Automated Electronic Toll System using License Plate Detection

Submitted in partial fulfillment of the requirements

Of the Degree of

Bachelors in Engineering

By

Aruna Shaju K	D19B	29
Yash Marathe	D19B	36
Nikhil Patil	D19B	46
Riya Thomas	D19B	67

Supervisor:

Prof. Dr. Nadir Charniya

V.E.S.
Since 1962

Electronics and Telecommunication

Vivekanand Education Society's Institute of Technology

Mumbai University

2019-2020

Project Approval for B.E

Project entitled Automated Electronic Toll System using License Plate Detection is approved for the degree of Bachelor of Engineering.

Examiners

1. ____________________

2. ____________________

Supervisor

____

Date:

Place:

Declaration

We declare that this written submission represents our ideas in our words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misinterpreted or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

____

(Aruna Shaju K)

____

(Yash Marathe)

____

(Nikhil Patil)

____

(Riya Thomas)

Date:

Acknowledgment

We would like to take this opportunity to express our sincere thanks with deep sense of gratitude to our project guide Prof. **Dr. Nadir N. Charniya**, for his valuable support, co-operation, and guidance which helped us in the time of evolution of this report.

We sincerely thank our H.O.D Prof. **Mrs. Shoba Krishnan** and all the department staff for giving us the valuable suggestion and appraisal of my report.

We would also like to convey our gratitude to principal **Dr. Mrs. J.M. Nair** and the teaching and non-teaching staff of **Vivekanand Education Society's Institute of Technology** for permitting to carry out this work in this prestigious institute. The library and lab facilities were of enormous value to this work.

We gratefully acknowledge the support given by **India Innovation Council Design Contest 2019** and **Texas Instruments** for providing the components for the project work.

We would like to express our heartfelt thanks to our loving parents for their blessings and all other concerned individuals who have directly or indirectly extended their valuable support and contribution for completion of our work.

Aruna Shaju K

Yash Marathe

Nikhil Patil

Riya Thomas

Table of Contents

Abstract

Majority of the unnecessary traffic that exists in India is due to the improper management of toll systems. Manual collection of toll is practiced in over 80% of the existing systems. This has proved to delay and crowding near the toll systems majorly caused due to the manual money exchange and improper lane management.

The other problems with current toll systems are that they are slow, inefficient and unregulated. In automatic systems, vehicles are scanned in a fraction of a minute and the payment is done. In manual systems, barriers have to be lifted or needs to be slid or pushed. Manual collection of toll implies that no detail is recorded into a computer and mostly written down in log sheets. Hence its retrieval is difficult.

Through the project being proposed, a complete automatic e- toll system is proposed using license plate detection and e- wallet linkage for faster and practical implementation of tolls.

List of Figures

List of Tables

Chapter 1
Introduction

With the growth in the number of vehicles, the need for expansive roads catering to thousands of motor vehicles transverse across India has become inevitable. The revenue required for the same is collected mainly through toll booths. However, considering the present situation the modern toll system has some limitations due to the manual collection. This has proved to delay and crowding near the toll systems majorly caused due to the manual money exchange and improper lane management. The implementation of 'FASTag' has hardly proved to reduce this traffic and hence it is required to automate the system using the License Plate of the vehicle which is an existing and unique feature of any vehicle.

1.1 Motivation:

Collection of toll has become burdensome for commuters and seldom travelers. The digitizing of wallets has led to people carrying less cash and paying in cash at an unexpected toll becomes tiring. The manual collection of toll even leads to traffic and unnecessary delay at these places. Implementation of RFID based tags 'FASTag' have had its drawbacks like non tag vehicles entering the tag reserved lanes and the tag itself not working which makes the situation worse.

With the growing number of drivers and an increase of vehicles on the road come problems associated with traffic. Some of these problems, such as accurate bridge and highway tolling, parking lot management, and speed prevention, can now be solved using machine learning. This project will explore the use of deep learning for the purpose of vehicle tracking and license plate recognition.

Fig. 1.1 Current scenario at toll centers

1.2 Proposed Solution:

Complete removal of toll centre structures has been proposed. Instead of the toll centers, cameras would be put to capture the license plate of vehicles passing by. The license plate would be detected and matched with the vehicle registration details. The user has to install an android application in the device to be able to pay the toll amount from e-wallet. One has to add information of the vehicle i.e. license number and vehicle name. The e-wallet will be linked to the bank account from where one can add money. If the vehicle is registered with the e-toll app, the amount would be automatically deducted from the account. If a vehicle is not registered, notification would be sent to the vehicle owner via mail to pay the toll within a particular amount of time, failing to do so, would be considered a crime.

If a forged or blacklisted license plate is detected by the system, an alert signal will be sent to the concerned authorities. By removing toll centers and promoting payment through application, the delay and disturbance created due to manual collection would be removed.

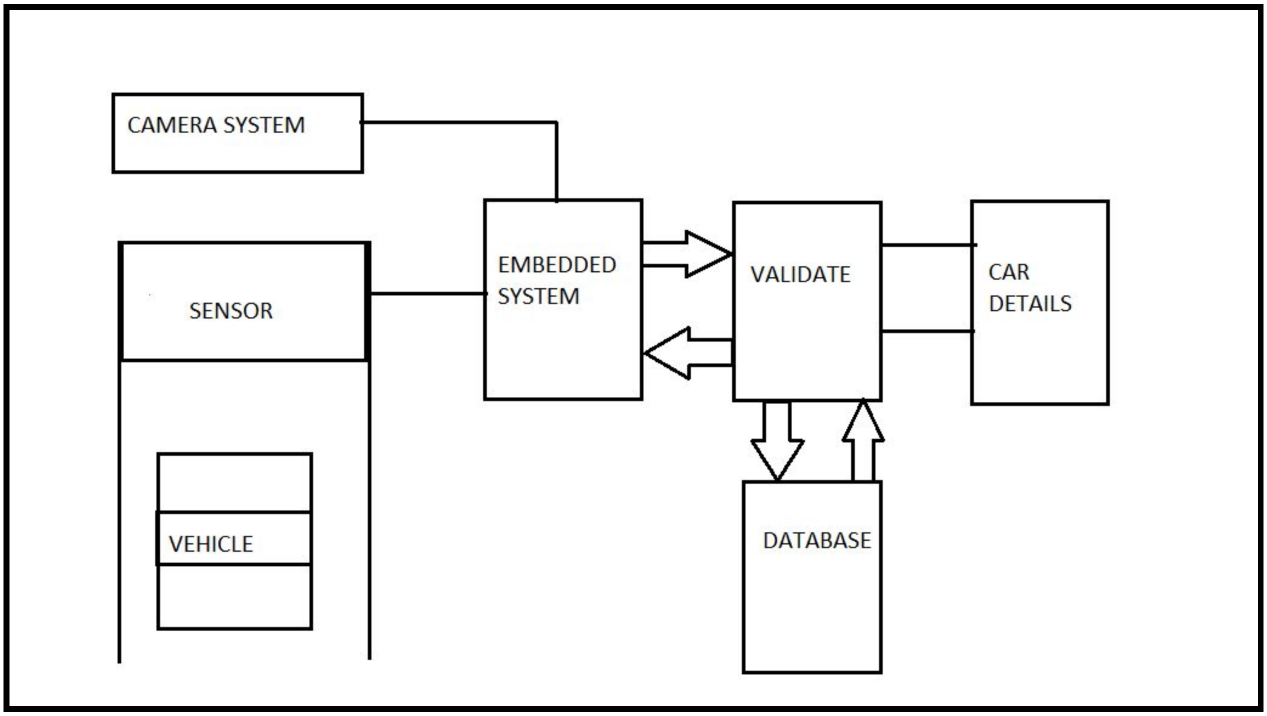


Fig. 1.2. Block Diagram of Proposed System

1.3 Relevance of the Project:

It has been observed that manual collection of tolls is practically inefficient and tedious. Automating the system using RFID tags has proved to be a failure due to the mentioned issues. Thus, instead of introducing a new system like RFID tags, using the already existing license plate and their database is a more practical solution. In Automated Toll Systems using RFID tags, many users have to wait for a long time at the toll due to certain technical failures which leads to chaotic situations. In our solution, due to the data synchronization used in real time cloud storage, every update is received by the devices/clients and at the ANPR System in no time, so the goal of low-latency is achieved and the transaction happens in seconds, so the user does not have to wait longer. Due to automation of the system, electronic record of every vehicle passing through the toll center is made available. Hence any stolen or blacklisted vehicle can be easily traceable using this system. As the system uses e-wallet application, it removes the requirement of cash to be paid at toll centers thus encouraging a cashless economy. This system truly proves to be a user friendly system.

1.4 Report Organization:

The report is organized as follows: Chapter 2 provides details of the recent work done in the field of License Plate Recognition, Android Application and E-wallet Implementation using various methods. Chapter 3 provides the proposed system of our project. Chapter 4 describes in detail the methodology of each step mentioned in the proposed system. Chapter 5 explains about the System design using microcontrollers and sensors. Chapter 6 includes the results of the work done, its advantages, disadvantages, applications and the overall system accuracy. Chapter 7 draws the conclusion in this work and future scope of the work.

Chapter 2

Literature Survey

The work developed is an image processing based work for Automatic Recognition of Number plate from the image of the vehicle taken into consideration which allows one to recognize the unique number plate of the vehicle for an intelligent traffic or vehicle management system. With the rapid development of highways and the wide use of vehicles, people have started to pay more and more attention to the advanced, efficient and accurate Intelligent Transportation Systems (ITSs). The Automated Number Plate Recognition (ANPR) task is quite challenging from vehicle images due to the viewpoint changes, when vehicle bodies and license plates have similar color, multi-style plate formats, and the non uniform outdoor illumination conditions during image acquisition.

The different systems proposed in License Plate detection are based on different properties. Some techniques use simple rules based on deterministic methods, detection contours, morphological operations and statistical analyses, while others opt for learning and classification systems. Belongie et al. used shape descriptors, called "context shapes", who described the distribution of the rest of the forms relative to a given point on the contour. Seeking the correspondence between two forms was then equivalent to finding the point on the other form that had a "shape context" similar to each point on a shape.

Parallel to this, Carmichael et al. (2002) showed the variation of the context shapes in order to differentiate between the form and the content. Another approach utilized morphological operations on gray-scale images. On the other hand, H Bai and M Sun combined a contour detector with morphological operations to search the rectangles that were considered a candidate License Plate. Moreover, Kim et al. (2005) proposed a method based on the extraction of the contour to localize a License Plate on images taken in low light conditions. Gao and Zhou also put forward a method for the detection of the License Plate of a vehicle image with a complex background. They used the histogram equalization to find the threshold in order to improve the quality of the image that contained the License Plate.

The ANPR is used widely for detecting speeding cars, security control in restricted areas, unattended parking zones, traffic law enforcement, and electronic toll collection, etc. The major steps to accomplish the proposed work can be given as

1. Image acquisition
2. Number plate recognition
3. Edge Detection
4. Character Segmentation
5. Character Recognition
6. Integration of the above algorithm with database and application

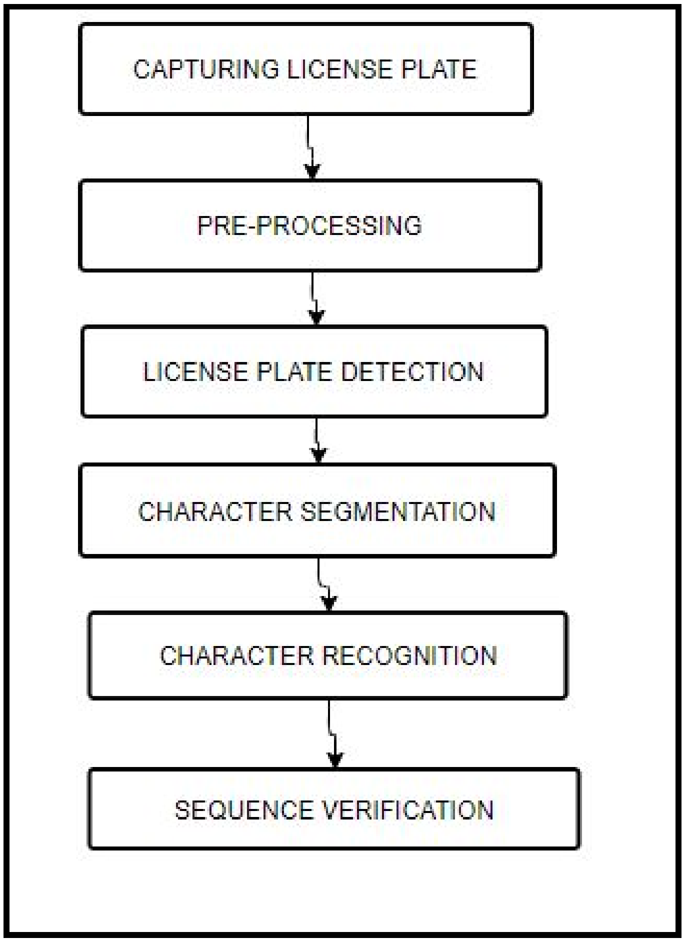


Fig. 2.1 Flowchart of LPR algorithm

The solution is proposed to deal with the problem in toll collection. Human intervention makes the system very much prone to mistakes and inefficient. That's why we propose a system which will automatically capture the vehicle’s image and also pre process it by removing the effect of noise and blur with the help of image pre-processing activities. This pre-processed image which is the partial output is given to the next process i.e. for finding the region of interest which is our number plate. To do that, we are using the Contour tracing algorithm. This process will detect the number plate from the image taken.

This partially processed image which is the output of the previous step will be given to the next step which is sharpening of the edges of the number plate. This sharpening is done with the help of Canny Edge detection algorithm which along with sharpening of edges also enhances the quality of image using Gaussian kernel as its filtering part. The sharpened image is then inputted to a segmentation algorithm which separates out individual characters from the output of the previous step. From that we will find out individual characters that are present in the number plate and we will match them individually with the help of Tesseract which are already trained Optical Character Recognition (OCR) models.

The final output of the system will be the string form of the characters of the license plate of vehicles. This output is then matched with the database to deduct the toll amount directly from the e-wallet without any further delay. The integration of an android application with the system is a challenging one. Also, other advantages like database of passed vehicles in electronic form, list of blacklisted vehicles, etc can be easily retrieved using this system.

Optical character recognition or optical character reader (OCR) is the electronic or mechanical conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo (for example the text on signs and billboards in a landscape photo) or from subtitle text superimposed on an image.

The advantages of the proposed system include online access to information such as the total number of vehicles currently present at the toll center, the number of authorized vehicles, etc. Moreover, too much information of vehicles traffic statistics can be extracted from the given

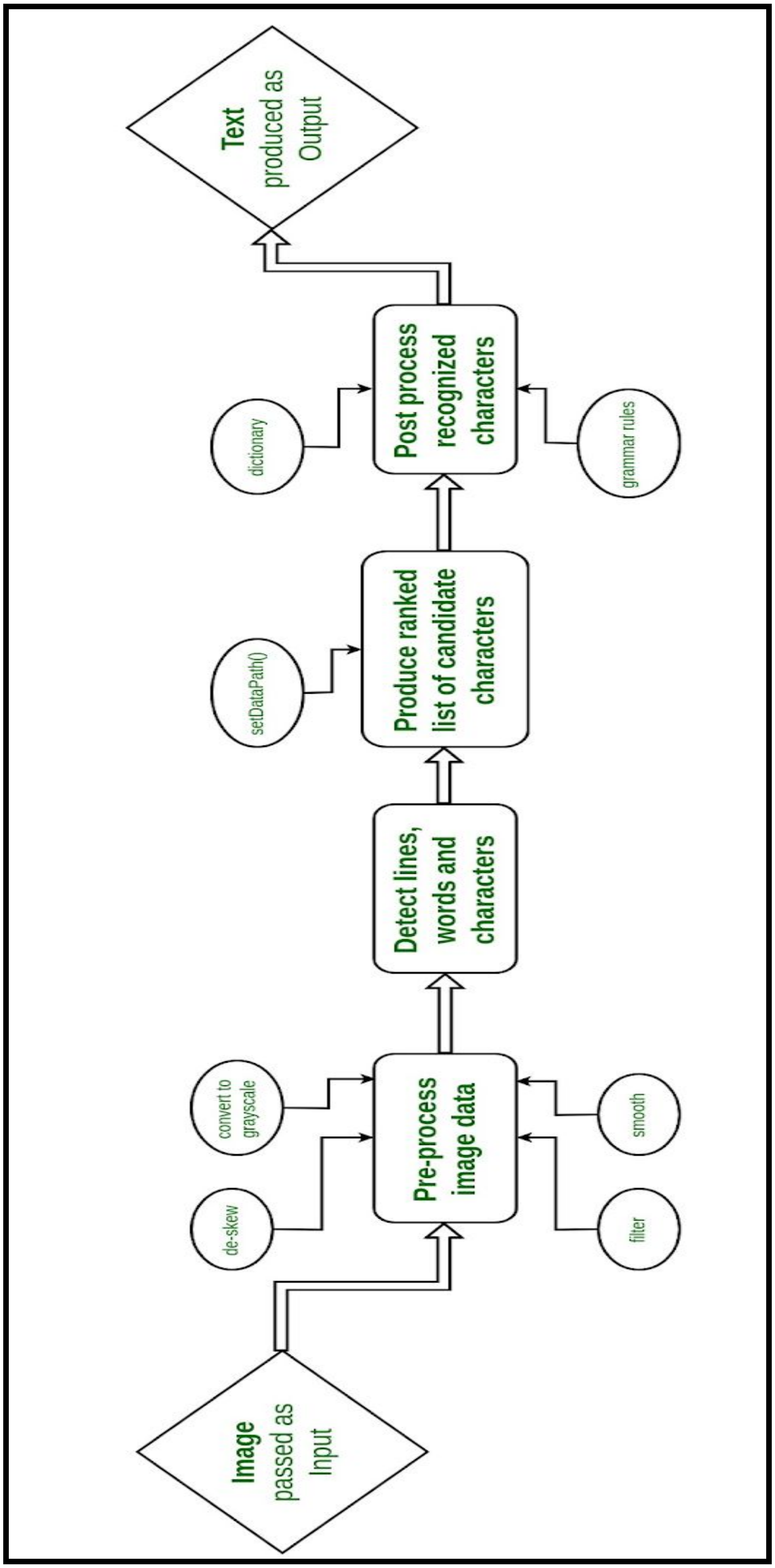


Fig. 2.2. Basic flowchart of OCR system [14]

system. The cameras used in the system can be deployed under all weather conditions and are equipped with powerful infrared radiation units for identifying vehicle license plates in absolute darkness.

Moreover, these cameras are equipped with thermometers which detect ambient temperature and activate the cooling system of the camera at very high temperatures so as to provide high-quality imaging in the -50 °C to +70 °C temperature range. The camera case is built in accordance with the IP66 standard and is fully water and dust proof. The lenses used in the camera provide high-quality image processing capabilities.

After the recognition of License Number from the Number Plate, it is integrated with the Firebase and hence with the application through which the deduction of toll amount is carried out. The complete procedure of automating the toll gates creates data at runtime and this dynamic data needs to be stored. So, for creating a dynamic database, Firebase is used. Cloud framework used is shown in the figure given below.

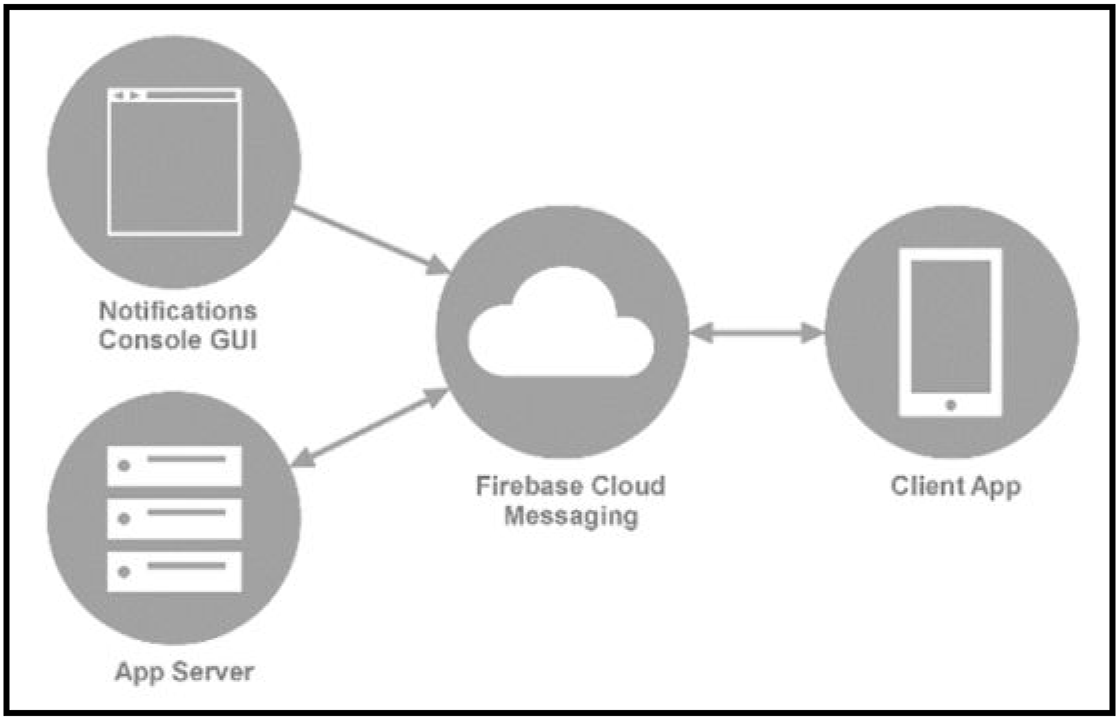


Fig 2.3 Firebase and Application Connectivity [1]

Firebase is Google provided API to create a database and fetch from it in real time with only a few lines of code. Data is stored as JSON and is accessible from all the platforms. It can be synced into android, IOS or web applications. Its back end service provides application developers an API that allows data to be synchronized across clients and stored in Firebase's cloud. The company provides client libraries that enable integration with Android, iOS, JavaScript, Java, Objective-C, Swift and Node.js applications. The database is also accessible through a REST API and bindings for several JavaScript frameworks such as AngularJS, React, Ember.js and Backbone.js. The REST API uses the Server-Sent Events protocol, which is an API for creating HTTP connections for receiving push notifications from a server. Developers using the real time database can secure their data by using the company's server-side-enforced security rules. Android Studio was used for Application development. Android Studio is the official IDE for android application development. It works based on IntelliJ IDEA.

