

Automated Parking Management system using Number plate recognition using Vision API mapped with local database

Yashwanth Krishna

B.Tech, ECE, Final Year

SRM Institute of Science & Technology

Chennai, Tamil Nadu, India

yashwanth3170@gmail.com

Prof.S.Indhumathi

Asst.Prof, ECE

SRM Institute of Science & Technology

Chennai, Tamil Nadu, India

Abstract—This Proposed system uses the concept of Number plate recognition from vehicles to keep track of the parking duration of the specific vehicle in the given slot. Also, the use of a local database helps to automate the process of allowing only those vehicles which are already stored in it. For the other outside vehicles, a manual override button is given to facilitate the entry of the vehicle.

Keywords—Google Cloud Vision API, MySQL, Raspberry PI 3, USB Webcam, Database, Parking Management System.

I. INTRODUCTION

This system is very practical in Shopping malls, where they give us a small card which records our entry time into the mall and calculate the parking charges accordingly. In case of that card being damaged or misplaced, a heavy fine is levied. To avoid all those, this proposed system uses the Number plate of the vehicle to automatically save the entry and exit time, and calculate the parking fees accordingly. Also, a local database stores a list of allowed vehicles that will be allowed automatically as it enters.

II. EXISTING SYSTEM

Before we go any further into this system, let us take a look at the present available systems. As far as India is concerned there is no such Automated parking management system that uses this concept fully. Even if something of these sorts is there, there is usage of RFID which is used to detect all the known vehicles. Usage of RFID cards only becomes more expensive and difficult to manage.

III. PROPOSED SYSTEM

The new proposed system makes use of the Number plate of every vehicle, which is by itself unique to each and every one. This reduces the hassle of usage of RFID tags or even parking slips that is used in Shopping malls. Based on the number of each vehicle, the system is able to store its entry time, exit time, and calculate the number of vehicles present in the parking space.

IV. HARDWARE REQUIREMENTS

A. Raspberry Pi 3

Raspberry Pi v3 is a pocket sized computer that is able to run Linux based operating system. Since we will be needing a computer to do all the processing work as well as store and display the results, The Pi is a perfectly suited choice for this purpose.

B. USB Webcam

A HD Camera is used to scan the Number plates of the vehicle while entering and exiting of the vehicle. This camera will capture the image, store it and then send the image for text detection in it.

C. Servomotor

A small Servomotor is used as the gate open close mechanism that will work on the bases if the input obtained after the processing done by the Raspberry Pi 3. A small cardboard piece is attached to the servomotor to show the gate open close action. PWM commands are used to control the servomotor through the Pi's GPIO pins.

D. LED

Two LEDs are used to show the driver the status of the gate. A RED light indicates that the gate is closed, or the vehicle is not stored in the database. A GREEN light indicates that the gate is open and that the vehicle is allowed to pass.

E. POWER SOURCE

A small battery is used to run the Servomotor and LED because taking the 5V power from the PI directly is very unstable to use in real time.

