Smart Parking System

Mr. KARAN DADRA, MR. AKASHDEEP SINGH

Abstract
With increase in number of vehicles, finding a proper parking space for an individual is getting difficult. This project deals with the development of an application which recognizes the Vehicle License Plate and aids to manage the parking space. Smart parking system uses Image Processing technique to identify number plate of vehicle. The system provides automatic door closing and opening operation only for authorized vehicles. This system also gives information related to vehicle, parking space and also provides security against theft and robbery. The main objective of this project is to identify authorized vehicles, parking security and space management. The project combines the capabilities of hardware and software into one integrated automatic system. The main component of this system is Raspberry Pi for processing and controlling the whole system operation. Ultrasonic sensor is being used to detect any object availability at the parking door. Once any object is sensed by the sensor, it sends a signal to the Raspberry Pi which activates the pi-camera to capture the image. The system is intelligent system as we have integrated Artificial Intelligence(AI) and AI checks whether the object in front of the gate is a vehicle or not. If the image is of vehicle, the captured image is then processed by raspberry pi and characters are extracted from the vehicle license plate. The characters of the number plate are matched with the cloud database and if a match is found, Raspberry Pi sends signal to the servo motor to open the door for certain interval of time. Same process repeats when a vehicle exits. LCD display provides information on availability of parking space currently for parking. This project creates an organized parking only for authorized vehicle and collect the information on the density of vehicle along with the space available for parking.
Keywords: Image processing, Artificial Intelligence, Cloud Database.

1. Introduction

1.1Background

Cities with large number of people often find difficulty in placing their vehicle in safe place. Major cities with larger number of vehicle, one can barely find a proper place to park his/her vehicle. Moreover, wasting time on finding a place for settling vehicle would not only be time consuming but also increases the chances of theft and robbery. With the evolution of science and technology one may find various techniques for dealing up with these complications related to parking. Despite of all these efforts there are still minor errors one faces which brings huge loss and defect of that system.

Our project attempts to solve some issues with parking space in a typical organization. Our project displays the total parking space available and at the same time allows only the authorized vehicle to enter in the parking lot. This is achieved by using Image Processing technique. The system is intelligent enough to distinguish the object in front of the gate is either a vehicle or not.

1.2 Objective

The main objectives of our project are: to implement Image Processing for segmentations of characters in the number plate of the vehicle, to compare the obtained characters with the cloud database and to allow only the authorized vehicle into the parking lot, to distinguish either the object in front of the entrance gate is a vehicle or not.

2. Material and Methods

2.1 Related Theory

Our project combines the capabilities of hardware and software into one integrated automatic system. The aim of our project is to provide parking management system for vehicle license plates using several algorithms such as Canny Edge Detection, Connected Component Labelling, Thresholding, Hough Transformation, Image Segmentation, OCR etc.

2.2 Image Processing

ANPR uses optical character recognition (OCR) on images taken by cameras. When Dutch vehicle registration plates switched to a different style in 2002, one of the changes made was to the font, introducing small gaps in some letters (such as P and R) to make them more distinct and therefore more legible to such systems. Some license plate arrangements use variations in font sizes and positioning. ANPR systems must be able to

cope with such differences in order to be truly effective. More complicated systems can cope with international variants; though many programs are individually tailored to each country. The cameras used can be existing road-rule enforcement or closed-circuit television cameras, as well as mobile units, which are usually attached to vehicles. Some systems use infrared cameras to take a clearer image of the plates.

The input is an image, various processing occurs in stages, and the output is the possible plate numbers in the image. The pipeline stages occur in the following order as below.

2.2.1. Detection:

The detection phase happens one time for each input image. It uses the LBP algorithm (generally used for face detection) to find possible license plate regions (x,y, width, height). Each of these regions is sent to the later pipeline phases for further processing.

The detection phase is usually the most processing-intensive phase. It can be GPU accelerated to improve performance.

2.2.2. Binarization:

This phase (and all subsequent phases) occurs multiple times once for each possible license plate region. The binarization phase creates multiple binary images for each plate region. The reason multiple binary images are used is to give us the best possible chance of finding all the characters. A single binarized image may miss characters if the image is too dark or too light for example. Binarization uses the Wolf-Jolien method as well as the Sauovola method with various parameters. Each of the binary images are processed in subsequent phases.

