نظام مرأب السيارات يعمل بتقنيه انترنت الاشياء باستخدام الراسبري باي مع تعلم الاله

IoT Car Parking System Using Raspberry pi with Machine Learning

Ahmed Qasim A. , Dr. Haider M. Al-Mashhadi

Information System Dept., College of Information Technology, University of Basrah.

ABSTRACT

Parking system is one of the most popular systems around the world. When we study this system we found that there is many problems like the crowded parks, the dependence on workers, delaying which happened by humans, and etc. So, in this project we will try to solve these problems through using the below procedure: Opening the Entrance gate through checking the car number either reserved in DB or not, if it was recorded so we won't deal with this car as a guest the pi is going to say "welcome" by the LCD monitor, and open the gate immediately because it is already has a dedicated parking place. After that we should record the car number and entrance date into the database. If the car number wasn't recorded in the DB as a reserved number the pi will check if there is any empty slot to park this car so, if there is an empty slot the pi going to show the number of free parking slots on the LCD monitor, open the entrance gate, and write the car information and entrance date into the database else the entrance gate won't be opened!. The system using machine learning to recognize the car plate number and process the data according it to allow open the gate or not.

Keywords: rasspbeerry Pi, Internet of Things IoT, Cloud Computing, Machine Learning.

1.Introduction:

Searching for a parking slot has become a daily habit for a lot of people all around the globe. It is usually a time consuming and frustrating process that a lot of citizens have to go through. Next to parking being such a daily pain, it also has a big impact on the pollution of the planet.

After naming the various problems that arise from looking for parking spots, only one question remains: How can this problem be fixed? The answer is Smart Parking, but what is it?

The Definition of Smart Parking

In a nutshell, Smart Parking is a parking solution that can include in-ground Smart Parking sensors, cameras or counting sensors. These devices are usually embedded into parking spots or positioned next to them to detect whether parking bays are free or occupied. This happens through real-time data collection. The data is then transmitted to a smart parking mobile application or website, which communicates the availability to its users. Some companies also offer other in-app information, such as parking prices and locations. This gives you the possibility to explore every parking option available to you.

Smart Parking and its Smart Parking Sensors can be seen as a part of smart cities. These smart cities are cities that are driven by an IT infrastructure and by using this infrastructure, cities can enhance the quality of life and improve economic development for its inhabitants. Becoming a smart city can be a good way to collect historical data in a relatively easy way. By collecting this data, cities can analyze how processes, like parking can be optimized.

As a result of using Smart Parking, people who are looking to find a parking spot will find it in the most efficient way possible and companies or municipalities can optimize their parking territories. It also makes cities more livable, safer and less congested.

Advantages of Smart Parking for Drivers

- **Optimizing the driving experience:**

using a Smart Parking system saves a lot of time for drivers since they know where to find a vacant parking spot. The amount of time you spend while looking for a parking spot will be minimized. By using the Parkeagle technology of the Smart Parking sensors, you will be able to find the parking spot you are looking for, without having to browse through the streets.

Advantages of Smart Parking for Cities

- **Less pollution**: Smart Parking contributes to a cleaner environment. Reducing the time that is necessary to find a parking spot will reduce the amount of fuel that is used when looking for a parking space. This makes the process of finding a parking spot contribute to less pollution, which is beneficial for everyone.

The space of a municipality will be utilized more efficiently: because Smart Parking sensors transmit live-data, drivers will have a real-time overview of the occupancy of parking bays. This means that free spots can be filled quicker, which will reduce the time that a parking spot is empty.

Safety: The use of Smart Parking Sensors can optimize safety within cities. As a result of placing, for instance, on-ground sensors on parking bays, people will not be as stressed as when they are looking for parking spaces. Because these people will know where they are going, they can simply navigate to their parking spot and they will not have to stress out about it.

Real-time parking analytics for cities: Parking space will become intelligent by use of the smart parking sensors on the parking bays. This means that as a city you're able to see historical data which is stored and you're able to make data driven decision and predictions based on the parking sensor data.

In general, Smart Parking solutions, such as sensors, give municipalities and companies the opportunity to make parking a more fluid and efficient process. Furthermore, it saves people a great amount of time, money, and reduces the frustration that a person might have when wanting to find a parking slot.

2. Statement of the Problem :

As the money flow increases, and more and more individuals and companies expand their ownership of vehicles, the complexities and conflicts of parking swells. While cooperation and coordination are crucial for a smooth functioning of the parking lot areas, there are a number of things that the parking lot owners could do to speed up their processes. Parking management solutions address parking problems and fully automate parking operations.

In the cities, the supply-demand ratio leads to some deep parking issues for the parking area providers. From tracking, processing, checking to management everything seems to be messed up! Before moving to parking lot problem solving, let's first read 5 major problems faced in parking lot management. Further, we will move forward to suggesting innovative solution to vehicle parking.

1- Manual Checks: Parking managers perform manually intensive work of counting permit and non-permit cars. There is manual checking of vehicle status and details and handwritten tickets. Such a manual procedure leads to 5% entry errors, further resulting in huge losses to the bottom line.

2- Paper Records: It is difficult to sieve through the large volumes of information. For accomplishing this task, the parking lot managers have to spend hours searching files for the exact information. So, these paper records create a lot of problems.

3- High Labor Costs: Reading, writing and entering data is labor-intensive and time consuming. Unnecessary capital expenditure is increased due to the money spent on labor that performs repetitive manual tasks.

4- Waiting Customers: The outdated mode of management troubles the customers and makes them wait in long queues when they need to enter and exit the parking lot. Due to this, precious time of the clients is wasted, and their sustainability gets shaken.

5- Unauthorized Access: The parking manager in-charge issue handwritten paper tickets that can be duplicated easily. No security alerts are raised to the authorized personnel if any unauthorized vehicle enters the parking lot.

3.Project Goals:

The main aim of this project is enhancing the passing, traffic, and management operations that failed or complicated because of the manual way of parkng cars inside garags comparing to the modern progress.

And the most important goals are:

1- Replacing the current parking systems with the ANPR (Automatic number plate recognition) system.
2- Devloping a desktop application to managing the anpr system and providing the ability to the instant access to DB and the other information about the cars.
3- Devloping an online application to providing an online reservations.

The tools that used:

We used micro and mini ships will mention them below;

- Microcontroller (Raspberry pi).
- Raspberry pi camera.
- Ubuntu mate OS.
- IR sensors.
- Leds and resistors.
- Power supply.
- Servo motors.
- LCD monitor 16x2.
- SQL DBMS.
- Opencv Library.
- Python OOP and AI Recognition algorithms.

4.Raspberry pi:

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote teaching of basic computer science in schools and in developing countries.

The Raspberry Pi is one of the best-selling British computers. As of December 2019, more than thirty million boards have been sold. Most Pis are made in a Sony factory in Pencoed, Wales, while others are made in China and Japan.

