NYU | TANDON SCHOOL OF ENGINEERING

ME-GY 6933 – Advance Mechatronics

Navigation Shopping Cart

Project report

Group Member:
Kuan Li (kl3744)
Jui Yu Tseng (jyt258)
Yuhang Lin (yl7094)

Project Supervisor:
Dr. Vikram Kapila

Report Date:
05/ 14/ 2020

Number of Pages

Introduction

In this day and age, a great deal of emphasis is given to science and innovation to improve the regular man's life. The advancement of robots has been imperative to mechanical as well as our day by day use. We see the utilization of electronic segments in pretty much every non-living item around us in the expectation of making everything computerized one day. Taking up the Advance Mechatronics class this semester has given us a decent understanding of the present limit of innovation and gives us how a specialist can breathe life into his thoughts by utilizing straightforward sensors and electronic circuits.

As a piece of the desire and subjected by the need of finishing a mechatronics venture, we endeavored to investigate plenty of thoughts that were planned for streamlining an individual's life accordingly expanding profitability and utilizing time. The area which we decided to focus on was the navigation shopping cart since it is inconvenient to push a shopping cart while shopping in a crowded supermarket. In the future, this technology can be used widely in every kind of supermarket, retailer and the Unmanned store.

Navigation is a process of monitoring and controlling the movement of a vehicle from one place to another. It has been researched and developed for a long time and implemented on robots and self-driving cars. With this technology and several sensors, such as cameras and microphone, we can achieve our goal of making a Voice control Navigation shopping cart. When the customer says a specific product, the smart cart will lead him to the item shelf. This brand new tool can bring the customers a better shopping experience. In the future, it can also improve the efficiency of checkout by adding a new function that the camera reads the QR code of the items which the customer put in the cart. When the customer goes through a door, the payment would be done automatically.

Method

Our goal is to build an auto-navigating smart shopping cart with voice and QR code recognition functions. There are two boards used in the project, Raspberry Pi 4B and Arduino Uno. Raspberry Pi is mainly used for acquiring and processing data, and Arduino is used for driving the cart. These two boards are connected by a usb serial port. Raspberry Pi will detect the QR code and voice, and convert these data into states signal: forward, backward and stop. After that, Arduino will drive the navigation shopping cart based on those states.

We utilized open source libraries DeepSpeech for voice recognition and OpenCV for QR code detection. DeepSpeech is an open source A.I. based embedded Speech to Text engine, using a model trained by machine learning techniques and TensorFlow to make the implementation easier. One of the largest advantages is that it is able to run offline on resource constrained environments like embedded systems. OpenCV (Open Source Computer Vision Library) is the most well known library in the computer vision field. There are many OpenCV's applications, including image processing.

Here is how the whole system works. After the system starts, the camera will keep capturing QR code in its sight, and keeps track of the robot current position. When the user speaks to the navigation shopping cart, the voice will be transcribed to text. The system will search the keyword in the text to see if there is a specified destination. Once the destination is set, it will be compared with a map that is prestored in the program to find out which direction it should go. To simplify the problem, we make this navigation shopping cart only face in one direction, which means it can only go forward or backward in a straight line. Therefore, when it knows where its current position is and its destination, the cart is able to decide which direction it should move. Finally, the cart will either go forward or backward and keep searching for the corresponding QR code which represents the position as what the customer wants.

Table 1: Bill of materials

Material	Quantity	Unit Price
1. Raspberry pi 4 B	x1	$35.00
2. Arduino Uno R3	x1	$15.98
3. LoveRPi 5MP Camera	x1	$9.99
4. SunFounder USB 2.0 Mini Microphone	x1	$7.69
5. VEGET Raspberry Pi 4 Case with Fan	x1	$14.79
Total	$83.45	

Result

The Navigation Shopping Cart shows in figure 1 has a good performance in the demonstration. When we speak to the mic, it takes a few seconds to recognize the audio and transfer it to text. At the same time, it recognizes the QR code, and then it will either run forward or backward to the assigned position as the adio. The list included: electronic, vegetable, water, paper and apple.

First, we put the car in front of the QR codes and it will start to recognize the code. Then when it recognized the audio correctly, the cart run in the correct direction immediately. At the same time, the QR code that it passes will be recognized as well. When driving, the voice would not be recognized. Therefore, it cannot change the final position by any interruption. The advantage is that it can ignore other audio and finish the mission in the noisy environment. When it doesn’t recognize the audio, it will not change the destination.

