TERM PAPER

ON

IMAGE SEGMENTATION

SUBMITTED TO:

AMITY INSTITUTE OF ENGINEERING AND TECHNOLOGY, KOLKATA

SUBMITTED BY:

PRASUN DUTTA **A91005218026**

NAME OF FACULTY GUIDE:

DR. SUDARSHAN NANDI

DECLARATION BY THE CANDIDATE

I do hereby declare that

1. The work contained in this report is original and has been done by me under the guidance of my supervisors.

2. The work has not been submitted to any other Institute for any degree or diploma as per best of my knowledge.

3. The guidelines provided by the Institute in preparing the thesis have been followed.

4. The norms and guidelines given in the Ethical Code of Conduct of the Institute have adhered to.

5. Wherever materials (data, theoretical analysis, figures and text) used from other sources, due credit has been given by citing them in the text of the thesis and giving their details in the reference.

(Student's Signature)

Name: PRASUN DUTTA

Enrolment No.: A91005218026

Stream: B. Tech (CSE)

Batch: 2018-2022

Semester: 3

Paper Name: Mini Project

Paper Code: ETMI100

CERTIFICATE BY FACULTY GUIDE

This is to certify that the project "Autonomous Car" submitted by PRASUN DUTTA to Amity University, Kolkata, West Bengal, India, is a record of original research work carried under my supervision and is worthy of consideration for the award of the degree of Bachelor of Technology (Computer Science & Engineering) from the University.

Prof. Dr Sudarshan Nandi

Assistant Professor,

Department of CSE, ASETK,

Amity University, Kolkata,

West Bengal, India

Prof. (Dr.) Neeraj Sharma,

Head of Department,

Department of CSE, ASETK,

Amity University, Kolkata,

West Bengal, India

Dr. Birajashish Pattanaik,

Head of Institution,

Amity School of Engineering and Technology,

Amity University, Kolkata,

West Bengal, India

Acknowledgement

I would like to express my special thanks of gratitude to my faculty guide Prof. Dr. Sudarshan Nandi, our Head of Department, Prof. Dr. Neeraj Sharma and Head of Institution, Dr. Birajashish Pattanaik, who gave me the golden opportunity to do this wonderful project on Image Segmentation, which also helped me in gaining a lot of knowledge and I came to know about many new things. I am really thankful to them.

Secondly, I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

INDEX

1. ABSTRACT

This mini project is focus on the making of Bluetooth controlled car. The car is driven by four DC motors. These motors are controlled by a headless computer Raspberry Pi 3 B. The Raspberry Pi 3 B is used to run python codes that will be controlling the motors using a L298N motor driver. The Bluetooth module included in the Raspberry Pi 3 B is used to communicate with a Bluetooth controller like a smart phone. Using the Bluetooth controller, the car will be driven wirelessly. Later a Pi camera and ultra-sonic camera will be introduced using which we will be attempting autonomous driving by implementing artificial intelligence code in python.

2. OBJECTIVE

1. The objective of the mini project is to build a Bluetooth controlled car
2. Understanding which hardware to choose
3. In-Depth understanding and working of Raspberry Pi 3 B
4. In-Depth understanding and working of L298N
5. In-Depth understanding and working of HC-SR04
6. Designing a python code to run the car using smart phone

3. REQUIREMENTS

1. Raspberry Pi 3 B+
2. L298N Motor Driver Module
3. HC-SR04 Ultra sonic Sensor
4. 300 RPM DC Motor
5. Car chassis
6. Wheels
7. 12 V Battery
8. Jumper Cables
9. Solder Iron

Raspberry Pi

It is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It can run most programming language. It has general purpose input output pins which helps in controlling hardware and take input from sensors. This gives the device immense real-world application and opens the door for many DIY developers to create creative projects.

