Self-Parking Car

Using obstacle detection, image processing and machine learning

Team- KLUDGE (C2/A/26)
Anukriti Singh
Divya Priayadarshni
Aditi Chandra
B.Tech 2nd Year
Bajaj School of Automation
Banasthali University
Jaipur, India
anukriti.runjhun@gmail.com

***Abstract*— The purpose of the project is to build an autonomous self-parking car using Raspberry pi as a micro-processor. Infrared Sensors are placed along with the USB camera to feed the data of the real world to the car and avoid the obstacles in its way. Further, the data is used for learning and analyzing the pattern for making the car smarter with data handling and machine learning. Hence, the combined application enhances the car to park safely.**

***Index Terms*—** Raspberry PI, Obstacle Detection, Image processing, Machine Learning

1 INTRODUCTION

With increasing population, demand of cars is increasing. Car saves time in a long route but when trying to get the errands done, parking space issues are always present which contrarily uses more time. Traffic and complicated road systems make human errors one of the top reasons for accidents happening in daily life. Thus, it has become inevitable to aid the mankind.

To reduce all the above complexities, the optimum solution is to fortify the machine with intelligence. If the car has the data about the roads and type of cars, it will not only evade but also overcome the unfortunate accidents.

2 HARDWARE FIXTURES

2.1 PREREQUISITE

- Raspberry pi 3 Model B Vi.2
- Infrared Sensors
- L293D IC
- Two Servo Motors
- Batteries for power supply
- Jumper Wires (male to male, male to female, female to female)
- Chaises
- Logitech C510 Webcam

2.2 DELINEATION OF HARDWARE AND SOFTWARE

2.2.1 Raspberry Pi

Raspberry Pi is a microprocessor with 1GB RAM, built-in Wi-Fi and Bluetooth, Display Serial Interface (DSI) and Camera Serial Interface (CSI). It has 40 GPIO headers out of which VCC and GND are separately specified. It acts as an interface hub for embedded circuits and compilers for the Raspbian Operating System.

2.2.2 Infrared Sensors

The proximity sensors work on the principle of transmitting infrared radiations and receiving the radiations from the obstacle after reflection. IR Sensors serve to be the basic edifice of automation by detecting the obstacles.

2.2.3 L293D IC

L293D is a H-bridge driver circuit with in-built current amplifiers. This is used to rotate the motor in either of the directions by following the X-OR logic.

2.2.4 Logitech C510 Webcam

It is a Universal Serial Bus camera which is connected to the Raspberry Pi Port. The pro camera module can be used to take high-definition images and videos with easy to integrate feature.

2.2.5 RPi.GPIO Python Library

The RPi.GPIO Python Library is an easily importable library used to setup the input and output headers of the RPi, enabling the read-write configuration within the python script.

2.2.6 Open Source Computer Vision (OpenCV)

OpenCV is a cross-platform library of programming functions, serving an amalgamation between real-time

versus digital time. CV has comprehensive and efficient set of optimized algorithms for processing and tracking of live images and videos. It supports multiple languages like C++, Java and Python.

2.3 HARDWARE CONNECTIONS

There are two wheels in the bot connected separately with individual motors, driven by L293D IC which make the wheels rotate in either of the clockwise or anticlockwise direction [Table 1]. The IC contains 2 EN pins, 2 inputs and 2 output pins which is further integrated with the GPIO pins of Raspberry pi for pin set mode [Figure 1]. The moving bot detect obstacles (i.e. vehicles) sideways by Infrared Sensors and moves forward if there are obstacles on both sides, moves left if there is obstacle on the right side and moves right if there is obstacle on the left side. IR sensors have 3 connection pins- 1 GND, 1 VCC and 1 pin to the Raspberry pi GPIO. The GPIO header is coded accordingly in the python shell of Raspberry pi for the above given conditions. The positive and negative supply in the circuit is provided by a 9V battery (~6-7 mA current passage). This whole circuit along with the Raspberry pi is glued on the top of bot for easy and neat connections.

The USB camera is used in place of the pi camera module for the preview model of the space area, with the codes fetched for the face recognition, edge and number plate detection for the live number of obstacles. The camera is fitted on the top of bot with a rotating motor moving in all directions to access and capture the images of the whole set area.