Chapter 3

Proposed Automated E-Toll System

The proposed system majorly consists of License Plate Recognition and the Android Application. It is designed such that the response time of the system is as little as possible. By using other techniques like multithreading, the response time can be further decreased. The implementation of e-wallet helps to integrate with Paytm, PayPal and Internet Banking. Hybrid lanes, which are yet to go cashless after the latest extended relaxation of FASTag for a month, are experiencing long queues resulting in a massive traffic jam at the various toll plazas.

The proposed system contains majorly of 5 steps:

1. Capturing of License Plate
2. Detection and Extraction of License Plate
3. Transfer of extracted License Number to Database
4. Implementation of E-wallet application
5. Linking the database with E-wallet system

As a vehicle approaches the toll structure, the sensor senses it accordingly and signals the microcontroller connected to it. Following this, the microcontroller triggers the camera to capture a video of two seconds. The position of the sensors will be decided after analysis of the time taken for the above step and accordingly configure it to enable the vehicle to be at optimum position when the camera takes the video. The captured video is sent to the microcontroller where processing, comparison and elimination algorithms are applied to each frame to finalize the most probable license plate boundary. The image pertaining to the boundary detected is cropped from actual frames and validated against standards. On failing any standard, the succeeding frame in the video is processed else it comes out of the loop. This is followed by sending the number obtained to the database through which toll is automatically deducted, or a notification is sent to registered email id or phone number according to whether the vehicle owner is a registered user or not. The frame at which the vehicle was confirmed is also updated in the vehicle user's application as proof.

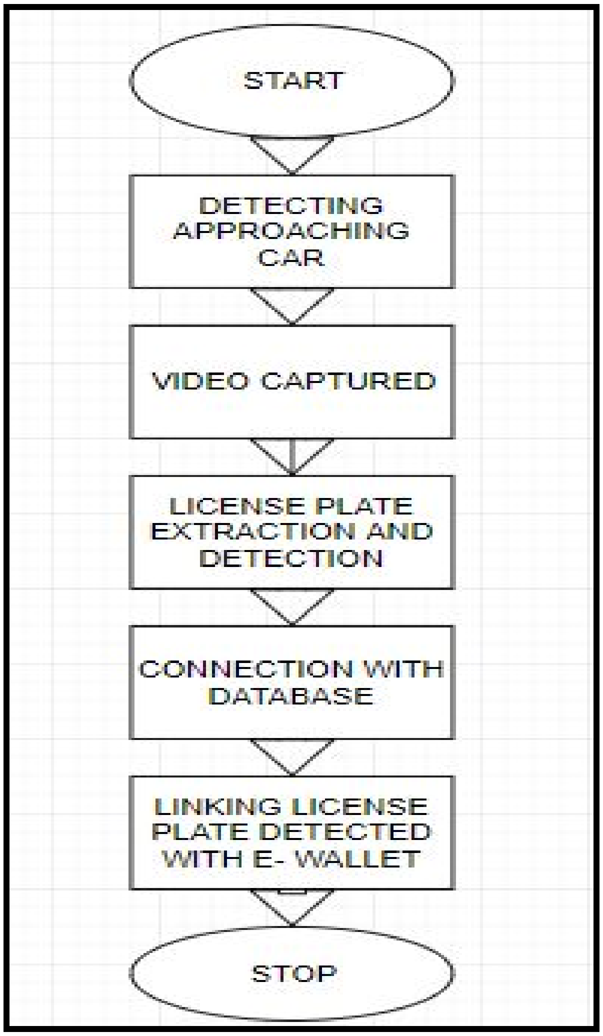

START
DETECTING APPROACHING CAR
VIDEO CAPTURED
LICENSE PLATE EXTRACTION AND DETECTION
CONNECTION WITH DATABASE
LINKING LICENSE PLATE DETECTED WITH E- WALLET
STOP

Fig. 3.1. Flowchart of Proposed System

3.1 Capturing Of License Plate:

A threshold sensor (working on pressure or light sensitivity) is installed for detecting an approaching car. The example for a pressure sensor can be a load cell. It is a type of transducer, specifically a force transducer. It converts a force such as tension, compression, pressure, or torque into an electrical signal that can be measured and standardized. As the force applied to the load cell increases, the electrical signal changes proportionally. A light sensor would simply measure the intensity of visible light and alert when a specific threshold has reached. When a vehicle is over the sensor, the light intensity falling on the sensor is very less.

As the vehicle is detected by the sensor, the camera would be triggered and would be recording for a few seconds. This captured video is sent to the license plate processing unit for license plate detection and extraction. Number plate recognition starts with the acquisition of images from an image source, desirably from a surveillance camera. The image acquisition technique determines the captured image quality of the number plate with which the detection algorithm has to work. Better the quality of the acquired images, the higher the accuracy.

3.2 Detection and Extraction of License Plate:

The video captured is processed in this step. The video is converted to frames. Each frame is checked for accuracy and on obtaining the proper plate, the loop is exited. For detection, each frame undergoes preprocessing. This is followed by contour analysis to fit contours for a license plate. The detected plate region is then cropped. The cropped image is sent to the Tesseract OCR module for extraction of license plate. This step accounts for the majority of the processing time in the system. The conversion of videos to frames takes time depending on the frames per second of the camera used. So, a two second video on an average generates 80-90 frames. Thus, the processing of 80-90 frames takes ample amount of time. In order to reduce the processing time, the frames are processed from backwards hence increasing the possibility of getting the correct number plate with the standard aspect ratio.

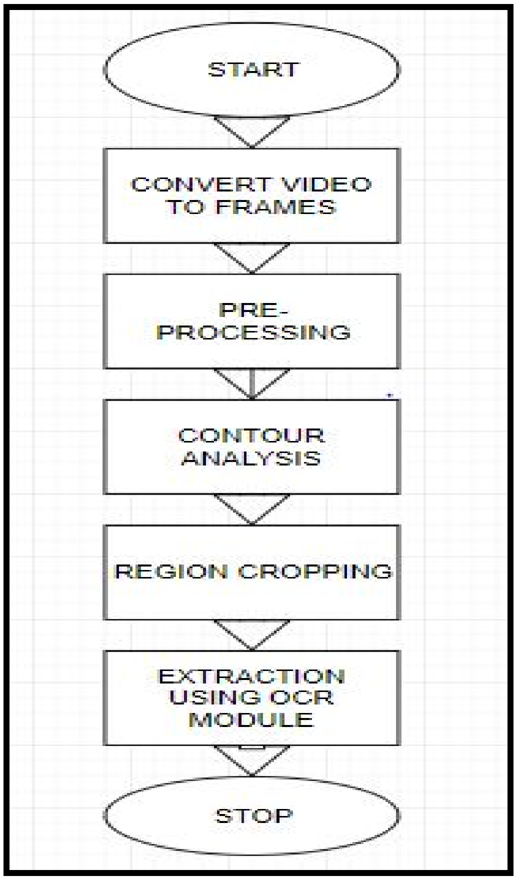
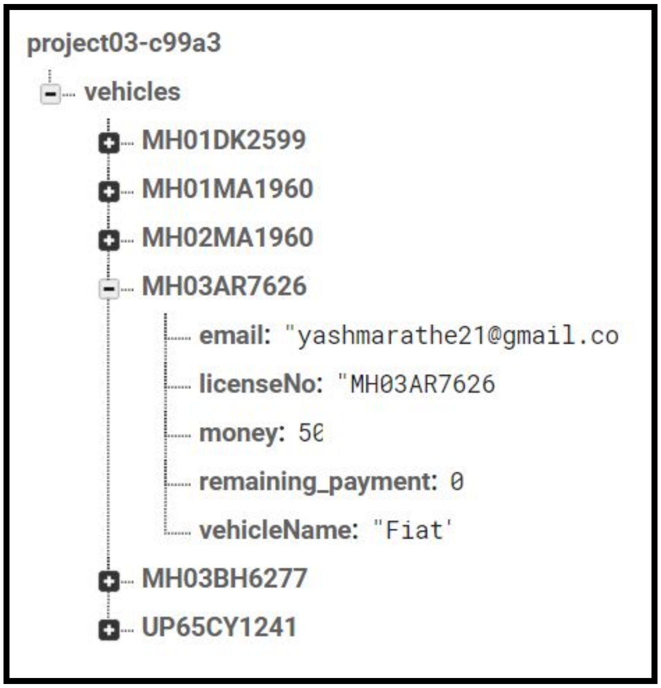


Fig. 3.2. Flowchart of license plate extraction

To improve efficiency, changes are made in the obtained license plate number to remove the need of processing another frame taking into account the Indian License Plate structure. The number obtained is checked against standards such as length, character and number positions etc. The standard Indian License plate has an aspect ratio of 4.727 which helps in identifying the number plate. The light intensity falling on the number plate during capture can also vary the processing time required in the recognition of Number Plate. The consecutive frame in the loop is processed if it does not fit the standards.

3.3 Transfer of Extracted License Plate to Database

The extracted license plate is uploaded to the Real-time Database. The ANPR System is linked with a Firebase Real time Database. The reason why Firebase Database has been chosen is because it is supported by various languages and platforms. It can be used for building a back-end for various platforms like android, iOS, Web, etc. Firebase is a Cloud hosted database, i.e. it runs on a cloud and access to the user is provided as a service. It stores data in JSON (JavaScript Object Notation) format, a format to store or transport data. Due to the data synchronization used in real time, every update is received by the devices/clients in no time. As the database can be accessed directly from the mobile device or any system, there is no need for an application server. The structure of Firebase used in the system is shown in the figure below.



```
project03-c99a3
  vehicles
      MH01DK2599
      MH01MA1960
      MH02MA1960
      MH03AR7626
          email: "yashmarathe21@gmail.co
          licenseNo: "MH03AR7626
          money: 5
          remaining_payment: 0
          vehicleName: "Fiat'
      MH03BH6277
      UP65CY1241
```

Fig 3.3 Structure of Firebase

From Fig 3.3 we find that the key parameter is the License Number of the vehicle. For each License Number, a valid email ID, money in the e-wallet, the remaining payments and name of the vehicle is related. This helps in keeping track of the toll amount deducted both by the system as well as by the user. It is also possible to list the vehicles that passed a toll center in one day. Such statistics helps in determining the toll revenue also. The blacklisted or stolen vehicles can be easily identified using this list.

3.4 Implementation of E-Wallet Application

The e-wallet is implemented so that the toll amount can be directly deducted. An online wallet application allows users to make electronic commercial transactions swiftly and securely. It functions much like a physical wallet. A digital wallet has both a software and information component. The software provides security and encryption for personal information and for the actual transaction. Typically, digital wallets are stored on the client-side and are easily compatible with most e-commerce transactions. A server-side digital wallet, known as thin wallet, is the one that an organization creates and maintains on its servers. The information component is basically a database here, Firebase, of user inputted information. This information consists of email ID, remaining amount and other information.

3.5 Linking the Database with E-Wallet System

For creating an E-Wallet System, the user needs to store all the details such as name, mobile numbers, address, bank details, transaction details, etc in his profile. These features play a vital role as it offers personalized service and ease in accessing the app. The E-Wallet can also be linked with Paytm or PayPal Software Development Kits (SDKs) for ease of access to the bank accounts. Linking the E-Wallet to such mobile payments and commerce platforms will provide a better experience to the user.

The linking of the application with the Paytm SDK can be implemented using the Paytm All-in-One SDK. It provides a swift, secure and seamless payment experience to users by invoking the Paytm App (if installed on user’s smartphone) to complete payment for order. Paytm All-in-One SDK enables payment acceptance via Paytm wallet, Paytm Payments Bank, saved Debit/Credit cards, Net Banking, BHIM UPI and EMI as available in customer’s Paytm account. If Paytm App is not installed on a customer's device, the transaction will be processed via web view within the All-in-One SDK. Now, it also enables subscription payment acceptance via all the mentioned pay modes, specifically Paytm wallet, Paytm Payments Bank, saved Debit/Credit cards, Net Banking, and BHIM UPI.

There are two methods by which the application can be integrated with.

1. Integrating All-in-one SDK
2. Integrating Application Program Interface(API) for App Invoke

INTEGRATING ALL-IN-ONE SDK

This All-in-One SDK is meant for making payments seamlessly with Paytm regardless of Paytm App being installed on device or not. This SDK handles 2 flows:

A. App Invoke Flow: In case Paytm App is installed, it will be launched to complete the transaction and give the result back to merchant App.

B. Redirection Flow: In case Paytm App is not installed, All-in-One SDK will open a web view to process transactions and give results to merchants.

INTEGRATING API FOR APP INVOKE

The following steps are followed in order to integrate the API for App Invoke

1. On the mobile App, the user proceeds to checkout.
2. Application calls the ‘Initiate Transaction API’ from the backend to generate transaction tokens.

   In case of Subscription payment, Application calls ‘Initiate Subscription API’ from backend to generate transaction token.
3. Paytm shares a transaction token with the user.
4. Check if Paytm App is installed on user's phone.
5. If Paytm App is installed, invoke the App.
6. User completes payment on the Paytm App using saved credentials.
7. If Paytm App is not installed, call the ‘Show Payment Page API’ and redirect the user to Paytm’s hosted payment page.
8. Paytm processes the transaction with the user’s bank and returns the transaction response.
9. Application calls the ‘Transaction Status API’ from the backend to verify the transaction response.

   In case of Subscription payment, application calls ‘Fetch Subscription Status’ API from your backend to verify the transaction response
10. Notify the payment status to the user and proceed with the service fulfillment.

Chapter 4

Methodology for License Plate Recognition

Capturing of License Plate

In the first step for capturing the License Plate, we have considered a camera connected to a microcontroller which will record a video of 2 seconds for capturing the license plate. The first type of sensor that can be used is a pressure load sensor. The camera will be triggered by the load cell which will in turn be activated by pressure applied on it i.e. when the vehicle shall pass over it.

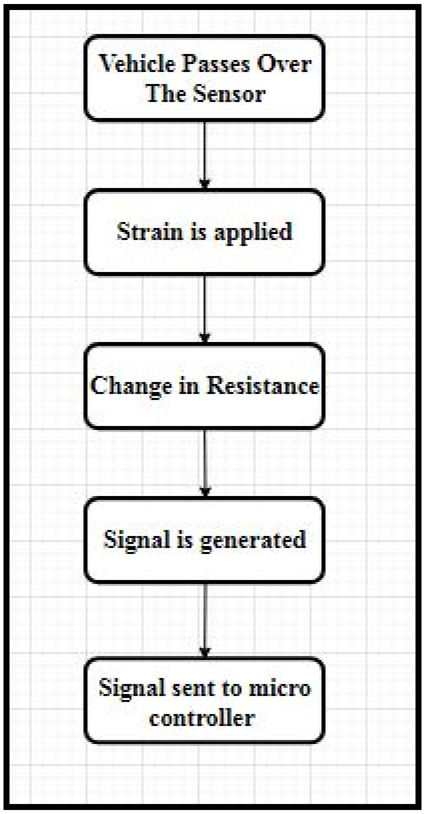

Vehicle Passes Over The Sensor
Strain is applied
Change in Resistance
Signal is generated
Signal sent to micro controller

Fig. 4.1 Flowchart of Load cell working

For detection of an approaching vehicle, a load cell (strain type) sensor is used. As the vehicle passes over the sensor, strain is applied on the load cell which creates a deformation in the spring of the sensor. This deformation creates a resistance mismatch in it. As it works on the Wheatstone bridge implementation, a corresponding signal is generated which would notify the microcontroller connected to it of the vehicle that has moved over it.

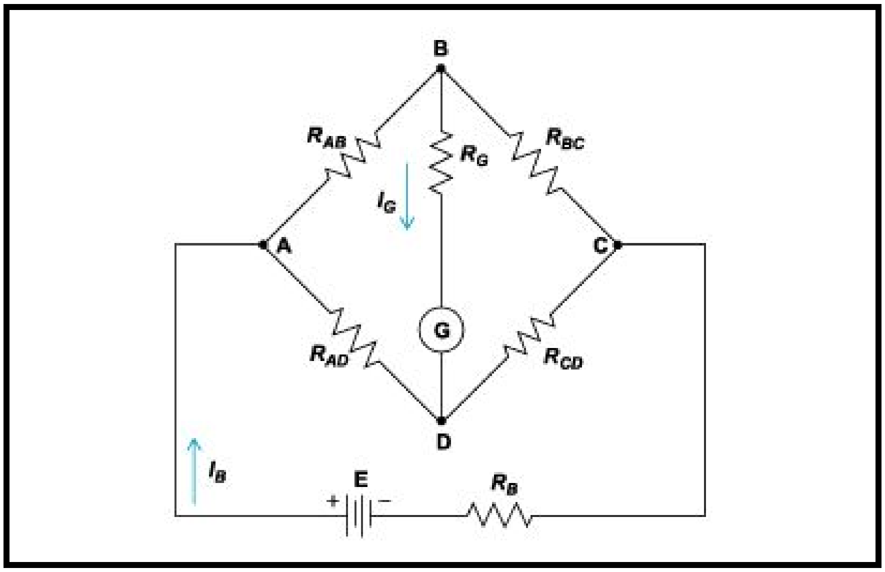


Fig. 4.2. Wheatstone bridge implementation in load cell

Referring to the above figure, when the load cell is not applied any pressure, voltage difference between points B and D is zero as the Wheatstone is balanced. Under pressure, the resistance in the four arms changes due to deformation in the cell due to strain leading to mismatch in the Wheatstone bridge. This creates a voltage difference between points B and D. This voltage appearing between B and D shall notify the microcontroller of a vehicle present above the load cell.

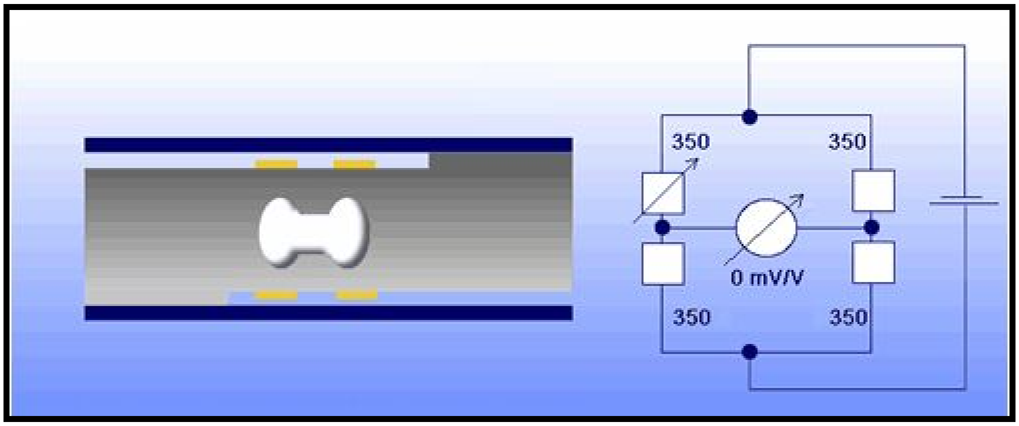


Fig. 4.3. Load cell under no stress

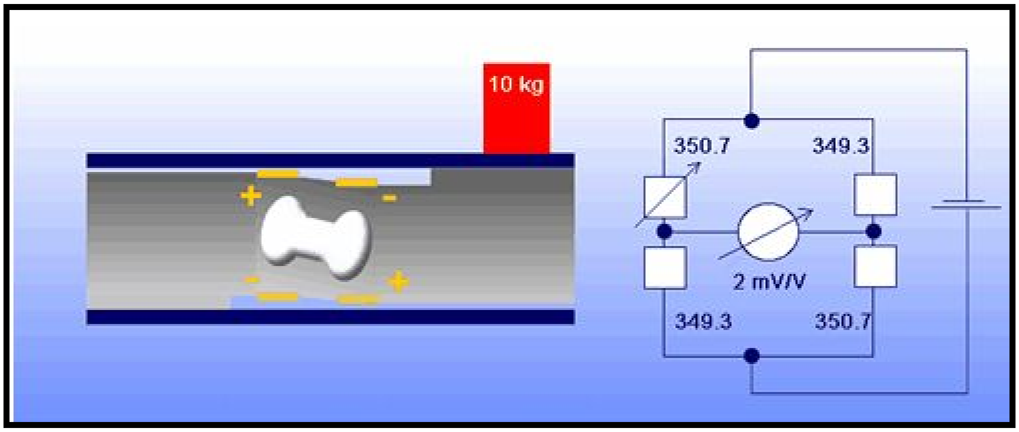


Fig. 4.4. Load cell under stress

The microcontroller system, after being signaled by the load cell triggers the camera which will record a video of 2 seconds. This video is sent to the microcontroller which converts it into frames and applies pre-processing on it before the license plate detection algorithm is applied.