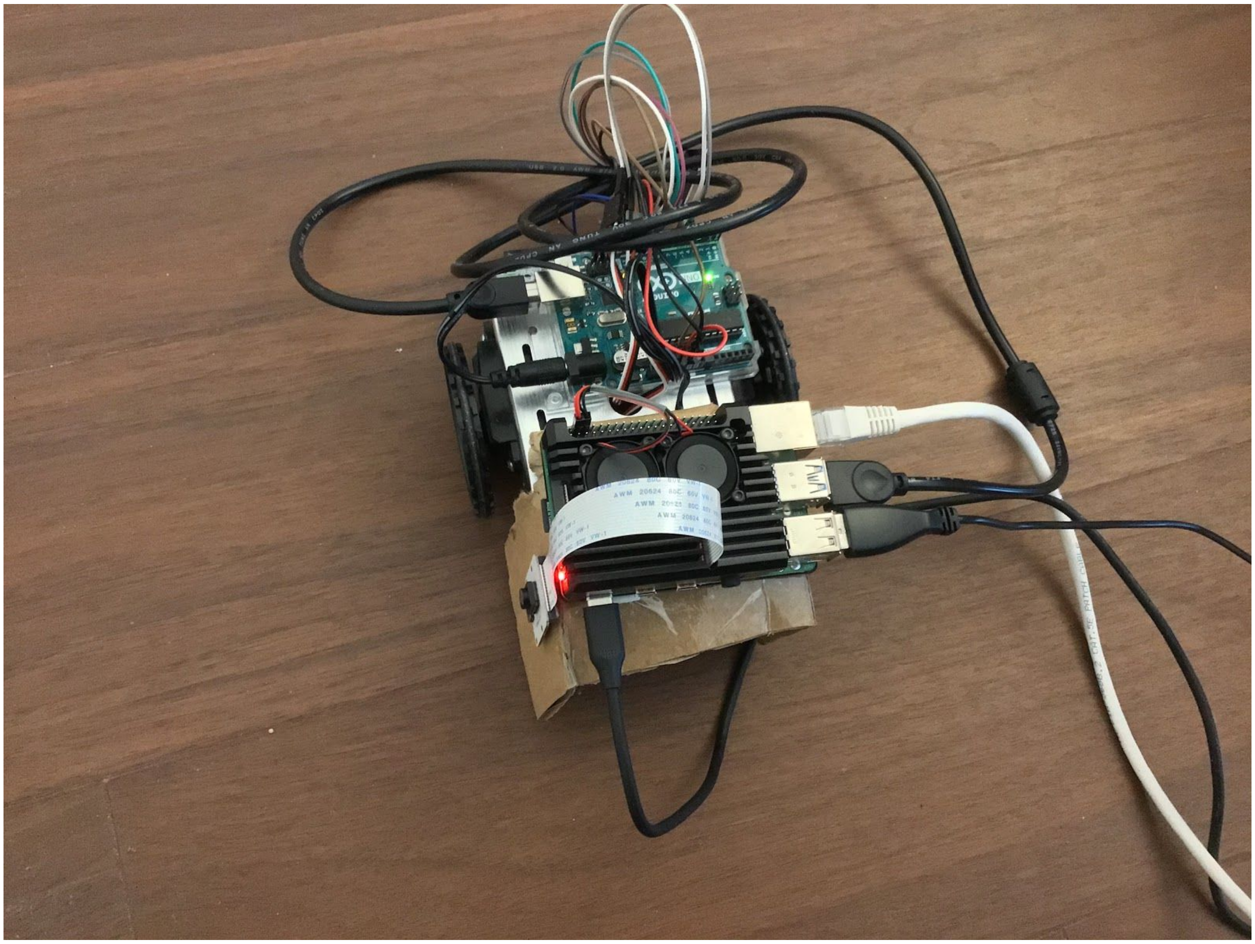


Figure 1: Navigation Shopping Cart

Conclusion

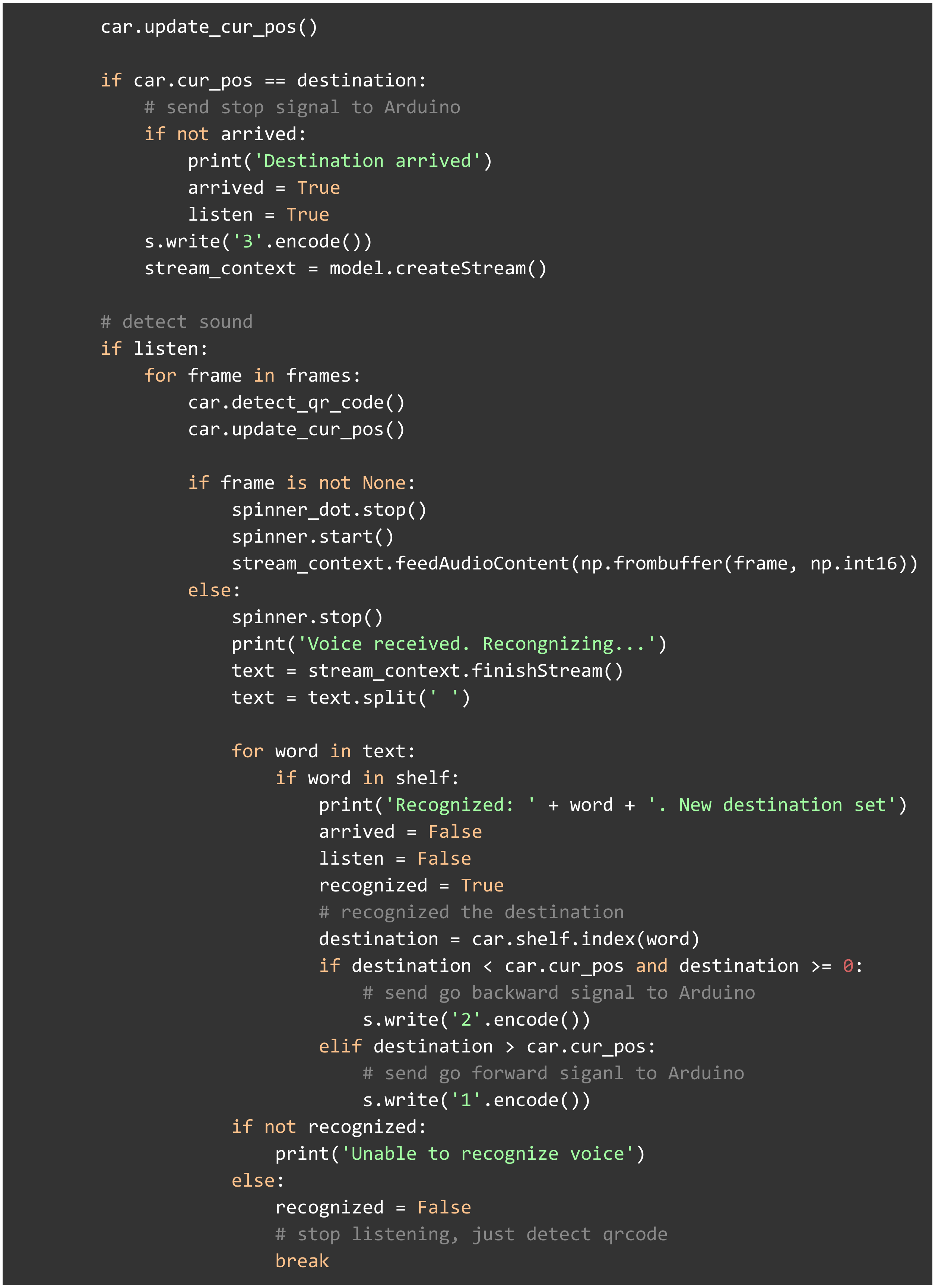
The Navigation Shopping Cart is a workable solution to lead people to the place they want to visit during shopping. In the future, it is accomplished that the website or database could be a better solution with a larger area. It can increase the operation speed by using an online voice recognition library and provide a better microphone. Then it can run with lower resources and higher accuracy.