2.2.3. Character Analysis:

Character analysis attempts to find character- sized regions in the plate region. It does this by first finding all connected blobs in the license plate region. Then it looks for blobs that are roughly the width and height of a license plate character and have tops/bottoms that are in a straight line with other blobs of similar width/height. This analysis is done multiple times in the region. It starts by looking for small characters, then gradually looks for larger characters. If nothing is found in the region, then the region is thrown out and no further processing takes place. If it finds some potential characters, then the character region is saved and further processing takes place.

2.2.4. Plate Edges:

The next phase is to find the edges of the license plate. Keep in mind that the detection phase is only responsible for identifying a possible region where a license plate may exist. It often is going to provide a region that is a little larger or smaller than the actual plate. The plate edge tries to find the precise top/bottom/left/right edges of the license plate. The first step is to find all of the Hough lines for the license plate region. Platelines.cpp processes the plate image and computes a list of horizontal and vertical lines. It uses this list as well as the character height (computed in Character Analysis) to find the likeliest plate line edges. It uses a number of configurable weights to determine which edge makes the most sense. It will try using a default edge (based on the ideal width/height of the plate) to see if that makes a good match.

2.2.5. De-skew:

Given the plate edges, the de-skew stage remaps the plate region to a standard size and orientation. Ideally this will give us a correctly oriented plate image (no rotation or skew).

2.2.6. Character Segmentation:

The character segmentation phase tries to isolate all the characters that make up the plate image. It uses a vertical histogram to find gaps in the plate characters. This phase also cleans up the character boxes by removing small, disconnected speckles and disqualifying character regions that are not tall enough. It also tries to remove "edge" regions so that the edge of the license plate doesn't inappropriately get classified as a'1'.

2.2.7. Optical Character Recognition (OCR):

Optical character recognition (OCR) is the electronic conversion of handwritten, typewritten, or printed text from still or motion images to machine-encoded text. Common uses include scanning of books for electronic retrieval

2.2.8. Post Processing:

Given a list of all possible OCR characters and confidences, post processing determines the best possible plate letter combinations. It is organized as a top N list. Post processing disqualifies all characters below a particular threshold. It also has a "soft" threshold characters that are below this threshold will still be added to the possible list, but they also add a possible blank character since it's possible that the low confidence character is not really part of the plate.

2.3 Components Requirement

For our projects, the component requirements are divided into two parts. They are software and hardware component.

2.3.1 Hardware Requirements:

1. PI Camera

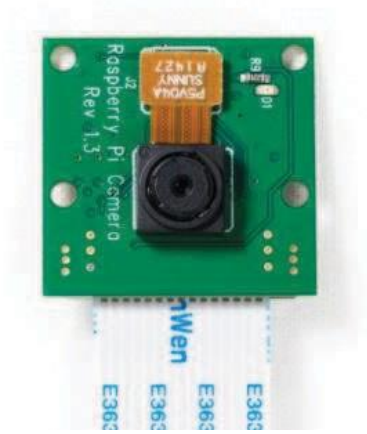


Figure 2.3.1: Pi Camera

Pi cam is the plug-and-play-compatible device, making it perfect for time-lapse photography, recording video, motion detection and security applications. Connect the included ribbon cable to the CSI (Camera Serial Interface) port on your Raspberry Pi, and you are good to go!

The board itself is tiny, at around 25mm x 23mm x 9mm and weighing in at just over 3g, making it perfect for mobile or other applications where size and weight are important. The sensor has a native resolution of 8 megapixels, and has a fixed focus lens on board. In terms of still images, the camera is capable of 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and 640x480p90 video.

2. Ultrasonic Sensor

Figure 2.3.3: Ultrasonic Sensor

HC-SR04 Ultrasonic (US) sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module.

3. Servo Motor

Servo motors are great devices that can turn to a specified position. Usually, they have a servo arm that can turn 180 degrees. Using the Arduino, we can tell a servo to go to a specified position and it will go there. Servo motors were first used in the Remote Control (RC) world, usually to control the steering of RC cars or the

flaps on a RC lane. With time, they found their uses in robotics, automation, and the Arduino world.