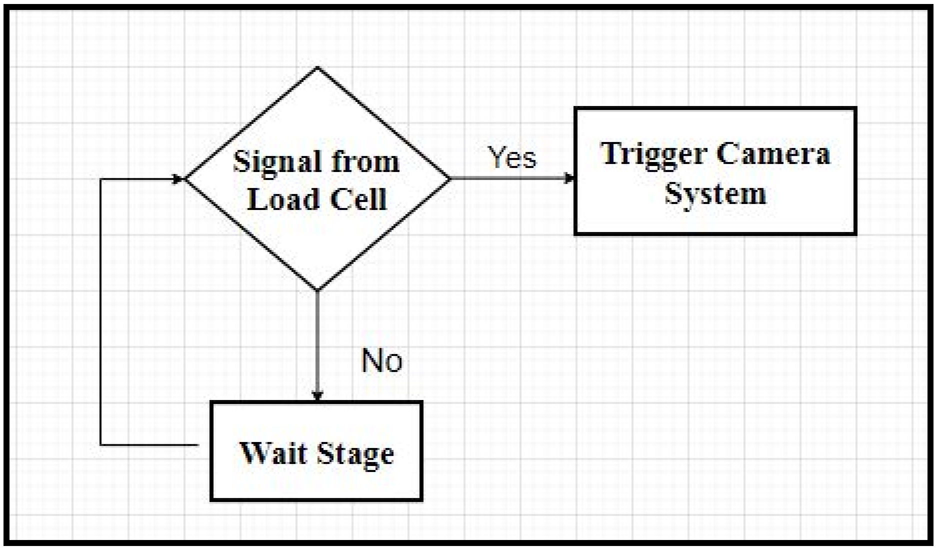


Fig. 4.5. Activation of Camera System by Load Cell

By carefully selecting a threshold value in the lower range of a vehicle i.e. around 1200kg, we can avoid any false triggers that may occur by stray animals or pedestrians on the path the load sensor is sensing. As the weight of any false positive elements is considerably lower than a vehicle, chances of the same applying pressure for minimum threshold signal is very low.

A similar system can be implemented by using an ambient light sensor. This can be used during daytime during rush hours because of its fast response output. It is a photo-detector that is used to sense the amount of ambient light present, and appropriately produce a signal that can be processed by the microcontroller to determine if an obstacle has been detected over the region.

Varying thresholds of this output may be set in the controller for different periods of the day. This improves the efficiency in implementation. The ambient light sensor shall be put at the centre of the lane at slightly below the road level. This is done to protect the sensor from excess pressure from vehicles and also remove the possibility of false positives if it was at the side. As the vehicle moves over the sensor, there is a difference in the light incident on it due to the shade provided by the vehicle. This phenomenon is used for the detection of the vehicle.

After trial runs, suitable threshold values of incident light are fed to the controller to identify the ambient light corresponding to the shade from the vehicle. In this way, an ambient light sensor can be used to identify vehicles at the toll centre and this signal is further processed in similar steps to that described for the load sensor.

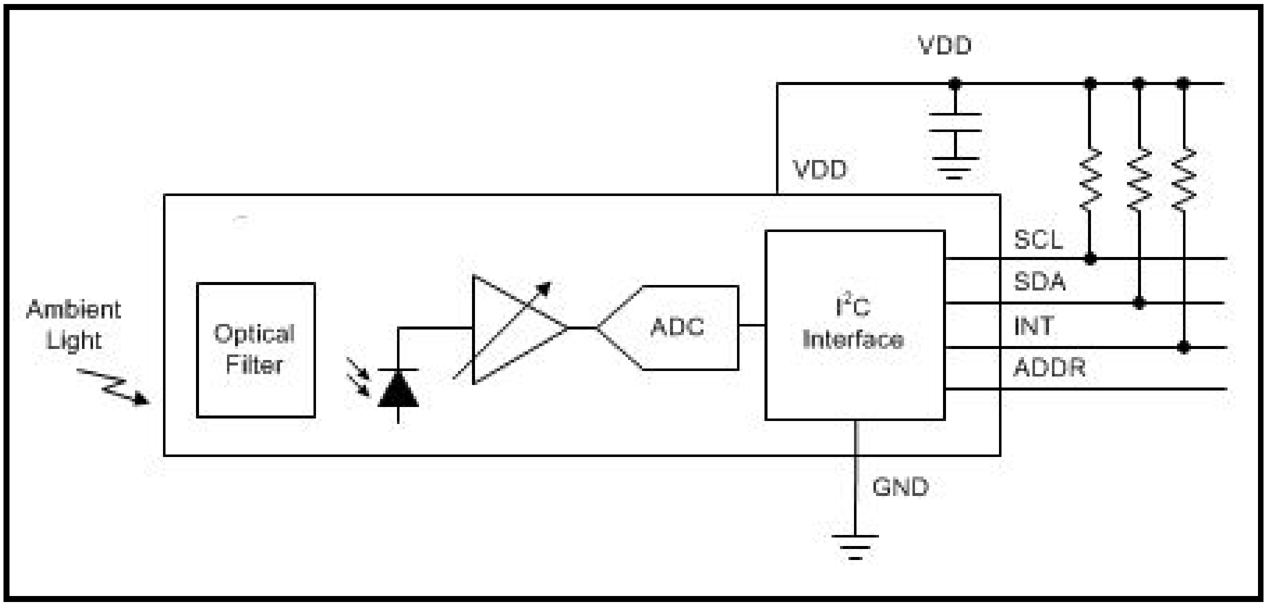


Fig. 4.6. Block Diagram of Ambient Light Sensor

The microcontroller system, after being signaled by the light sensor triggers the camera which will record a video of 2 seconds. This video is sent to the microcontroller which converts it into frames and applies pre-processing on it before the license plate detection algorithm is applied. The decision tree for the sensor is the same as shown in figure 4.5.

Detection and Extraction of License Plate

1. FRAME GENERATION:

The video generated by the camera is sent to be executed by a program that converts video into frames. Each frame is sent to the pre-processing unit after which it is run for license plate detection and extraction. Each frame generated is saved as ‘framei’ with i being the ith frame generated from the video.

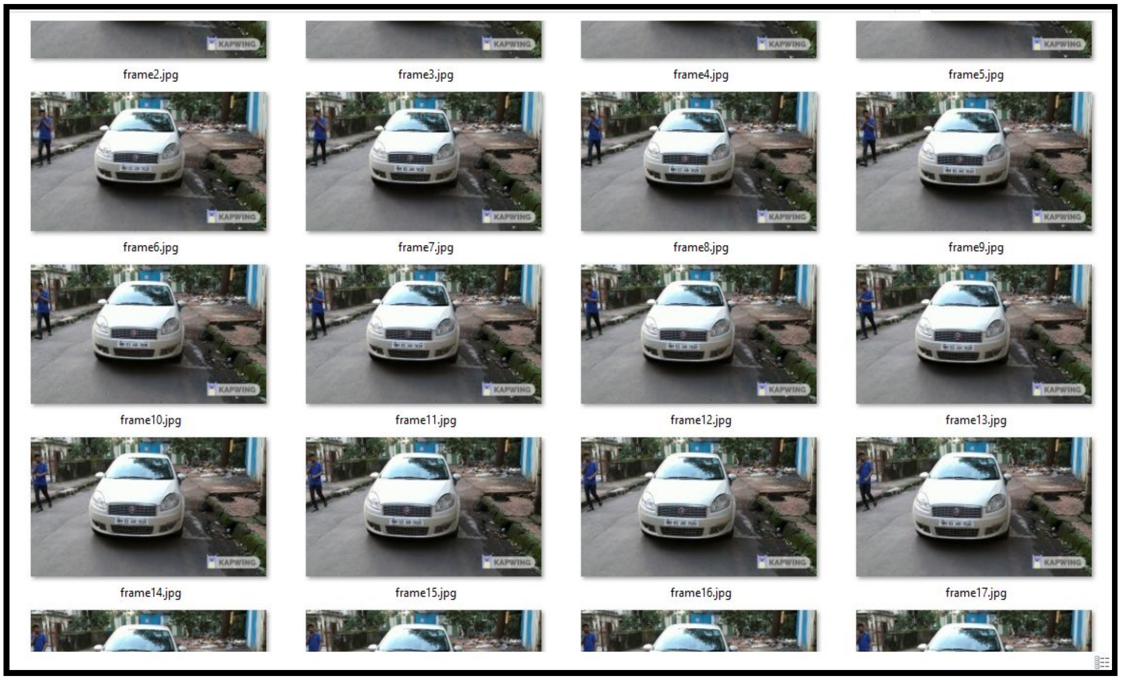


Fig 4.7 Generation of Frames

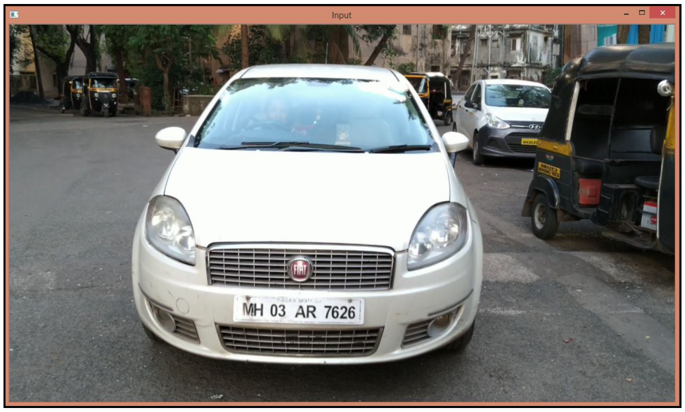


Fig 4.8 Generated Frame

2. PRE- PROCESSING:

The generated frame is first pre-processed with the following steps to attain a Thresholded image for contour analysis. The steps for pre-processing are shown in the figure below.

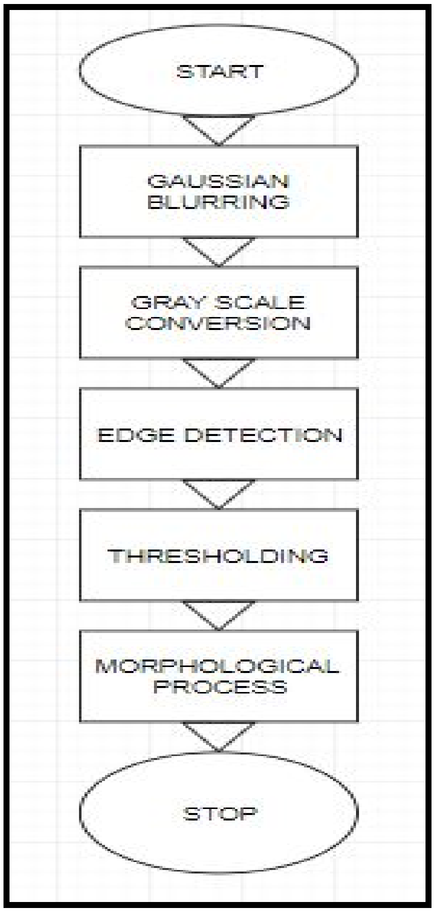


Fig. 4.9. Flowchart of Pre-processing steps

The first step involved is

a. GAUSSIAN BLURRING: Blurring is implemented using Gaussian filters to smoothen out the image obtained. This is done to remove the salt and pepper noises present which can affect the detection algorithm. The blurring step removes any unnecessary contours that may be needed to be analyzed if passed onto the next step. Specifically, a Gaussian kernel (used for Gaussian blur) is a square array of pixels where the pixel values correspond to the values of a Gaussian curve (in 2D).Each pixel in the image gets multiplied by the Gaussian kernel. This is done by placing the centre pixel of the kernel on the image pixel and multiplying the values in the original image with the pixels in the kernel that overlap. The values resulting from these multiplications are added up and that result is used for the value at the destination pixel.

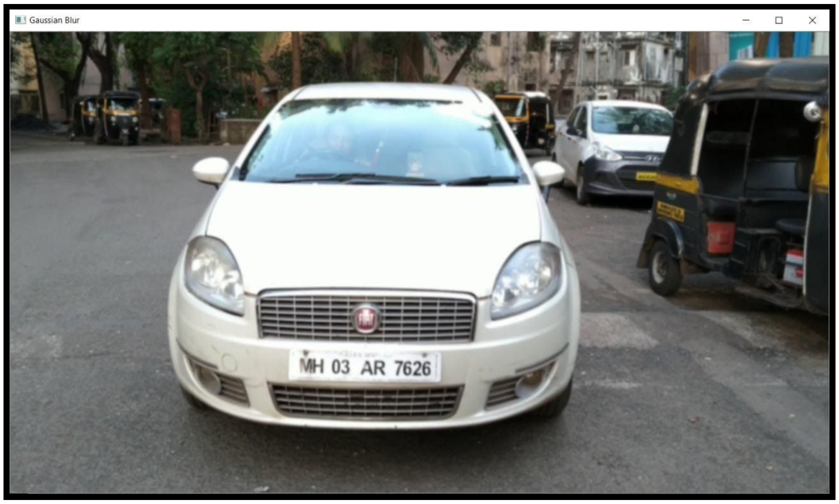


Fig 4.10 Gaussian Blurring on Generated Frame

b. GRAY SCALE CONVERSION- The edge detected image is converted to gray scale.

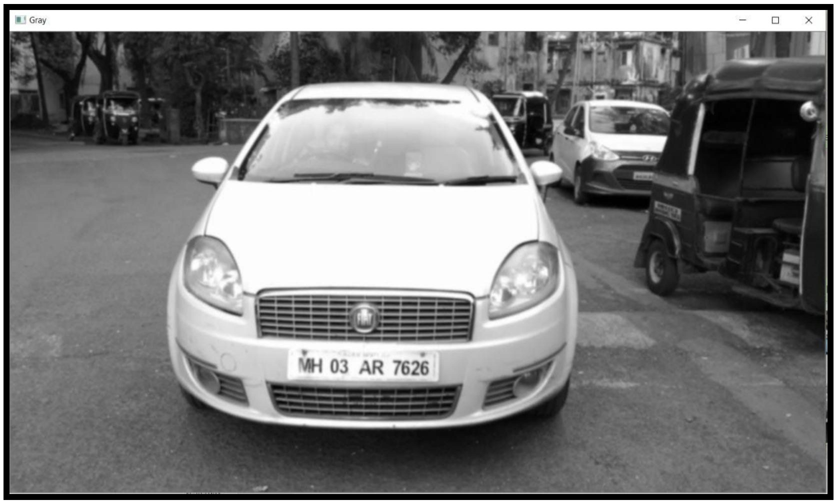


Fig 4.11 Gray Scale Converted Image

c. EDGE DETECTION- As the boundary of the license plate is a clearly defined boundary, edge detection can be used to obtain the same. Sobel mask using horizontal basis is used in the program. Edge detection works on the principles of high pass filters. The sharp transition in intensity of pixels on and around a boundary makes it to have high frequency properties which are passed by the edge detector whereas stopping the continuous intensity pixels.

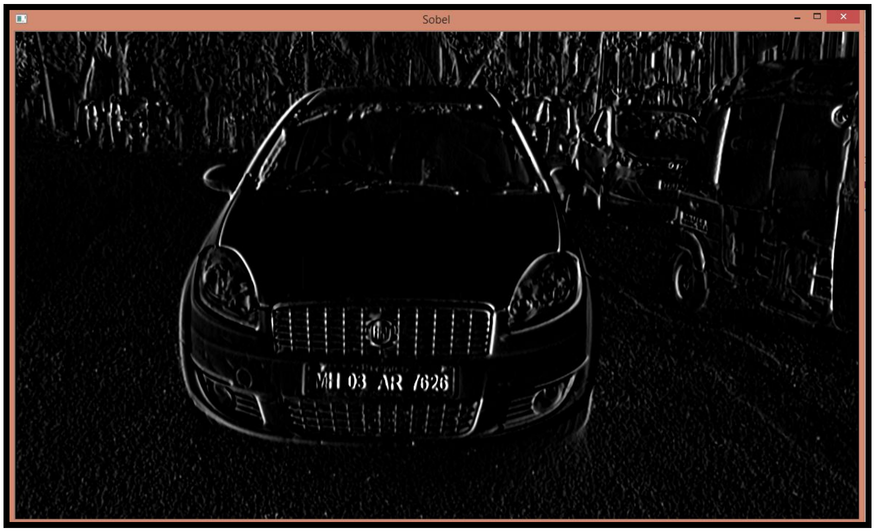


Fig 4.12 Sobel Edge Detected Output

d. THRESHOLDING- The image is Thresholded into binary format using Otsu's method of thresholding. Otsu's thresholding method involves iterating through all the possible threshold values and calculating a measure of spread for the pixel levels each side of the threshold, i.e. the pixels that either falls in foreground or background. The aim is to find the threshold value where the sum of foreground and background spreads is at its minimum. This is done by analyzing histograms for the background and foreground spread of the contending threshold values.

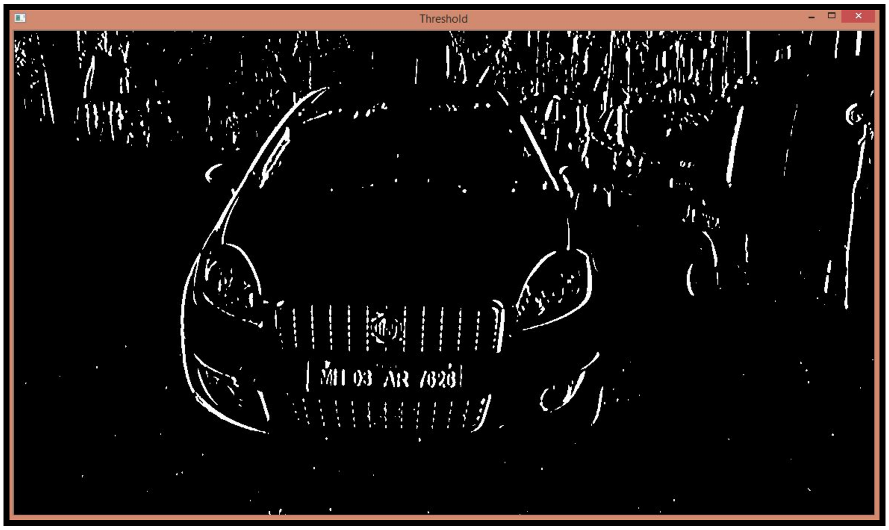


Fig 4.13 Thresholded Image

e. MORPHOLOGICAL PROCESS- Dilation is applied on the threshold image with a structuring element of size 17x3. Dilation process widens the thickness of the boundary detected, i.e. Morphological dilation makes objects more visible and fills in small holes in objects. This makes it easier for us to handle the edges detected in the image which contains the license plate boundary.

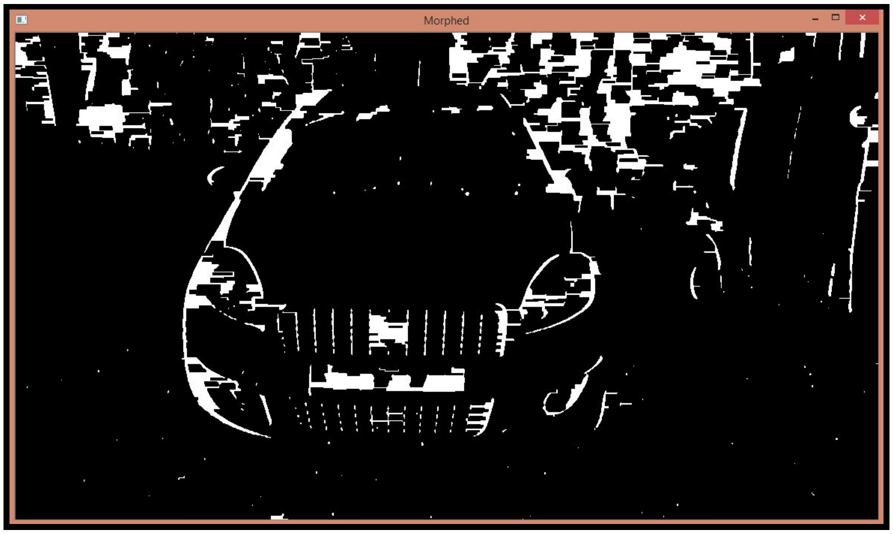


Fig 4.14 Dilated Output

3. CONTOUR ANALYSIS:

The contours are extracted in a matrix and analyzed one by one. Several steps are conducted to eliminate the non- probable license plates.