Appendix

Raspberry Pi

```python
import time
import collections, queue, os, os.path
import deepspeech
import numpy as np
import pyaudio
import wave
import webrtcvad
from halo import Halo
from scipy import signal
from mic_vad_streaming import Audio, VADAudio

import serial
import cv2


class Car():
    def __init__(self, shelf):
        self.shelf = shelf
        self.cur_pos = -1
        self.qr_data = None
        self.cap = None
        self.detector = None

    def enable_camera(self):
        self.cap = cv2.VideoCapture(0)
        self.detector = cv2.QRCodeDetector()
        print('Camera enabled')

    def detect_qr_code(self):
        _, img = self.cap.read()
        self.data, _, _ = self.detector.detectAndDecode(img)

    def update_cur_pos(self):
        if self.data:
            if self.data != self.shelf[self.cur_pos]:
                print('Now at: ', self.data)
                self.cur_pos = self.shelf.index(self.data)
            else:
                print('Same position')
        else:
            print('Unable to detect QR Code')
```

```
def main():
    # define variables
    shelf = ['electronic', 'vegetable', 'water', 'paper', 'apple']
    destination = -1
    arrived = -1
    listen = True
    recognized = False
    car = Car(shelf)
    car.enable_camera()

    # connect with Arduino
    s = serial.Serial('/dev/ttyACM0', 9600)
    try:
        s.open()
    except:
        s.close()
        time.sleep(5)
        s.open()
    time.sleep(5)
    print('Arduino boot completed')

    # load deepspeech model
    model_name = 'deepspeech-0.7.0-models.tflite'
    scorer_name = 'deepspeech-0.7.0-models.scorer'
    rate = 44100  # mic rate

    model = deepspeech.Model(model_name)
    model.enableExternalScorer(scorer_name)

    # Enable camera to detect qrcode
    car.detect_qr_code()
    car.update_cur_pos()

    # Start audio with VAD
    vad_audio = VADAudio(aggressiveness=3, input_rate=rate)
    print("Listening (ctrl-C to exit)...")
    frames = vad_audio.vad_collector()
    spinner = Halo(spinner='line')
    spinner_dot = Halo(text='reading qrcode...', spinner='dots')

    stream_context = model.createStream()

    # Start detect sound and qrcode
    while True:
        # detect qrcode
        spinner_dot.start()
        car.detect_qr_code()
```



```
        car.update_cur_pos()

        if car.cur_pos == destination:
            # send stop signal to Arduino
            if not arrived:
                print('Destination arrived')
                arrived = True
                listen = True
            s.write('3'.encode())
            stream_context = model.createStream()

        # detect sound
        if listen:
            for frame in frames:
                car.detect_qr_code()
                car.update_cur_pos()

                if frame is not None:
                    spinner_dot.stop()
                    spinner.start()
                    stream_context.feedAudioContent(np.frombuffer(frame, np.int16))
                else:
                    spinner.stop()
                    print('Voice received. Recongnizing...')
                    text = stream_context.finishStream()
                    text = text.split(' ')

                    for word in text:
                        if word in shelf:
                            print('Recognized: ' + word + '. New destination set')
                            arrived = False
                            listen = False
                            recognized = True
                            # recognized the destination
                            destination = car.shelf.index(word)
                            if destination < car.cur_pos and destination >= 0:
                                # send go backward signal to Arduino
                                s.write('2'.encode())
                            elif destination > car.cur_pos:
                                # send go forward siganl to Arduino
                                s.write('1'.encode())
                    if not recognized:
                        print('Unable to recognize voice')
                    else:
                        recognized = False
                        # stop listening, just detect qrcode
                        break
                    recognized = False
                    stream_context = model.createStream()


if __name__ == '__main__':
    main()
```

Arduino

```
#include <Servo.h>
int left = 5;
int right = 6;
int value;
Servo Servo1, Servo2;

void setup()
{
  Serial.begin(9600);
  Servo1.attach(left);
  Servo2.attach(right);
  Stop();
}

void loop()
{
//  Serial.println("start");
  if(Serial.available())
  {
    value = Serial.read();
    //value = 3 ;  //Signal from raspberry
    //Serial.println(value);
    //Stop();
    switch(value)
    {
      case '1': forward(); break;
      case '2': back(); break;
      case '3': Stop(); break;
    }
  }
}
```

```
void forward()
{
  Serial.println("forward");
  Servo1.write(150);
  Servo2.write(30);
  delay(100);
}

void back()
{
  Serial.println("backward");
  Servo1.write(30);
  Servo2.write(150);
  delay(100);
}

void Stop()
{
  Serial.println("stop");
  Servo1.write(90);
  Servo2.write(90);
  delay(100);
}
```