Table 1: Logic Table for Motor-Driver

Input A	Input B	Motor State
High	Low	Turns clockwise
Low	High	Turns anti-clockwise
High	High	Braking occurs
Low	Low	Braking occurs

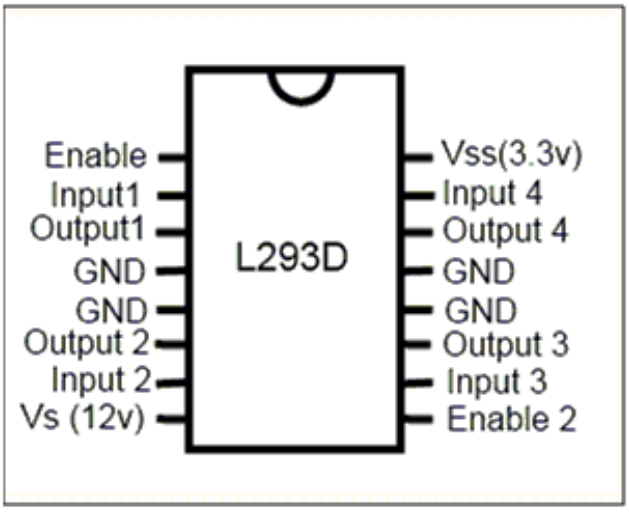

Fig 1: Pin Diagram of L293D IC

3 SOFTWARE DESCRIPTION

The software used in the project are divided into 3 parts: Obstacle detection with Automation, Image processing of the arena and Machine Learning by the analyzed data.

3.1 OBSTACLE DETECTION

The obstacle detection algorithm simply uses GPIO pins of pi coded in the python shell of raspbian OS. The code imports the GPIO library to meet the logic [Figure 2].

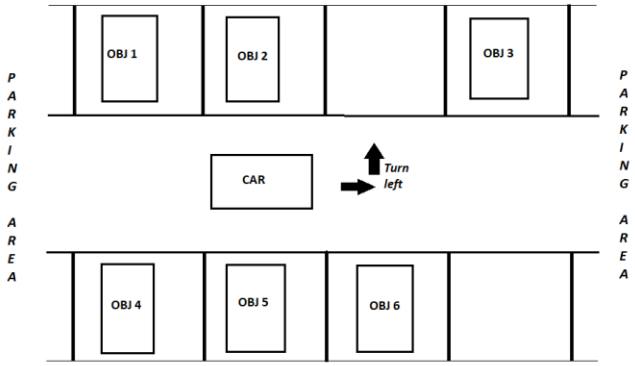

FIG 2: Drive Function of the car

3.2 IMAGE PROCESSING

In the car parking arena there are several people and other vehicles which are previewed using a camera module. This serves as the fore look before the car enters the arena and also acts as the raw data for the machine learning part. The code used here imports face cascade and license plate .xml files using OpenCV library for python [Figure 3 & 4].

