Automatic Vehicle Counting For IoT Based Smart Traffic Management System

Adarsh S Navalgi[1], Farhana Kausar[2]
[1, 2] Dept of Computer Science and Engineering
[1, 2] Atria Institute of Technology, Bengaluru, Karnataka, India

Abstract- *In a smart city management, the key aspect is efficient handling of road traffic. If the numbers of vehicles that are to pass through a crowded junction can be pre-estimated in time, traffic congestion can be managed effectively. 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 an urban city. 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 and the clustering algorithms model which is based on KNN algorithm. Using this algorithm new model will be liable to determine expected required timing as per provided inputs to the signal which is vehicles count. The input of these systems is vehicles counts on each side of the road from crossing signal. The performed experiments demonstrate the effectiveness of this Internet of Things (IoT) based technology.*

Keywords- IoT, Firebase, Blob Detection, Raspberry Pi,k nearest neighbours.

I. INTRODUCTION

Considering Indian city managing system, it is a combination of many interdependent systems, in which traffic management playsa significant role. 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, it 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 civilization. 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 with the help of economical raspberry – pi board. Further, implementation of Intelligent Transportation System (ITS). ITS improves Vehicle to vehicle and Vehicle to Infrastructure communication for improving road facilities rather than increasing road capacities or developing new roads. This is possible because of ITS, it utilizes advanced information and communication, and this communication will be helpful for decreasing traffic congestion and to reduce the accidents on the road, which is dangerous in the urban areas.

Managing traffic signal timing is one if the key thing in the urban areas. Managing to time on the road will decrease the waiting time of the drivers on the road, and that will help to reduce the fuel consumption. This is done with the help of the ITS.” In this system, we are going to use IR Sensors. IR sensor is also called as an Infra-Red spectrum. IR sensors have 2 parts in it, one is the transmitter and second is a receiver. The transmitter is used to transmit the light and receiver keeps on receiving the light. When this connection is interrupted, the counting process is started, i.e., when the receiver does not receive the light transmitted by the transmitter it is said that the object is there in between transmitter and receiver. The line of sight concept is used in this approach.

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 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.

II. RELATEDWORK

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. PROPOSEDSYSTEM AND ARCHITECTURE

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 inOpen-CV.

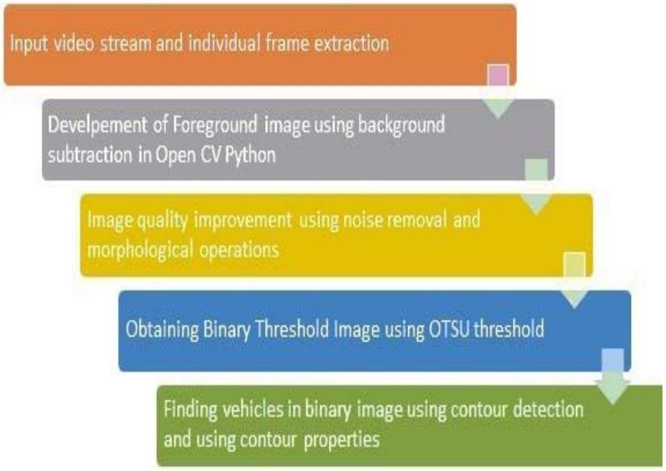

Input video stream and individual frame extraction
Develpement of Foreground image using background subtraction in Open CV Python
Image quality improvement using noise removal and morphological operations
Obtaining Binary Threshold Image using OTSU threshold
Finding vehicles in binary image using contour detection and using contour properties

Fig. 1: Systematic flow diagram to illustrate the image processing operations for obtaining the count the present vehicles

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 imageframe.
- 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 ‘find Contours( )’ 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 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 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)) usingPython.
- 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 CVto Firebase cloud using JSON (JavaScript Object Notation) through Python with the Time Stamp.