1. Checking angle parameter - The contours satisfying the angle parameters of the rectangle are considered.
2. Checking aspect ratio- The aspect ratio is found to be 4.727 which is the ratio of width to height to be the threshold for license plate. The min and max parameters are defined using this. The area of the contour is compared and determined if they satisfy the condition. The contours unsatisfying criteria are removed from the contour matrix.
3. Average Value of Contour- The average value of the pixels in the defined contour is found out. If it exceeds the value of 115, the contour is considered for the next step else, discarded.
4. The contours are analyzed again and the most probable one is selected.

4. DETECTION OF LICENSE PLATE:

The selected contour region is cropped and considered as the license plate for further analysis. The coordinates of the selected contour region is noted and a rectangle is drawn on the generated frame.

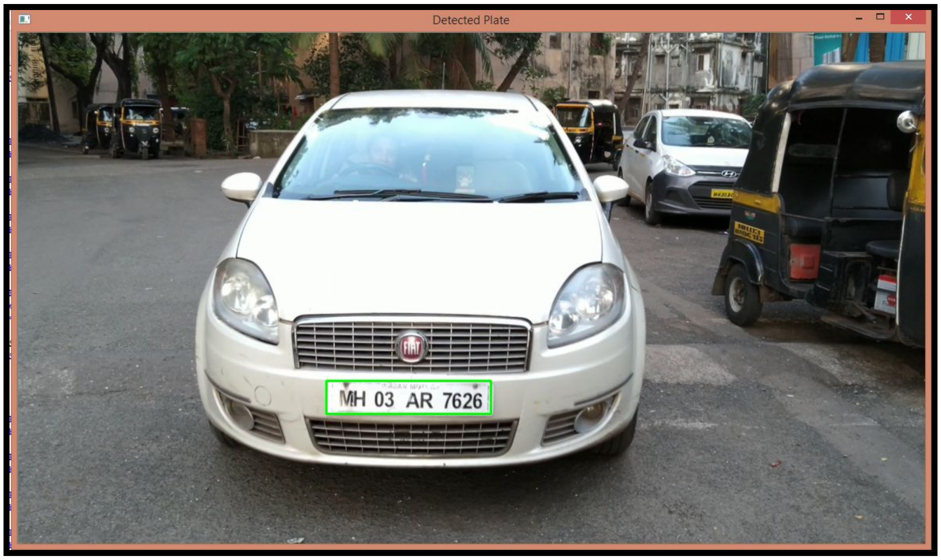


Fig 4.15 Final Contour generated

5. EXTRACTION OF LICENSE PLATE:

The license plate obtained after cropping is sent to the Tesseract OCR module for character recognition.

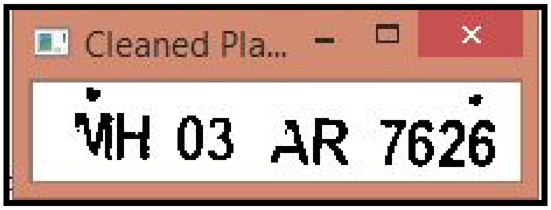


Fig 4.16 Input to OCR module

The Tesseract package contains an OCR engine - libtesseract and a command line program - Tesseract. Tesseract has Unicode (UTF-8) support, and can recognize more than 100 languages. It can be trained to recognize other languages. Tesseract supports various output formats: plain-text, html. Tesseract up to and including version 2 could only accept TIFF images of simple one column text as inputs. These early versions did not include layout analysis and so inputting multi-columned text, images, or equations produced a garbled output. Since version 3.00 Tesseract has supported output text formatting, OCR positional information and page layout analysis. Tesseract can detect whether text is monospaced or proportional. Tesseract can be trained to work in other languages too. Tesseract is suitable for use as a backend, and can be used for more complicated OCR tasks including layout analysis by using a frontend such as OCRopus. In this project, Tesseract is used as the final step for OCR after the image has been sufficiently processed so as to get optimum output.

6. VALIDITY OF LICENSE PLATE EXTRACTED:

Primarily, the length of the license plate number string obtained is checked against standards. The validity of the license plate is determined by comparing the first two characters extracted with all the RTO codes in India. To improve efficiency, changes in the recognized number are made in places where the detected character at a position is obviously wrong and accordingly rectified keeping the Indian License Plate as standard.

The changes are:

1. ‘O’ detected at positions of a number is changed to zero and vice versa.
2. ‘S’ detected at positions of a number is changed to 5 and vice versa.

The validity of the license plate is determined by comparing the first two characters extracted with all the RTO codes in India. The no. of characters extracted and their respective character and number positions are also compared with the ideal license plate. If all the conditions are satisfied, the loop is terminated and the detected plate is displayed.

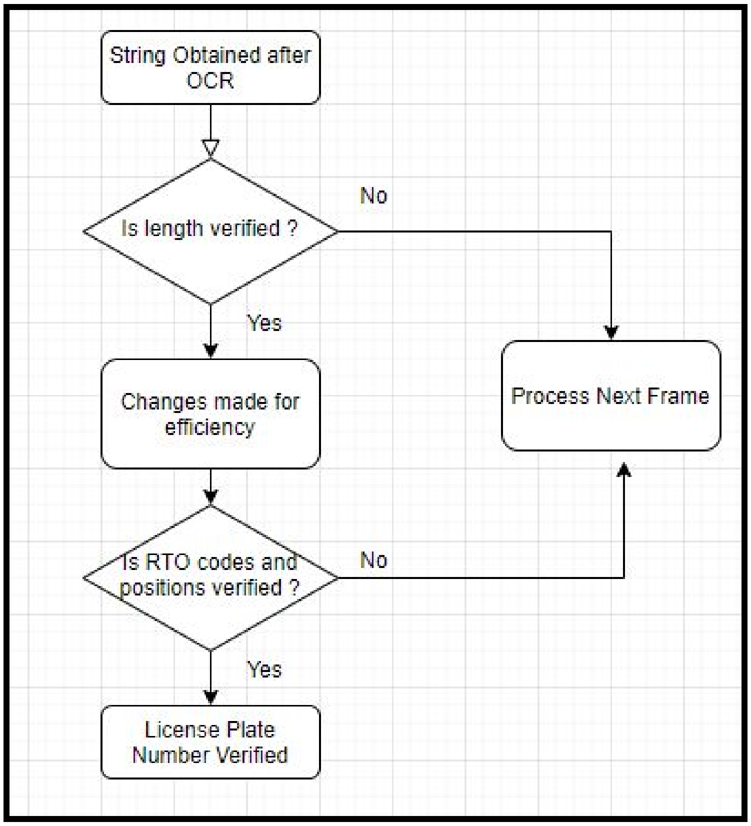
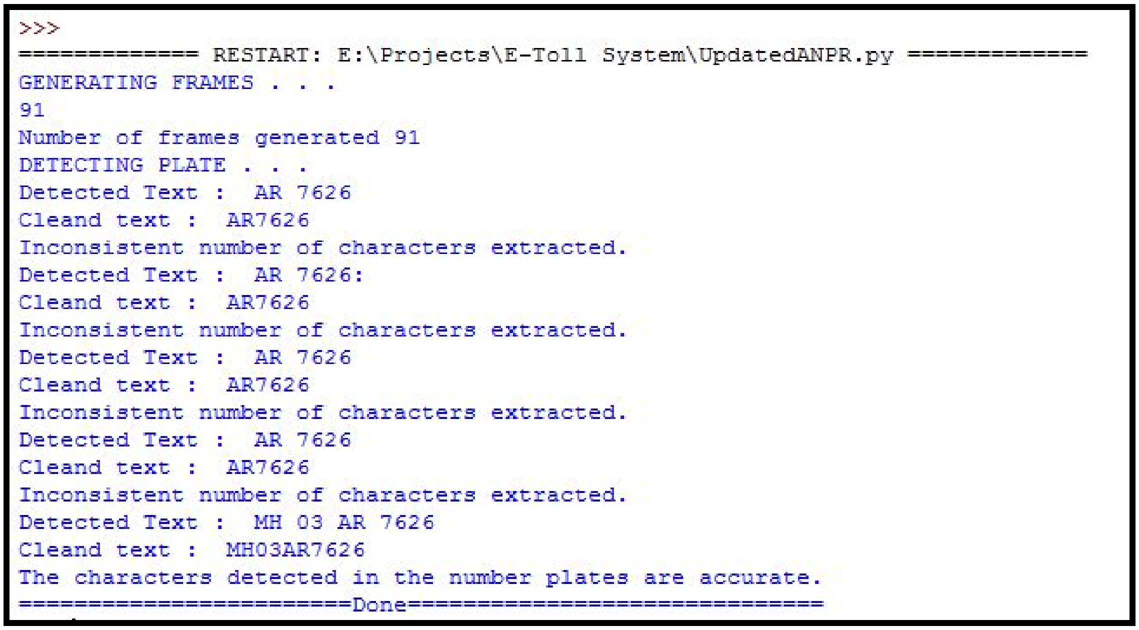

String Obtained after OCR
Is length verified ?
No
Yes
Process Next Frame
Changes made for efficiency
Is RTO codes and positions verified ?
No
Yes
License Plate Number Verified

Fig 4.17. Flowchart of Validity of License Plate Extracted



```
>>>
============= RESTART: E:\Projects\E-Toll System\UpdatedANPR.py =============
GENERATING FRAMES . . .
91
Number of frames generated 91
DETECTING PLATE . . .
Detected Text :  AR 7626
Cleand text :  AR7626
Inconsistent number of characters extracted.
Detected Text :  AR 7626:
Cleand text :  AR7626
Inconsistent number of characters extracted.
Detected Text :  AR 7626
Cleand text :  AR7626
Inconsistent number of characters extracted.
Detected Text :  AR 7626
Cleand text :  AR7626
Inconsistent number of characters extracted.
Detected Text :  MH 03 AR 7626
Cleand text :  MH03AR7626
The characters detected in the number plates are accurate.
========================Done===============================
```

Fig 4.18. Validated License Plate Number

Transfer of Extracted License Number to Database

In the 3rd step, the extracted and verified license plate number is uploaded onto a Real time Database (Firebase). The number is checked against the existing users registered and using the E- wallet Application. On finding a match, automatic deduction of toll is done through the application. If the existing user has insufficient funds, he/she is notified about the same in the application. If the license plate registered does not have a pre-existing account, an official notification concerning the payment of toll due is sent to the registered phone or email-id of the vehicle owner. For the purpose of providing a proof to the customer, the system uploads the frame at which the correct number plate was detected on the firebase storage which is accessible to the customer on his android application.

The ANPR System is linked with a Firebase Real time Database. It is a Backend-as-a-Service — BaaS — that started as an YC11 Startup and grew up into a next-generation app-development platform on Google Cloud Platform. Firebase frees developers to focus crafting better user experiences. There is no need to manage servers and to write APIs. Firebase is the server, API and the data store, all written so generically that it can be modified to suit most needs.

The reason why Firebase Database has been chosen is because it is supported by various languages and platforms. It can be used for building a back-end for various platforms like android, iOS, Web, etc. Firebase is a Cloud hosted database, i.e. it runs on a cloud and access to the user is provided as a service. It stores data in JSON (JavaScript Object Notation) format, a format to store or transport data. Due to the data synchronization used in real time, every update is received by the devices/clients in no time. As the database can be accessed directly from the mobile device or any system, there is no need for an application server.

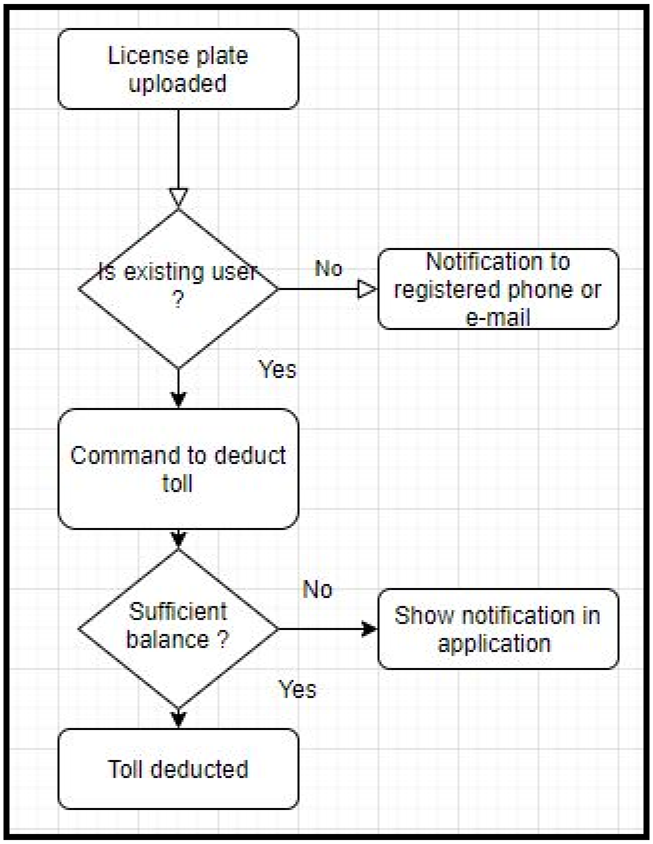


Fig. 4.19. Flowchart of Toll Deduction Process

When the developer builds cross-platform apps with Firebase iOS, Android, and JavaScript SDKs, all the clients share one Real time Database instance and automatically receive updates with the newest data. Instead of typical HTTP requests, the Firebase Real time Database uses data synchronization - every time a data change, any connected device receives that update within milliseconds. Provide collaborative and immersive experiences without thinking about networking code. Firebase apps remain responsive even when offline because the Firebase Real time Database SDK persists your data to disk. Once connectivity is reestablished, the client device receives any changes it missed, synchronizing it with the current server state.

Chapter 5

Methodology for Android Application

Android App Implementation

From the user’s side, one has to download the application for the toll system. The user has to then sign-in giving his email id and password followed by adding his vehicle details. Most apps need to know the identity of a user. Knowing a user's identity allows an app to securely save user data in the cloud and provide the same personalized experience across all of the user's devices.

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more. Firebase Authentication integrates tightly with other Firebase services, and it leverages industry standards like OAuth 2.0 and OpenID Connect, so it can be easily integrated with your custom backend. The Firebase Authentication SDK provides methods to create and manage users that use their email addresses and passwords to sign in. Firebase Authentication also handles sending password reset emails.

The Firebase Real time Database can be accessed directly from a mobile device; there’s no need for an application server. Security and data validation are available through the Firebase Real time Database Security Rules, expression-based rules that are executed when data is read or written.

The steps are:

1. Signup and Login - The user can register using his email id and a password for his account. The user will be uploading the vehicle details through this account created. This process is carried out using the Firebase Authentication SDK. During login, after passing the details of email id and password, Firebase will verify the credentials and in return,

will give a response that tells the application if the authentication is successful or not. The whole process is divided into three activities:

- Signup Activity: This activity is used to register the user into the application and after successful registration, transfer the user to the Main Activity.
- Login Activity: This activity is used to login the user into the application and after successful login, transfer the user to Main Activity.
- Main Activity- This is the stage a user reaches on successful login or sign-up and contains the main features of the application.

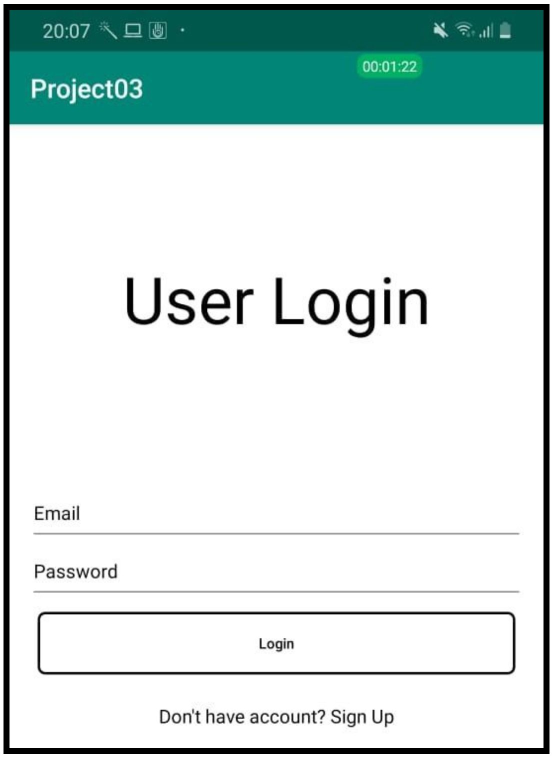
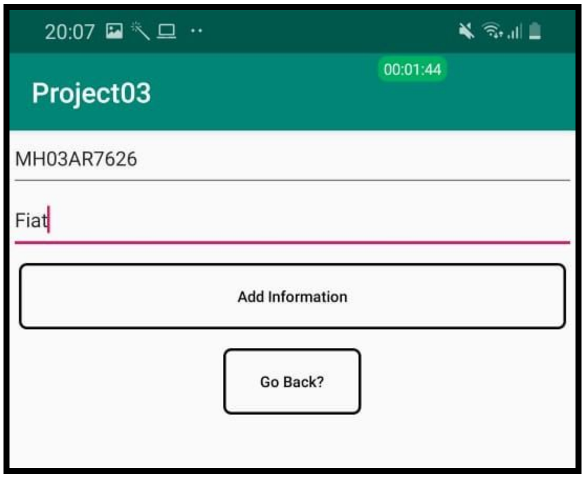


Fig 5.1. Login Page

2. Upload Vehicle Information- After the login verification, the user first needs to upload the information of his vehicle, i.e. the license plate number and the vehicle name. As a single user can have multiple vehicles, information of several vehicles can be uploaded to the database under a single account. All deductions of toll on detection of the vehicle would be through the same account and would be displayed separately.

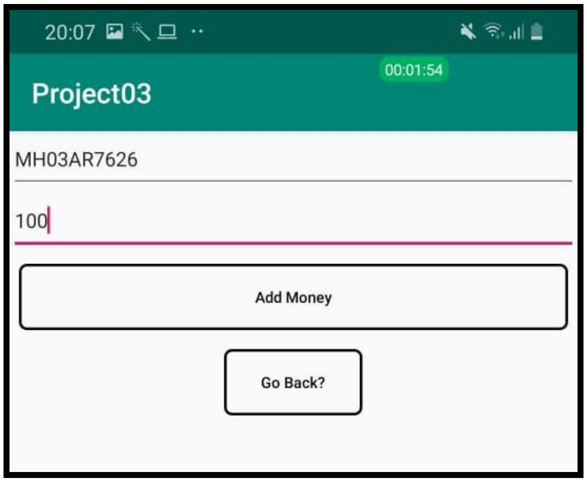


MH03AR7626

Fiat

Fig 5.2. Adding Vehicle Information

3. Adding Money- The user needs to add money to his e-wallet. No constraints have been added regarding the amount of money that can be added to the e-wallet as of now.

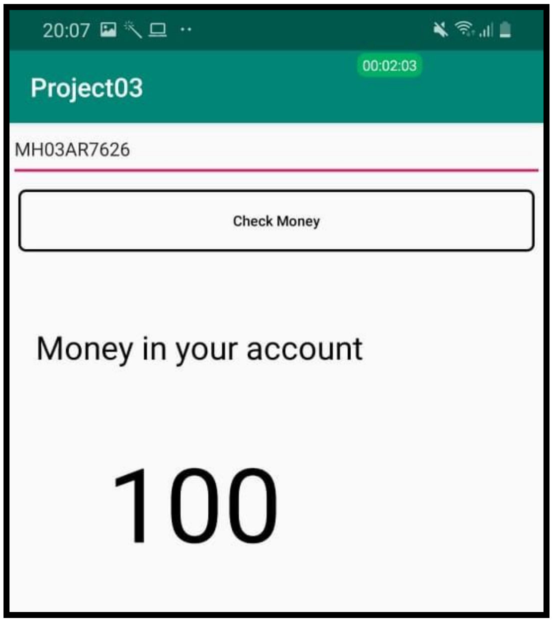


MH03AR7626

100

Fig 5.3. Adding Money

4. Deducting Money - Whenever a vehicle linked with the user's account has been spotted at any toll, a notification will be sent to the user and money will be directly deducted from his e-wallet. If there is no sufficient money in the e-wallet, he will get a warning regarding the same.