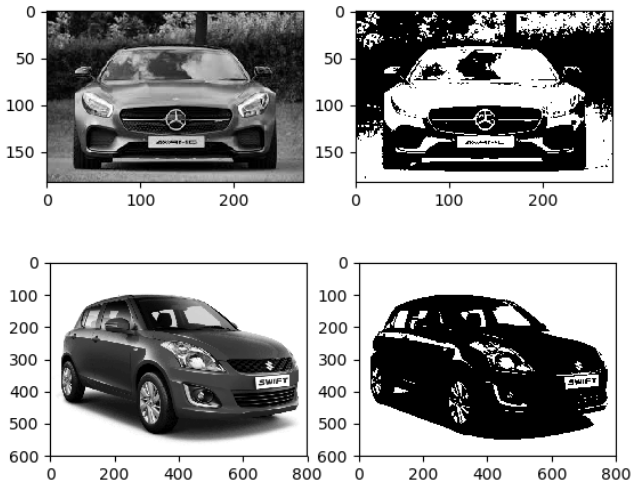

FIG 3: Gray to Binary processing of the car image

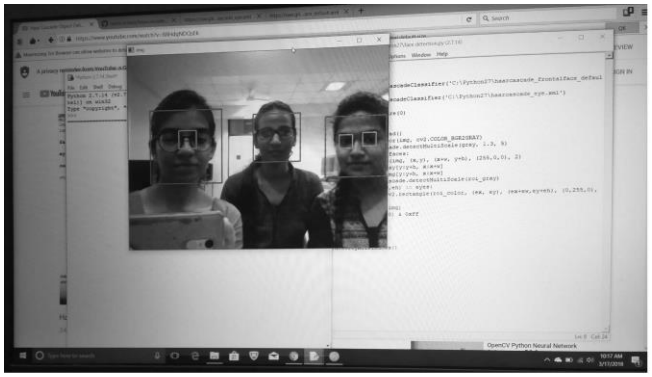

FIG 4: Face recognition

3.3 DATA HANDLING AND MACHINE LEARNING

With the ascertained data obtained, the Automatic License Plate recognition of any vehicle is done in 3 major stages: License Plate Detection, Character Segmentation and Character Recognition. The first stage is to determine the position of the license plate from the image input. Once, the license plate is detected it is segmented into individual section of images which are further used for the identification using the machine IQ.

4 WORKING

The project proceeds in individual sections with the applied conditions [Figure 4]. The first processing starts from capturing the live image of the parking arena clearly indicating the objects (both human and vehicles) and differentiating the obstacles with the fetched programming functions in the python shell of microprocessor. Once the image is captured the current situation and availability of slots is easy to detect and the bot moves according to it by initiating the motors. In the incipient condition it moves in the forward direction. If there are obstacles detected in the way, it keeps moving forward until it finds an empty slot with the help of sensors.

The data being captured with the camera recognizes the face of the persons and number plate of the cars for the preview as well as keeps on observing this whole cycle again and again over some duration of time. The repeated data is being stored in the memory of microprocessor (cloud on a larger scale) which turns out to be the raw information of the machine learning algorithm. Once the data handling starts, the optical character recognition of the characters on the license plate can be under covered by the ML algorithm. However, this algorithm can be developed can be without machine learning, using template matching or feature extraction but using ML improves the system accuracy by improving the training process i.e., the character recognition aspect of mapping a character image to its actual character and differentiate between As, Bs, etc.

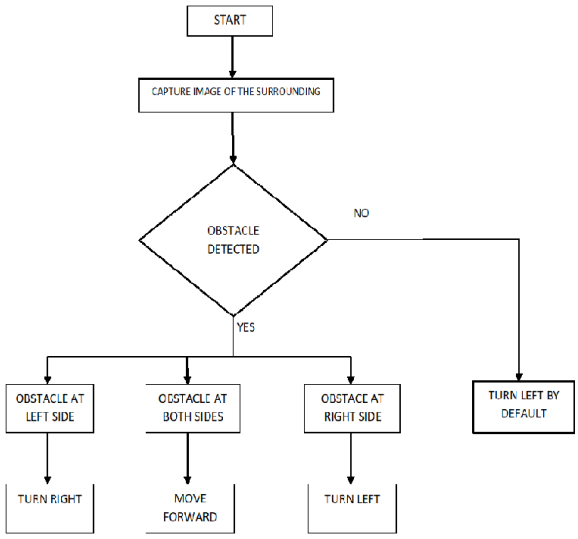

FIG 4: Flowchart for the implementation of idea

5 FUTURE WORK

The present algorithms works on all types of vehicles ranging from small to large scale. It is optimized and practical but still with the training of machine learning algorithms for a period of time it can be made more efficient. Every time the car moves through a certain path it can memorize the data by noting the characteristic features, hence avoiding on-spot calculations and unnecessary mistakes. The accuracy of parking can be increased with more cases and scenario.

6 REFRENCES

[1] Gurjashan Singh Pannu, Mohammad Dawud Ansari & Pritha Gupta, "Design and Implementation of Autonomous Car using Raspberry

pi," International Journal of Computer Applications (0975-8887), vol. 113, No.-9, March 2015.

[2] Ayush Rathi, Ashay Aawte, Narendra Shiradkar & Milind Kothikar, Electronic Design Project Report on " Self- parking car", Electrical Engineering Department, IIT Bombay.

[3] Basavaraju S R, On "Automatic Car Parkin System", Department of Information science and Engineering, International Journal of Scientific and Research Publications, Volume 5, Issue 12, December 2015