Figure 2.3.4: Servo Motor

4. LCD Display:

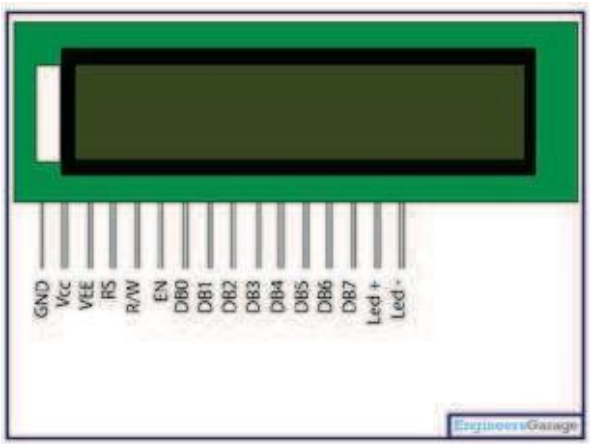


Figure 2.3.5: LCD Display

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. The reasons being: LCDs are

economical; easily programmable; have no limitation of displaying special & even custom characters, animations and so on.A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task

5. Arduino UNO

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

Figure 2.3.6: Arduino

6. Bluetooth Module

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Blue core 04-External single chip Bluetooth system with CMOS technology and with AFH(Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm.

2.4 Software Requirement:

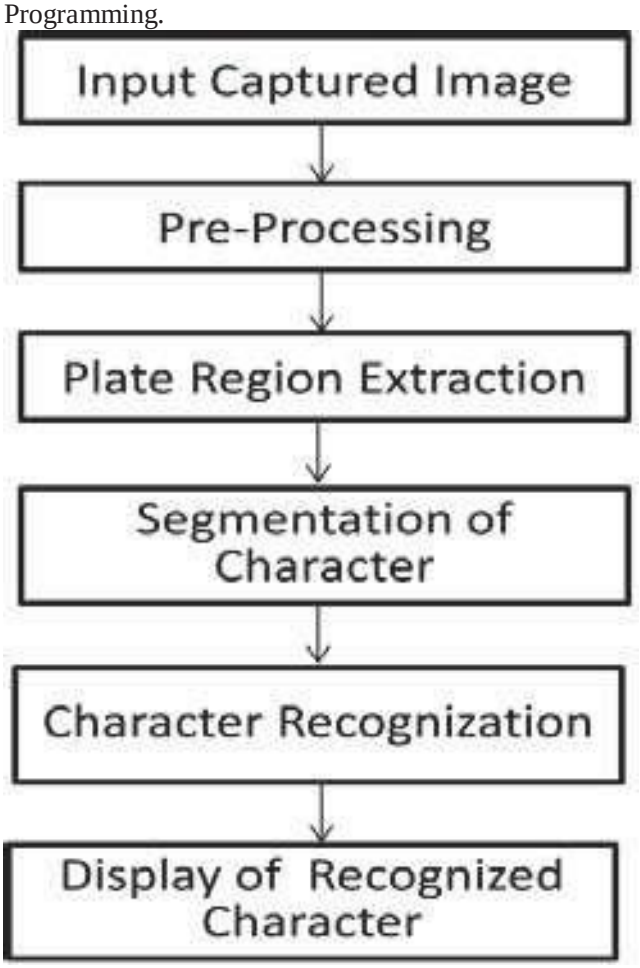
The OS used in the project is the Raspbian OS. Likewise, the main programming language is python. The core of our project is image processing. For this, we have using the library function; OPENCV, which includes several hundred of computer vision algorithms. For data storage and retrieval, SQL is used. For the determination whether the object is vehicle or Something else, AI (Artificial Intelligence) is used.

2.5: METHODOLOGY

The working of full NPR system can be divided in to two broad sections which are the hardware part and the software part. The working mechanism of all the parts is described in details below.

2.5.1 Software Model

The first and the most important part in this process is the software model. The software model uses the image processing technology. The algorithm is divided into following parts: Capture image, Pre-processing, Plate region extraction, Segmentation of character in the extracted number plate, Character recognition, Comparison with database and Indicate result. The flow chart of license plate recognition system implementation in this work is shown in the following figure. There are various steps in this approach and these are implementation in Python Programming.



Workflow Process

Capture of Image: The first step is the capture of image. The image is captured by electronic device Pi Camera or Webcam. The image captured is stored in JPEG format. Later on it is converted in to gray scale image.

Pre-processing: The next step after capturing the image is the pre processing of the image. When the image is captured there is lot of disturbances and noises present in the image for which the image can't be used properly. So in this step the noises from the image are required to be cleared to obtain an accurate result.

a. Gray Processing: This step involves the conversion of image in to Gray levels. Colour images are converted in to Gray image. According to the R, G, B value in the image, it calculates the value of gray value, and obtains the gray image at the same time.

b. Median Filtering: Median filtering is the step to remove the noises from the image. Gray level cannot remove the noises. So to make image free from noise media filtering.