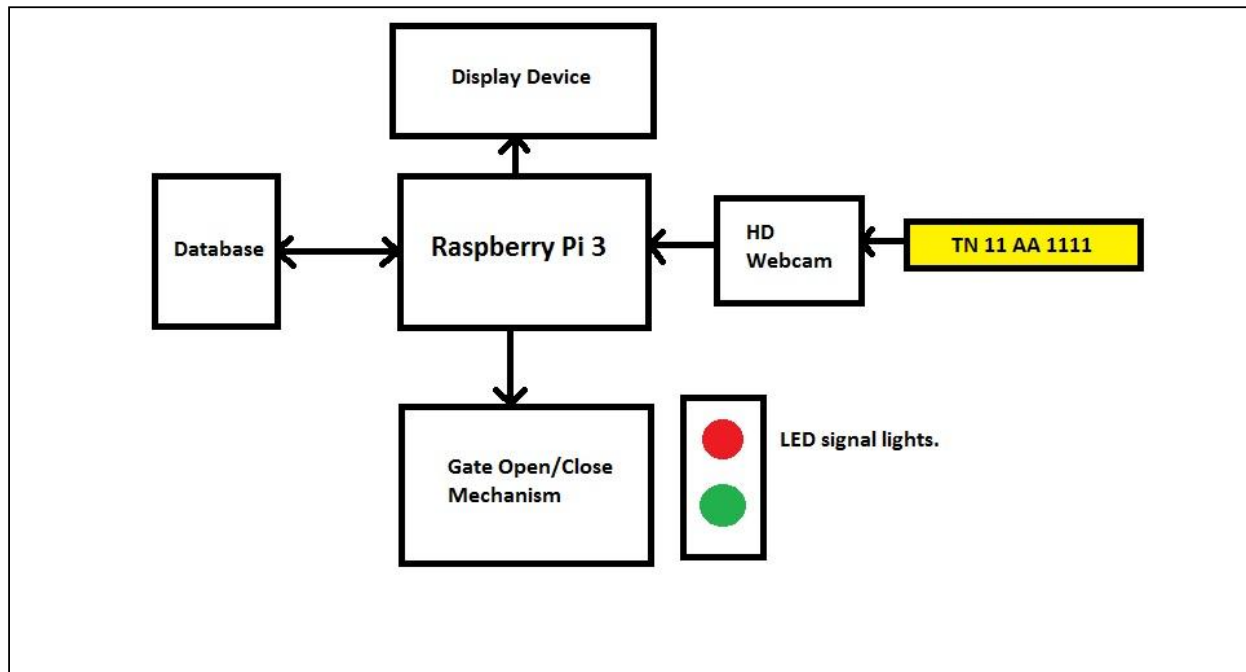


Fig. 1. Block Diagram of the Proposed System

This is done so that the records need not be viewed in the MySQL command prompt.

V. SOFTWARE REQUIREMENTS

A. GOOGLE CLOUD VISION API

Cloud Vision is an Application Program Interface that allows user to obtain OCR, labels and even description of the given input image. The image can be a locally stored one, or directly fed from a camera in Real-Time, as it is done here. Since it a cloud service, it requires an active internet connection to operate.

B. RASPBIAN

Raspbian is the default OS that comes bundled with the Raspberry Pi 3 when we purchase the NooBS SD Card. But there are several other alternatives that can be installed in the Pi board, but all Operating systems are based on Linux only. Raspbian is a altered version of Debian which is customised for Pi specifically.

C. PYTHON

Python is a very simple, easy to use programming language that is able to do the entire tasks that other programming languages are capable of. Also, Python is more of direct English like language which is very easy to understand and implement.

D. MYSQL

MySQL is a Database application based on MariaDB. We use this database service to mainly create three tables: Allowed vehicles (that stores the list of stored vehicles for immediate access), Entry timelogs (list of entered vehicles) and Exit

E. FLASK

Flask is a web based server application that is used to display the record of the database in a local host webpage.

VI. INITIAL PREREQUISITES

A. Fancy Number Plates

Here we show the photo of the front of a car with its number plate visible. Here the plate itself need not be in any specific format or whatsoever, but since this system entirely relies on the recognition of the text from the plate itself, number plates that employ a fancy design of representing the numbers or any other illegible format will not work properly. Even if it does recognize, chances are very highly possible that it may not be detected the same way the next time.

B. Database Entry

Before we start with the entry and exit of vehicles in the parking, we need to first run a database program where we will be able to feed the list of allowed vehicles that will have unrestricted access to and from the parking lot. Now this data entry will not be done by simply entering the number of every vehicle directly into the system but we will be using the images of the same car as the input. This will be captured by the camera and then the text will be obtained from that. This is done to ensure that the camera is able to capture this number in this format when it is actually is put into place while controlling the gate mechanism.

C. Pricing

Since there are no restrictions on the Database part, we are allowed to add almost infinite number of vehicles into the list. However, the usage of Google Vision API has some restrictions. While the API is free for Non Commercial and Project purposes, when this system is implemented for a proper Real-Time application, Google will start to charge a nominal fee which is as follows. The first 1000 requests every

month is free of cost, after which, for every subsequent 1000 requests made, it costs around \$1.50, which reflects to around 100 INR. This is way cheaper compared to other API provided by other vendors.

VII. WORKING CONCEPT

Once the database entries are made, it's time for the actual working of the program to start. Now the Camera is positioned exactly at a point where it is able to capture the front part of the arriving vehicle. This is done using trial and error. This part is very crucial because every car will have the plate positioned in different places. Once this is done, the system can go online.

Now, when a car approaches the gate, it has to stop at a predefined distance before the gate where the camera is fixed. This is done to ensure the no part of the plate is missing. This predefined position will be marked with the help of a painted line on the ground. After that, the person responsible to operate this system will run the program that will execute in the following order

a) *Decision:* The person will have to choose whether to run the entry program or the exit program depending upon the situation. This provision is given because the proposed system uses only a single gate for entry and exit and a single camera for both purposes.

b) *Execution:* Once the choice is made, the system captures the image with the help of the camera, feeds it into the system for processing. The Pi processes the image and returns the text from the image in the form of a value. Now the system maps this value with every value stored in the database that is stored already. If there is a match found, then the gate opens with the green LED glowing and a welcome message for the driver. Also, this plate value along with its entry or exit time is pushed into the respective databases. In case, a match is not found, then the system displays that there is no match and the gate will not open with the Red LED glowing. Now the person has the option of doing a manual override process, where he can allow the gate to open by the click of a single button. If he presses this button, then the gate will open, with the green LED glowing. The vehicle is allowed to pass and its plate value and its time of entry or exit also is simultaneously pushed into a different database which only stores the timelogs of all vehicles passed. All of this happens with a single press of that manual override button. Depending upon whether a vehicle enters or exits, the total number of vehicles present inside gets added or subtracted accordingly.