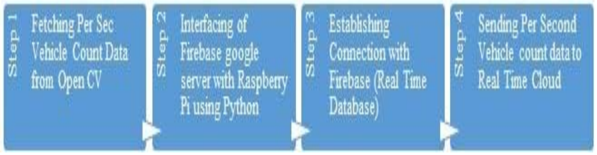
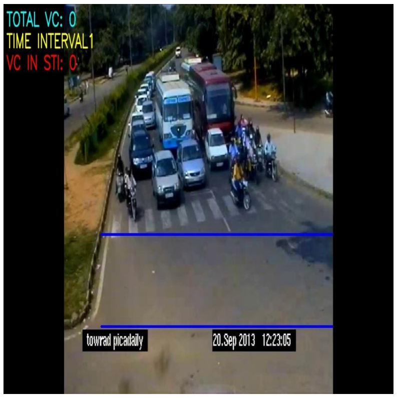

Step 1
Fetching Per Sec Vehicle Count Data from Open CV
Step 2
Interfacing of Firebase google server with Raspberry Pi using Python
Step 3
Establishing Connection with Firebase (Real Time Database)
Step 4
Sending Per Second Vehicle count data to Real Time Cloud

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



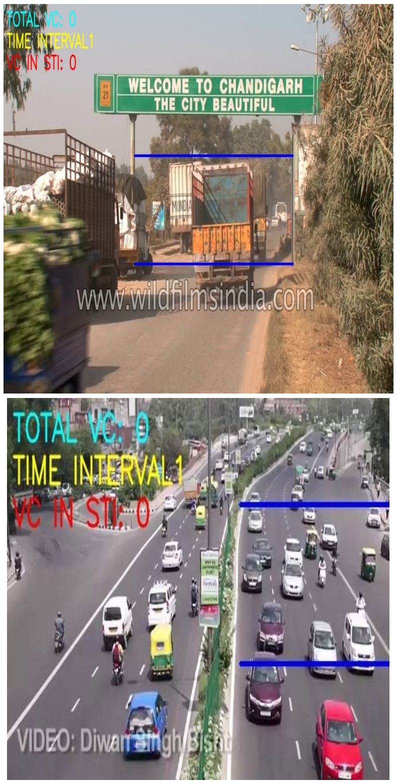


Fig.3:Glimpseofvideosequencestakenforexperimentalpurpose; Thesearethecitytraffic footage.

Once, the data had been send 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. Everyday traffic congestion bigger issues are a daily basis. So automation systems are currently not available in India. We need of IoT to utilize in the traffic signal monitoring systems and to control it in an advanced controlling system. Any system is designed to act smartly with higher control features for all four side way traffic systems. Every road towards heavy traffics of vehicles in higher counts. We need to define the priority level of traffic in our TMS on the basis on which least or highest priority. Traffic management system-TMS key appliances to control over traffic as per population of vehicles ID that particular area. So every road lane needs IR sensor to monitor and capture data of vehicles count in that lane. In this proposed system depends on the count of vehicles from the road lane IR data we are allocating higher time rate for that signal. This systems model using more numbers of IR sensors, for automation control microcontroller, with Bluetooth controller, as well as Android mobile device and finally PC-server. Any of these sensors surround with IR transmitter & receiver for placing in both directions of roadlane.