MH03AR7626

Money in your account

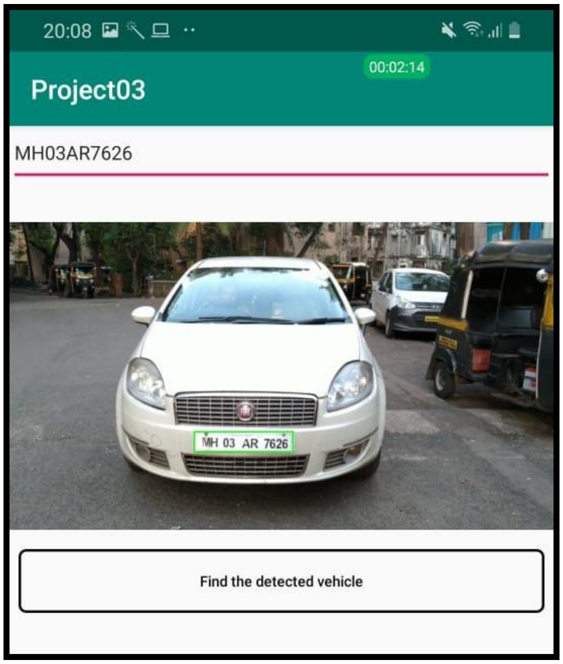
100

Fig 5.4. Money in the E-wallet

5. Verification of Detected Vehicle

The user can verify his vehicle on the app by checking the frame which is uploaded on the firebase storage by the ANPR System. Because of Cloud Storage, all this happens in real time and the need of low-latency is achieved.

Cloud Storage for Firebase is a powerful, simple, and cost-effective object storage service built for Google scale. The Firebase SDKs for Cloud Storage add Google security to file uploads and downloads for the Firebase apps, regardless of network quality. Developers can use Firebase SDKs to store images, audio, video, or other user-generated content. On the server, they can use Google Cloud Storage, to access the same files.



MH03AR7626

Fig 5.5. Verifying the vehicle

Thus, on successful verification the payment is done as the vehicle passes the toll center. The time required for the vehicle detection and payment of toll is very less as compared to the manual collection. This helps reduce the traffic at the toll center and the vehicles are moving at a faster rate.

Chapter 6

Automated E-Toll System

The system makes use of Raspberry Pi as the microcontroller board for image acquisition and processing, which also acts as the server for the Firebase. The camera used for the capture of video is connected to the Raspberry Pi. The video captured is thus processed further by the controller. The connection diagram for the system is as shown below.

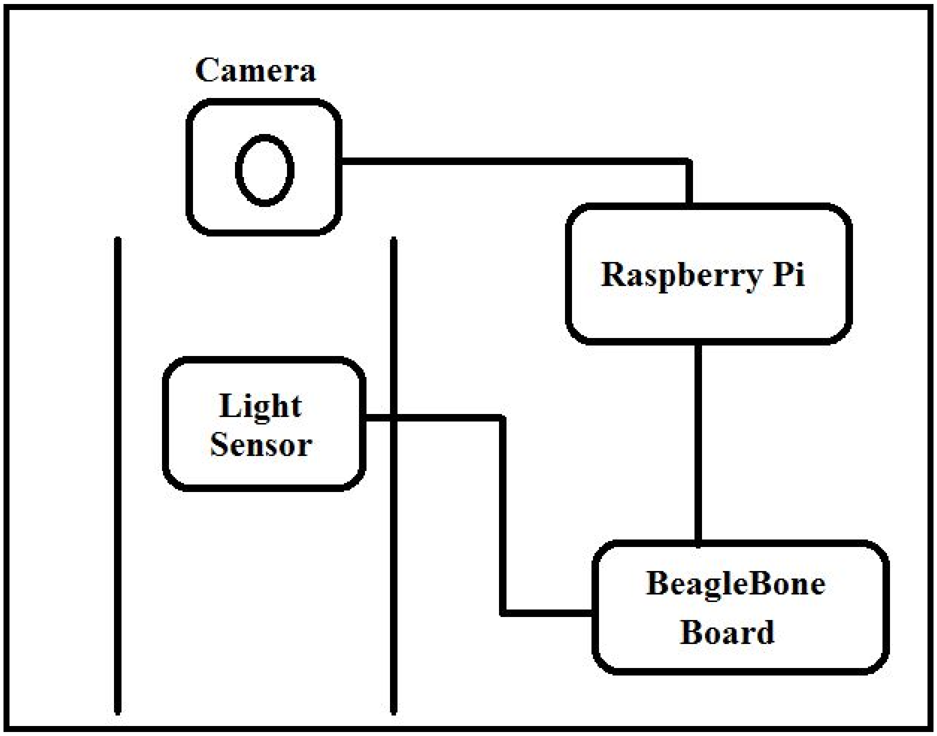


Fig 6.1 Connection Diagram of Proposed System

Connection between Sensor and Beagle Bone Board

The sensor used here is OPT3001 which is an ambient Light Sensor provided by Texas Instruments. Measurements can be made from 0.01 lux up to 83k lux without selecting full-scale ranges by using the built-in, full-scale setting feature. This capability allows light measurement over a 23-bit effective dynamic range. The connection is Inter-Integrated System (I2C) which is a serial communication protocol, so data is transferred bit by bit along a single wire via the Serial

Data (SDA) line. Like SPI, I2C is synchronous, so the output of bits is synchronized to the sampling of bits by a clock signal shared between the master and the slave.

Beagle Bone Black is an Open Source Hardware Software Development Platform using Texas Instruments' AM335x System-on-Chip device. It provides an easy to use Software Development Platform based on an ARM Cortex-A8 CPU together with many other peripherals integrated into the die. The connection between Beagle Board and OPT3001 is as shown below.

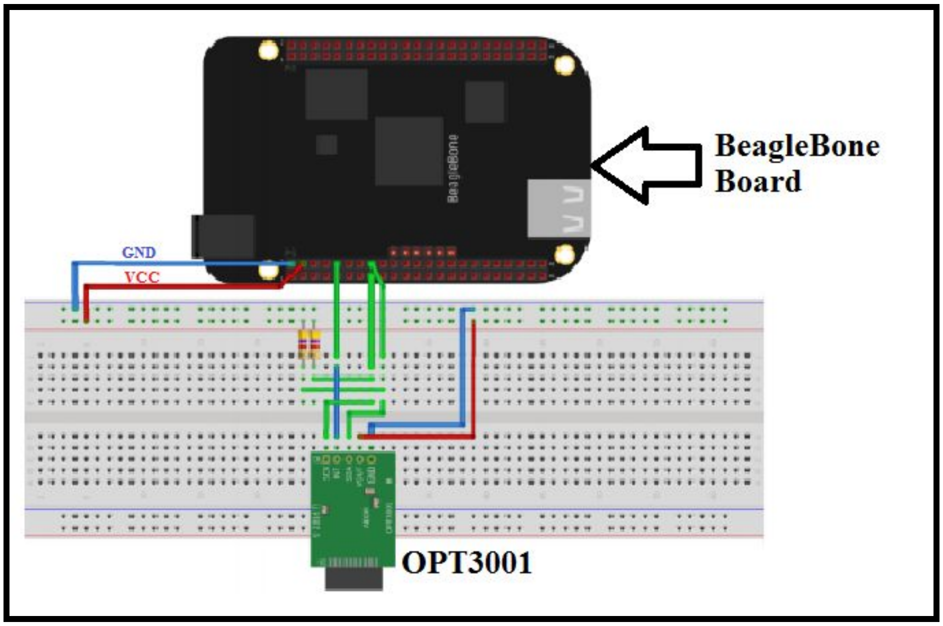


Fig 6.2 Connection between Beagle Bone Board and OPT3001

One of the GPIO pins in Beagle Board is connected with Raspberry Pi. This signal is continuously checked by Raspberry pi in order to trigger the camera connected to it.

<u>Connection between Camera and Raspberry Pi</u>

Raspberry Pi has its own camera module that can be used. Rather than buying this module, we can just use a USB camera which is available everywhere. The configuration of such a camera is very simple. Just connect the USB camera to one of the USB ports of the Raspberry Pi. After doing that, we can check whether the camera works well or not by capturing images using the "fswebcam" package. First, we need to install that package using the following command: `sudo apt-get install fswebcam.` After that, we can use it to capture an image. A very basic way to do that is as follows: `sudo fswebcam test_image.jpg.` This accesses the camera, captures an image, and saves it to the SD card in the current directory with the name “test_image.jpg”.

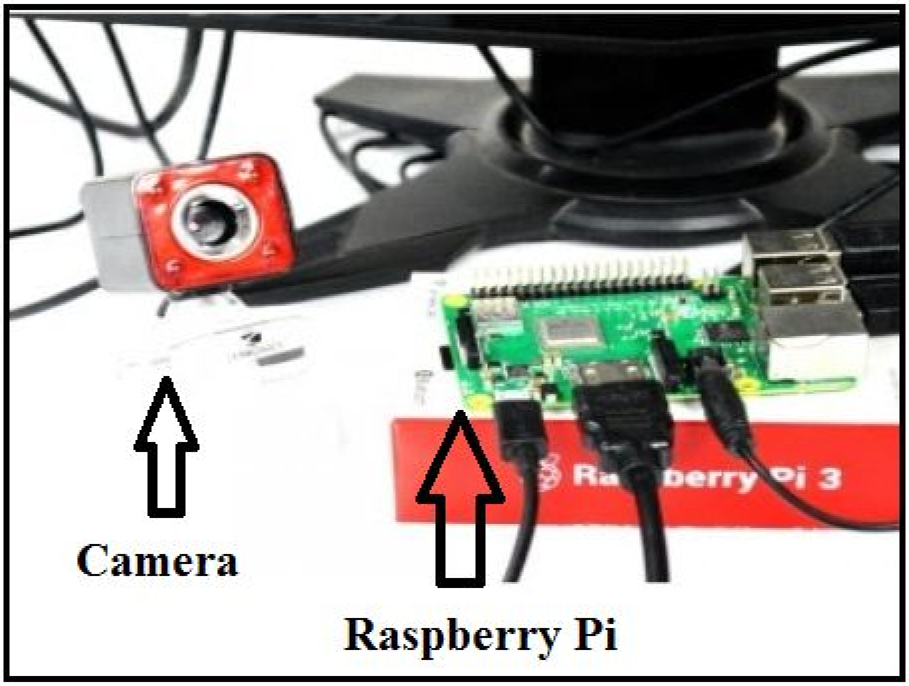


Fig 6.3 Connection between Raspberry Pi and camera

In order to record a video of 2 seconds, the package “ffmpeg” needs to be installed. The installation step is: `sudo apt-get install ffmpeg.` After this running the following command will record a video for 2 seconds and save it in the current working directory.

```
ffmpeg -t 2 -f v4l2 -framerate 25 -video_size 640x80 -i /dev/video0 output.mkv
```

The "-t time" flag specifies the duration of the recording in seconds. The name of the captured video will be output.mkv with a frame rate of 25 and resolution of 640*80.

The captured video is then taken as input in the License Plate Recognition program written in python and we get required output.

Auto run Raspberry Pi on Boot Up

The raspberry pi receives the captured video periodically and processes the same to acquire the license plate number, which it updates on the Firebase Database through the same python program. To make the system portable, we have configured the raspberry pi to auto run the python program on boot.

The use of the rc.local method is preferred as the microcontroller will be running in a headless mode. The steps required are:

1. Edit the rc.local folder. To do so type '**sudo nano /etc/rc.local**' with root permissions.
2. Add commands required to configure other devices with raspberry pi - here USB camera and Beagle Bone trigger signal connected to ambient light sensor.
3. Exit the file with exit 0.
4. Type '**sudo python /home/…folder directory.../ANPRsystem.py &**', ANPR system being the python program running the system. Ampersand is used as the Automated E-Toll System runs an infinite loop and it keeps sensing the Beagle Bone signal. If the ampersand is not included and if the program runs continuously, the Pi will not complete its boot process. The ampersand allows the command to run in a separate process and continue booting with the main process running.
5. Type '**sudo reboot**' to reboot the system.

Chapter 7

Results and Analysis

7.1 Result

The approaching vehicle is detected by the sensor which is placed on the surface over which the vehicle is moving.

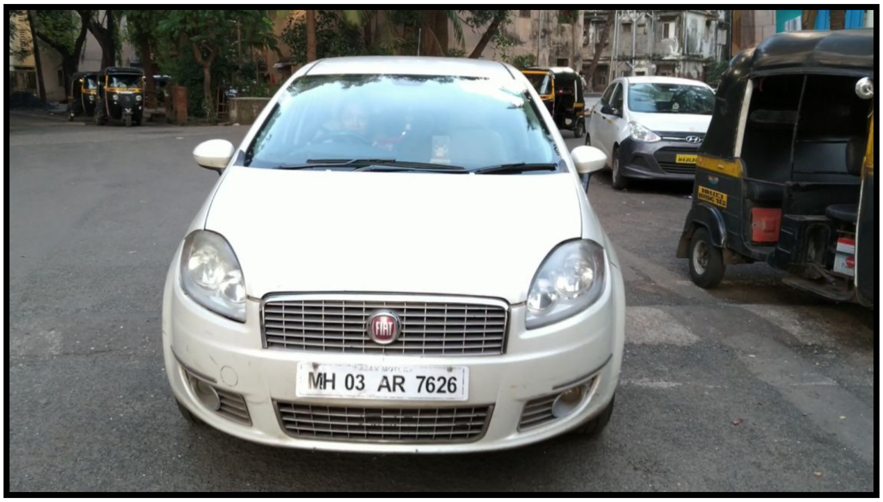


Fig 7.1 Detecting Approaching Vehicle

On sensing the vehicle, the signal is sent to the microcontroller to trigger the camera system to capture/record a video of 2 seconds. The User Interface (UI) of the system is as shown below.

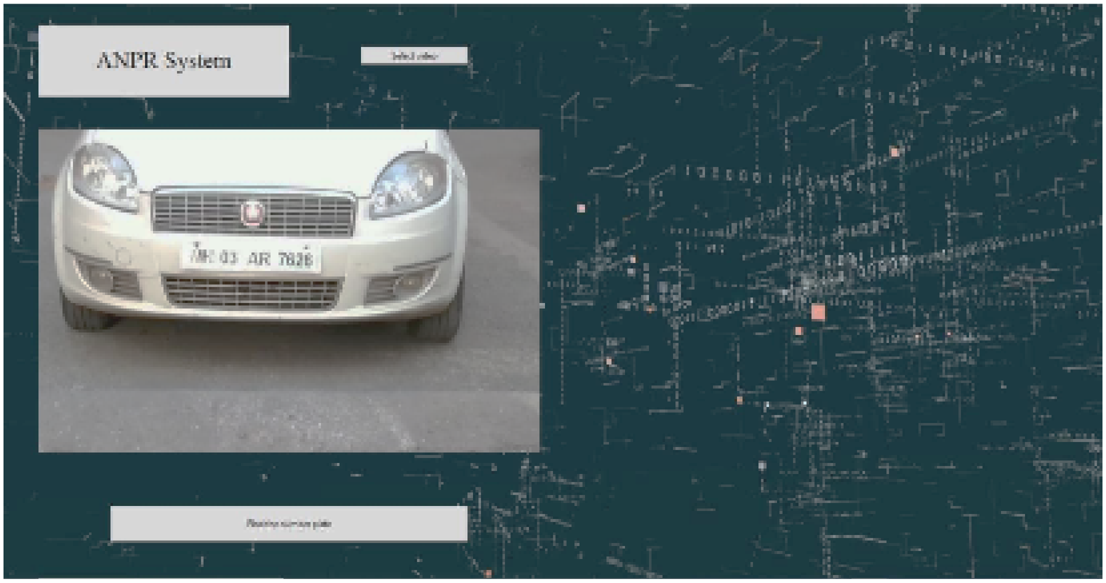


Fig 7.2 User Interface capturing the video

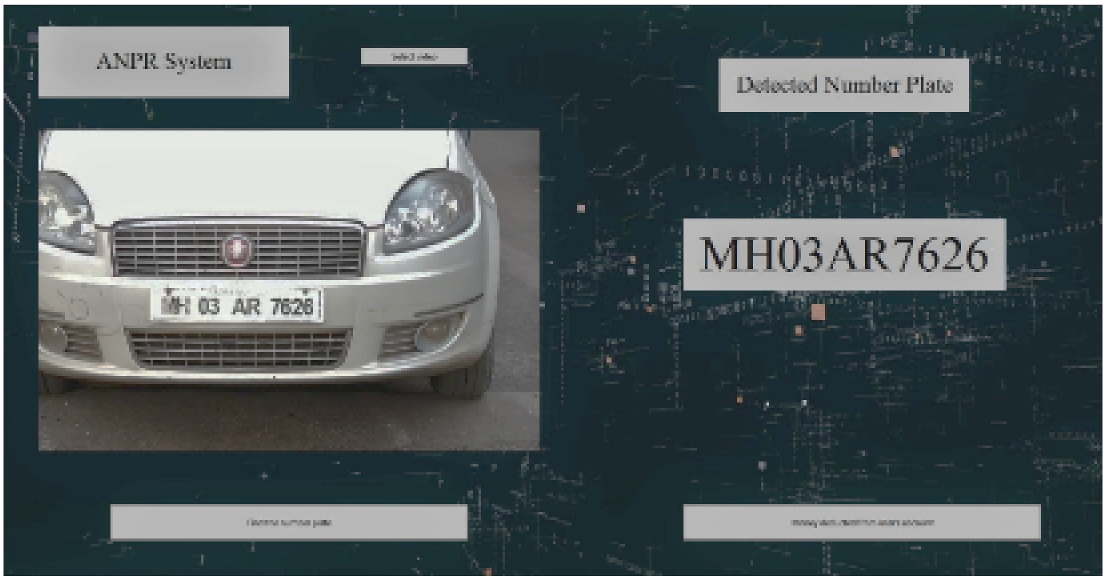
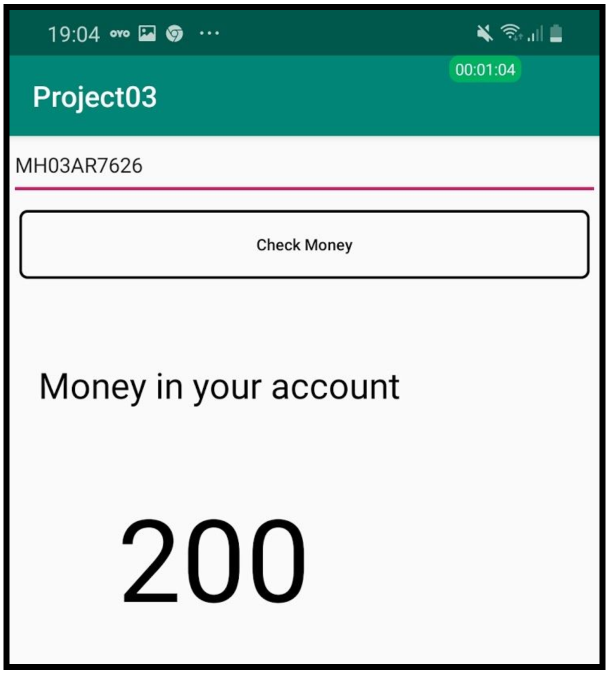


Fig 7.3 License Number detected on the User Interface

As soon as the License Number is detected at the system the required toll amount is deducted from the e-wallet. The notification regarding the vehicle detected at toll is also shown in the android application.

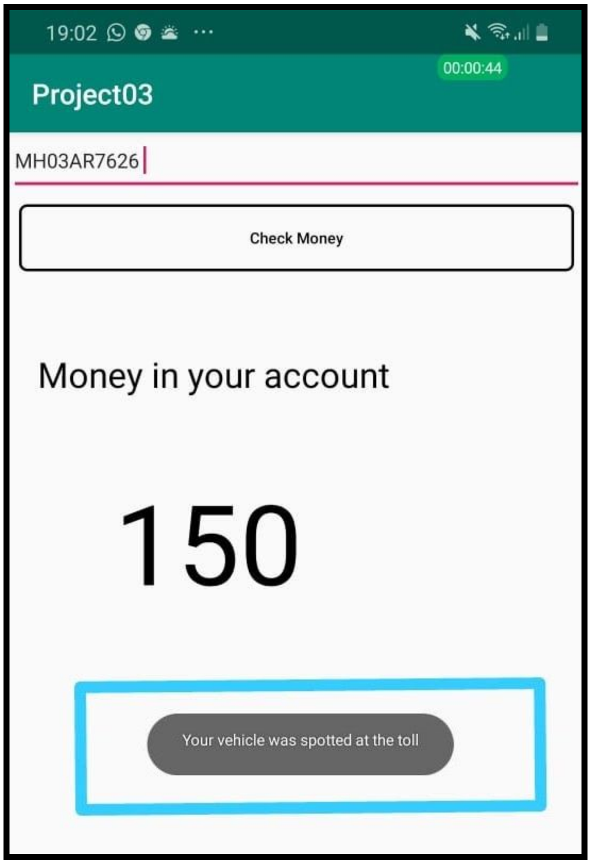


MH03AR7626

Money in your account

200

Fig 7.4 Money in E-wallet before Detecting



MH03AR7626

Money in your account

150

Your vehicle was spotted at the toll

Fig 7.5 Money in E-wallet after Detecting

This system proves to be purely user friendly and the payment is done in a fraction of seconds. The vehicle owner gets notification regarding the vehicle reaching the toll center. The user also gets a picture of the detected vehicle. Hence, the transparency of the application is improved manifold.