Specifications:

1. Processor: Cortex-A53 (ARMv8) 64-bit
2. RAM: 1GB
3. WLAN: 2.4GHz and 5GHz and Bluetooth 4.2
4. Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
5. Extended 40-pin GPIO header
6. Full-size HDMI
7. 4 USB 2.0 ports
8. CSI camera port for connecting a Raspberry Pi camera
9. DSI display port for connecting a Raspberry Pi touchscreen display
10. 4-pole stereo output and composite video port
11. Micro SD port for loading your operating system and storing data

GPIO- pins which stands for "General Purpose Input/Output". These types of pin found on an integrated circuit and does not have any specific functionality. While most pins have a dedicated purpose, such as sending a signal to a certain component, the function of a GPIO pin is customizable and can be controlled by software. In a Raspberry Pi GPIO pins can output (3.3V) and (0V) as high and low. A GPIO pin designated as an input pin can be read as high (3.3V) or low (0V). These pins can also be used for PWM (pulse-width modulation), Serial Peripheral Interface, Inter-IC, Serial. These pins can be programmed in Raspberry Pi by using programming language such as Python, Scratch, Processing3, lipgpiod.

L298N Motor Driver Module

The L298N is an integrated monolithic circuit in a 15- lead Multi-watt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic level sand drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the in-put signals. The emitters of the lower transistors of each bridge are connected together rand the corresponding external terminal can be used for the connection of an external sensing resistor. An additional Supply input is provided so that the logic works at a lower voltage.

Specifications:

1. High operating voltage, which can be up to 40 volts;
2. Large output current, the instantaneous peak current can be up to 3A;
3. With 25W rated power;
4. Two built in H-bridge, high voltage, large current, full bridge driver, which can be used to drive DC motors, stepper motors, relay coils and other inductive loads.
5. Using standard logic level signal to control.
6. Able to drive a two-phase stepper motor or four-phase stepper motor, and two-phase DC motors.
7. The module can utilize the built-in stabilized volt tube 78M05 to obtain 5v from the power supply. But to protect the chip of the 78M05 from damage, when the drive voltage is greater than 12v, an external 5v logic supply should be used.
8. Drive voltage: 5-35V; logic voltage: 5V

HC-SR04 Ultra sonic Sensor

The HC-SR04 uses non-contact ultrasound sonar to measure the distance to an object. It consists of a ultrasonic transmitters, a receiver, and a control circuit. The transmitters emit a high frequency ultrasonic sound, which bounce off any nearby solid objects, and the receiver listens for any return echo. That echo is then processed by the control circuit to calculate the time difference between the signal being transmitted and received. This time can subsequently be used, to calculate the distance between the sensor and the reflecting object.

Specifications:

1. Operating voltage: +5V
2. Theoretical Measuring Distance: 2 cm to 450 cm
3. Practical Measuring Distance: 2 cm to 80 cm
4. Accuracy: ±0.3 cm
5. Measuring angle covered: <15°
6. Operating Current: <15mA
7. Operating Frequency: 40kHz

4. WORKING OF THE PARTS

Raspberry Pi 3B

For this project we will be using the following pins:

1. 2 × 5-volt supply pins
2. 2 × ground pins
3. GPIO PIN 18
4. GPIO PIN 23
5. GPIO PIN 24
6. GPIO PIN 25
7. GPIO PIN 21
8. GPIO PIN 26

The 5-volt power supply with supply power to the logic boards of L298N and HC-SR04.

The ground pins are used to complete the circuit of the supply pins

GPIO pins 18 and 23 are used to control the logic input pins 1 and 2 of the L298N which controls the forward and backward movement of the right motor.

GPIO pins 24 and 25 are used to control the logic input pins 3 and 4 of the L298N which controls the forward and backward movement of the left motor.

GPIO pin 21 is used to set trigger the HS-SR04 which then activates itself to use its SONAR capabilities

GPIO pin 26 is a input pin which receives input from the HS-SR04 to get the time given by ultrasound ranging.

L298N Motor Driver Module

For this project we will be using the following pins:

1. 12 V DC input
2. 5 V DC input
3. Ground
4. 4 × logic input pins
5. 2 × output pins

The 12 V DC input is used to provide power to the motor

The 5 V DC input is used to provide power for the logic board

The ground pin is used to complete the circuits from the power sources

The logic input pins 1 and 2 is used to control the right motor. The right motor turns forward when pin 1 is high and 2 is low. The right motor turns backward when the pin 2 is high and pin 1 is low. If both are low the right motor stops.