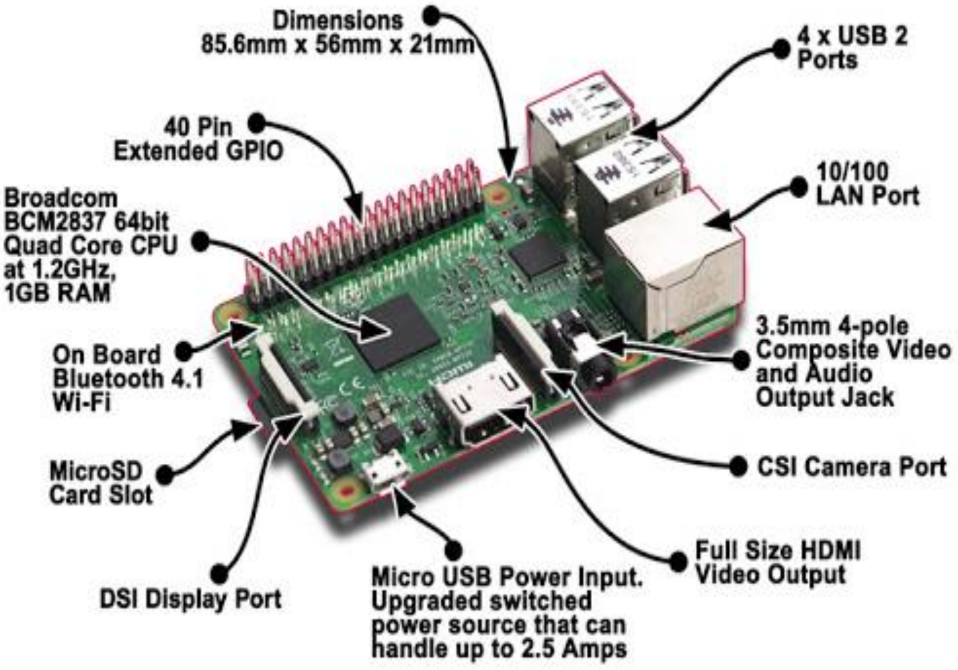

Dimensions
85.6mm x 56mm x 21mm
4 x USB 2 Ports
40 Pin Extended GPIO
10/100 LAN Port
Broadcom BCM2837 64bit Quad Core CPU at 1.2GHz, 1GB RAM
3.5mm 4-pole Composite Video and Audio Output Jack
On Board Bluetooth 4.1 Wi-Fi
CSI Camera Port
MicroSD Card Slot
Full Size HDMI Video Output
DSI Display Port
Micro USB Power Input. Upgraded switched power source that can handle up to 2.5 Amps

Figure (1) Raspberry pi

5.python programing language:

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system with reference counting.

Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

The Python 2 language was officially discontinued in 2020 (first planned for 2015), and "Python 2.7.18 is the last Python 2.7 release and therefore the last Python 2 release." No more security patches or other improvements will be released for it. With Python 2's end-of-life, only Python 3.5.x and later are supported.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

6. OpenCV:

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV's deployed uses span the range from stitching street view images together, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

7. Systematic analysis

The parking process is a really important process in many places like cinemas, theaters, super markets, private companies, private schools, and universities, etc.

Our smart garage is designed depending on algorithm that we made it depending on algorithms from previous studies and through the current needs for smart international car parks.

The system consist of six stages:

1- The car entering process.
2- The plate recognition.
3- Comparing the car number with DB.
4- Inserting car number and date to the DB.
5- Closing the entrance gate.
6- The car exiting process.

The car entering process:

The car needs two things to enter the garage, firstly it needs to be near from the entrance sensor, and firstly it needs to be either reserved car (already have a parking slot inside the garage) or there is an empty slot inside the garage.

After specifying the car type (guest or reserved car) through the recognition process, the LCD screen should print “Welcome” and open the entrance gate.

The plate recognition:

We will use an anpr algorithm called “KNN”. It is an AI algorithm depending on deep learning and neural networks and has many sub-stages to get the best recognition on the car number.

Comparing the car number with DB:

After the camera recognition process we should compare the car number with our database to know if the number is reserved or not. If it was reserved number so the system will open the entrance gate automatically. But if not, (guest car) we should use a function to know if there is any empty parking slot or not to opening the gates.

Inserting car number and date to the DB:

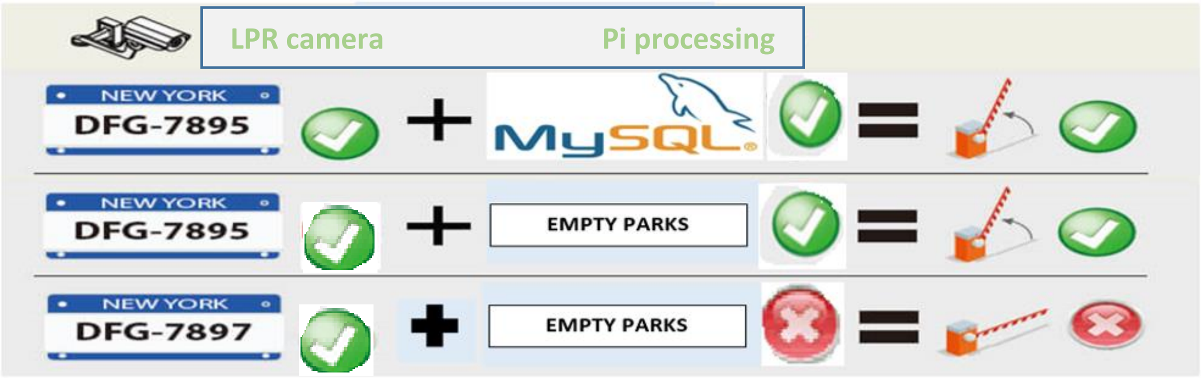
After the car is entered to the park we should insert these information to the database. We will just insert the car number and the entering date.

Closing the entrance gate:

After inserting the car information to the database, now we should close the entrance gate before that anybody could be able to enter without permission and information recording. So, the best solution to this problem is to put an IR sensor after the entrance gate. This sensor duty is to calling a function that is responsible to close this gate.

The car exit process:

This process does not cost a lot. We should just put another IR sensor next to the exit gate where it will call a two functions. The first will open the exit gate and have a sleep for 2 seconds, and the second will just close the door and print “Good by” on LCD screen.