The performance evaluation of the proposed system is done here. Since video of the incoming vehicle is taken, the vehicle need not stop at the toll center. The following table gives the results of the videos taken using the proposed system.

Table 1. Results from videos of different vehicles

True Number Plate	**Predicted Number Plates**	**No. of frames required**	**Time Taken (sec)**
MH01DK2599	MH0lDK2599 MH01BK2599 MH01DK2599	3	7.3323
MH03AR7626	AR7626 MH03AR7626	2	6.923

Table 1 shows the predicted number plates and the time taken to detect the license number of moving vehicles in seconds. Thus, the total time taken is around 7 - 8 seconds for each vehicle at the toll center. The average time taken to recognise a vehicle is 7.12 seconds. Considering the average time of 7.12 seconds, in 1 hour around 505 vehicles can be recognised at a single lane on the toll booth.

There is provision of storing the video captured of a vehicle detected at the toll although the microprocessor may be processing the frames of the preceding ones. Thus, vehicles can keep passing through the lane without any reduction in speed even if the processor may not have completed generating a license plate number for the preceding vehicles.

It should be noted that the average time of 7.12 seconds includes the time taken to generate frames. If faster processors are used, generation of frames will be faster and thus this time can be reduced significantly. Another factor which affects the speed of recognition of vehicles is the frame at which the potential license plate boundary is found. The camera should be set at optimum angle and the sensor needs to be triggered at the proper time, taking into consideration its distance from the camera, to be able to detect the probable boundary from the starting frames of the video. Keeping all these parameters into account, the system is suited for real time execution on roads.

The number plate recognition algorithm was also tested on still images of vehicles. The algorithm works for a fraction of seconds and gives instant output. The test images were taken at different light conditions and different angles. Some extreme angles are not detected by the system which is a limitation of the system. However, it can be rectified by applying some tilt operations for better output.

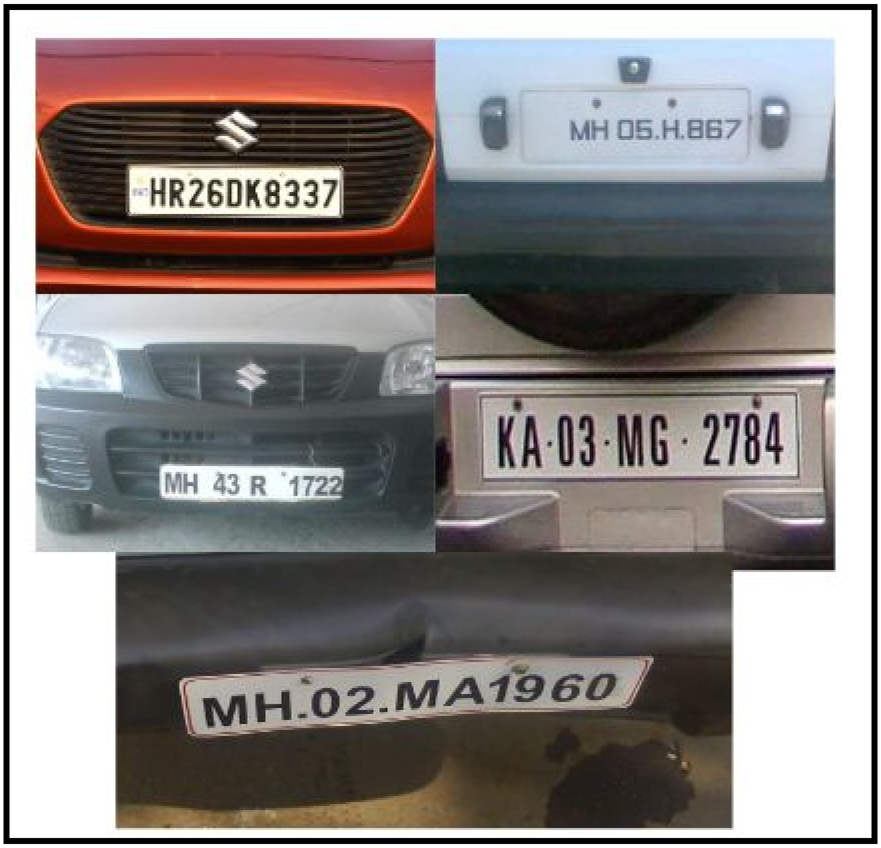


Fig 7.6 Still Images of License Plates used for testing

Table 2. Results from still images

True Number Plate	Predicted Number Plates	Time Taken (sec)
HR26DK8337	HR26DK8337	0.29507
KA03MG2784	KA03MS2784	0.32603
MH05H867	MH0S867	0.21769
MH43R1722	MH43R1722	0.30376
MH02MA1960	None	0.02

Table 2 gives the predicted number plates and time taken for still images of License Plates. The images are as shown in Fig. 7.6. The average of 0.285 seconds is recorded which indicates that the recognition technique is fast enough. Compared to the manual and RFID based toll collection systems, these results show that the processing time is very less in case of License Plate recognition. Thus, from the above results we can record an accuracy of 85.71 %.

7.2 Analysis

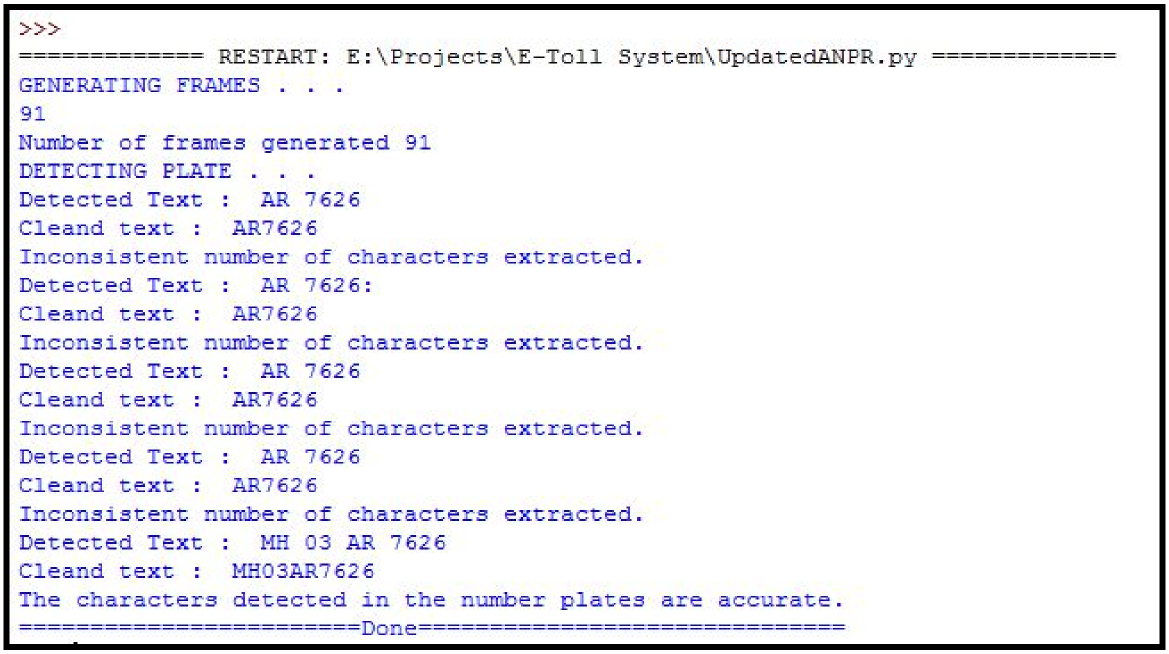
In order to reduce the execution time, the generated frames are sent for pre-processing in the reverse order. That means the last generated frame is processed first. This increases the chance of getting the license number correct.

During the execution of the python program in the background, various strings are produced at the output. This leads to inconsistency in the system. This is avoided by adding verification conditions in the program to get the correct license number.

Example, the obtained string is checked whether the first 2 characters are string, next 2 are integers, next 2 strings and last 4 to be integers. Also, the first two characters are checked whether they are among the state codes in India.

The changes are:

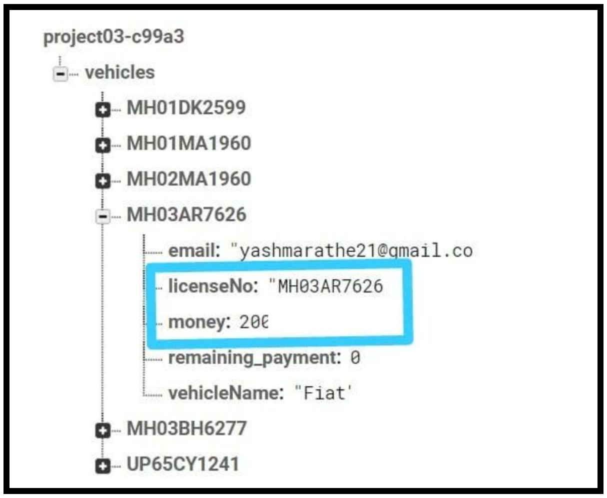
1. ‘O’ detected at positions of a number is changed to zero and vice versa.
2. ‘S’ detected at positions of a number is changed to 5 and vice versa.



```
>>>
============= RESTART: E:\Projects\E-Toll System\UpdatedANPR.py =============
GENERATING FRAMES . . .
91
Number of frames generated 91
DETECTING PLATE . . .
Detected Text :  AR 7626
Cleand text :  AR7626
Inconsistent number of characters extracted.
Detected Text :  AR 7626:
Cleand text :  AR7626
Inconsistent number of characters extracted.
Detected Text :  AR 7626
Cleand text :  AR7626
Inconsistent number of characters extracted.
Detected Text :  AR 7626
Cleand text :  AR7626
Inconsistent number of characters extracted.
Detected Text :  MH 03 AR 7626
Cleand text :  MH03AR7626
The characters detected in the number plates are accurate.
========================Done==============================
```

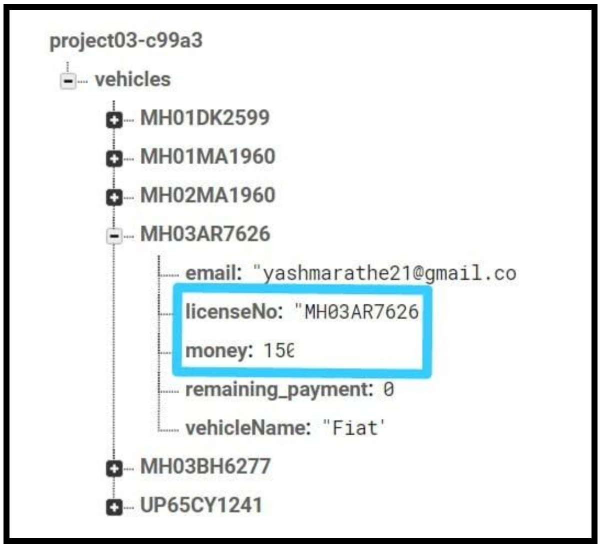
Fig 7.7 Execution of program at console

During the payment, the database updates the new value as soon as the vehicle is detected at the toll center. The database screenshots before and after the payment are shown below.



```
project03-c99a3
  vehicles
    MH01DK2599
    MH01MA1960
    MH02MA1960
    MH03AR7626
      email: "yashmarathe21@gmail.co
      licenseNo: "MH03AR7626
      money: 200
      remaining_payment: 0
      vehicleName: "Fiat'
    MH03BH6277
    UP65CY1241
```

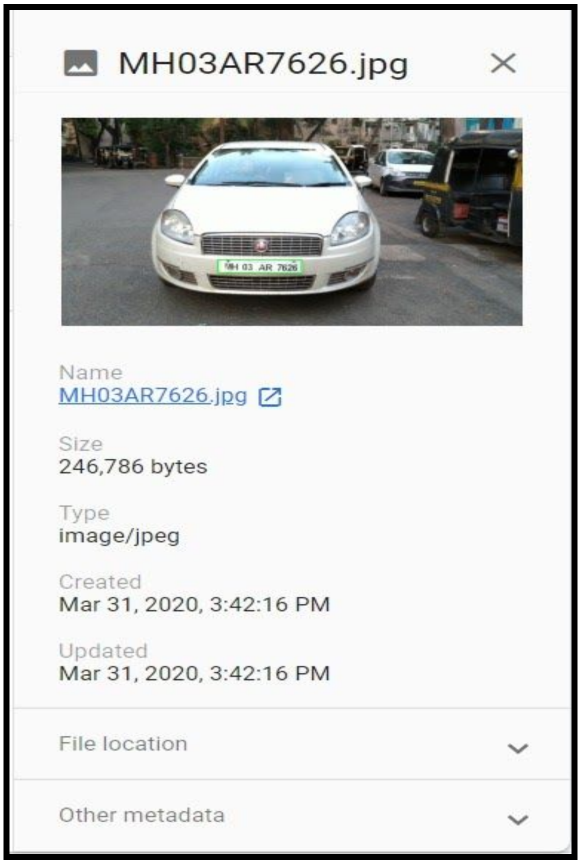
Fig 7.8 Database before payment



```
project03-c99a3
  vehicles
    MH01DK2599
    MH01MA1960
    MH02MA1960
    MH03AR7626
      email: "yashmarathe21@gmail.co
      licenseNo: "MH03AR7626
      money: 15€
      remaining_payment: 0
      vehicleName: "Fiat'
    MH03BH6277
    UP65CY1241
```

Fig 7.9 Database after Payment

The screenshot of the frame at which license plate was obtained is saved in the database to increase the transparency of the system. The details related to the image are also saved. This is also made available to the user in the android application.



MH03AR7626.jpg

Name
MH03AR7626.jpg

Size
246,786 bytes

Type
image/jpeg

Created
Mar 31, 2020, 3:42:16 PM

Updated
Mar 31, 2020, 3:42:16 PM

Fig 7.10 Generated frame in Database

7.3 Advantages of the system

The unique feature of the system is that it uses the existing feature of a vehicle for toll collection. Unlike RFID based systems, where a tag has to be attached to a vehicle, our system eliminates the requirement. Due to automated systems, the human labor required at the toll center is considerably less which leads to more revenue for the toll contractors. This system also helps in tracing blacklisted or stolen vehicles which will be easily identified by the system. The traffic at the toll center is reduced to a great extent and hence the system works faster and efficiently. This encourages the general public in using the toll centers with this system more frequently again leading to more revenue. The system also encourages digital payment for the users which helps in striving towards a cashless economy. This eliminates manual cash exchange also. On providing discounting through application, there will be more downloads of the application, hence leading to a more successful system. The system also provides a real time verification of the License Number which helps the user to verify the transaction of toll amount from the e-wallet application.

7.4 Limitations of the system

This system provides minimum limitations due to its user friendly nature and quick response time. However, this system cannot be used for number plates with fancy fonts or other languages. The examples of such license plates are shown below.

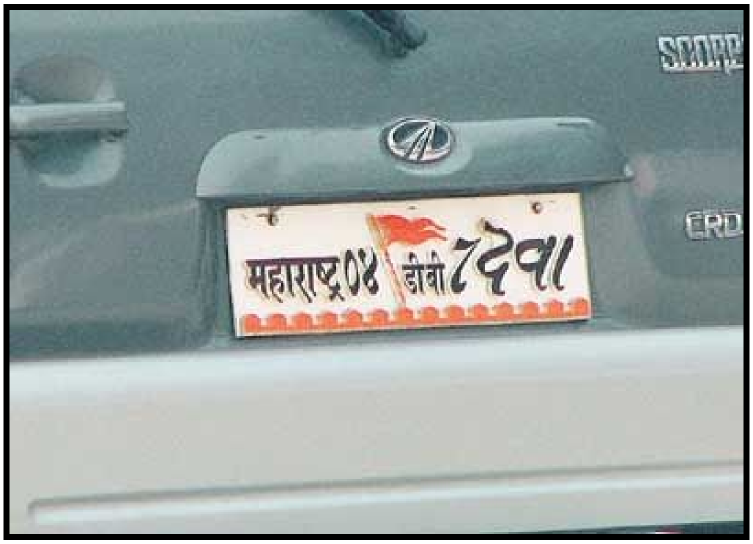


Fig 7.11 Vanity Number Plate

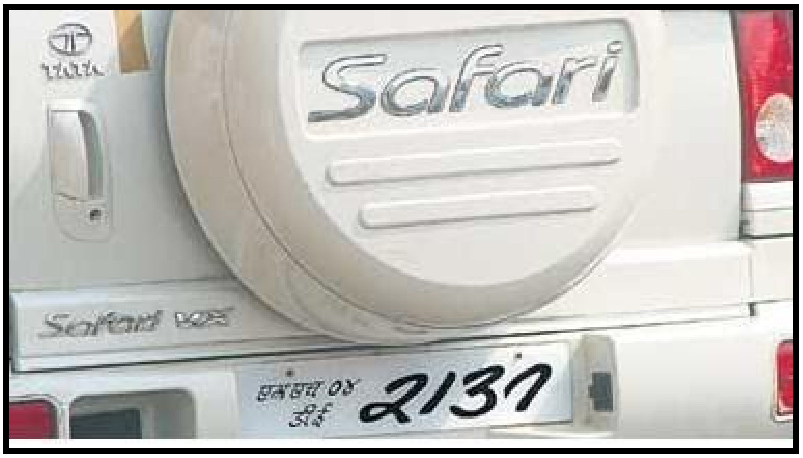


Fig 7.12 License Plate with other characters

This system is designed for the working of toll centers in India specifically. The string validations are as per the standards set by The Government of India.

Although the system has been made to be highly efficient and reliable, there are chances that vehicle plates may not be recognized due to environmental lightning factors or vernacular font. These number plates would then have to be uploaded to another database from which manual entry of the vehicle needs to be made of the license plates that are recognizable by humans. Although illegal, if a vehicle does not have a license plate number, the vehicle cannot be recognized in any way by the system.

The system relies heavily on the toll users being registered users of the application. Although the implementation at the road is not affected by it, overhead of notifications to be sent to the unregistered users increases that decrease the intended system performance. An added task of tracking pending dues by them is also added.

Chapter 8

Conclusion and Future Scope

8.1 Conclusion

In this project, the technique to recognize the License Number of vehicles is presented. For this, introduction to image capture, preprocessing, edge detection, segmentation, character resizing and finally recognizing the characters of the number plate specifically on Indian number plates is done. On successful identification of License Number, the toll amount is directly deducted from the e-wallet. The android application hence provides a user friendly interface to verify the toll amount and the vehicle detected. The database used is Firebase which is integrated with the number plate detection and Android application. The string validations helped improve the efficiency of the system by removing erroneous characters. Dataset creation consists of a number of images and videos which are collected at real times. It has been found that other characters do exist in the number plate besides the vehicle's own registered number. To deal with such erroneous characters, character analysis has been applied using our machine learning algorithm. For such systems, a few changes are to be made to the proposed system. This system proves to be a user friendly technique for the toll collection system.

8.2 Future Scope

In future perspective, the speed of license plate recognition can be increased using multithreading. The system, at this level, is processing each frame at a time which leads to more time for the detection. If frames are processed in parallel, then the computing time can be reduced. This system can also be used at parking lots in residential areas, malls and at corporate complexes. The efficiency of the system can be improved more by using a better quality camera for better results.

Neural networks can be used in the license plate extraction for character recognition. This not only improves the efficiency of the system with more and more vehicles, but also helps us to remove the limitation of vernacular font. The network can be trained heavily for different languages and fonts to make our system more independent. The system can also be trained for very low or very bright lighting environments which previously would have been an issue thus bettering the efficiency of the system

With better real-time vehicle detection algorithms and cameras, the sensors can be removed which is presently needed to notify the camera of an approaching vehicle. With this system, the camera need not be triggered i.e. it will continuously record a video of the road and automatic real-time recognition of vehicles shall be done. This removes our dependency on the sensor.

References

Proceeding Paper,

1) Etqad Khan, Dipesh Garg, Rajeev Tiwari and Shuchi Upadhyay, “Automated Toll Tax Collection System using Cloud Database”, 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU), February, 2018, India

2) R Palekar, S Parab and D Parikh. Real Time License Plate Detection using OpenCV and Tesseract, International Conference on Communication and Signal Processing, April, 2017, India.

3) A K Singh and S Roy. Automatic Number Plate Recognition Indian System using Surveillance Cameras, March 2015, New Delhi, India.

4) M Olsen, J Hedman and Ravi Vatrapu. Designing Digital Payment Artifacts, International Conference on Electronic Commerce, August 2012, Singapore.

5) J. Bernsen, “Dynamic thresholding of grey-level images”, Proc. Eighth Int’l Conf. on Pattern Recognition, pp. 1251-1255, Paris, 1986

Links,

6) Addition of Firebase to Android System, https://firebase.google.com/docs/android/setup Last accessed: August 08, 2019.