The logic input pins 3 and 4 is used to control the right motor. The left motor turns forward when pin 3 is high and 4 is low. The right motor turns backward when the pin 4 is high and pin 3 is low. If both are low the left motor stops

The output pins are connected to DC motors which rotate when the logic board receives signal

HC-SR04 Ultra sonic Sensor

For this project we will be using the following pins:

1. VCC
2. Trig
3. Echo
4. Ground

The VCC is a 5 V DC power input which is used to power the logic board and the ultrasonic speaker and receiver.

The ground pin is used to complete the circuit of the power supply

The Trig pin or trigger pin is used to set this device active. When a high signal is sent to the trigger pin for a micro second the device is activated. It sends out an ultrasonic wave of 40 kHz from its speaker and waits for the receiver to receive the reflected sound.

The Echo pin is set active only when the receiver which is in active mode receives a ultrasonic sound of 40 kHz and is set active for the time that was required for the wave to go and come back.

5. WIRING OF PARTS

Connecting Raspberry Pi 3B to L298N

1. Connect 5-volt output of Raspberry Pi to 5-volt input of L298N
2. Connect GND pin of Raspberry Pi to GND pin of L298N
3. Connect GPIO pin 18 of Raspberry Pi to Logic pin 1 of L298N
4. Connect GPIO pin 23 of Raspberry Pi to Logic pin 2 of L298N
5. Connect GPIO pin 24 of Raspberry Pi to Logic pin 3 of L298N
6. Connect GPIO pin 25 of Raspberry Pi to Logic pin 4 of L298N
7. Connect positive side of 12-volt battery to 12-volt input pin of L298N
8. Connect negative side of 12-volt battery to GND pin of L298N

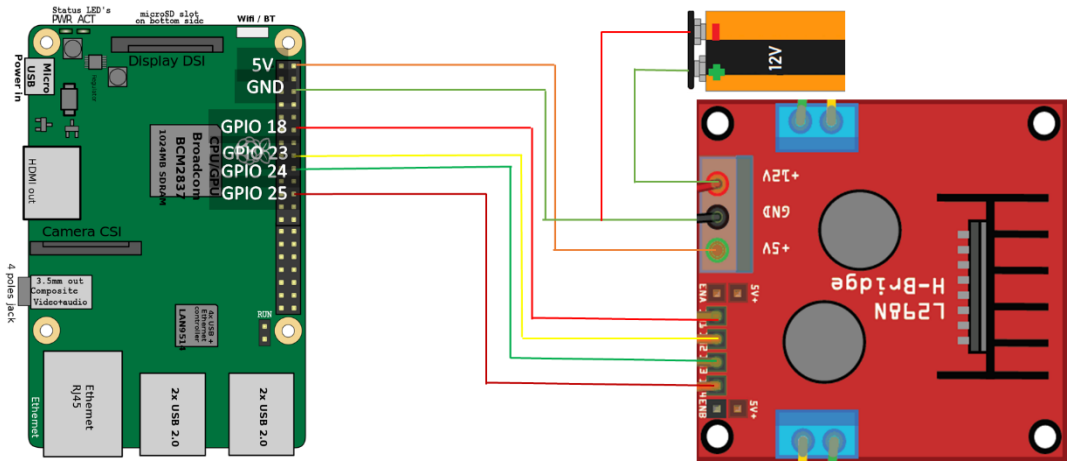

Fig: 5.1 Connecting Raspberry Pi 3B+ to L298N

Connecting L298N to DC motor

1. Connect positive of DC motor 1 and 2
2. Connect negative of DC motor 1 and 2
3. Connect the positive wires to L298N positive part of output pin
4. Connect the negative wires to L298N negative part of output pin
5. Connect positive of DC motor 3 and 4
6. Connect negative of DC motor 3 and 4
7. Connect the positive wires to L298N positive part of output pin
8. Connect the negative wires to L298N negative part of output pin