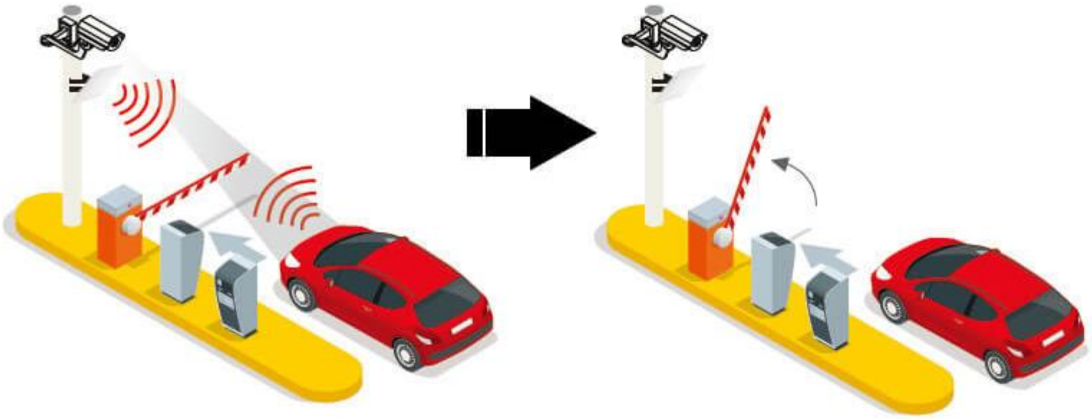


Figure (3) systematic aspect

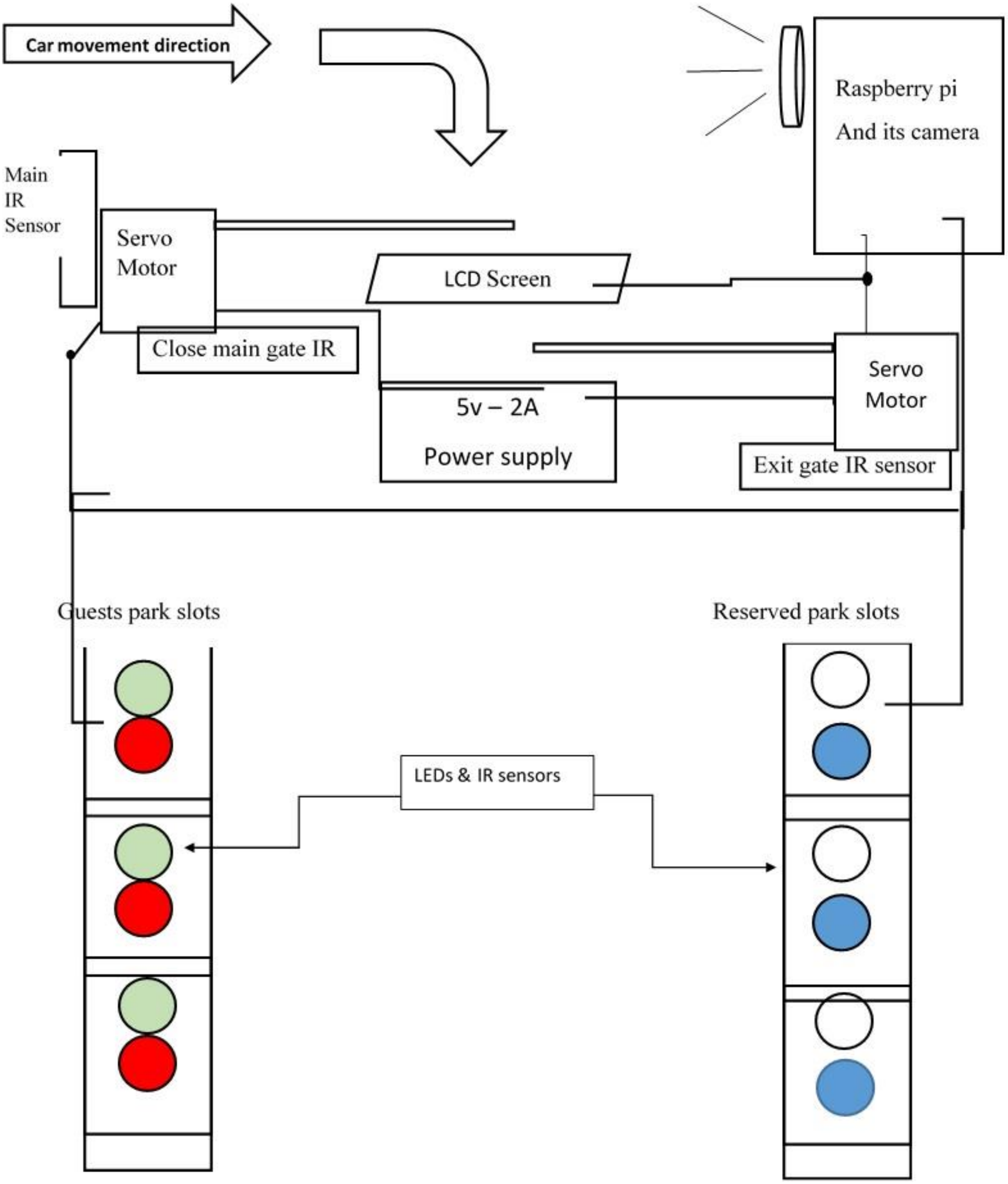


Figure (3) General aespct for the system