Plate region extraction: The most important stage is the extraction of number plate from eroded image significantly. The extraction can be done by using image segmentation method. There are numerous image segmentation methods available in various literatures. In most of the methods image binarization is used.

Character segmentation: In this step get the o/p of extracted number plate using labelling components, and then separate each character and split the each and every character in the number plate image by using split and also find the length of the number plate, then find the correlation and database if both the value is same means it will generate the value 0-9 and A - Z, and finally convert the value to string and display it in edit box, and also store the character in some text file in this code. Following figure shows the segmented characters

The character recognition is now used to compare the each individual character with the character stored in the database. OCR uses the correlation method to match the characters. And if both the character matches then it displays the authorized otherwise it will display the unauthorized.

2.6 Hardware Model

We used different components for hardware model such as Raspberry Pi, sensors, pi camera, servo motor and others. We used the Raspberry Pi as a main component for controlling all the hardware components and sensors.

2.6.1 Workflow Process

The main component of this system is Raspberry Pi which processes and controls the whole system. Ultrasonic sensor is being used to detect any object availability at the parking entrance. If object is found, a check is carried out (by capturing the image of the object) to determine the object in front of the entrance gate, is either a vehicle or something else. If the check returns the object as vehicle, then the subject goes under further processing. Raspberry PI which is responsible for controlling the overall operations of thesystem. It is responsible for performing image processing and AI. It is mini pc where we can load OS and can perform task accordingly.

The ultrasonic sensor is a small four pin device and is used to detect object near the gate.
As the name suggests, Pi Camera is used to capture the image of object (can be vehicle or something else). The LCD display is an electronic display module which is used to show the availability of parking space. The for opening and closing of the gate.
The database is used to store the vehicle credentials.

The captured image of vehicle is processed using Image Processing technique in which the characters of the vehicle are segmented. The characters of the number plate is matched with the online database and if the match is found, Raspberry Pi sends signal to the servo motor and the door opens for 10 second and the counter is decreased by 1.
Similarly, while existing from the parking again the Ultrasonic sensor inside detects the object and door opens for 10 seconds and again closed. And hence the counter is increased by 1.

