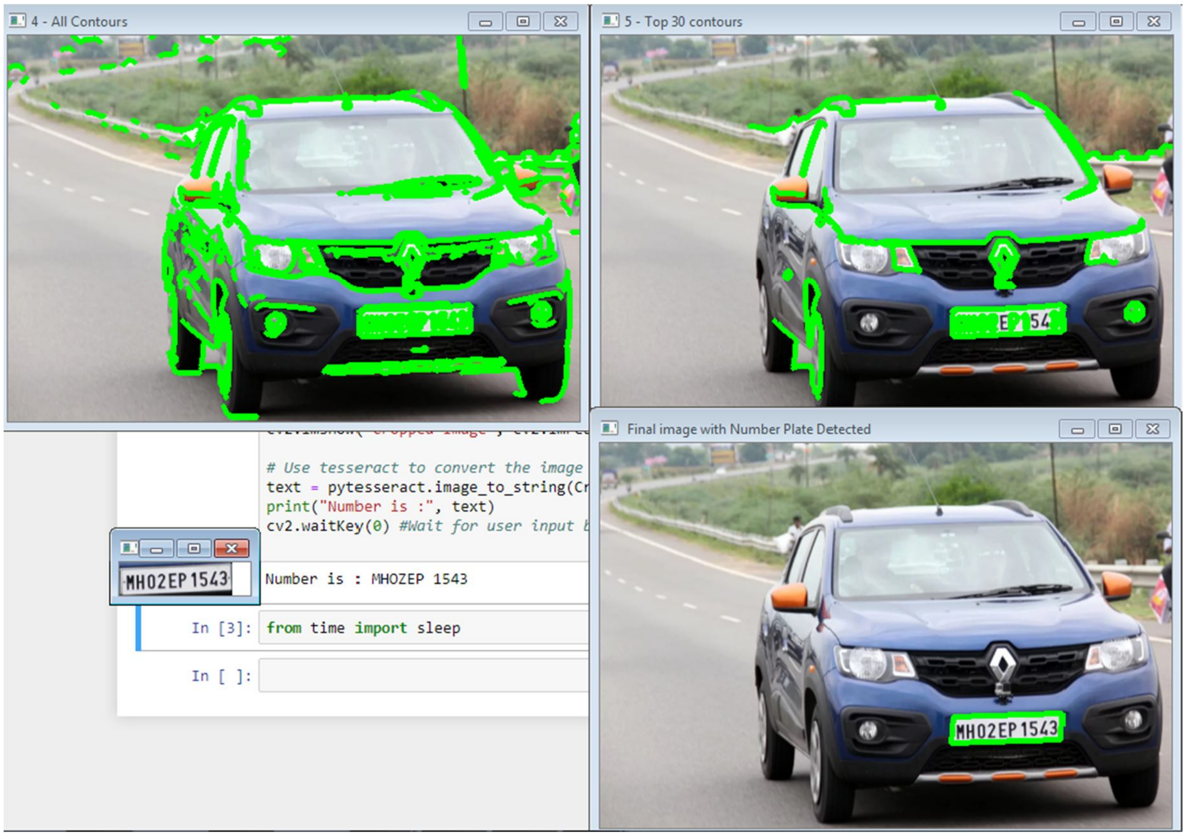


Figure 3: Automatic Number Plate Recognition system acquiring the number plate

Figure 2, 3 shows the result of automatic number plate recognition (ANPR). i.e., how the number plate is extracted from the image. The first slide is the original image, the second one is the grayscale conversion of the original image. The third image is the result of the bilateral filter used to remove details, the fourth one is the output result of Canny edge detection. The fifth image overlaps the detected contours on the original image whereas the sixth image includes the only top thirty contours and the seventh image is that detecting the number-plate. The eighth image includes cropped number-plate which is gained by cropping the sixth image with the coordinates of the contour and then the pytesseract is used to obtain the number-plate.

Figure 4: Emergency vehicle detected using Pytesseract

Figure 3, shows the emergency vehicle detected using Pytesseract, here when the vehicles re waiting on the signal; the separate algorithm is used to detect the ambulance by using the pytesseract. This process is in the loop and is done for every red light assignment for each lane.

VII. CONCLUSION

Thus our system efficiently provides the smooth traffic flow as well as records the number plate of a vehicle violating the traffic rules, also it provides speed monitoring in the night time. The traffic flow is smooth because the density of the lanes is checked in each iteration. With proper guidance from the local traffic department, the system can be successfully implemented also, as a future scope the ANPR system can be used to automatically delivering fine messages to the vehicle owner by the use of GSM module and legal local vehicle owner database. As the system can be used anywhere with the same modifications considering updating a local database, numberplate structure. At this point all the recorded data of vehicles is hard stored, in future the cloud storage can be used for this to improve data safety.