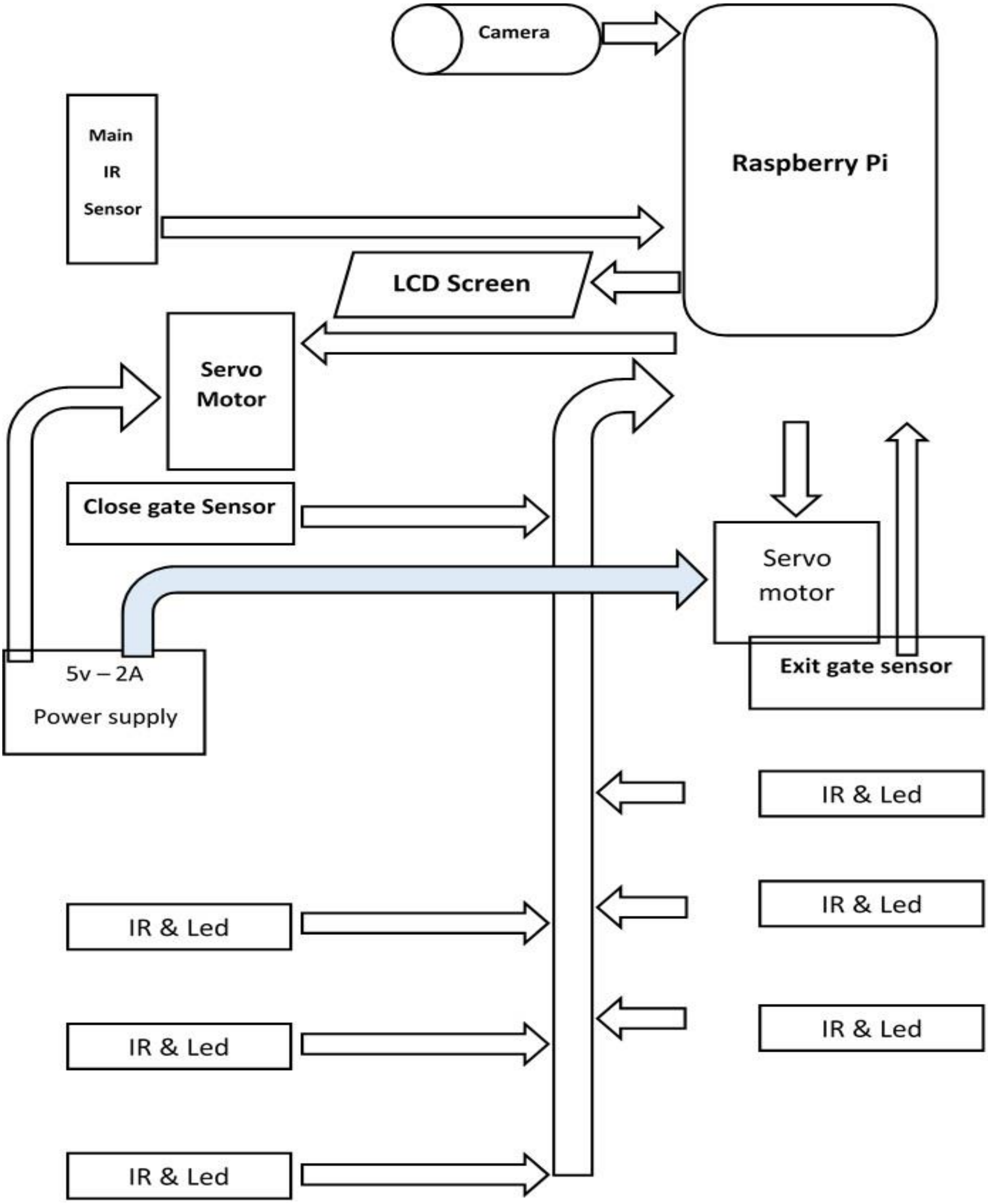


Figure (4) System block diagram

System algorithm:

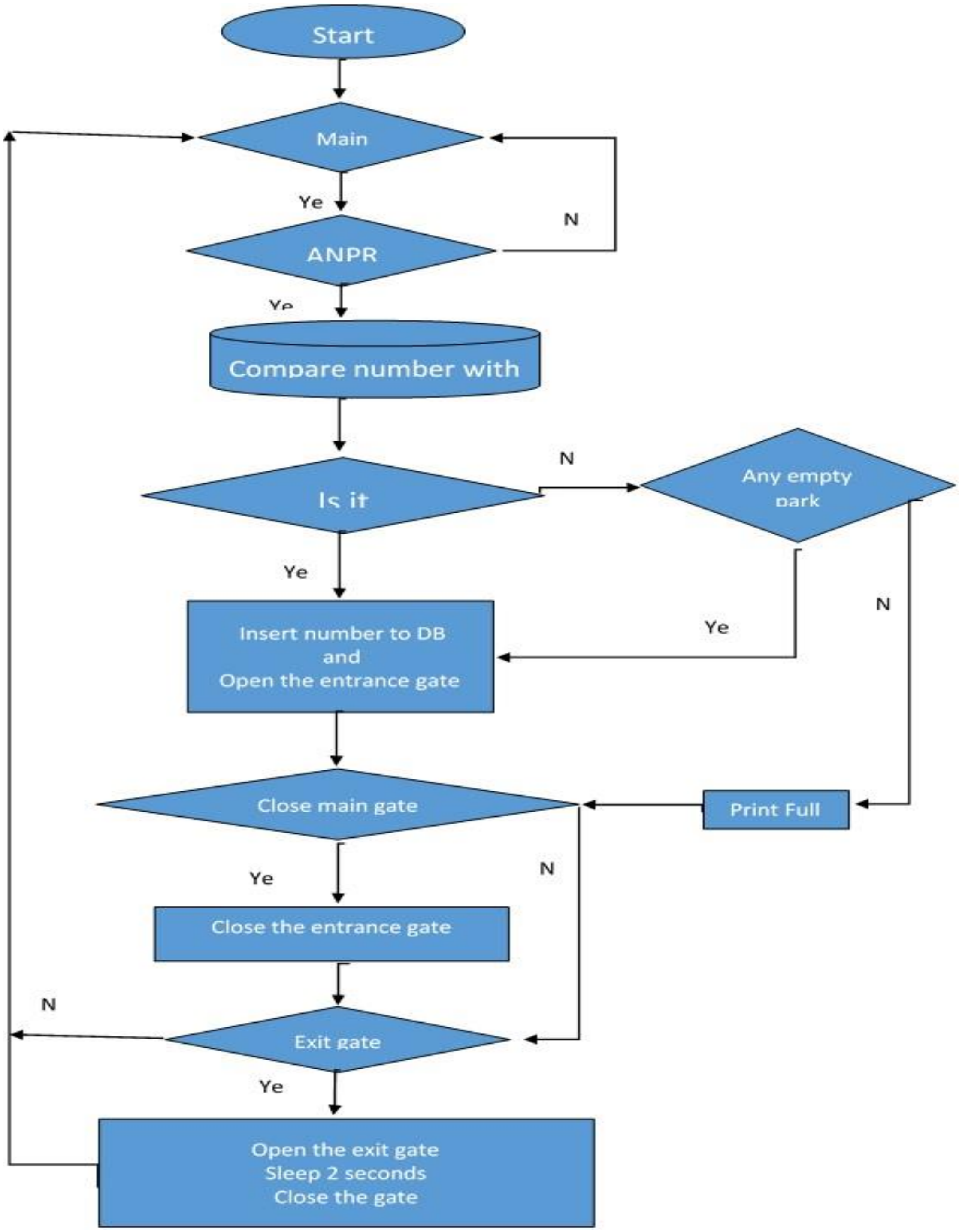
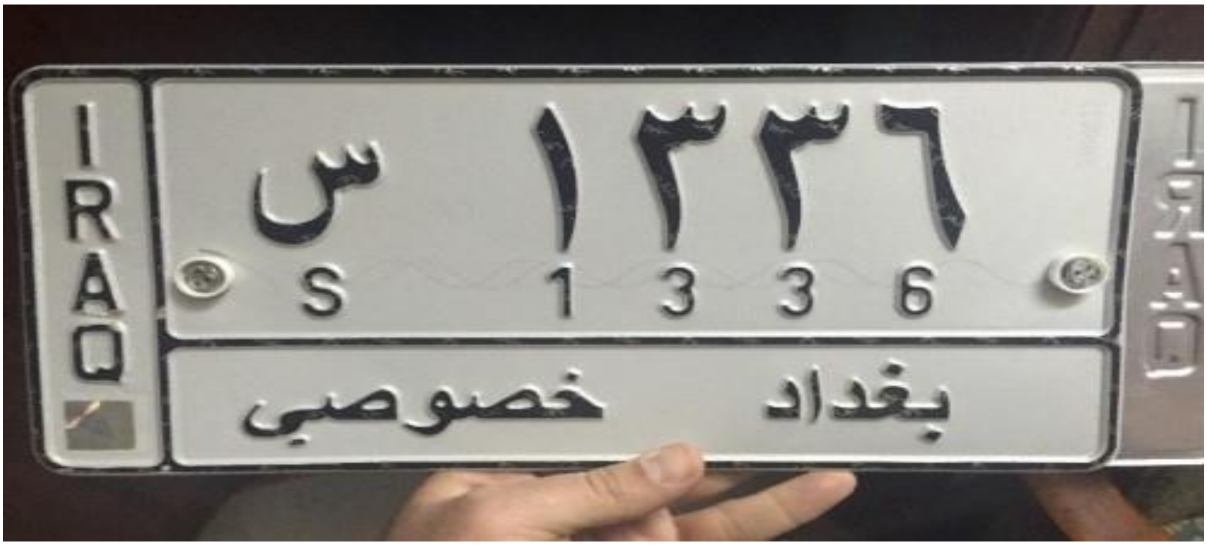


