



## COMPUTER VISION BASED INTELLIGENT TRAFFIC LIGHT SYSTEM

**Mr. Dhanushprashath.P,**

Student, Commerce and IT Department, Jain (Deemed to be University)

Bangalore, Karnataka-560069.

Guided by Dr. Swati Sah

Jain (Deemed to be University)

Bangalore, Karnataka-560069.

**Abstract**— Unresolved traffic is one of the main contributors to the India' declining productivity, which affects both its citizens and many industrial sectors. The nation has made several attempts to control traffic, involving the construction of highways, the extension of roads, and the execution of several traffic plans. The answer to the drawbacks of conventional traffic signal systems is being researched as one of the research directions. Existing research on traffic light systems has led to the development of intelligent transportation systems (ITS), which normally rely on real-time data on traffic density to function but are only implemented with a certain level of control. This work focused on a technique for developing traffic signaling systems that may prioritize congested lanes based on real-time data on traffic density, coupled with automatic and human control, and implemented in a mobile Android-based application. The system collected traffic images from CCTV cameras positioned in each of the intersection's lanes and delivered them to the Raspberry Pi 3 microcontroller for use in determining traffic density using image processing. To control how traffic lights operated, it used an Android mobile app store and a traffic monitoring system. The system was evaluated, and the average vehicle detection rate for daylight and nighttime, respectively, was 92.83% and 85.77%. Additionally, during daylight and nighttime testing, an overall system dependability of 92.82% and 85.77% were achieved based on the Android GUI, lane prioritization, and traffic light response. Future development included connecting the traffic light system to the Internet of Things (IoT) for a more extensive, networked application.

**Keywords:** computer vision, image processing, Intelligent transportation systems, and mobile Android- based applications

### 1. INTRODUCTION

Traffic congestion has long been one of the biggest unresolved.

crises that the India' citizens and industrial sectors have faced, and it gets worse every year. As the number of automobiles rises, this is starting to become a global phenomenon. The productivity and economics of a nation are negatively impacted by this traffic problem. According to statistics from only the United States, up to

\$300 billion was really spent on time and gasoline, and the corporate sector lost up to 70 million dollars a year as a result of traffic congestion. In addition, traffic in Manila, India, will approach "standstill levels" by 2022, meaning that during peak hours, average automobile speeds will be 5 mph or less, predicts Boston Consulting Group. According to research by the Japan International Cooperating Agency (JICA), the growing traffic congestion in Metro Manila, India, costs the economy P3.5 billion daily in missed work hours and commercial opportunities. The government has made several attempts to control the traffic issue, including the construction of new highways and roads and the execution of various traffic plans [3]. The answer to the drawbacks of conventional traffic signal systems is being researched as one of the research directions. The main objective of the intelligent transportation system (ITS), which has been proposed as a possible replacement for the traditional traffic signal system, has been to increase traffic efficiency. In order to further enhance traffic efficiency, traffic lights (also known as traffic signals) are used to control traffic flow at intersections, pedestrian crossings, and other sites. The conventional traffic light system, also known as the standard traffic light system, frequently employs fixed-timed cycles, in which the signaling lights alternate with one another at predetermined intervals [4]. It is inefficient since the traffic situation is a variable event that is always changing as a result of many circumstances, such as rush hours, accidents or collisions, rallies, roadshows, etc. ITS was created in response to these limitations. Depending on the traffic circumstances at any given time, ITS is intended to adaptably control traffic flow operations. A typical type of intelligent traffic signal system prioritizes the lanes that are busier and require more attention by detecting and computing the traffic densities of each one using sensors or cameras placed at crossings. However, for these kinds of systems to completely operate, a significant degree of human participation is necessary, which is why automated computer vision-based alternatives were developed. In order to collect

photos from various lanes, the study employed a rotational camera positioned in the middle of the junction. The images were then processed using edgedetection-based image processing methods. For the purpose of operating and controlling traffic signal lights, the traffic density and timing were determined. Another investigation photographed a traffic crossing using a camera mounted on a DC motor. The image processing techniques employed included item counting, edge detection, thresholding, picture cropping, and image enhancement. However, using computer vision-based traffic signal systems is still a difficult undertaking since the majority of these systems were not created for harsh weather conditions (stormy, foggy, and wet), which affect the accuracy of vehicle recognition. Additionally, several studies that use cameras for image processing had trouble estimating the volume of traffic at night, and controls were often made by computers or microcontrollers, which had restricted mobility and access. This study suggests creating an intelligent traffic signal system that can function at any time of day by measuring traffic density using computer vision. The system uses CCTV cameras placed at each junction lane to record traffic pictures, which are then transferred to the Raspberry Pi 3 microprocessor for image processing to calculate traffic density. For increased mobility and accessibility, traffic light management is offered both automatically and manually through an Android application. This essay is divided into the following six parts: The study is described in the first part, along with some current instances of intelligent traffic signal systems. The suggested framework for the computer vision-based intelligent traffic signal system with Android monitoring and control is covered in the second chapter. In Chapters 3 and 4, the algorithms used in image pre-processing, traffic density prediction methods, and the development of an Android application for monitoring and control are discussed. Results and analysis from experiments are shown in Chapter 5. The last chapter concludes the topic of the article by outlining the suggested work's future directions.

## II. PROPOSED STRUCTURE

The recommended framework for the method is shown in Figure 1. As shown in Figure 2, the researchers set up CCTV cameras to take pictures of the traffic at the junction's four lanes, then uploaded the data to a server. The data obtained by the server was then gathered by the microcontroller and made ready for processing. Following an image processing analysis of the captured images, the microcontroller transmitted the data on traffic density over Bluetooth to the Android mobile application. Afterwards, a text message was shown on the mobile application along with the system's car count, indicating if there was light, moderate, or heavy traffic. The mobile application may be used by an authorized user, who can then view the current traffic situation at a junction. The following are the elements of the suggested framework that are discussed:

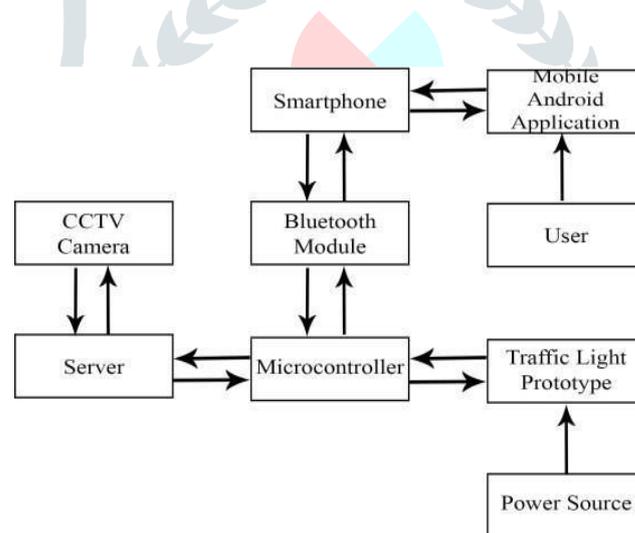


fig. 1. suggested system architecture.

**CCTV Camera:** One of the most significant components of this system is selecting the imaging equipment that will be utilized to take traffic images since both daytime and nighttime operations will be evaluated. To enhance efficiency and guarantee that the system was always fully functional, the project incorporated an infrared CCTV camera. Figure 3:

A sample capture is shown. - The host (server) operated as a link device between the microcontroller and the cameras since the microcontroller only had one Ethernet port and the quad cameras were physically linked via UTP cables.

Network Video Recorder (NVR) was used by the proponents as the system's server

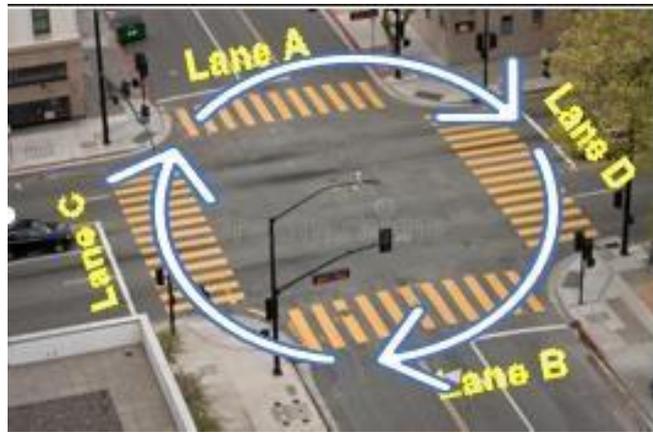


fig 2. a camera installation at a crossroads

- Pi 3 Module for the Raspberry Pi The collection, handling, processing, and transmission of data into further components

Mobile App Powered by Android: An Android app served as the controller. and the project's region for traffic monitoring. The app has the ability to manage the traffic signals. Based on the demands of the circumstances, a prototype or based on the user's preference. That's assuming the project has a traffic monitoring area. The Android program allows you to see the current traffic density.

- Bluetooth Module: To establish a wireless connection between the microcontroller and the smartphone, the Bluetooth module acts as an intermediate device.

Following the completion of all calculations and processes, the microcontroller would communicate the results to the traffic light prototype, along with the user's requests. The "Go" and "Stop" signals for each lane were emitted by this prototype, whether the system was being operated manually or automatically.



fig. 3. traffic image capture

### III .IMAGEPROCESSINGTECHNIQUES

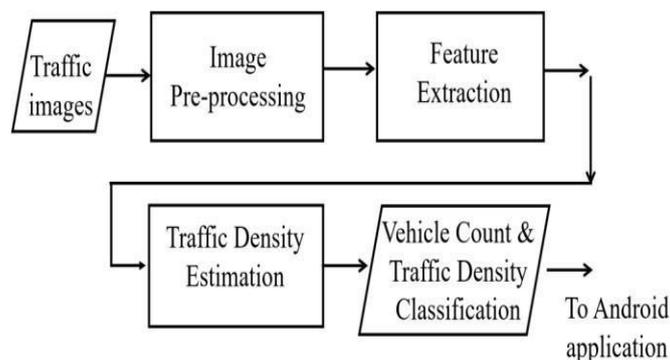


fig. 4. review of image processing techniques

To help the traffic light system decide which lane to prioritize, it was critical to take traffic photographs of the junction. The pictures were captured using a CCTV camera placed close to the roadside under scrutiny. After being captured, the pictures were sent to the CPU for image processing. Figure 4 illustrates the three primary steps in the picture processing for this study. It covers the following: traffic picture input, image pre-processing, feature extraction, traffic density estimate, and vehicle counting and traffic density classification output. Prior to feature extraction, image pre-processing seeks to enhance the input pictures by reducing undesirable characteristics or strengthening crucial ones. The process of extracting useful parameters from an image is known as "feature extraction," and it is often carried out after picture segmentation. Segmentation separated the foreground photos of the vehicles from the environment's backdrop, which contained people and buildings. The number of cars recognized during feature extraction was necessary for estimating traffic density. Image preprocessing is the initial step in the procedure. First, picture cropping was used to process the camera-captured images. Only the desired portion of the picture is processed by the system as a consequence of the procedure that involves eliminating the undesirable portions of the image. After cropping, a grayscale transform was used to convert the RGB values of the pixels to their grayscale equivalents. Following this, Gaussian blurring was used to facilitate binarization. By combining the clipped picture with a Gaussian filter, the blur effect was created. Edge detection is the subsequent step. The goal of this method is to identify object boundaries, which are defined by the white pixels that highlight an item in an image. This study makes use of canny edge detection, with results similar to those in Figure 5. This kind of edge detection approach locates the local maxima of the picture gradient to identify the image edges.

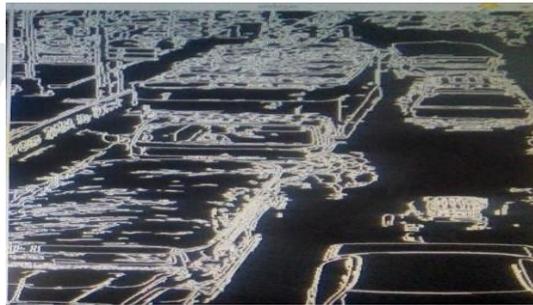


Fig. 5. picture that has been binarized and has edges identified

The technique of "segmenting" a picture involves dividing it into sections that are of interest and parts that are not. The target items to be segmented in the photos are the automobiles. Background pixels and vehicle pixels were separated and given different labels using connected component analysis. Instead of labelling each individual member point, the boundary points of the binary regions were established via contour-following techniques utilizing two vectors. The contour array, which consists of a set of boundary points for the different binary sections in the picture, is one vector. The second vector is an array of indices that enabled the contours to be treated hierarchically, allowing for the consideration of certain contours that represented holes in an item that were located inside of other contours. Following that, the contour segmentation binary areas of the cars were tallied to determine the traffic density.

#### IV. MOBILE ANDROID-BASED APPLICATION PROTOTYPE DESIGN AND ARCHITECTURE

The newly created intelligent traffic light system was combined with the "e-Trapiko" android application, which helped traffic enforcers monitor the functioning of the intersection's traffic flow.

The sample display that might be viewed via the created mobile application is shown in Figures 6 to 9. Figure 6 depicts the login screen where a certified traffic enforcement official might log in using their ID number and password of choice. Only authorized users were allowed to access the program, which decreased security and prevented chaos in traffic flow.



fig. 6. security options for android app

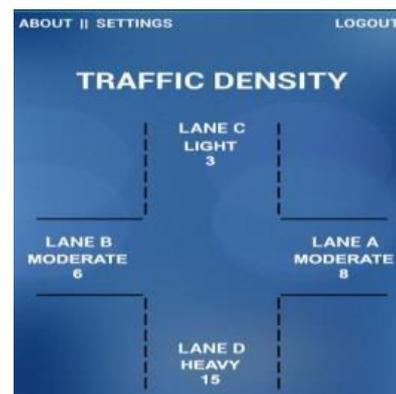


fig. 7. illustrative traffic density display

In Figure 7, which shows the traffic surveillance screen for the smartphone app, it is shown how many vehicles are now in the intersection's different lanes, as well as how densely they are currently travelling. There were two modes of operation available: automatic and manual. The system's autonomous operating function aims to make it easier for traffic enforcers to monitor and manage the flow of traffic by projecting real-time traffic statistics along the various lanes of a junction. Whether there is low, moderate, or heavy traffic in each lane, the switching of signal lights is entirely dependent on that traffic density. The system's manual operating capability was created to ensure continued operation even under adverse to severe weather conditions, such as stormy or wet conditions. The camera could not be as accurate in poor weather as it is in good weather, which might lead to mistakes in the pace at which vehicles are detected. The safety of the traffic enforcer was another consideration when introducing the manual operating option to the system. By remotely utilizing the built-in Android application, traffic enforcers were able to regulate traffic flow by turning on and off traffic lights. It is safer for the traffic enforcer since they do not need to be in the middle of the junction, where there is a greater risk of being struck by a car. Additionally, automated operation would not work in the event of a collision along any of the intersection's lanes since the impacted lanes would likely have heavy traffic and could need to take precedence over other lanes in order to completely control the flow of traffic. When a traffic officer is on duty at the junction, the choice to switch the signaling lights to manual operation would exclusively be up to the officer's discretion.

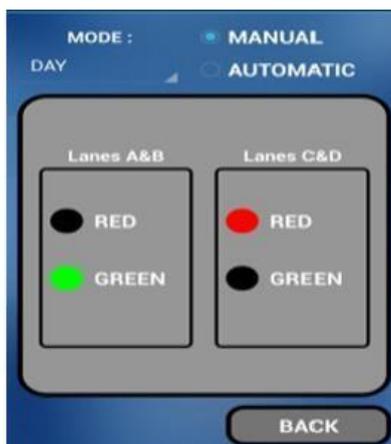


fig. 8. sample display with manual setup



fig. 9. sample display with automatic setup

Fig 8 depicts the application's manual mode, in which a traffic enforcer used a smartphone and Bluetooth to manually operate a traffic light prototype. Figure 9 depicts the application's automated operating settings. The traffic enforcer may determine the time limit and vehicle allotment for each traffic density assignment based on the width of the intersection's roads.

## V. INFORMATION AND OUTCOMES

Figure 10 depicts the model testing location in Bangalore, Electronic City, Karnataka.

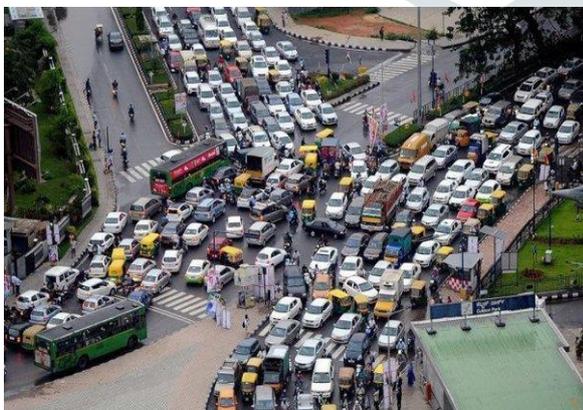


fig. 10. system testing site



fig. 11. capture daytime



fig. 12. nighttime

The actual test consisted of 84 daylight trials and 30 midnight trials, respectively. Figures 11 and 12 show examples of photos that were captured during the day and at night, respectively. A traffic light prototype test, a test of traffic density estimation (actual vs. systems), a test of the functionality of the application interface, and a test of lane prioritization were all performed. A preliminary test was performed before the main test to ensure that the traffic signal model and smartphone app were correctly integrated and working smoothly with manual operation. To ensure that there was no bias in the test results, the second test was carried out under the observation and direction of traffic enforcement officers. This test was performed to evaluate the precision and dependability of the system's image processing methods. The final test checked to see whether the Android application's GUI accurately reflects the number of real vehicles and the traffic density that goes along with it. The lane prioritization test came last. This experiment was carried out to see whether the project's primary goal—prioritizing lanes that were much more crowded than the other lanes—was really accomplished.

Equations used for the calculation of the accuracy and reliability of the system: For the Accuracy:

$$\text{Accuracy} = (1 - \text{Error rate}) * 100 \%$$

$$\text{Error rate} = [(M - S) / M] * 100 \%$$

Where:

M – Manual vehicle count present

S – System count presented in the application

For the Reliability

$$\text{Reliability} = (\text{No. of Success} / \text{No. of Trials}) * 100\%$$

The Android browser estimated traffic density based on the number of vehicles listed in Table 1 and the time allotted for the "Go" traffic signal.

**TABLE 1**

traffic density and timer

traffic density	vehicle quota	timer
low	0-5	10 sec
normal	6-10	30 sec
high	11-above	40 sec

**TABLE 2.**

System traffic and vehicle count accuracy

<b>bays</b>	<b>day screening</b>	<b>night screening</b>
A	90.49%	85.70%
B	90.71%	85.74%
C	92.78%	85.59%
D	92.70%	85.79%
average	92.84%	85.77%

The accuracy for each lane during daylight and nighttime testing is shown in Table 2. The system's typical vehicle detection rates during the day and at night are 92.70% and 85.77%, respectively.

**TABLE 3.**

System reliability summary

<b>bays</b>	<b>day screening</b>	<b>night screening</b>
android to traffic light control	92.71%	85.67%
Android GUI	90.56%	84.64%
Lane Prioritization	92.70%	85.59%
Average	92.62%	85.77%

Table 3 displays the system's overall dependability during both daylight and nighttime testing. The system achieves an average of 92.71% and 85.77% reliability using Android traffic light control, the Android interface, and lane prioritization.

## CONCLUSION

The creation of the e-Trapiko system opens the door to a creative strategy for enhancing the current traffic light system by fusing the most recent ITS innovation with Android technology. The technology not only allows law enforcers to successfully monitor the traffic condition in theregion through a mobile app application named "e-Trapiko,"but it also allows the system to regulate road traffic operations using crowd density calculation techniques via image analysis technologies. Computer vision has been successfully used to estimate traffic volume and integrate it into an automated traffic signal system. The system successfully detected vehicles 92.71% of the time during daylight operation and 85.77% of the time during nighttime operation. A smartphone application built on the Android operating system is being developed and tested to display the volume of traffic in each lane and control the traffic lights automatically or manually. By enabling the traffic police to manage traffic at a junction without having to physically stand at the intersection, the mobile application successfully increased personal safety. With time and vehicle limits that could be adjusted for the right traffic densities, the system operated well, even in automated mode. The system performed reliably overall, with daytime performance of 92.71% and nighttime performance of 85.77%. Future studies will combine the ITS with the Internet of Things for more technical adaptability and to be capable of keeping up with the lightning-fast pace of technological progress.

**REFERENCES**

- [1] P. Jadhavm P. Kelkar, K. Patil and S. Thorat, “Smart Traffic Control System Using Image Processing”.International Research Journal of Engineeringand Technology (IRJET). 2016
- [2] Rahishet, A. Sahoo, A. Indore, V. Vdeshmukh. 21stIRF International Conference. 2015
- [3] A. R. Farhadian, M., Miyake, S., Analysis of the mental workload of city traffic control operators whilemonitoring traffic density: A field study, InternationalJournal of Industrial Ergonomics, pp. 170-177, 2016
- [4] Tyler, Jessica , “Traffic jams, commuting around the world: pictures “.business insider, Accessed May 7, 2018[Online]  
Available: <http://www.businessinsider.com/traffic-jams-commuting-world-citiespictures-2018-1>
- [5] MMDA Programs and Projects, Accessed June 4, 2018[Online]  
Available: <http://www.mmda.gov.ph/images/pdf/Transparency/V/Progra ms-Projects>.
- [6] Saleh, A. Adeshina, S.,Galadima A., Ugweje, O., Anintelligent traffic control system, 2017 13th InternationalConference on Electronics, Computer and Computation (ICECCO), pp. 1 - 6 , 20