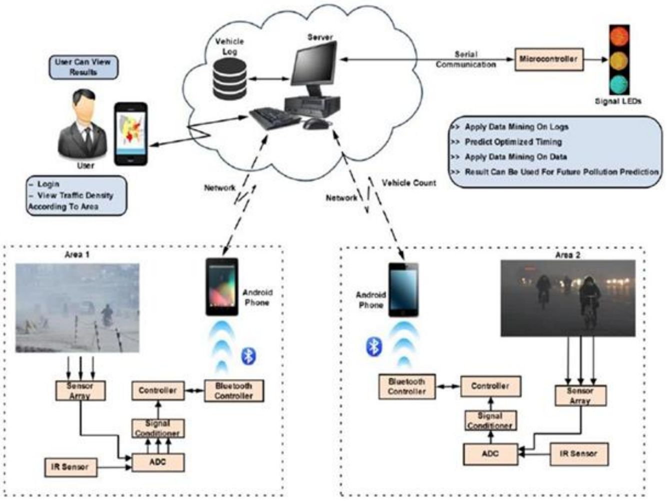


Fig. 1: TMS architecture diagram

Traffic Monitoring System–TMS using multiple IRs in IOT model to learn vehicles count in real time then updates signal timing of every side traffic lights according's to the predicting factors using KNN algorithms. It uses IRs to collect signals from numerous inputs on basis IOT model setup to learn vehicles count available in a traffic signal. So it will get cleared depends on the dependability of vehicles count to either increase or decreases timing of particular signals using KNN algorithms by data analysis.

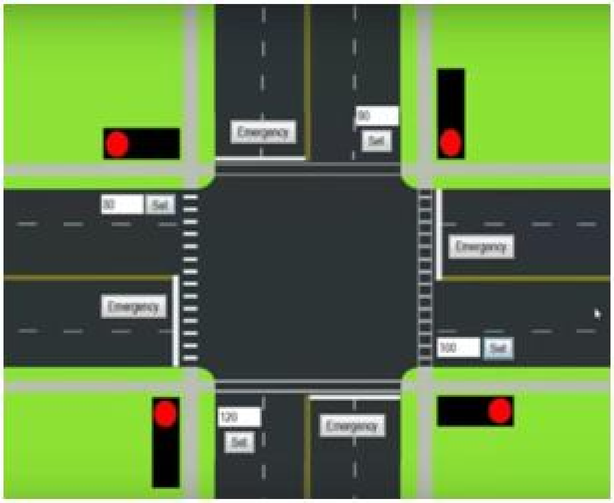


Fig. 3: Signal model **-** IR sensors for vehicles count

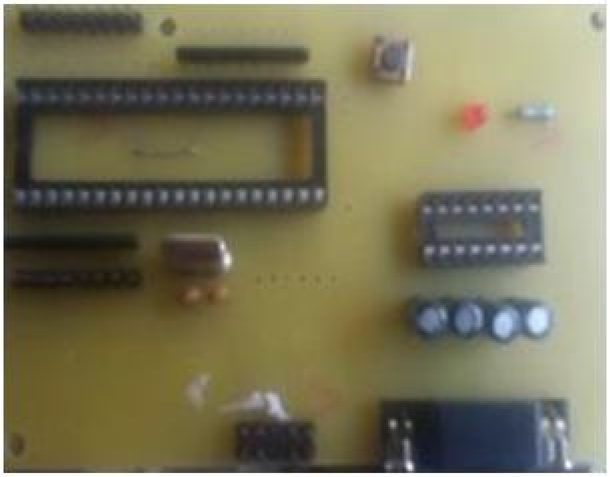


Fig. 2: Microcontroller Kit in TMS

In this implemented application considered three major works as modules processes of the project.

These modules describes in detail about work done in the project which is listed below:

1. Android Application
2. Server Side Communication
3. User Communication

1. Android Application: In this Android 1st module about android kit utilized to transferd at a from on side themicro-controller to the other side server communication. Then it's coded for the android application. The server performing communication operation towards data transaction.

2. Server Communication: This module is used to perform algorithms computation using clustering technique from data given inputs using an android device, which is used to store all the previously stored data i.e., like the history of all the previous vehicles already used the same signal crossing. KNN model algorithm is utilized for clustering model extraction from the given inputs, which is working on non- parametric methods for classification & regression model of data analyses. This input which contains k closest outputs of training examples for the feature space. K-NN working on the instance-based learning those function is approximated as local values. This project always collects data for four side road crossing as the vehicle counts. According to the count as inputs, this algorithm going to decide the signal timing intervals to get higher time limits for that particular signal to avoid the traffic queuing in a dense number of vehicles.