REFERENCES

[1] "Intelligent Traffic Light Control System Based on Real-Time Traffic Flows" Zhijun Li, Chunxiao Li, Yanan Zhang and Xuelong Hu School of Information Engineering, Yangzhou University, Yangzhou, 225127, China 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)

[2] "Image Processing Based Traffic Light Control" Omkar Ramdas Gaikwad, Anil Vishwasrao, Prof. Kanchan Pujari, Tejas Talathi International Journal of Science, Engineering and Technology Research (IJSETR) Volume 3, Issue 4, April 2014

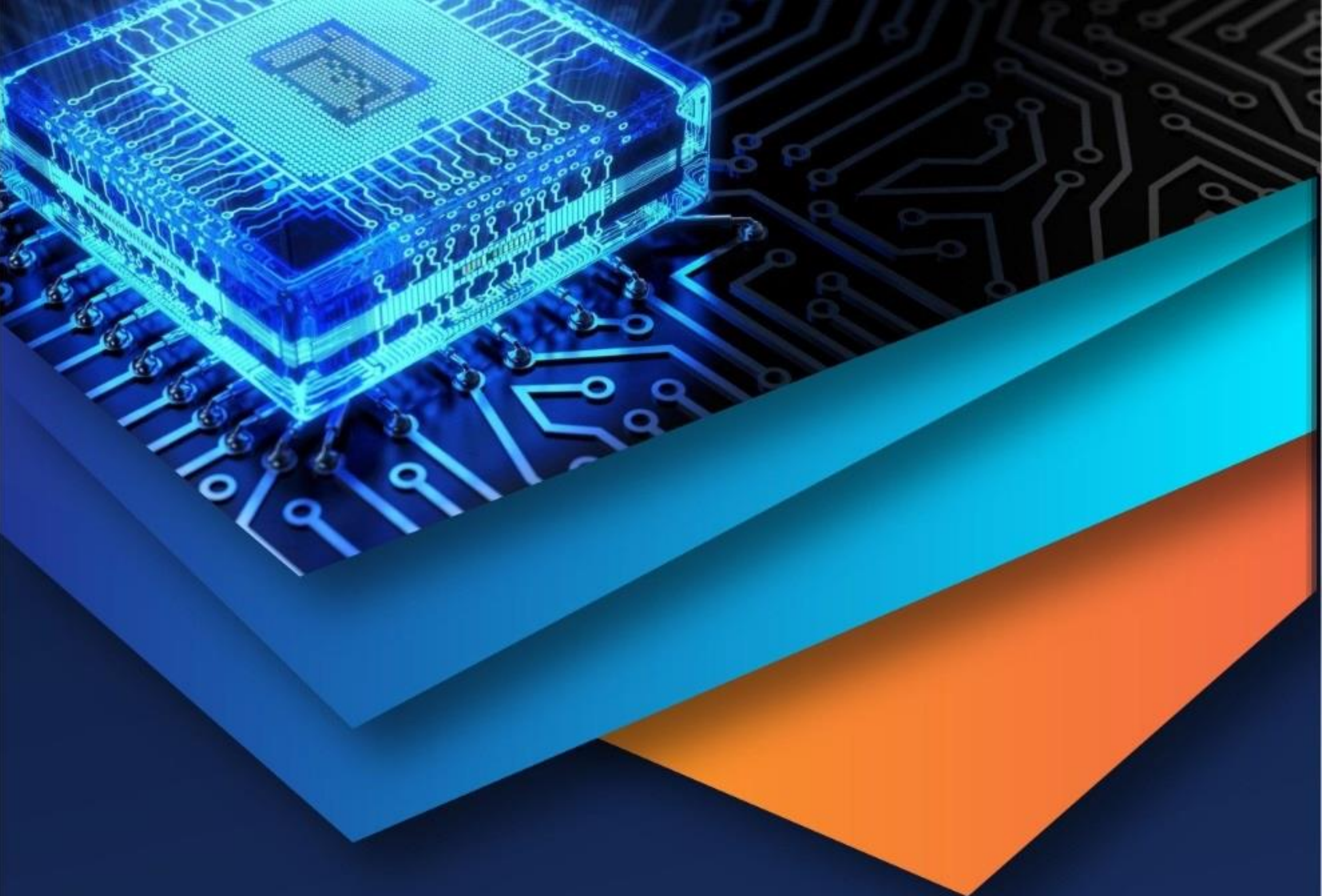
[3] "Density Based Traffic Light Control System Using Image Processing" D.Prakash[1], B.Sandhya Devi[2], R.Naveen Kumar[3], S.Thiyagarajan[4], P.Shabarinath[5] Assistant Professor, Department of Electrical and Electronic Engineering, S.A. Engineering College, Chennai, India1 UG Scholar, Department of Electrical and Electronic Engineering, S.A. Engineering College, Chennai, India [2,3,4] IJAREEIE Vol. 6, Issue 3, March 2017

[4] "Image Processing Based Intelligent Traffic Control System by Using Raspberry Pi" Santoshi Gupta[1], Utkarsha Raikar[2], Bhavna Mohite[3], Priyanka Patil[4], Renuka Molavade[5] Electrical Department[1, 2, 3, 4, 5], SPPU University. Volume 6 Issue IV, April 2018

[5] "Traffic Light Control System Using Image Processing" Kavya P Walad, Jyothi Shetty M.Tech 3rdSemester, Department of Computer Science and Engineering, Srinivas School of Engineering, Mukka, India. Vol.2, Special Issue 5, October 2014

[6] You Only Look Once: Unified, Real-Time Object Detection Joseph Redmon University of Washington pjreddie Santosh Divvala Allen Institute for Artificial Intelligence Ross Girshick Facebook AI Research Ali Farhadi University of Washington.

[7] "Image Processing Based Intelligent Traffic Controller", Vikramaditya Dangi, Amol Parab, Kshitij Pawar & S.S Rathod Undergraduate Academic Research Journal (UARJ), ISSN : 2278 – 1129, Volume-1, Issue-1, 2012.



10.22214/IJRASET

INDEX COPERNICUS

45.98

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY