

# Automatic Vehicle Counting for IOT based Smart Traffic Management System for Indian urban Settings

Vinaykumar H R<sup>1</sup>, Saravanan C<sup>2</sup>

Student, Master of Computer Applications, RV College of Engineering, Bengaluru, India<sup>1</sup>

Professor, Master of Computer Applications, RV College of Engineering, Bengaluru, India<sup>2</sup>

**Abstract:** For smart city management, efficient handling of road traffic is one of the key aspects. Traffic congestion can be managed effectively, if the numbers of vehicles that are to pass through a crowded junction can be pre-estimated in time. The proposed method presents a framework, which has the capability to continuously convey the vehicle count and generate an alarm in case of large vehicle gathering to the controlling station in the Chandigarh or alike urban Indian cities. The number of vehicles passing through a location well before the required traffic junction can be estimated using the help of image processing techniques. Further, the monitoring details can be shared to a distant controlling centre situated anywhere in the city through internet usage. The performed experiments demonstrate the effectiveness of this Internet of Things (IoT) based technology.

**Index Terms:** IoT, Firebase, Blob Detection, Raspberry Pi.

## I. INTRODUCTION

Indian city managing system is a combination of many interdependent systems, in which traffic management plays a significant role. Moreover, it can be stated as one of the key aspects of the smart city. The world is moving very fast and it has to keep moving this way for continuous development. On contrary, modern transport is failing to provide smooth transportation system to the people. Excessive traffic jams lead to delays in reaching workplace or home, wastage of fuel, wear and tear on vehicles or even a road rage by the stressed and frustrated motorists. In addition, an increasing population is directly resulting in an increasing traffic related problems such as over speeding, accidents, hit and run, and so on. Criminal activities like mobile snatching at traffic signals also happen in metropolitan cities during long traffic jams. Therefore, an intelligent traffic management has evolved as a compulsory requirement for a prosper civilisation. Currently, smart and adaptable traffic control systems are being preferred over fixed time systems in most of developing nations. This type of traffic controlling is being monitored largely through centrally controlled systems/servers. Considering this, the Internet of Things, which has proved its worth in almost everything in our daily life, can be considered as a tool for managing the traffic through central server. In our proposed work, the count of vehicles passing through a passage that is at a few distance before the active traffic jam points can be conveyed to control station for managing the traffic flow. The timely received information for the traffic jam node in the city can be communicated using internet and cloud to manage the vehicle inflow. For counting of vehicles in real time, we have used image-processing applications in Open-CV software. We have proposed to solve the issue for Chandigarh traffic with the help of economical raspberry – pi board. The implementation of our proposal is inexpensive and asks for very little infrastructure. Therefore, the proposed system manages the traffic on local and centralized servers by exploiting the concepts of IoT and Intelligent Image Processing. In proposed system for Indian urban settings, the real time video data is acquired is firstly. Then, it is divided into frames, and then after binary conversion and noise removal, blob detection is performed and finally the count is estimated using the proposed vehicle counting method. The experiments are performed on Chandigarh city road scenes. The representation of traffic data in statistical form can also be helpful to authorities for real-time controlling and managing traffic. Moreover, it may also be helpful for future planning. The obtained vehicle count is then made available to a controlling station by using real time internet based database.

The rest of the paper is structured in four sections. Section II discusses the state of the art techniques. The proposed system's methodology is presented and discussed in Section III whereas a discussion on results is presented in Section IV. Section V concludes the research.

## **II. RELATED WORK**

The technology of IoT has been used recently in variety of applications. This section presents work carried out using IoT in traffic management.

An automatic traffic light control system was proposed in [1] using IoT, IR sensor and cameras. This work provided an IRsense based solution that makes traffic signals to shift the lights (red/yellow/green) dynamically. The sensed data gathered from IR sensor was transmitted by the Wi-Fi transmitter and was received by the raspberry-pi controller. The method had an advantage of ruling out the 'unwanted wait' for vehicles during traffic light operations in more crowded regions.

An IoT based traffic control system was presented in [2]. This system utilised MATLAB software for image processing and used WI-FI transceiver module for transmitting the vehicle count information . The next signal was dependent upon the inputs from the previous signal's traffic density. The hardware used were raspberry-pi and Arduino. The system can be made more effective, if instead of Wi-Fi transceiver, direct data communication through cloud have been preferred.

A system to control traffic was proposed in [3], which used wireless transmitter to transmit the images directly to main server. Then, the server system did the processing of images and found the traffic density. The time span of red light for a particular lane of the intersection was determined by traffic density on road. However, the method can be improved if the information to be transmitted is not in the form of images. Rather, the processed output information is transmitted itself, it will save lot of time and load of communication.

A method to control the traffic congestion was proposed in [4] by using sensor sources that transmitted continuous infrared light. The presence or absence of the vehicle was detected on the basis of the intensity of reflected light back to the sensors. The method proposed to provide the framework for a smart traffic routing systems. Here, the sensors used had a drawback that their output is dependent on changing temperature and humidity conditions.

In [5], a progressive Internet of Things (IoT) based traffic management system was developed. In this, embedded circuit, which operated using RFID with clustered system, was used. For working with big data analytics, Hadoop was incorporated. Along with this, supervised learning methodologies were used in this research.

A similar approach was presented in [6] that was aimed at detecting vehicles and get their location by using sensors and RFIDs. In this, the data acquired by sensors was sent to centralized controlling centre by using a wireless connection for further processing.

An IoT based system for improved parking management was proposed in [7] for heterogeneous traffic condition countries like India. Image processing was used in form of Optical Character Recognition to verify employee, further extracted number plate was forwarded to server system. The system ensured that only registered candidates would be allotted the available parking lot. A micro controller facilitated a real-time analysis and optimization of vehicles.

A lane based traffic monitoring system was presented in [8] with the help of Ultrasonic sensors. The received data from sensors was processed using controller. After this, it was transferred to server through Wi-Fi module. Traffic was controlled by traffic signal control mechanism, which was dependent upon the detecting traffic levels at lanes.

An adaptive system designed for managing the traffic system in reference to the automatic passing of emergency vehicles was proposed in [9] using IoT technology. They used Raspberry PI, Node MCU, RFID Tag and Reader for making a system to change signal changes by properly communicating with the sensors in vehicle.

A scalable architecture for urban IoT environments was proposed in [10] that was primarily based on Lambda Architecture, ingesting the data through web services that provide a common interface to our system, and then, storing the data in a distributed, scalable, NoSQL data store. The researchers provided a study on how cloud-computing and big-data management can help in decision making for traffic management in smart cities. The work in this paper also implemented a concrete proof-of-concept, based on data publicly available from the city of Edmonton.

However, the proposed method focuses at providing the vehicle count data in Indian Urban cities to control station for a user specific time interval at user specified place that can help to manage traffic, which is going to affect the next busy junction. The next section provide the details of the procedure adopted during the proposed framework.

## **III. PROPOSED METHODOLOGY**

The automatic counting of vehicles passing through achosen destination was obtained by using image-processing techniques. A camera system was used to acquire the real time video footage of the traffic flow through road. After acquiring the video through camera, a portable raspberry-pi processing system was used to work on this data. Finally, after the average vehicle count per specified time interval has been found, the same processing system is used to communicate the same information to central control system. The complete details of how the work has been carried out are explained in following two sub sections:

### A. Vehicle counting using image processing

After obtaining the video footage from the camera system, the processing was done on raspberry pi using Open CV software. The flow of work done on image to acquire the number of vehicles passing through a region of interest area has been illustrated using figure 1. The description of steps adopted is provided as follows:

- As seen in above figure, the first step involves extraction of the individual frames from the video sequence This is performed using the 'cap.read()' command in Open-CV.

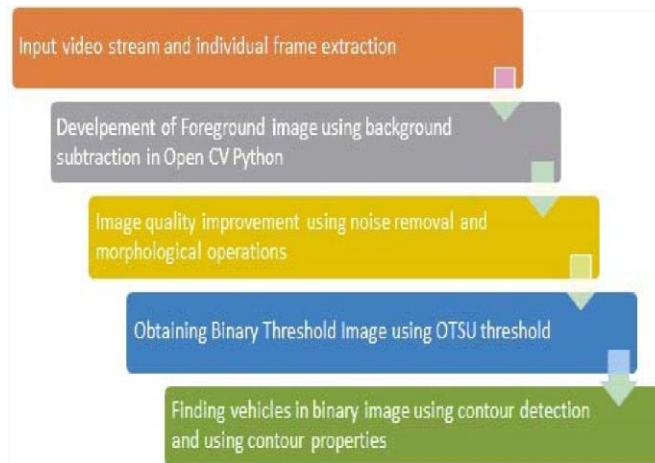


Fig. 1: Systematic flow diagram to illustrate the image processing operations for obtaining the count the present vehicles region. The vehicle counter only updated if the detected vehicle was present in this specified interest area.

Once, we found the total no. of vehicles present in single frame of the image, next step was to realise the average attendance of vehicles in the region of interest area. This information is particularly helpful in determining the load of traffic in a certain time interval. For this, per second interval was chosen to compute the average value of vehicle present during this time interval. The value of count to be communicated via internet was averaged in each time instant. Next part discusses the procedure of transferring the data from raspberry- pi processor to the central server system via internet.

### B. Transmission of information for traffic management system

After this, single image frame was worked upon to find the total number of vehicle objects present in the frame.

- The default background subtracting operation of OpenCV: 'create Background Subtractor MOG2()' was incorporated to generate a mask that has the ability to deduct the background from the image frame. Now, after application of above steps, only foreground objects were left in the image frame.
- To reduce the noise, foreground image was passed through a Gaussian filter. Further, a morphological operation of closing was applied to properly embrace the objects present in the image.
- In the next step, the processed image was converted to binary form, in which, the pixel value was converted either to '1' or '0' depending upon a threshold value. The OTSU bimodal threshold method was used to generate this threshold value. The binary threshold image is generated, because it is easier to detect the contours of objects in such kind of image.
- After this, the contours are detected using the 'findContours()' operation of Open-CV. Once, all the contours were detected in the image, their different properties could be used to find the desired class of objects. For our use, the size of the detected contours was taken as a property for classifying it as a vehicle or not. The region of interest was selected in the image by plotting two lines on the image frame: one at the upper bound of the region, and other at the lower bound of the using internet

This process of communication between the camera guided processing system and the final user server system via internet can be explained in four stages. The complete block representation is presented in figure 2 and each step is discussed in detail as follows:

- Fetching per second Vehicle Count Data from Open CV: The average value of vehicle present during respective time interval was fetched and stored inside a variable that updated after every fixed time interval.
- Interfacing of Firebase Google server with Raspberry Pi using Python: Next, to send processed information remotely to control room we must have a server and portable Raspberry system must have internet connection. For the server we have used Google Real Time Database server, i.e. Firebase. To use Firebase Real Time Database we must have compatibility with the Processor or Controller. To do so we have used Python language.

- Establishing Connection with Firebase (Real Time Database): After interfacing, next step was to establishing connection with our database. For that, we had created new database on firebase server. Then, connected this to database using Credentials (API keys (Application Programming Interface), End Point URL (Universal Resource Locator)) using Python.
- Sending Vehicle count data to Real Time Cloud: After connecting to database, our task was to send vehicle count data to server remotely. For that, we sent our vehicle count data obtained from Open CV to



Fig. 2: Systematic flow diagram to illustrate the image processing operations for obtaining the count of the present vehicles



Fig. 3: Glimpse of video sequences taken for experimental purpose; (a), (b) are the Chandigarh traffic footage and, (c) is the Delhi Traffic footage

Firestore using JSON (JavaScript Object Notation) through Python with the Time Stamp.

Once, the data had been sent using the above steps, the process repeats itself after every one second to continuously provide the monitoring details of the capturing system. The results of the work done and the output of the complete process is briefed in the following section of the paper.

#### IV. EXPERIMENTAL RESULTS DISCUSSION

This section provides insight into the experiments performed and the results obtained. The live traffic video can be captured using CCTV cameras or ordinary cameras. For the proposed work, already available videos of Indian cities have been taken. These videos resemble the exact conditions of road and the traffic in Indian urban settings. Two video sequences of Delhi and three sequences of Chandigarh are selected for demonstration of the proposed research work. As seen from figure 3, the video sequences provide a glimpse of the vehicular traffic in two urban Indian cities. It is clear from the above figure, that vehicle detection and counting is a challenging task in these types of settings, because of the presence of a large number of different types of vehicles along with the absence of a systemised lane rule procedure. Therefore, it becomes necessary to process the raw images acquired before starting object contour detection. The result of every step of execution of the proposed framework as mentioned in the work methodology section is discussed here by taking the example of one video sequence.

The figure 4 shown above depicts the resulting images obtained for the complete process. The process starts from extracting a single image frame from the real-time CCTV footage, which is shown in figure 4(a). Then, next shown is the foreground image obtained after applying Gaussian background subtraction through Open CV in figure 4(b). The noise from the foreground image is removed by passing it through a noise removal filter. Along with this, the morphological operation is applied that helps to solidify the object shapes properly, which is clearly seen in figure 4(c). The final figure in 4(d) shows the major information. The blue lines mark the region of the road selected for detecting the count flow of vehicles. The contours that lie in the region of interest area are selected. Further, these are classified as vehicles after they pass the size threshold level condition. As seen in this figure, two vehicles lying in the selected portion of road are detected and are shown by drawing green colour rectangles around them. After the average count of vehicles passing through a specified interest area was calculated, this information is communicated to the central traffic control system through

internet. The following figure 5 shows the output information fetched on the remote control system. In above figure, the real time firebase database updates its value after specified time interval. This real time server has the capability to update its data after every 30ms. However, for our purpose of traffic management, time interval of one second seems to be appropriate. Therefore, throughout the experimentation, the average vehicle count during time interval of 1 second was transmitted through Raspberry pi based processing system. The vehicle count per second for two different timing instants is displayed using figure 5(a) and figure 5(b) respectively. This information is updated regularly. So, in this way the information regarding the total number of vehicles present at every time instant is provided to traffic controlling station. When suddenly, the flow of traffic becomes large at some specified places well before an important junction/intersection point of city, the traffic jams can be avoided at that junction by utilising this information. The routes of traffic can be diverted, so that excessive accumulation of vehicles does not take place at

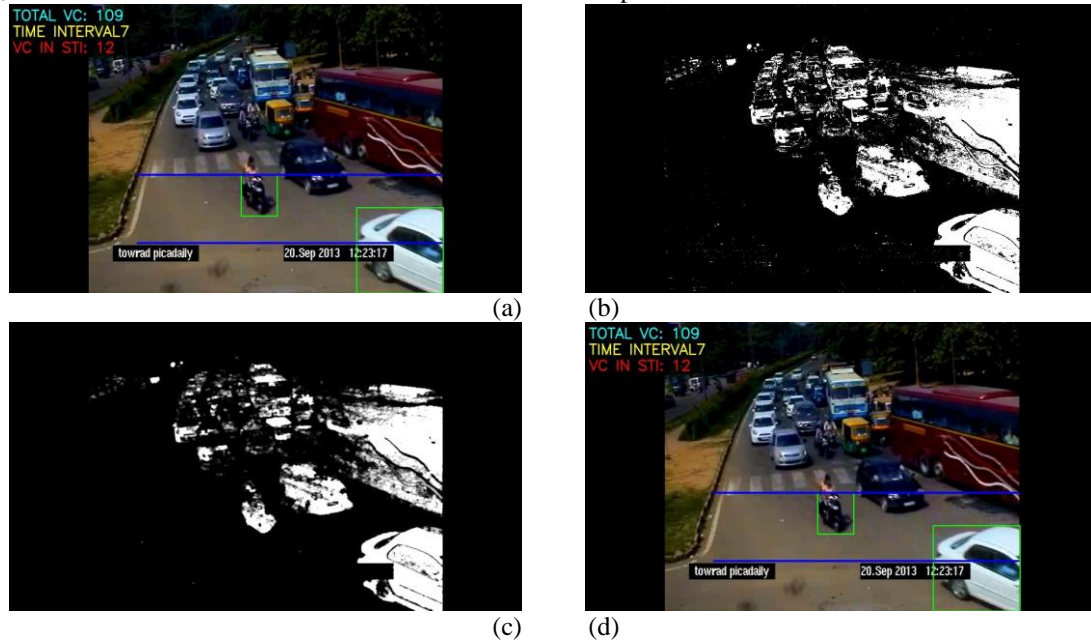
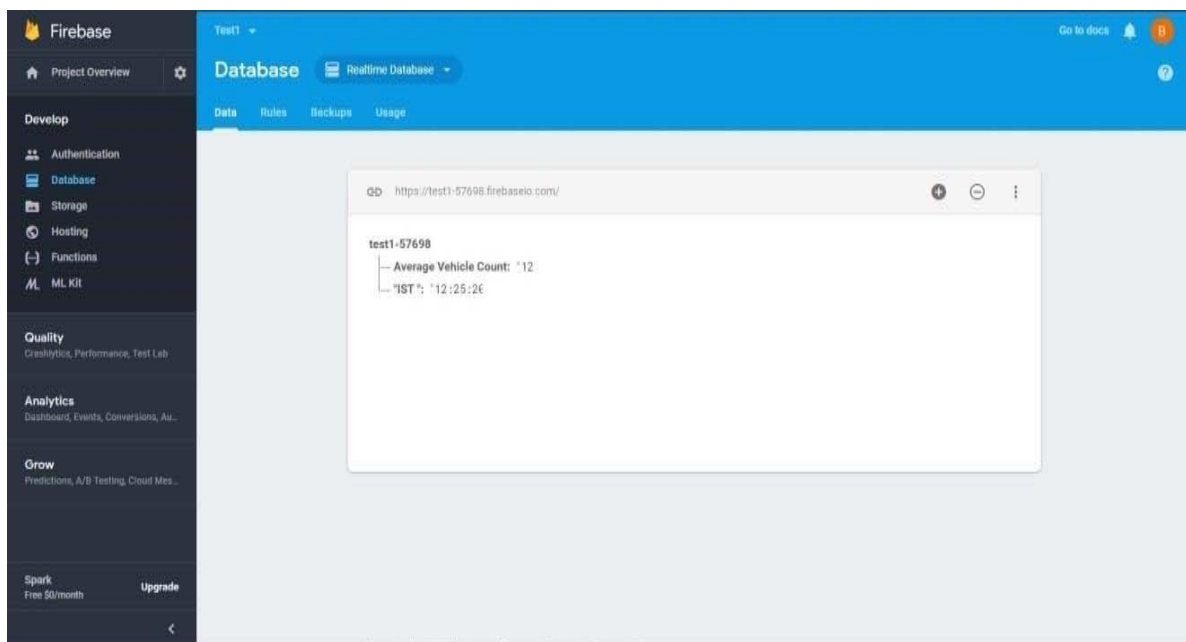
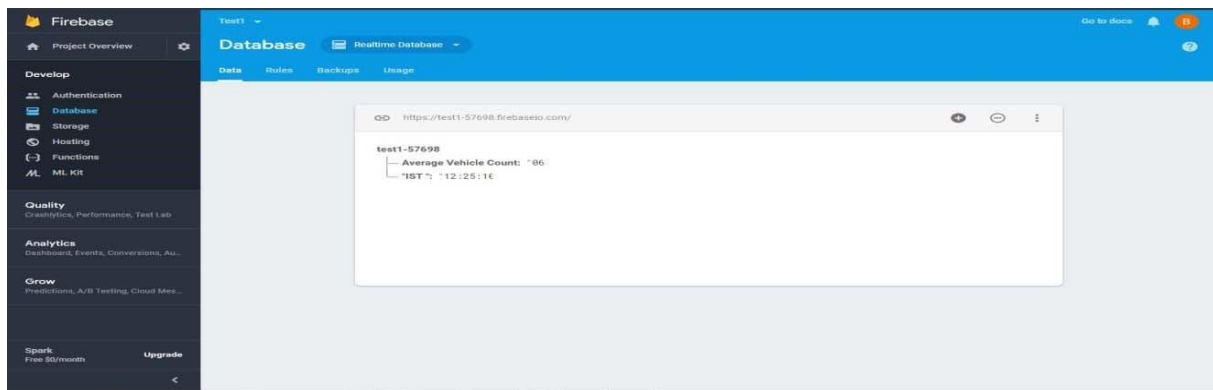


Fig. 4: Result of Image processing steps on a Chandigarh Road Traffic scene; (a) Original Individual Video Frame (b) Foreground Image (c) Image after noise removal and morphological operations (d) Vehicle Detected Image using contour detection



(a)



(b)

Fig. 5: Output acquired via internet on a remote control station indicating Traffic flow information at different time instants all.

## V. CONCLUSION

The amalgamation of computer vision technology with the IoT approach helps in developing an artificially intelligent system that can handle traffic in an effective manner. As we have used only raspberry-pi based system as hardware in the proposed framework. This seems to be a cost effective and portable system to help in decreasing the traffic congestion on intersections at peak points. Future work on the system can be to automate the complete vehicle diagnosis process that can further provide real time classification information on different type of vehicles. So, that the diversion of vehicle for traffic management can be done more robustly.

## ACKNOWLEDGMENT

The authors would like to express sincere gratitude to Director, CSIR-CSIO for providing infrastructural facilities. This study was supported in part by CSIR- CSIO under the FTT project MLP 0043.

## REFERENCES

- [1] L. P. J. Rani, M. K. Kumar, K. S. Naresh and S. Vignesh, "Dynamic traffic management system using infrared (IR) and Internet of Things (IoT)," Third International Conference on Science Technology Engineering Management (ICONSTEM), Chennai, pp. 353–357, 2017.
- [2] E. Basil and S. D. Sawant, "IoT based traffic light control system using Raspberry Pi," International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, 2017, pp. 1078–1081, 2017.
- [3] T. e. a. Osman, "Intelligent traffic management system for a cross section of roads using computer vision," in Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual, 2017.
- [4] V. S. A. M. S. D. K. K. Swathi, "Traffic Density Control and Accident Indicator Using WSN," International Journal for Modern Trends in Science and Technology, vol. 2, no. 4, pp. 2455–3778, 2016. [5] Mahesh Lakshminarasimhan, "IoT Based Traffic Management System", Mach 2016.
- [6] H. O. Al-Sakran, "Intelligent traffic information system based on the integration of Internet of Things and Agent technology," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 6, no. 2, pp. 37–43, 2015.
- [7] S. Rane, A. Dubey and T. Parida, "Design of IoT Based Intelligent Parking System Using Image Processing Algorithms," IEEE International Conference on Computing Methodologies and Communication, pp. 1049–1053, 2017.
- [8] V. S. Nagmode and S. M. Rajbhog, "An Intelligent Framework for Vehicle Traffic Monitoring System using IoT," IEEE International Conference on Intelligent Computing and Control, 23-24 June 2017.
- [9] S. V. Bhate, P. V. Kulkarni, S. D. Lagad, M. D. Shinde and S. Patil, "IoT based Intelligent Traffic Signal System for Emergency vehicles," International Conference on Inventive Communication and Computational Technologies, pp. 788–793, 2018.
- [10] D. Serrano, T. Baldassarre and E. Stroulia, "Real-time trafficbased routing, based on open data and open-source software," IEEE 3rd World Forum on Internet of Things (WFIoT), Reston, VA, pp. 661–665, 2016.