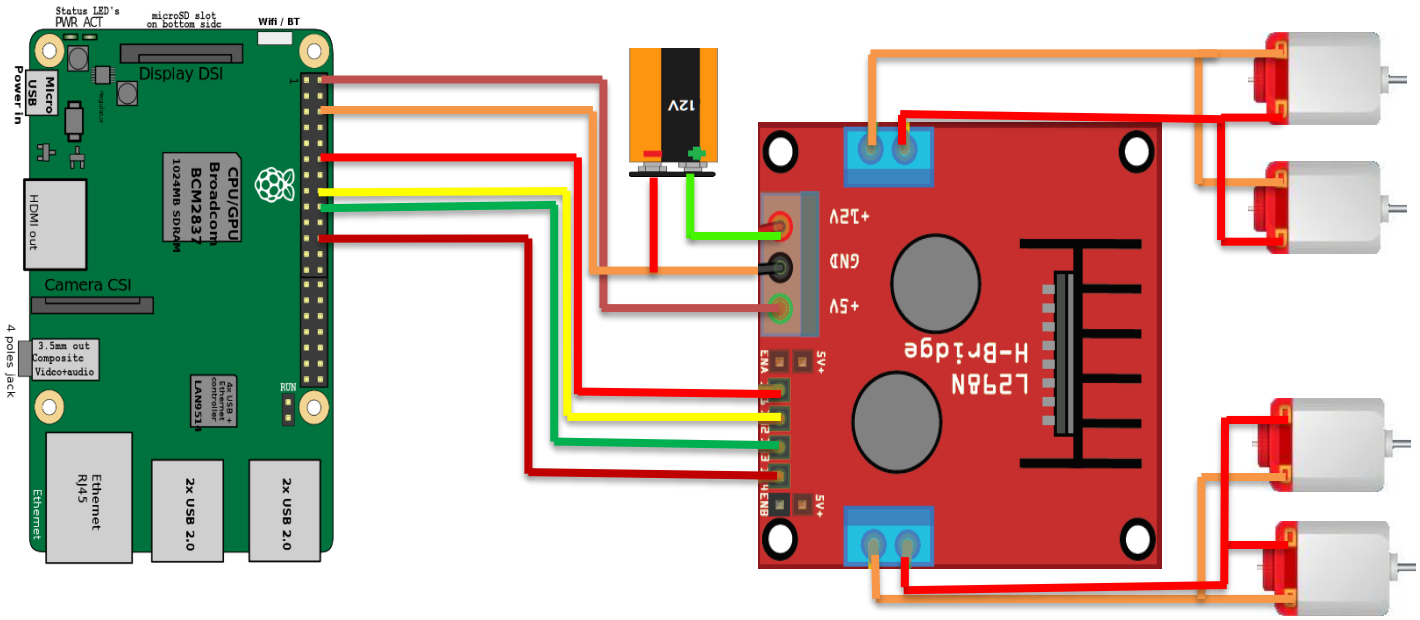

Fig: 5.1 Connecting L298N to DC motor

6. GPIO CODE IN PYTHON

Python Code:

```
1. import RPi.GPIO as GPIO
2. m1F=18
3. m1B=23
4. m2F=24
5. m2B=25
6. GPIO.setwarnings(false)
7. GPIO.setmode(GPIO.BCM)
8. GPIO.setup(m1F, GPIO.OUT)
9. GPIO.setup(m1B, GPIO.OUT)
10. GPIO.setup(m2F, GPIO.OUT)
11. GPIO.setup(m2B, GPIO.OUT)
12. GPIO.output(m1F , false)
13. GPIO.output(m1B , false)
14. GPIO.output(m2F, false)
15. GPIO.output(m2B, false)
```

Explanation

In 1st line one the Python library for GPIO controls is imported. The GPIO pins to be used is stored is variables m1F, m1B, m2F, m2B. GPIO warming are set to false because do not want to increase the complexity of the coding. Then GPIO naming mode is set to BCM. It can also be set in BOARD. For this project BCM naming conventions are used. Set the GPIO 18, 23, 24, 25 in output mode. Then initialize all the GPIO pins to output 0-volts or low.