Figure (5) System Algorthim

8.The used ANPR algorithm:

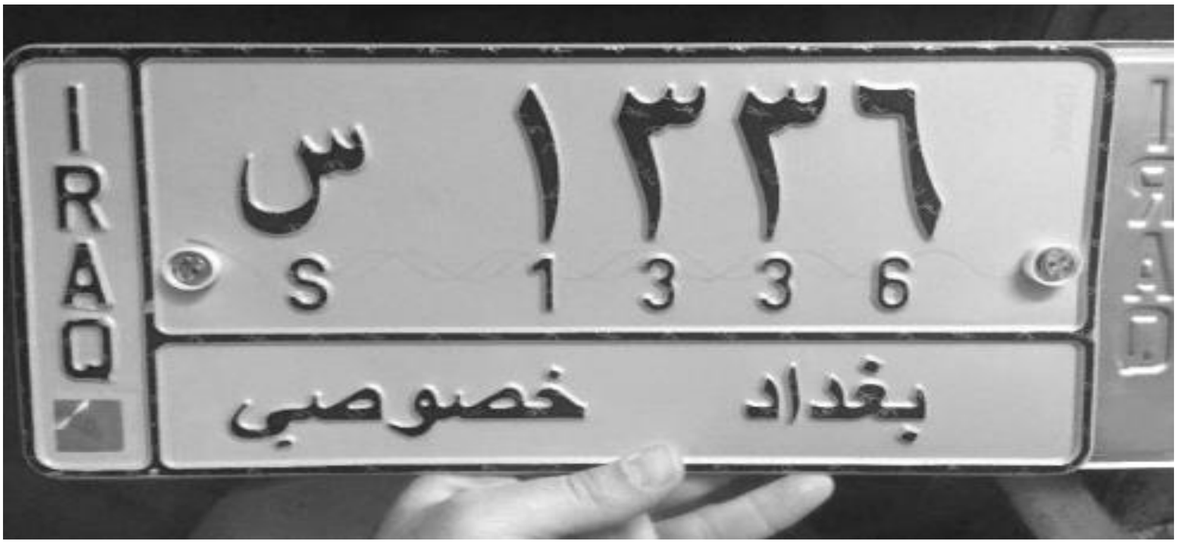
In fact, we will use a simple algorithm that take a capture firstly, then detecting the numbers and finally extracting the founded or possible characters.

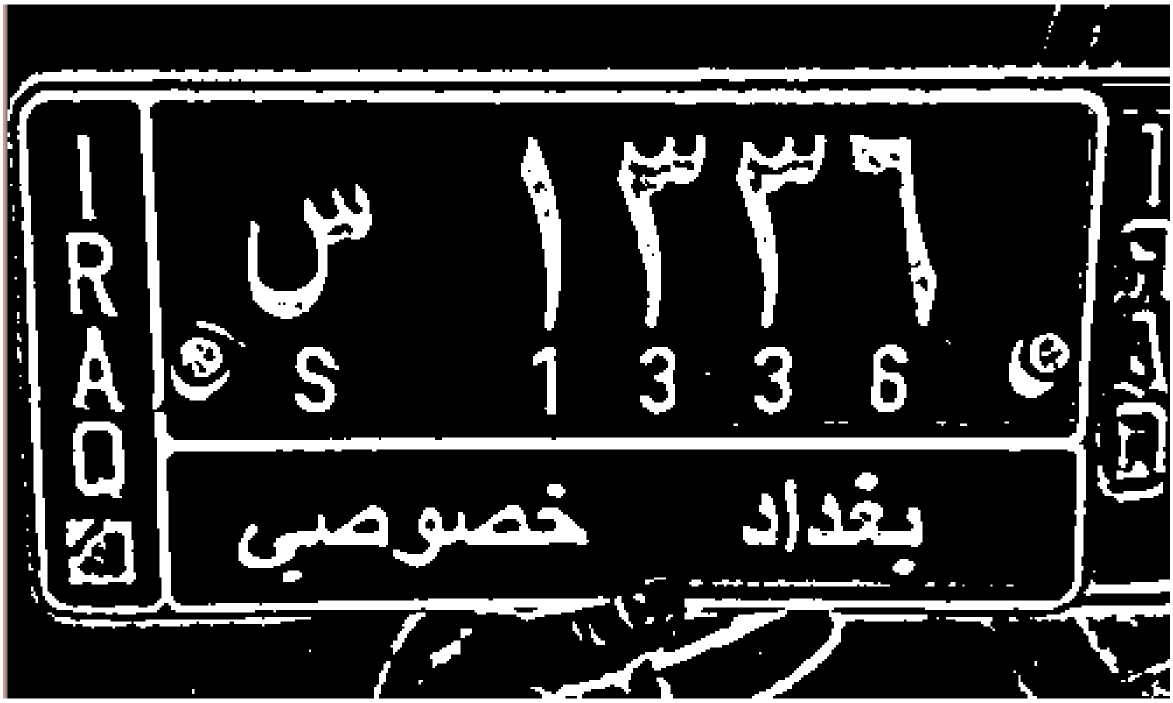
imgOriginalScene



preprocess()

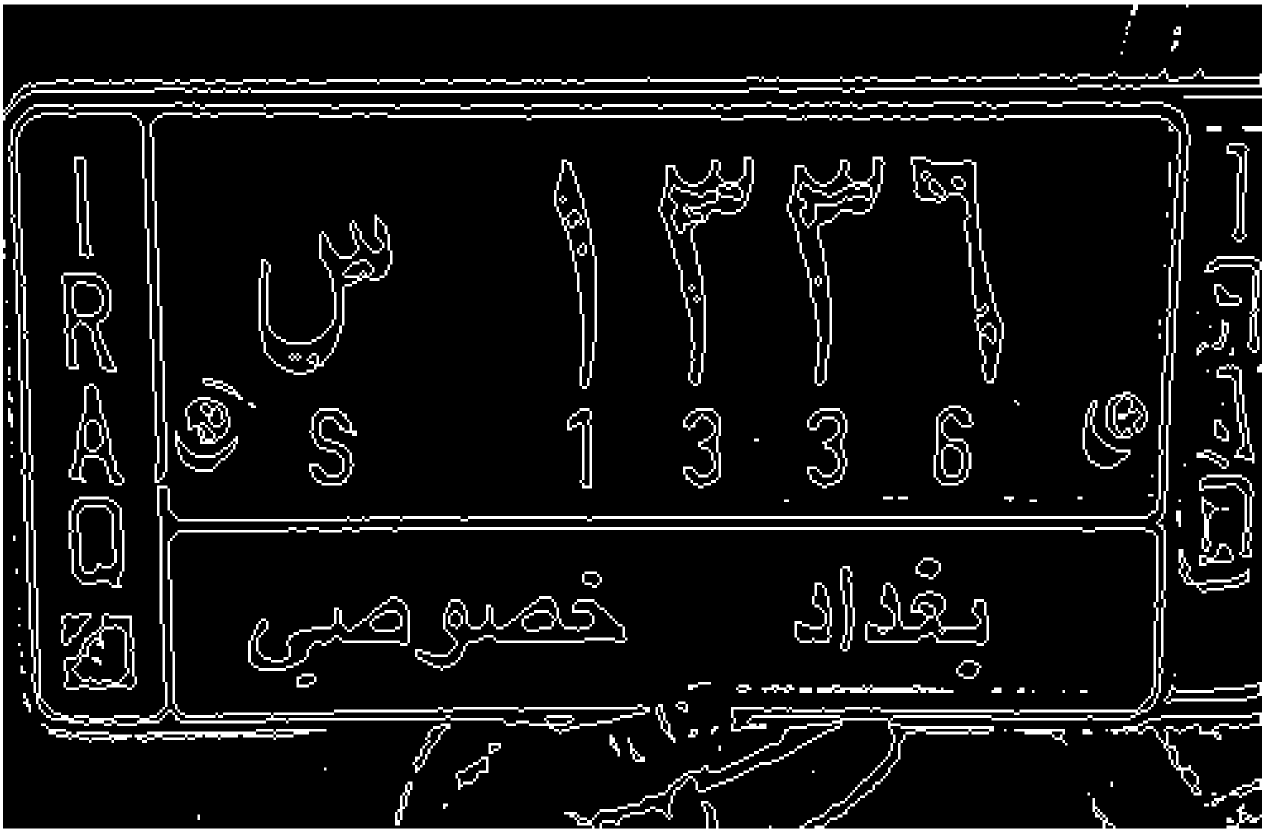
imgGrayscaleScene, imgThreshScene



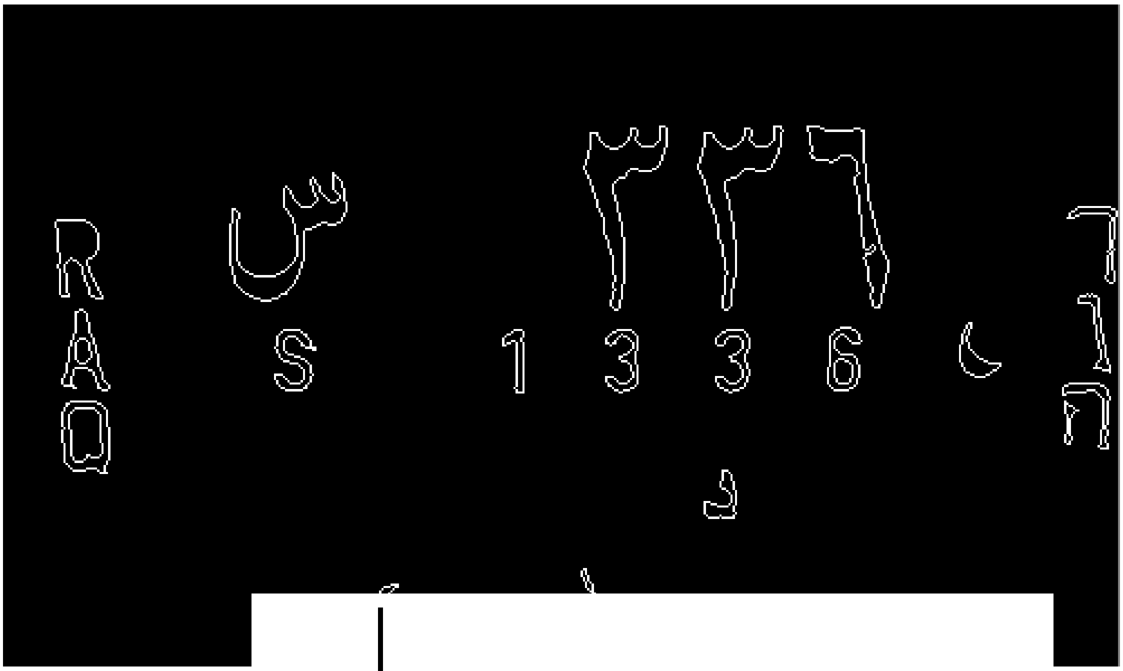


findPossibleCharsInScene()