3. User Communication: From the user's communication, they can able to view required information about traffic condition in that particular area. If the user decided to travel in the

particular area, then they can use current scenario of traffic conditions by this android application

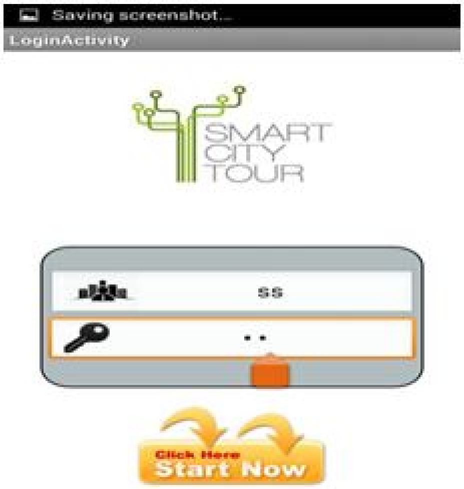


Fig. 4: TMS mobileapplication

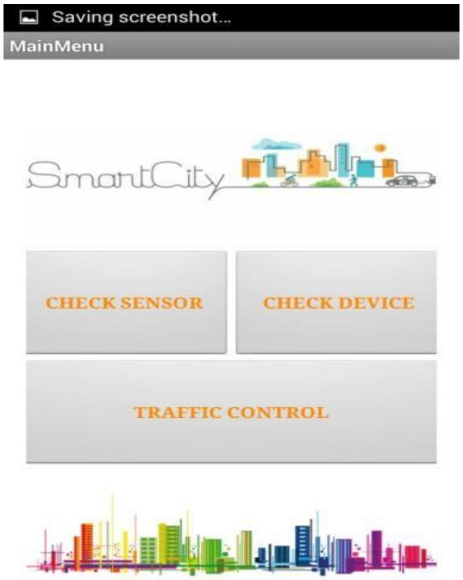


Fig. 5: TMS mobileapplication

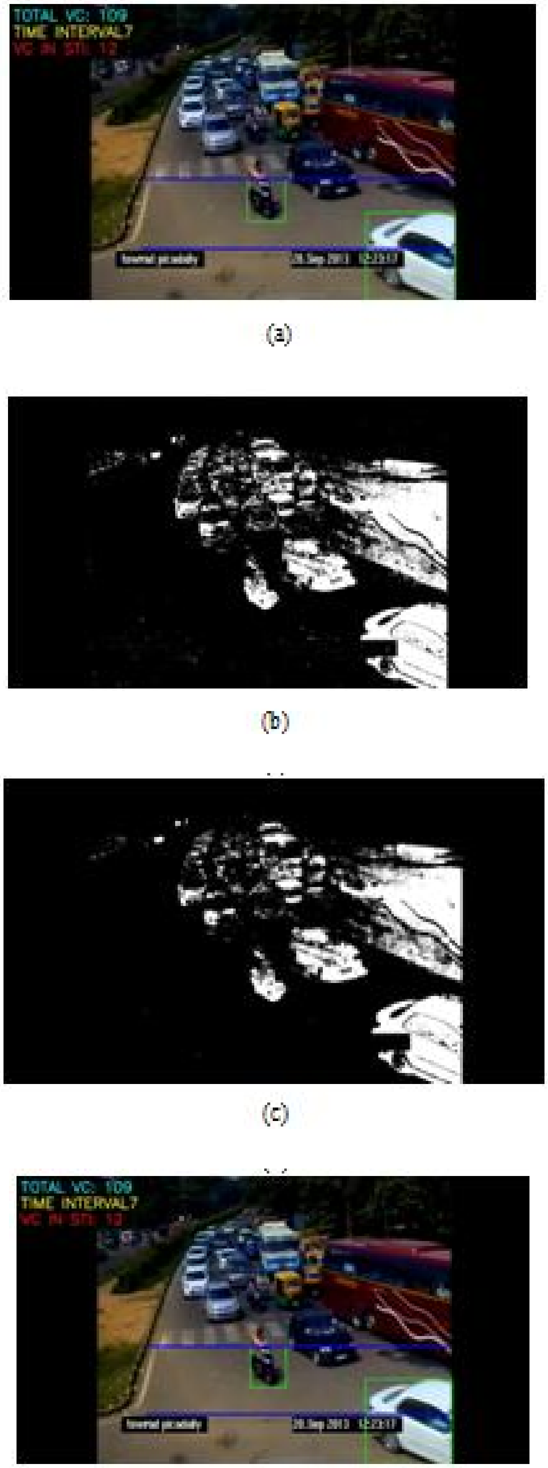
Android Mobile application for Traffic Monitoring System –TMS which is using multiple IRs for IOT model. To check IRs sensors status particular pages used from the admin side, Client Users can able to see traffic flow and timing for that particular road lane.