TABLE I. ENTRY TIMELOGS

Number	Time of Entry
AR19BA8892	2018-03-13 00:08:05
AP09BN7886	2018-03-13 00:08:34
TN45BA1065	2018-03-13 00:08:58

MH12DE1433	2018-03-13 00:09:41
TN99F2378	2018-03-13 00:10:01
TN07AJ5500	2018-03-13 00:10:51

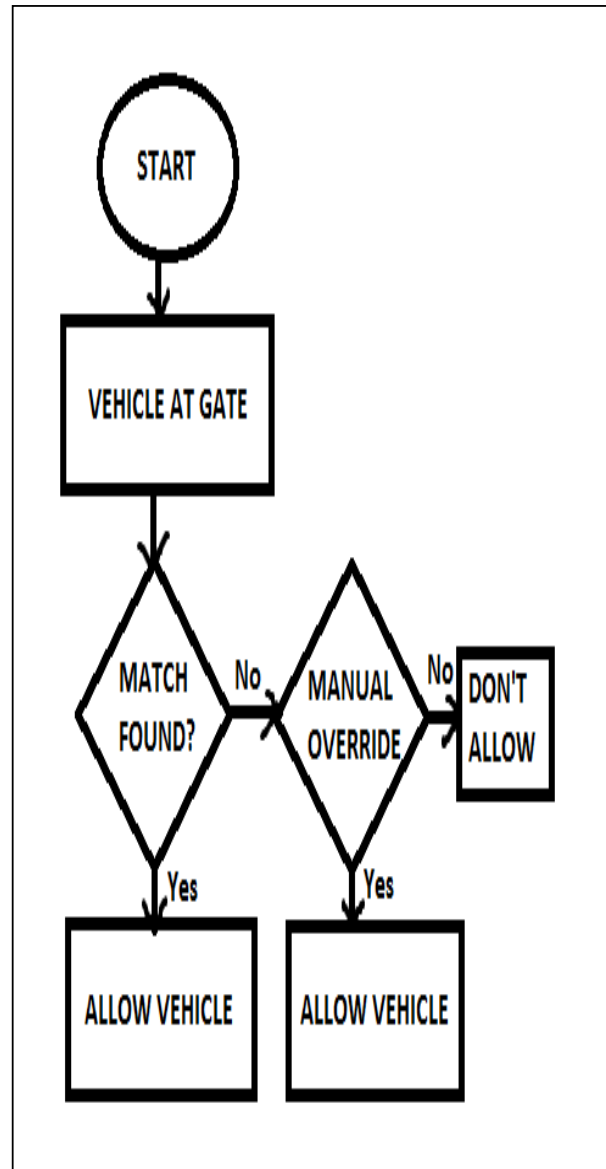


Fig. 2. Flow Diagram to describe process

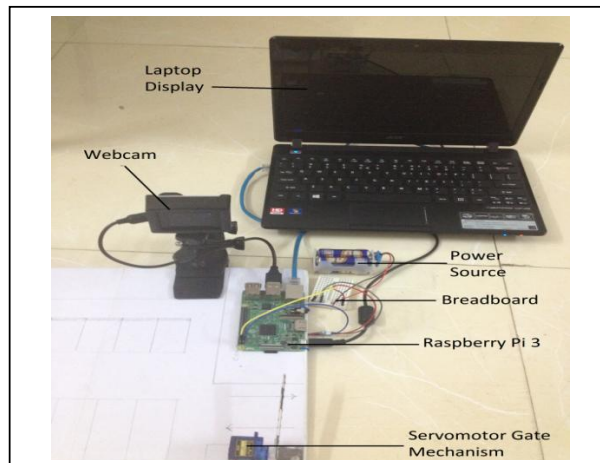


Fig. 3. Actual Experimental Setup



Fig. 4. List of allowed Vehicles



Fig. 5. Sample Car Number Plate

CONCLUSION

So we can finally understand that this proposed system is very efficient and quick too. This system removes the dependencies of using an external RFID tags for independent users or any other kind of identification. This system simply processes the number of each and every car that is in front of the gate and takes a decision accordingly. Very little human intervention is required in this process as mostly every step is automated by the system. Another salient feature of this system is that, since it is not based on any kind of a microcontroller, the Raspberry Pi PC can be running 24x7 without any hitches because it is designed in that way. Also, since we have exported all the details to a web browser to display all the records, it is easy for the person to operate and view all details on the screen easily. He can also search for a specific entry in that itself for cross checking. This entire prototype can almost be directly be implemented for Real-Time scenarios with just minor adjustments.

ACKNOWLEDGEMENT

Firstly, I would like to thank my guide Asst.Prof S.Indhumathi for her help and support throughout the implementation of this prototype. Also, I would like to convey my sincere gratitude to Mr. Ramganapathy for his constant support and encouragement for my prototype.

REFERENCES

- [1] Google Cloud Vision API Detect OCR Documentation: <https://cloud.google.com/vision>
- [2] Raspberry Pi 3 interfacing with Servomotor : <http://razzpisampler.oreilly.com/ch05.html>
- [3] Python and MySQL courses by GUVI: <https://www.guvi.in/>

