Exploring Computer vision capability on Raspberry pi 4
Raspberry Pi

- It is a basic SBC (single board computer that support raspbian os)
- It us in making IOT, Robotic and Automation projects
- Supports various programming language like C, Python, C++, JS, JAVA
- Can act as a web server, a stand alone computer
- Can run multiple codes and task at a single time
Raspberry Pi 4 GPIO

- 40 GPIO
- I2C
- UART
- JTAG
- SPI

(It also has DPI display parallel interface) takes 28 pins
Cv (Computer vision)

- It gives capability to analyse image
- Detect objects
- Recognise face
- Find pattern in image and video
- Can help in OCR (Optical character recognition)
- Helps in making decision based on image processing
Why we need CV (Computer vision)

- In robotics to recognise objects and analyse environment
- In Autonomous vehicles
- In security camera
- In analysing image and to study the pattern in image

Programming language good for this in Raspberry Pi

- Python
- C
Commonly Used Python Libraries and Modules for Computer Vision

- TensorFlow
- OpenCV
- ImageAI
- Numpy
- Scipy
Most Common Python Libraries For Robotics

Controlling Robot
- Gpiozero (Control GPIO of RPI)

Making Robot Talk
- Espeak
- GTTS

Making GUI for Robot
- Guizero
- pygame
Understanding the basics of python

- We can use pip for installing and uninstalling the libraries to python environment.
- Python is a scripting language the run line by line so here we do not need the opening and braces for starting of function.
- We need not to declare the data type of variables in python it automatically detects that.
- We need to take care of indentation in python.
- We need to take care of version of python.

Suppose if you want to print output in python 2 you can directly use print result to print.
But in python 3 you need to use brackets.
Designing A Object Detection Robot and Testing TensorFlow Module On Raspberry Pi 4

Choosing Right Board for Project?

- First decide purpose and features that your robot need
- Then go to specification of board that meet that feature

Features:
- Robot should take Image and video
- Robot can process Image and detect object
- Robot tell about detected object in form of audio and video output

Here we need to do Image processing for our project that requires a lot of RAM and Processing power.
So selecting RPI 4 with RAM 2 GB or grater
Designing a Object detection Robot Using Raspberry Pi TensorFlow

Selecting Libraries That Fulfill Our Requirement

For detecting Object and Processing object we use Open CV and TensorFlow.
For moving camera module we need to drive servo so we can use gpiozero.
For audio output of detected object we can use Espeak.
Libraries needed are:
Opencv
TensorFlow
Espeak
Gpiozero
Learn to use libraries to make robot

Controlling GPIO of raspberry pi using gpiozero

```python
from gpiozero import LED
from time import sleep
led = LED(17)
while True:
    led.on()  # for off led.off()
```

Controlling servo

```python
from gpiozero import AngularServo
from time import sleep
servo = AngularServo(17, min_angle=-90, max_angle=90)
while True:
    servo.angle = -90
```
How to make my Pi talk?

Import espeak

While True:
    espeak.set_voice("En") ///set the voice language
    espeak.set_voice("whisper") /// setting the type like loud
    espeak.synth("Hey hello ") /// voice to
Designing a Object detection Robot Using Raspberry Pi TensorFlow

```python
import os
import cv2
import numpy as np
from picamera.array import PiRGBArray
from picamera import PiCamera
import tensorflow as tf
import argparse
import sys
from guizero import App, TextBox, PushButton, Text, info
from tkinter.font import Font
from gpiozero import AngularServo
from time import sleep
from gpiozero.tools import sin_values
servo = AngularServo(17, min_angle=-90, max_angle=90)
```
Designing a Object detection Robot Using Raspberry Pi TensorFlow

```python
# Name of the directory containing the object detection module we're using
MODEL_NAME = 'ssdlite_mobilenet_v2_coco_2018_05_09'

# Grab path to current working directory
CWD_PATH = os.getcwd()

# Path to frozen detection graph .pb file, which contains the model that is used
# for object detection.
PATH_TO_CKPT = os.path.join(CWD_PATH, MODEL_NAME, 'frozen_inference_graph.pb')

# Path to label map file
PATH_TO_LABELS = os.path.join(CWD_PATH, 'data', 'mscoco_label_map.pbtxt')
```
Designing a Object detection Robot Using Raspberry Pi TensorFlow

```python
# Name of the directory containing the object detection module we're using
MODEL_NAME = 'ssdlite_mobilenet_v2_coco_2018_05_09'

# Grab path to current working directory
CWD_PATH = os.getcwd()

# Path to frozen detection graph .pb file, which contains the model that is used
# for object detection.
PATH_TO_CKPT = os.path.join(CWD_PATH, MODEL_NAME, 'frozen_inference_graph.pb')

# Path to label map file
PATH_TO_LABELS = os.path.join(CWD_PATH, 'data', 'mscoco_label_map.pbtxt')
```
Designing a Object detection Robot Using Raspberry Pi TensorFlow

```python
min_score_thresh=0.40

cv2.putText(frame, "FPS: {:0.2f}".format(frame_rate_calc), (30, 50), font1, (255, 255, 0), 2, cv2.LINE_AA)
name = str([[category_index.get(value) for index, value in enumerate(classes[0]) if scores[0, index] > 0.5])
print(name)
count1 = name.count(substring)
print(count1)
print("")
#str.split()
if (count1 >= 1):
    print("find the object in image")
else:
    print("no fruit")
servo.source = sin_values()
servo.source_delay = 0.1

cv2.imshow('Object detector', frame)
t2 = cv2.getTickCount()
time1 = (t2-t1)/freq
frame_rate_calc = 1/time1
if cv2.waitKey(1) == ord('q'):
    break
```
Designing a Object detection Robot Using Raspberry Pi TensorFlow
THANK YOU