Figure 2.5.2: Sample Number

2.7 Flowchart of the proposed system

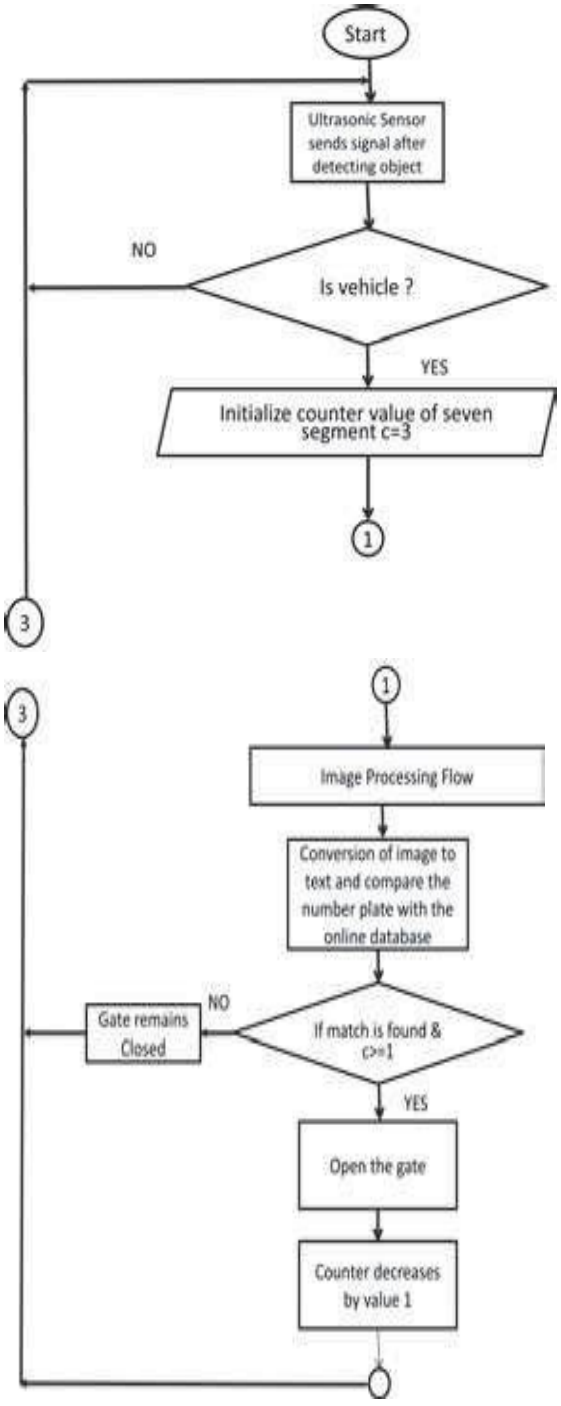
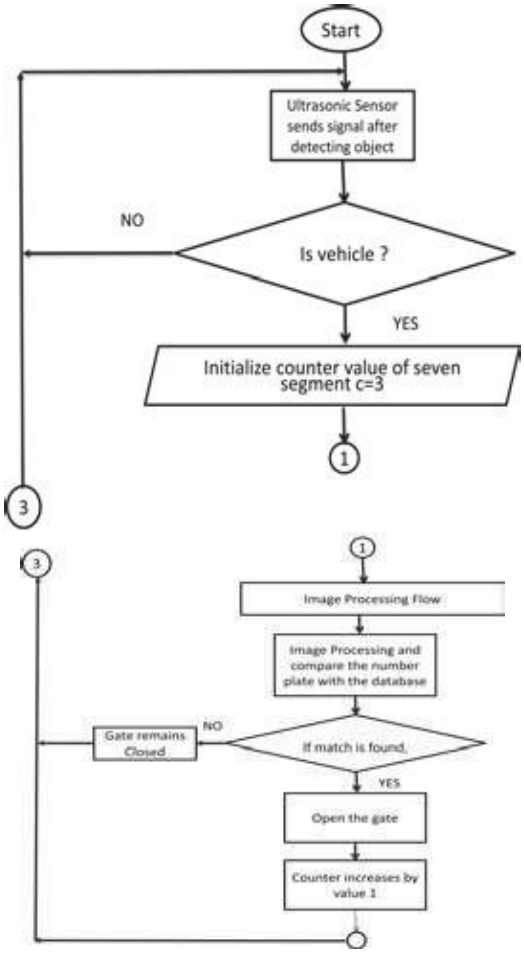


Figure 2.8.1: Vehicle Entrance Process



Vehicle Exit Process
Figure 2.8.2: Vehicle exit process:

2.1 Design of Parking System

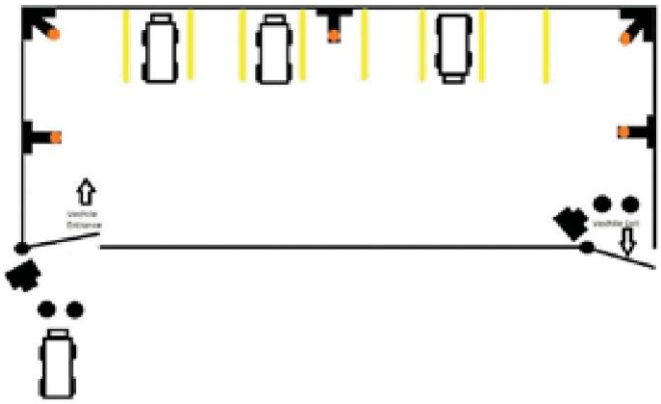


Figure 2.9: Parking Floor Plan

3. Result

Image processing was performed well with the help of library function, OPENCV. The output of image processing is the characters of license plate which is further compared with the database. We are able to pass the authorized vehicle if the characters obtained from image processing matched with the online database.

3. Discussion

Achieving the goal of our project was not an easy job. Several factors make our task more and more difficult. At first, all the devices listed should be in proper condition. Secondly, for AI and image processing the position of camera should be properly determined. It is better to use high resolution camera. The software dependencies are also major factor and hence one should use the correct version of the software. Also, proper voltage should be supplied to the devices else it will not work or might get damaged.

4. Conclusions

With the completion of this project, we are able to implement our ideas of recognizing the authorized vehicles and allowing them to enter into the parking space. We have concluded that this project helps in creating an organized parking space for authorized vehicles with the information of date and time the vehicle entered and exits the parking space.