7. BLUETOOTH CODE IN PYTHON

Python Code:

```
1. import Bluetooth
2. server_socket=bluetooth.BluetoothSocket( bluetooth.RFCOMM )
3. port = 1
4. server_socket.bind(("",port))
5. server_socket.listen(1)
6. client_socket,address = server_socket.accept()
7. print "Accepted connection from ",address
```

Explanation

In 1st line one the Python library for Bluetooth controls is imported. The next line creates a server socket that used RFCOMM protocol for communication between devices. Set port as 1. The next line tells the server socket to connect to any incoming device connection at the designated port. The server starts to listen for incoming connecting and waits for a device to connect. When ever a device connects the client is saved in client socket and the client address is saved in client address.

8. MOVEMENT CODE FOR CAR IN PYTHON

Forward Python Code:

```
1. def forward():
2.     print "FORWARD"
3.     GPIO.output(m1F , 1)
4.     GPIO.output(m1B , 0)
5.     GPIO.output(m2F , 1)
6.     GPIO.output(m2B , 0)
```

Explanation

Define a function forward (). Set GPIO pins m1F and m2F to high. As both the right and left motors rotate in forward direction the car moves in forward direction.

Backward Python Code:

```
1. def backward():
2.     print "BACKWARD"
3.     GPIO.output(m1F , 0)
4.     GPIO.output(m1B , 1)
5.     GPIO.output(m2F , 0)
6.     GPIO.output(m2B , 1)
```

Explanation

Define a function backward (). Set GPIO pins m1B and m2B to high. Set GPIO pins m1F and m2F to low. As both the right and left motors rotate in backward direction the car moves in backward direction.

Left Turn Python Code:

```
1. def left():
2.     print "LEFT"
3.     GPIO.output(m1F , 0)
4.     GPIO.output(m1B , 1)
5.     GPIO.output(m2F , 1)
6.     GPIO.output(m2B , 0)
```

Explanation

Define a function left (). Set GPIO pins m1B and m2F to high. Set GPIO pins m1F and m2B to low. As the right motor rotates forward and left motors rotates backward the car turns left.

Right Turn Python Code:

```
1. def l():
2. print "RIGHT"
3. GPIO.output(m1F , 1)
4. GPIO.output(m1B , 0)
5. GPIO.output(m2F , 0)
6. GPIO.output(m2B , 1)
```

Explanation

Define a function right (). Set GPIO pins m1Fand m2B to high. Set GPIO pins m2F and m1B to low. As the right motor rotates backward and left motors rotate forward the car turns right.

9. REST OF THE CODE

Python Code

```
1. data=""
2. while 1:
3.         data= client_socket.recv(1024)
4.         print “Received: %s" % data
5.         if (data == “w”):
6.                 forward()
7.         elif (data == “a”):
8.                 left()
9.          elif (data == “s”):
10.                backward()
11.        elif (data == “d”):
12.                right()
13.        elif(data == “q”):
14.                 break
15.                 GPIO.cleanup()
16.        else:
17.                        stop()
```

Explanation

Data is received from any Bluetooth controller. If data is “w” the car moves forward. If data is “s” the car moves backward. If data is “d” the car turns right. If data is “a” the car turns left. If data is “q” the program quits. If any other data is revived the car stops.

Connect to Bluetooth

1. Download BlueTerm from PlayStore
2. Connect Device “name of Raspberry Pi 3 B +”
3. Send command

10. CONCLUSION

Raspberry Pi 3 B+ is a small, compact and packs a lot of power for its size and power consumed. It gives a very beginner device with which we can introduce people to computers. It provides a lot of functionality though it's GPIO pins. A touch screen and camera can also be attached to it. It is very popular among the DIY enthusiast as the number of projects that can be done is limitless.

In the future a Pi camera will be attached to this module using which the video will be sent to a server. The server would process frames in the video and send autonomous command to the car. Thus, the car will be able to move and avoid obstacles, obey traffic rules without any human intervention.

11. REFERENCE

1. https://circuitdigest.com/microcontroller-projects/raspberry-pi-remote-controlled-car
2. https://www.raspberrypi.org/forums/viewtopic.php?t=230346
3. https://www.leenabot.com/Raspberry-Pi-Controlling-Robot-using-Bluetooth/
4. https://www.tutorialspoint.com/python/index.htm
5. https://www.pantechsolutions.net/android-mobile-controlled-robotic-car-using-raspberry-pi
6. https://www.raspberrypi.org/documentation/usage/gpio/