all contours

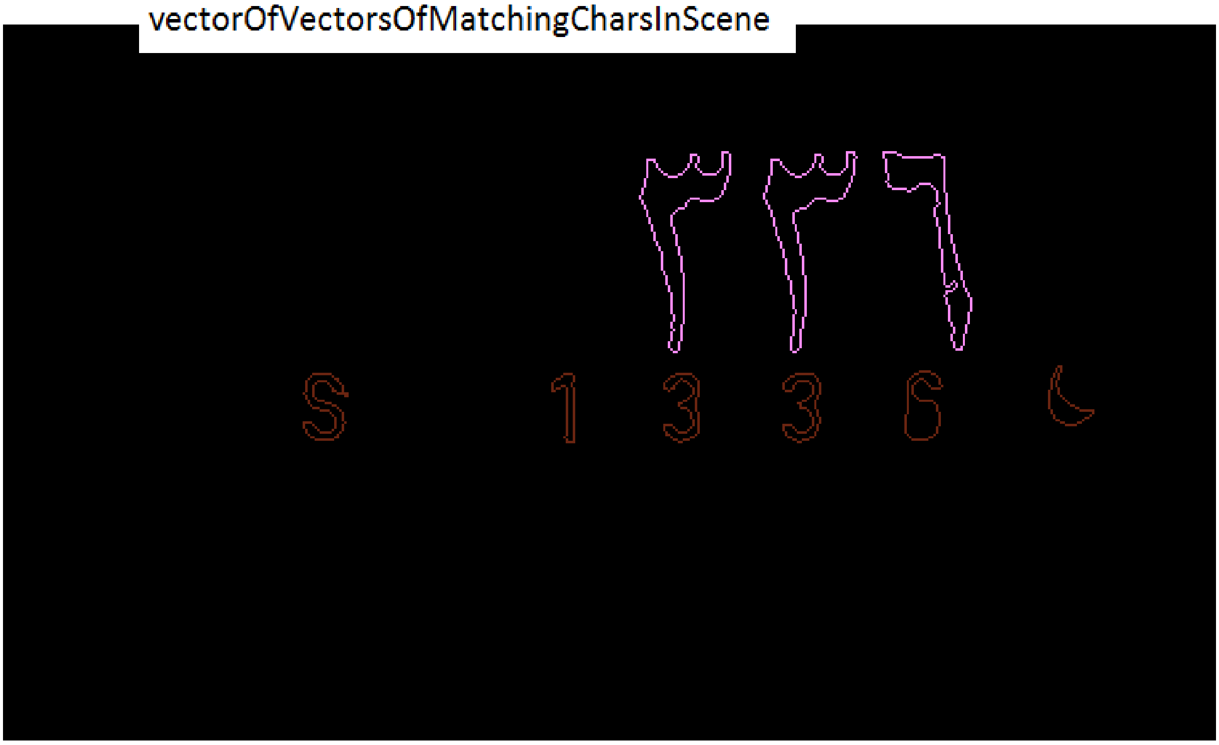


vectorOfPossibleCharsInScene



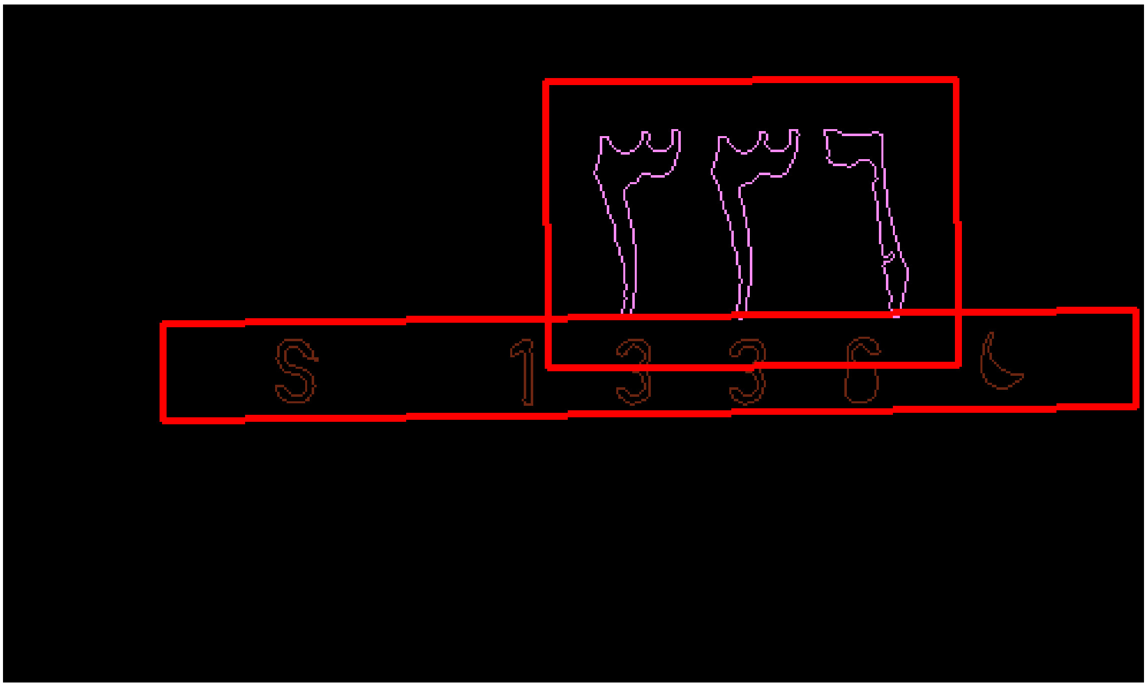
findVectorOfVectorsOfMatchingChars()

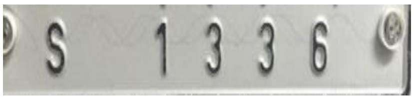
vectorOfVectorsOfMatchingCharsInScene

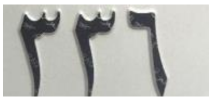


extractPlate()

vectorOfPossiblePlates

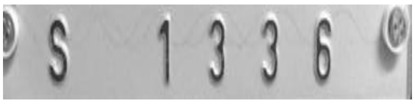


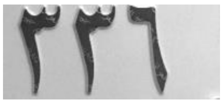




preprocess()

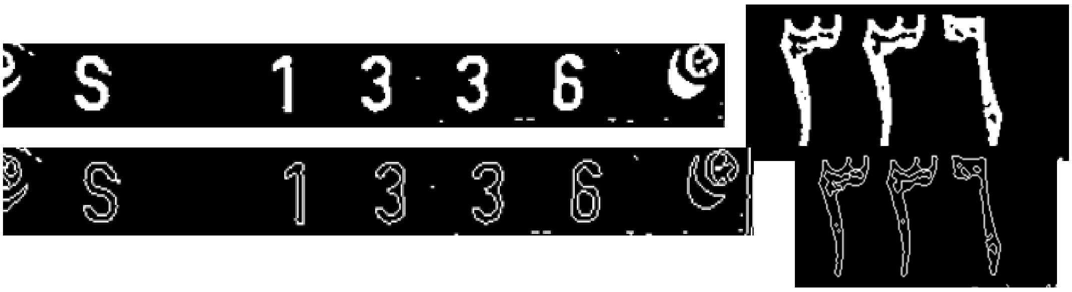
imgGrayscale, imgThresh





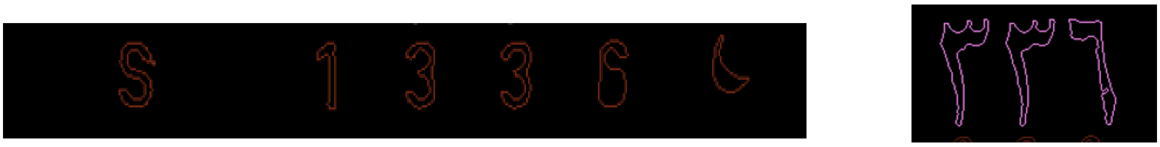
findPossibleCharsInPlate()

vectorOfPossibleCharsInPlate



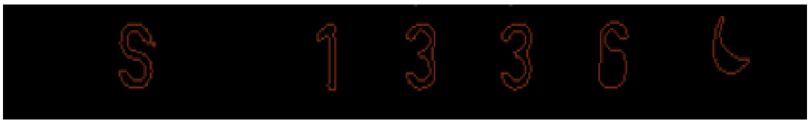
findVectorOfVectorsOfMatchingChars()

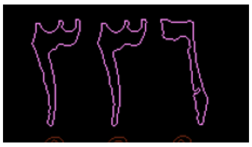
vectorOfVectorsOfMatchingCharsInPlate



removeInnerOverlappingChars()

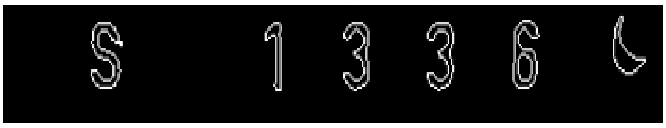
vectorOfVectorsOfMatchingCharsInPlate

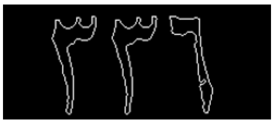




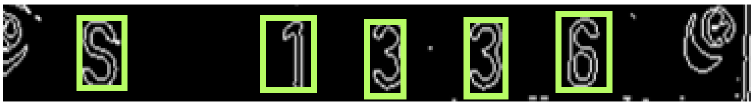
within each possible plate, suppose the longest list of potential matching chars is the actual list of chars

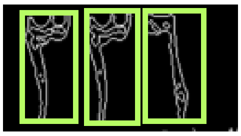
longestVectorOfMatchingCharsInPlate





recognizeCharsInPlate()