7) License Plate Detection using OpenCv and Python https://cvisiondemy.com/license-plate-detection-with-opencv-and-python/ Last Accessed : March 25, 2019

8) Oladeji, F., Developing a License Plate Recognition System with Machine, Learning in Python. Retrieved November 29, 2017, from https://blog.devcenter.co/developing-a-license-plate-recognition-system-with-machine-learning-in-python-787833569ccd

Journal Paper,

9) Roopali Batra, Neha Kalra, “Are Digital Wallets The New Currency?”, Apeejay Journal of Management and Technology, Vol 11, No 1, January 2016

10) Poonam Painuly and Shalu Rathi (2016) “Mobile wallet: An upcoming mode of business transactions “International journal in management and social science .volume 4 pp356-363

11) Ambarish Salodkar, Karan Morey and Prof. Mrs. Monali Shirbhate, “Electronic Wallet”,

International Research Journal of Engineering and Technology (IRJET), Volume 2, Issue 9, December 2015

12) Pinal Chauhan, “E-Wallet: The Trusted Partner in our Pocket”, International Journal for Research in Management and Pharmacy , Vol. 2, Issue 4, April 2013

13) Satadal Saha (2009)," License Plate Localization from Vehicle Images: An Edge Based Multi-stage Approach" International Journal of Recent Trends in Engineering.

14) Ray Smith, "An Overview of the Tesseract OCR Engine", Proc. of ICDAR 2007, Volume 2, Page(s):629 - 633, 2007.

15) M.A. Massoud, " Automated new license plate recognition in Egypt" Alexandria Engineering Journal 52, 319-326

Thesis,

16) R.W. Smith, The Extraction and Recognition of Text from Multimedia Document Images PhD Thesis, University of Bristol, November 1987.

Project Grant

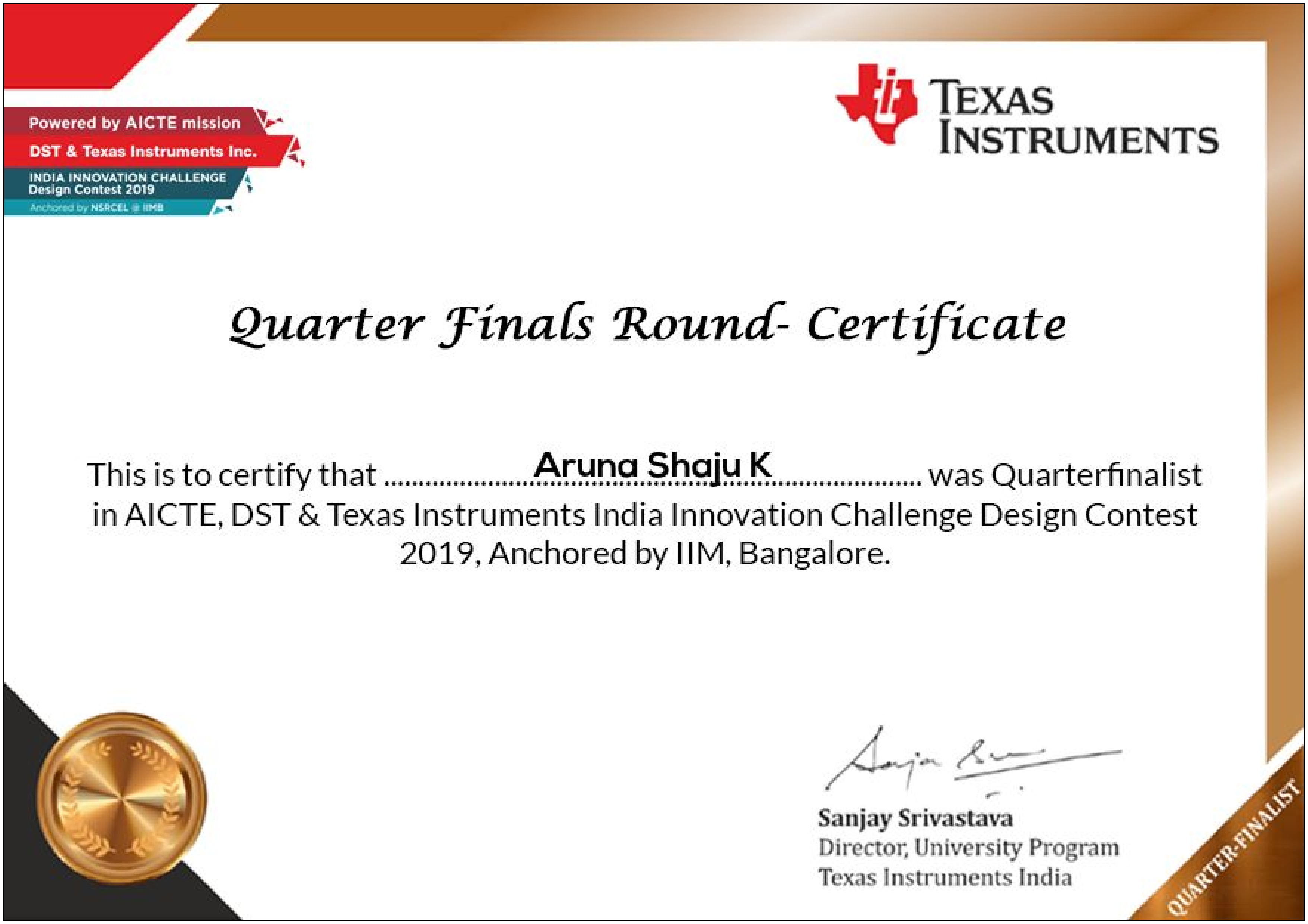
Our project received a grant in terms of components and kits from Texas Instruments via the India Innovation Challenge Design Contest 2019 anchored by IIM, Bangalore.

The details of components received are as follows:

1. **EVM BOOSTXL-SENSORS**

   It consists of OPT3001 Ambient Light Sensor, TMP007 Temperature Sensor, BMI160 Inertial measurement sensor, BMM150 Geomagnetic sensor, BME 280 Environmental Sensor

2. BeagleBone® AI AM5729 development board for embedded Artificial Intelligence



Powered by AICTE mission

DST & Texas Instruments Inc.

INDIA INNOVATION CHALLENGE
Design Contest 2019

Anchored by NSRCEL @ IIMB

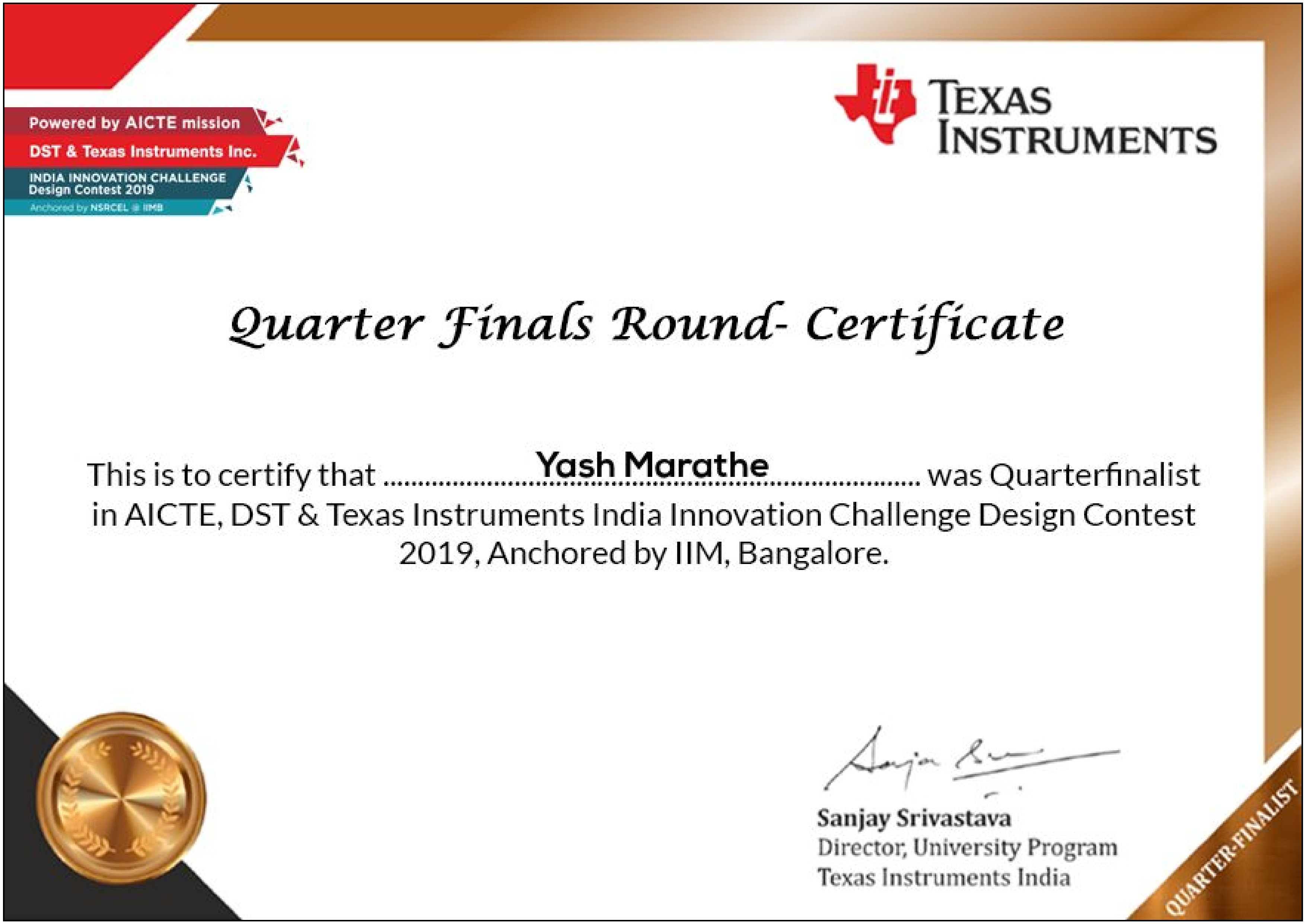
TEXAS INSTRUMENTS

Quarter Finals Round- Certificate

This is to certify that ....**Aruna Shaju K**................ was Quarterfinalist in AICTE, DST & Texas Instruments India Innovation Challenge Design Contest 2019, Anchored by IIM, Bangalore.

Sanjay Srivastava
Director, University Program
Texas Instruments India

QUARTER-FINALIST



Powered by AICTE mission
DST & Texas Instruments Inc.
INDIA INNOVATION CHALLENGE
Design Contest 2019
Anchored by NSRCEL @ IIMB

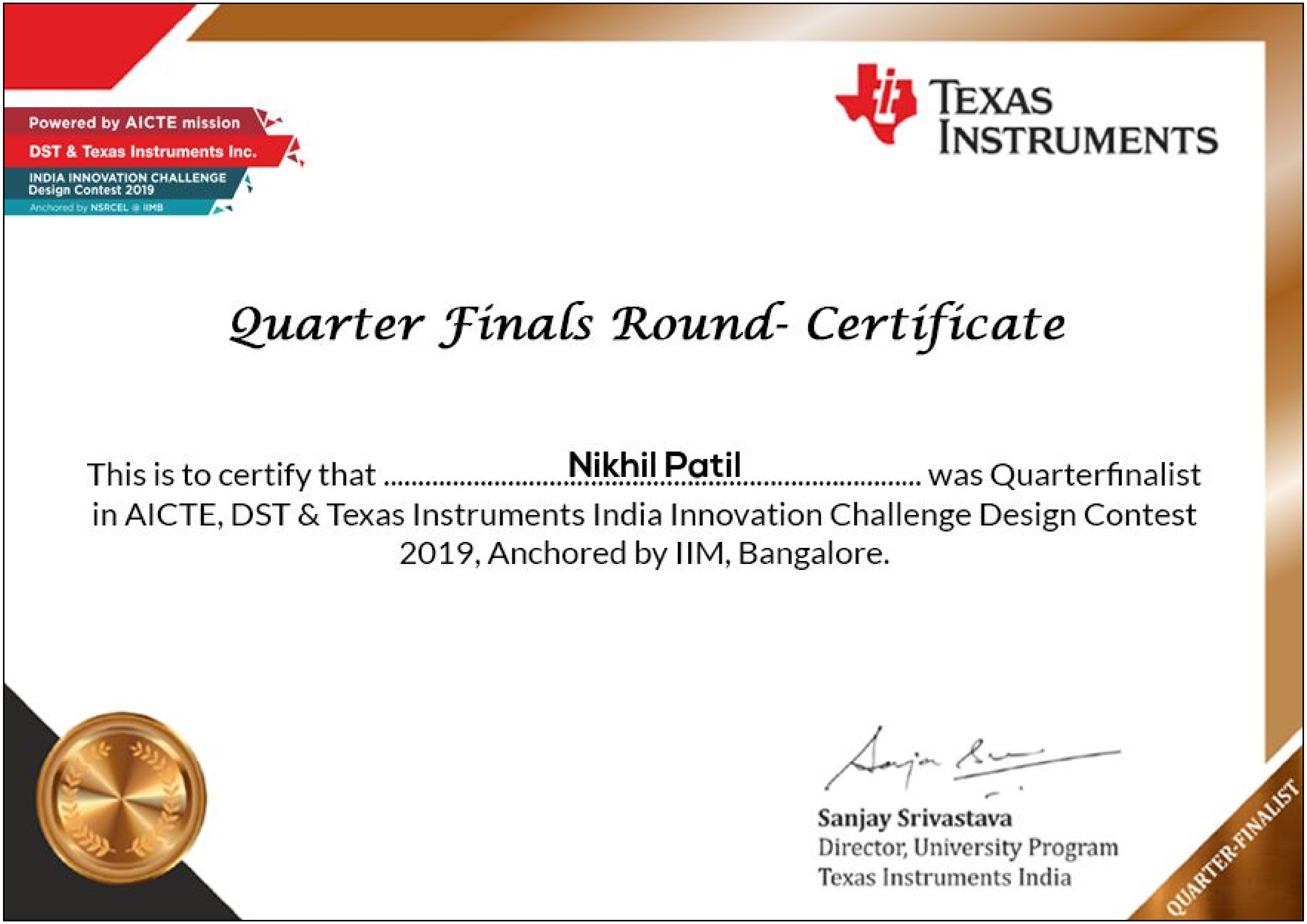
Texas Instruments

Quarter Finals Round- Certificate

This is to certify that ....**Yash Marathe**........................... was Quarterfinalist in AICTE, DST & Texas Instruments India Innovation Challenge Design Contest 2019, Anchored by IIM, Bangalore.

Sanjay Srivastava
Director, University Program
Texas Instruments India

QUARTER-FINALIST



Powered by AICTE mission

DST & Texas Instruments Inc.

INDIA INNOVATION CHALLENGE
Design Contest 2019

Anchored by NSRCEL @ IIMB

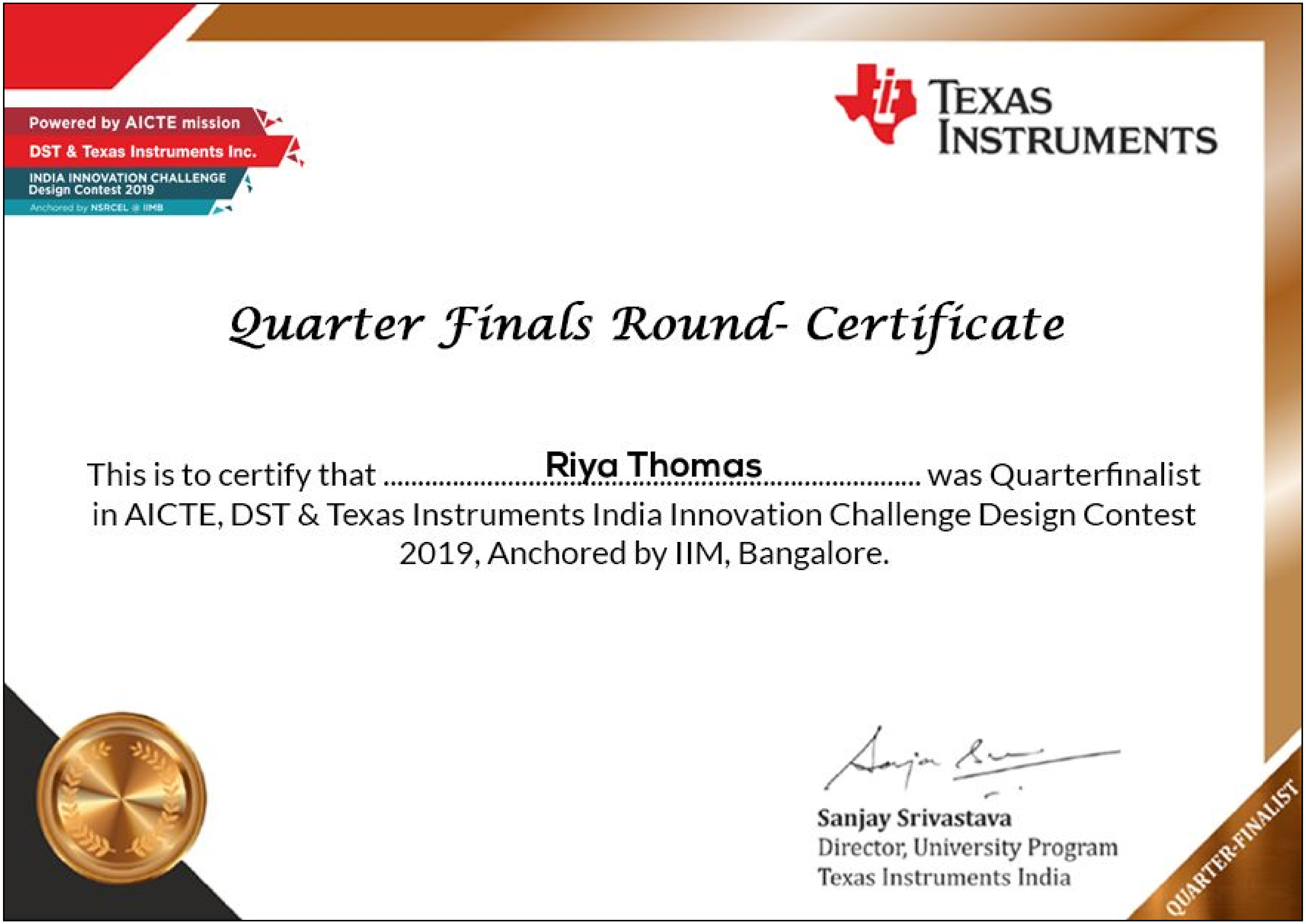
TEXAS INSTRUMENTS

Quarter Finals Round- Certificate

This is to certify that ....**Nikhil Patil**... was Quarterfinalist in AICTE, DST & Texas Instruments India Innovation Challenge Design Contest 2019, Anchored by IIM, Bangalore.

Sanjay Srivastava
Director, University Program
Texas Instruments India

QUARTER-FINALIST



Powered by AICTE mission
DST & Texas Instruments Inc.
INDIA INNOVATION CHALLENGE
Design Contest 2019
Anchored by NSRCEL @ IIMB

TEXAS INSTRUMENTS

Quarter Finals Round- Certificate

This is to certify that ....**Riya Thomas**.............................. was Quarterfinalist in AICTE, DST & Texas Instruments India Innovation Challenge Design Contest 2019, Anchored by IIM, Bangalore.