IV. RESULT

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 glimpse of the vehicular traffic in two urban Indian cities. It is clear from above figure, that vehicle detection and counting is challenging task in these type of settings, because of presence of large number of different type of vehicles along with absence of 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 backgroundsubtractionthroughOpenCVinfigure4(b).The noise from the foreground image 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 vehicle 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 placeat



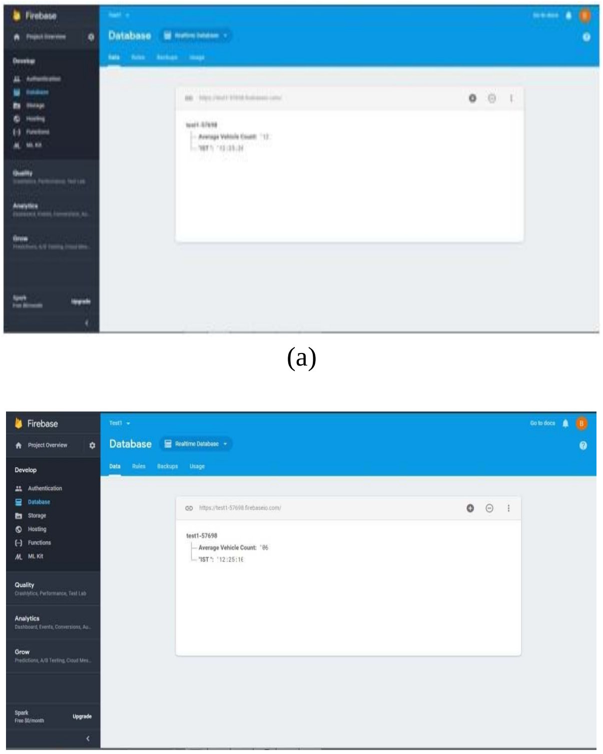
(a)

(b)

(c)

(d)

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



test1-57698
Average Vehicle Count: "12
"IST ": "12:25:31

(a)

test1-57698
Average Vehicle Count: "06
"IST ": "12:25:16

(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, it seems to be a cost effective and portable system to help in decreasing the traffic congestion on intersections at peek points. TMS timing has been developed by using multiple features of hardware components in IOT. Traffic optimization is achieved using IOT platform for efficient utilizing allocating varying time to all traffic signal according to available vehicles count in road path. TMS will help client to know timing arability and traffic flow count in any area of their nearby locality of any regions. 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. A model ambulance can able to communicate with all base station to get an easy free lane to rush up reaching the hospital on time for needy people.

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] Li Z, Shahidehpour M, Bahramirad S & Khodaei A, “Optimizing traffic signal settings in smart cities”, *IEEE Transactions on Smart Grid*, Vol.8, No.5, (2017),pp.2382-2393.

[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”, Mach2016.