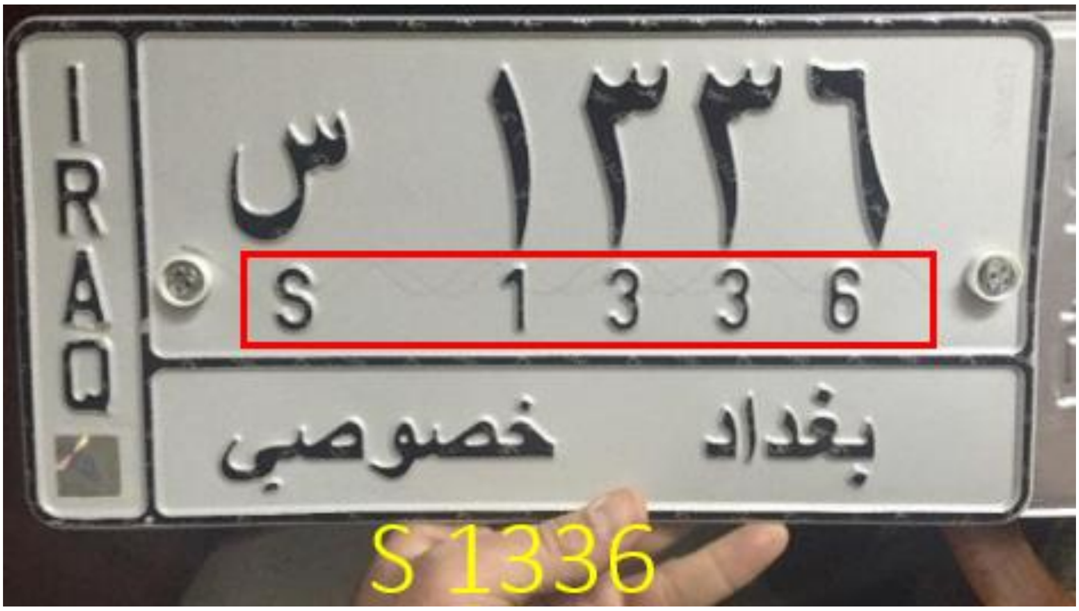




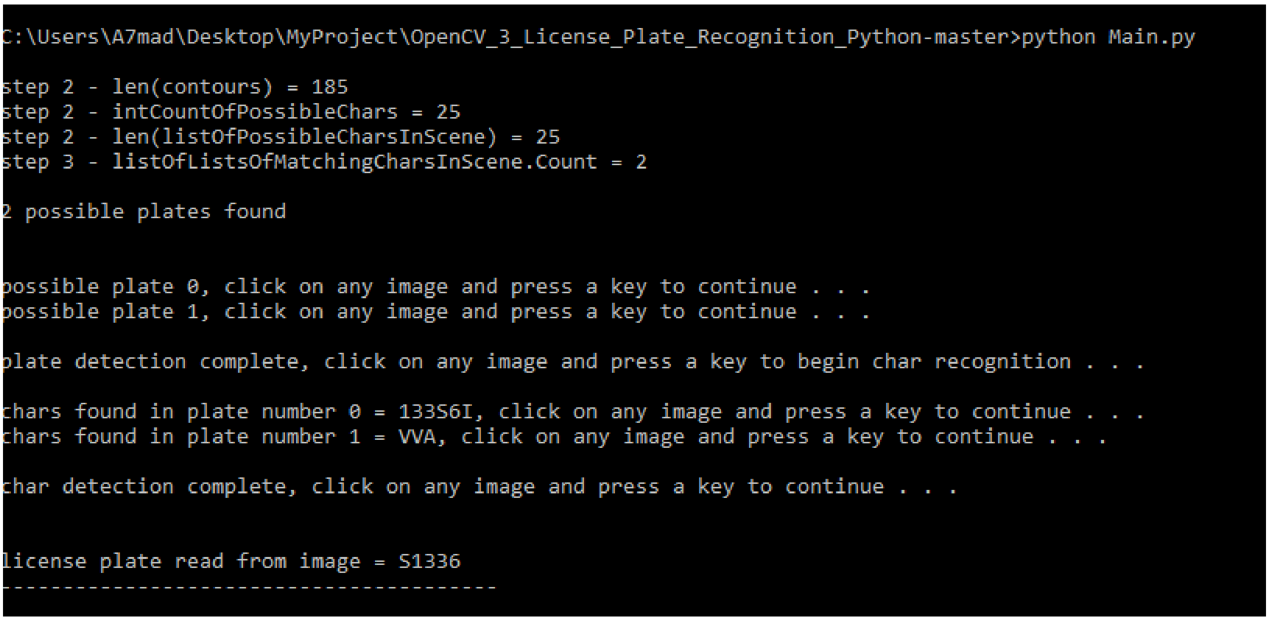
Chars found in plate number 0=S 1336 ,   chars found in plate number 5=336

possiblePlate.strChars

suppose the plate with
the most recognized
chars is the actual plate



After writing and running the code of this algorithm inside the pi using python, it will be able to extract the car number after pointing the camera towards car as shown in the figure below.



```
C:\Users\A7mad\Desktop\MyProject\OpenCV_3_License_Plate_Recognition_Python-master>python Main.py

step 2 - len(contours) = 185
step 2 - intCountOfPossibleChars = 25
step 2 - len(listOfPossibleCharsInScene) = 25
step 3 - listOfListsOfMatchingCharsInScene.Count = 2

2 possible plates found


possible plate 0, click on any image and press a key to continue . . .
possible plate 1, click on any image and press a key to continue . . .

plate detection complete, click on any image and press a key to begin char recognition . . .

chars found in plate number 0 = 133S6I, click on any image and press a key to continue . . .
chars found in plate number 1 = VVA, click on any image and press a key to continue . . .

char detection complete, click on any image and press a key to continue . . .


license plate read from image = S1336
----------------------------------------
```

9. Conclusion:

After that the project was done, We concluded that even with the difficulties and many negatives that faced us through using python libraries with raspberry pi that the using operating systems of the raspberry and the capabilities, needs more developing to be able to deal with python language the most language wich used with these types of microcontrollers.

But, in fact the python is the best language that used to deal with artificial intelligent and its libraries. as like as the rapberry is a good microcontroller but it is not the best.

10. The future work of the system:

The system can be developed through:

1- Putting an extra camera and extra gates in front of each a reserved gate to check the car number before opening the customed slot.

2- Changing the pi camera with a better type to get better results(the current camera marked as it works on 1080p but it is not(manufacturers cheats)).

References:

[1] A. Y. (Gus), "Raspberry Pi LCD using a 16×2 Liquid-Crystal Display," (in English), Pi My Life Up, Website Sep 19, 2016 2016. [Online]. Available: https://pimylifeup.com/raspberry-pi-lcd-16x2/

[2] C. Dahms, "OpenCV 3 License Plate Recognition Python full source code," (in English), GitHub, 9 Jan 2016 2016. [Online]. Available: https://github.com/MicrocontrollersAndMore/OpenCV_3_License_Plate_Recognition_Python

[3] https://ubuntu-mate.org/download/

[4] https://tutorials-raspberrypi.com/raspberry-pi-servo-motor-control/

[5] https://circuitdigest.com/microcontroller-projects/raspberry-pi-ir-sensor-tutorial

[6] https://www.python.org/downloads/

[7] http://mmv.boku.ac.at/refbase/files/mmv6_234_235.pdf

[8] https://www.trakaid.com/5-parking-lot-management-problems-solutions/