Sanjay Srivastava
Director, University Program
Texas Instruments India

QUARTER-FINALIST

Appendix

Raspberry Pi 3 b+:

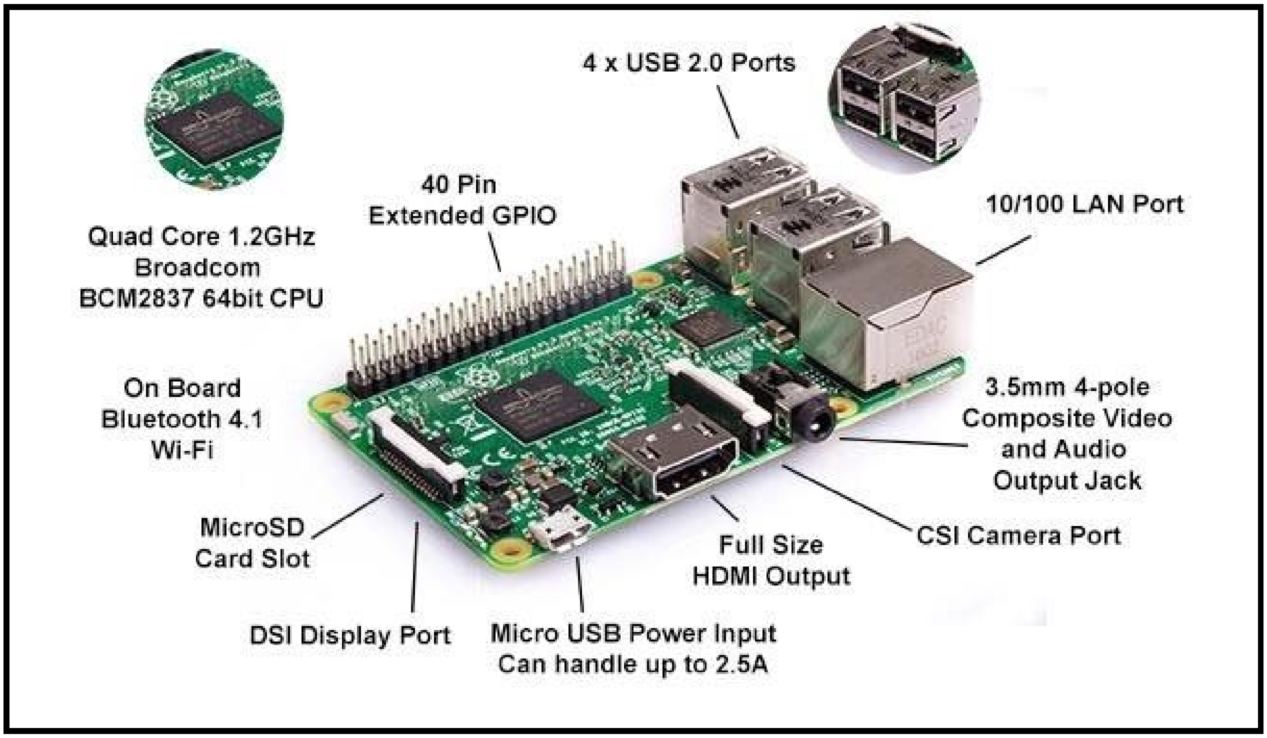

4 x USB 2.0 Ports
Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
40 Pin Extended GPIO
10/100 LAN Port
On Board Bluetooth 4.1 Wi-Fi
3.5mm 4-pole Composite Video and Audio Output Jack
MicroSD Card Slot
CSI Camera Port
Full Size HDMI Output
DSI Display Port
Micro USB Power Input Can handle up to 2.5A

Fig. 8 Raspberry Pi 3 Model B+

The Raspberry Pi 3 Model B+ is the latest product in the Raspberry Pi 3 range, boasting a 64-bit quad core processor running at 1.4GHz, dual-band 2.4GHz and 5GHz wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and PoE capability via a separate PoE HAT The dual-band wireless LAN comes with modular compliance certification, allowing the board to be designed into end products with significantly reduced wireless LAN compliance testing, improving both cost and time to market. The Raspberry Pi 3 Model B+ maintains the same mechanical footprint as both the Raspberry Pi 2 Model B and the Raspberry Pi 3 Model B.

Improved thermals on the Pi 3 B+ means that the CPU on the BCM2837 SoC can now run at 1.4GHz, a 17% increase on the previous Pi 3 model (which ran at 1.2GHz). Video performance on Pi 3 B+ is similar to the previous generation Pi 3, the Video Core being clocked at 400MHz for video processing and the 3D graphics processor running at 300MHz.

Table 3. Specifications of Raspberry Pi 3 Model B+

Parameters	**Specifications**
Processor	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
Memory	1GB LPDDR2 SDRAM
Connectivity	a) 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless b) LAN, Bluetooth 4.2, BLE c) Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps) d) 4 × USB 2.0 ports
Access	Extended 40-pin GPIO header
Multimedia	H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1
Environment	Operating temperature, 0–50°C
Input Power	5V/2.5A DC via micro USB connector 5V DC via GPIO header Power over Ethernet (PoE)–enabled (requires separate PoE HAT)

A significant change on the Pi 3 B+ compared to the Pi 3 is the inclusion of a new faster, dual-band wireless chip (CYW43455) with 802.11 b/g/n/ac wireless LAN and Bluetooth 4.2. The dual-band 2.4GHz and 5GHz wireless LAN enables faster networking with less interference (although the higher bandwidth has less range), and the new PCB antenna technology should allow better reception. Bluetooth allows you to use a wireless keyboard/track pad without extra dongles, keeping things nice and tidy.

Beagle Bone Black Development Board

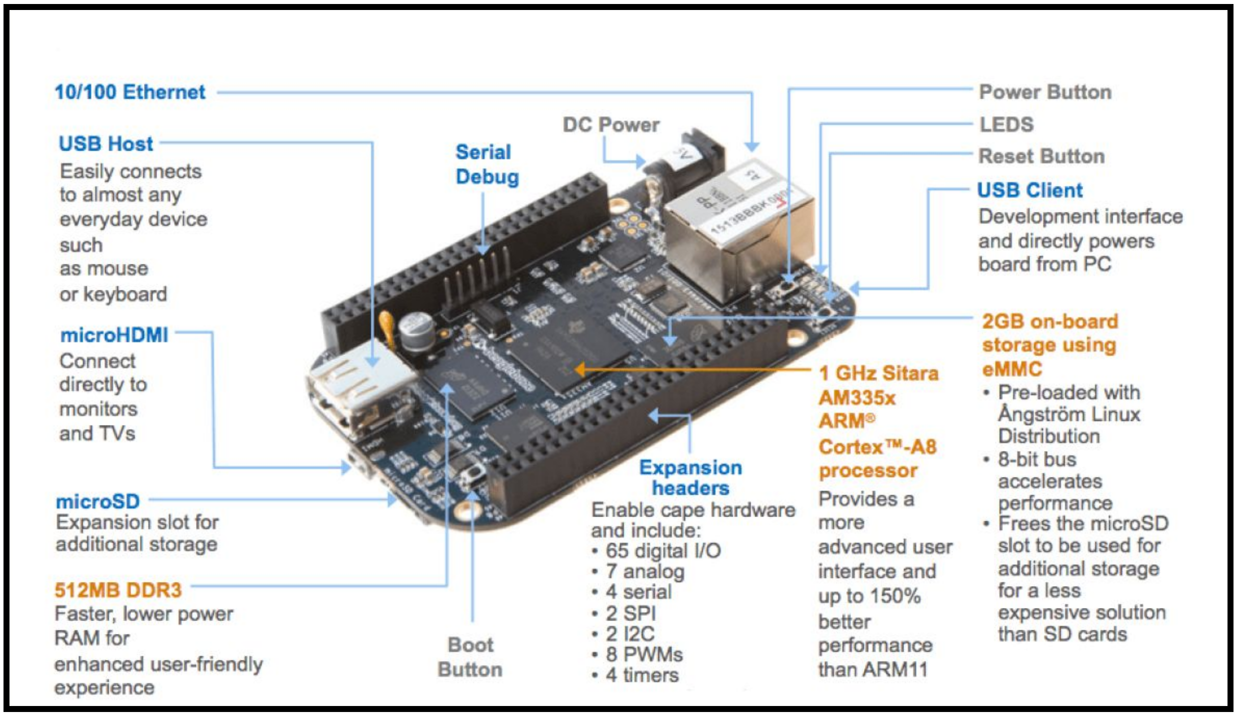

10/100 Ethernet
USB Host
Easily connects to almost any everyday device such as mouse or keyboard
microHDMI
Connect directly to monitors and TVs
microSD
Expansion slot for additional storage
512MB DDR3
Faster, lower power RAM for enhanced user-friendly experience
Serial Debug
DC Power
Boot Button
Expansion headers
Enable cape hardware and include:
• 65 digital I/O
• 7 analog
• 4 serial
• 2 SPI
• 2 I2C
• 8 PWMs
• 4 timers
1 GHz Sitara AM335x ARM® Cortex™-A8 processor
Provides a more advanced user interface and up to 150% better performance than ARM11
Power Button
LEDS
Reset Button
USB Client
Development interface and directly powers board from PC
2GB on-board storage using eMMC
• Pre-loaded with Ångström Linux Distribution
• 8-bit bus accelerates performance
• Frees the microSD slot to be used for additional storage for a less expensive solution than SD cards

Fig. 9 Beagle Bone Black Development Board

Beagle Bone Black is a low-cost, open source, community-supported development platform for ARM® CortexTM-A8 processor developers and hobbyists. Boot Linux in under 10-seconds and get started on Sitara™ AM335x ARM Cortex-A8 processor development in less than 5 minutes with just a single USB cable. Many other Linux distributions and operating systems are also supported on Beagle Bone Black including, Ubuntu, Android and Fedora

Beagle Bone Black's capabilities can be extended using plug-in boards called “capes” that can be plugged into Beagle Bone Black's two 46-pin dual-row expansion headers. Capes are available for, VGA, LCD, motor control, prototyping, battery power and other functionality. Sitara AM3358BZCZ100 is the processor for the board. Micron 512MB DDR3L or Kingston 512MB DDR3 is the Dual Data Rate RAM memory. TPS65217C PMIC provides the power rails to the various components on the board. SMSC Ethernet PHY is the physical interface to the network. Micron eMMC is an onboard MMC chip that holds up to 4GB of data. HDMI Framer provides control for an HDMI or DVI-D display with an adapter.

Table 4. Specifications of Beagle Bone Black

Parameters	Specifications
Processor	Sitara AM3358BZCZ100 1GHz, 2000 MIPS
SDRAM Memory	512 MB DDR3L 800MHz
Onboard Flash	4 GB, 8 bit Embedded MMC
Audio Out	Via HDMI Interface, Stereo
Video Out	16B HDMI, 1280 x 1024 (MAX)
User Input	Reset Button Boot Button Power Button
Environment	Operating temperature, 0–50°C

It also offers access to many of the interfaces and allows for the use of add-on boards called capes, to add many different combinations of features. A user may also develop their own board or add their own circuitry. Power Button alerts the processor to initiate the power down sequence. 10/100 Ethernet is the connection to the LAN.

BOOT switch can be used to force a boot from the microSD card if the power is cycled on the board, removing power and reapplying the power to the board. USB Client is a miniUSB connection to a PC that can also power the board. There are four blue LEDS that can be used by the user. Reset Button allows the user to reset the processor. A MicroSD slot is where a microSD card can be installed. MicroHDMI connector is where the display is connected to. USB Host can be connected to different USB interfaces such as Wi-Fi, BT, Keyboard, etc.

OPT 3001 Ambient Light Sensor

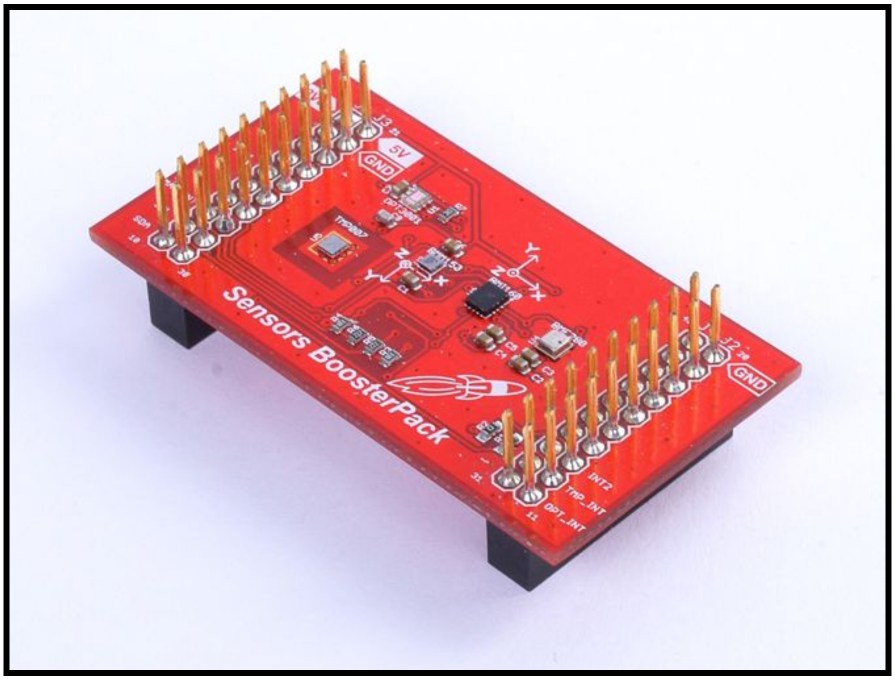


Fig. 10 Boostxl Sensors Booster pack with OPT3001 sensor

The BOOSTXL-SENSORS Booster Pack™ plug-in module is an easy way to add digital sensors to your Launch pad™ development kit design. MCU developers can use this Booster Pack module to start developing sensor applications using the on-board gyroscope, accelerometer, magnetometer, pressure, temperature, humidity and ambient light sensors.

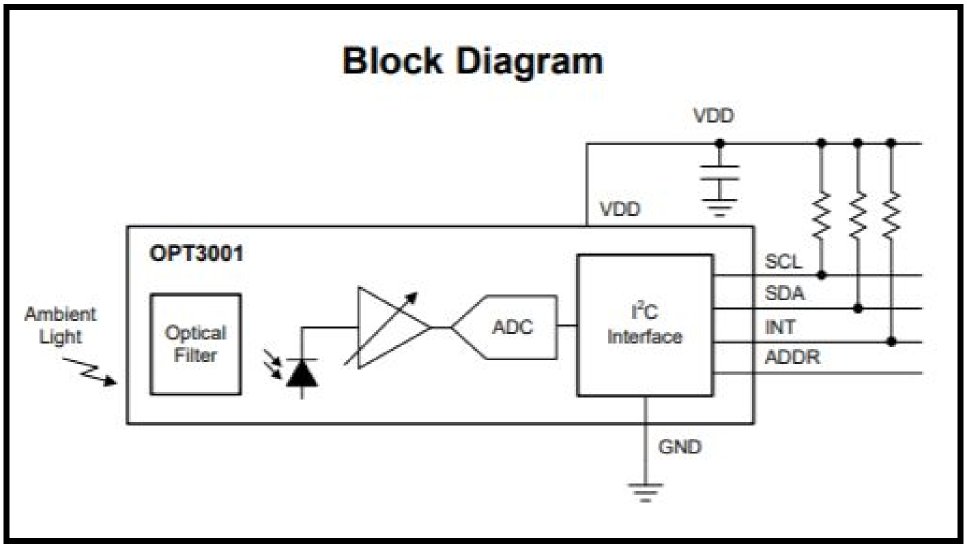


Fig. 11 Functional Block Diagram of OPT3001

The OPT3001 is a sensor that measures the intensity of visible light. The spectral response of the sensor tightly matches the photonic response of the human eye and includes significant infrared rejection. It is a single-chip lux meter, measuring the intensity of light as visible by the human eye. The precision spectral response and strong IR rejection of the device enables the OPT3001 to accurately meter the intensity of light as seen by the human eye regardless of light source. The strong IR rejection also aids in maintaining high accuracy when industrial design calls for mounting the sensor under dark glass for aesthetics. The OPT3001 is designed for systems that create light-based experiences for humans, and an ideal preferred replacement for photodiodes, photo resistors, or other ambient light sensors with less human eye matching and IR rejection.

Table 5. Specifications of OPT3001 Ambient Light Sensor

Parameter	**Specifications**
Measurements	0.01 lux to 83 k lux
Operating Current:	1.8 µA (typically)
Operating Temperature Range	–40°C to +85°C
Power-Supply Range	1.6 V to 3.6 V
Small-Form Factor	2.0 mm × 2.0 mm × 0.65 mm

OPT 3001 has high precision optical filtering to match human eyes. Its automatic full-scale setting feature simplifies software and ensures proper configuration and flexible interrupt system. Its applications include Display Backlight Controls, Lighting Control Systems, Tablet and Notebook Computers, Thermostats and Home Automation Appliances, Point-of-Sale Terminals, Outdoor Traffic and Street Lights, Cameras.

Camera for Image Acquisition

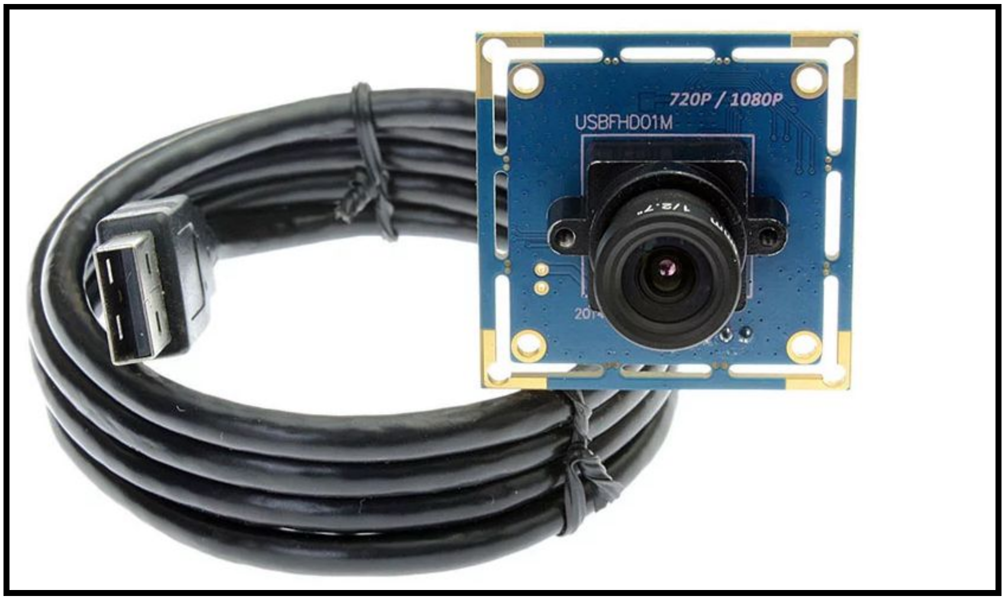


Fig. 12 USB Camera

This is a 1/2.7 inch OV2710 HD cmos camera module. Two megapixel high speed 100-120fps USB camera. Full HD real time free driver USB webcam for Windows, Linux, Mac with UVC, Compatible with raspberry pi, Ubuntu, Opencv, Amcap and many other USB webcam software, wide angle USB camera module with 180 degree fisheye lens, a 38x38/32x32mm mini micro USB board camera. USB camera module, well embedded in many machines, ATM machine, medical machine, automatic vending machine, industry machine.

The OV2710-1E supports multiple platform architectures and controllers with both parallel and MIPI interfaces. By allowing system designers to leverage the same opto-electrical design across various products and multiple market segments, the OV2710-1E significantly reduces product development time. Omni Vision's OmniPixel®3-HS pixel technology has already been proven in high quality webcam/video applications and is available in 1080p full HD in the OV2710-1E.

Eclipse Camera to avoid Glare Environment

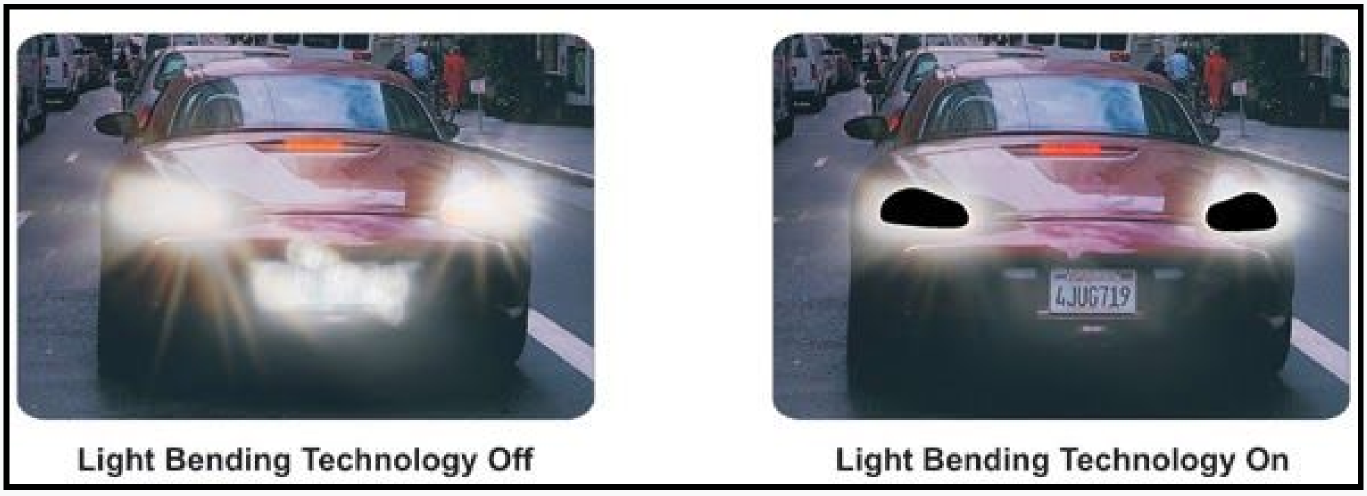


Fig 13. Solution for avoiding glare

Eclipse cameras can be deployed to reduce the headlight glare and reflect light from a vehicle. An eclipse camera is able to refract the light toward the vehicle where it can be used to produce a clear image. While the human eye cannot see in total darkness, a night vision camera is able to see in complete darkness, 0 LUX, without any light. Night Vision Cameras see beyond what the human eye can see because they see in the infrared spectrum. The human eye can only see in the visible light spectrum which is 400 - 700 nanometers. Thus, by using eclipse cameras, we can reduce the error that may occur in license plate detection due to the glare environment.