ROBOT VISION THROUGH CLOUD FOR EMBEDDED DEVICE

By

JEFFREY WONG HAW YIN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS)

COMMUNICATION AND NETWORKING

Faculty of Information and Communication Technology
(Kampar Campus)

MAY 2020

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: ROBOT VISION THROUGH CLOUD FOR EMBEDDED DEVICE

Academic Session: MAY 2020

I JEFFREY WOGNG HAW YIN

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)

(Supervisor's signature)

Address:

2172, Jalan Seksyen 2/4,
Taman Bandar Baru Barat
31900, Kampar, Perak

Dr. Goh Hock Guan

Supervisor's name

Date: 9th September 2020

Date: 10/9/2020

ROBOT VISION THROUGH CLOUD FOR EMBEDDED DEVICE

By

Jeffrey Wong Haw Yin

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS)

COMMUNICATION AND NETWORKING

Faculty of Information and Communication Technology
(Kampar Campus)

MAY 2020

DECLARATION OF ORIGINALITY

I declare that this report entitled "**ROBOT VISION THROUGH CLOUD FOR EMBEDDED DEVICE**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : J

Name : JEFFREY WONG HAW YIN

Date : 9th SEPTEMBER 2020

ACKNOWLEDGEMENTS

I have taken efforts in this project. It is impossible for me to finish this project without the kind support from my supervisors, Dr. Goh Hock Guan who has give me an opportunity to explore the new field in my university life. It is my first time exploring in robotic. A million thanks to you.

I would like to thank to my parents and my friend who give me support when I faced trouble in doing the project. Although they didn’t know have any knowledge in this field but they voice is my big support for me to continue work on my project.

ABSTRACT

In this project, an application of robot vision through the cloud will be developed as Artificial Intelligence had become the trend of the world. It is surrounded the human nowadays. When heard about AI, the image that comes into the mind of most of the people are a robot. In 2018, Asia was the world's largest industrial robot market over the world. Two out of three robots that are newly deployed in 2018 were installed in Asia. It shows that there is a high demand for robotic in Asia over the world. The installation rate of robots rose by 23% per year from the year 2013 to the year 2018. Human is using their eye to increases the efficiency of doing their jobs. It is also same as the robot. They are also needed to have their eye increases their efficiency in doing their jobs. At the end of the project, a path finding robot will be developed.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

OS	Operating System
AI	Artificial Intelligence
API	Application Programming Interface
GUI	Graphical User Interface
SD	Secure Digital

Chapter 1: Introduction

1.1 Problem Statement and Motivation

According to Alex, there are several problems toward the robot vision, which are lighting, deformation or articulation, background, movement. There is a problem when trying to take a digital photo in the low light. Imaging sensors are not as sensitive as the human eye, and the vision sensor is unable to detect the object correctly with the wrong type of lighting. Especially when the robot is moving, due to the difference in lighting conditions, it may change the colour, appearance, shading position, background, texture, and the motion of the object. It will decrease the accuracy of the robot’s vision.

Another problem for robot vision is the ability to distinguishing backgrounds and objects (Kerley, 2018). The robot will only be able to identify and track the object if they can recognize the environment and the object. The object will be determined more accurately when the background is bright or blank (Owen-Hill, 2017) As mention above, the lighting of the environment may be changed and caused the colour and the shading of the background to keep changing; it will also affect the robot's vision. The vision detection system must consider several potential background scenarios so that it could achieve a higher vision efficiency.

The third problem causes when the robot is moving, it will cause the blurring of the image that happened, especially when the robot is moving faster. This is because most of the robot is using the digital imaging sensor, which captures the image over a short period instead of capture the whole image instantaneously. Other than that, latency may also occur when the user wishes to use the live stream. The camera display may delay, especially when the internet connection is worst. In this case, if the robot is moving, the user may wait for a while for the camera display.

In this project, accuracy is the main issue to solve, especially when the background environment is worst. Accuracy is essential for robot vision. If the robot vision could not have high accuracy, it could cause other problems to occur when the robot needed to perform some tasks given.

1.2 Project Objectives

The project aims to develop a path finding robot. The robot can capture the image and send the copy obtained to the Google Cloud for recognition and find the item that the user needed by the robot itself. In this era of globalization, many jobs had been replaced by the robot, especially some big, dangerous, or dirty situations. Some hazardous jobs succeeded by the robot required high accuracy robot vision to perform. For example, bomb disposal jobs, it required the robot had high accuracy robot vision to prevent the bomb explodes during defuse period.

1.3 Project Scope

In this project, an application will be developed to build a path finding robot using robot vision through the cloud. Google Cloud is being used in the project to perform image recognition after the image capture by the robot. By the command by the user, the robot can find the item that requested by the user. There will be a list of items placed around of the robot. When the code is run, there will be a list of items are listed in the screen for the user to choose. After choosing the item, the robot will start to look around to find the item by itself and it is able to carry the item back to the user.

1.4 Impact, Significance, and Contribution

In the project, a path finding robot will be developed. In this era, Artificial Intelligence surrounds the daily life of humans. It connects the machines and humans. It makes life more convenient, for example, when we visit an e-commerce website, when we search for something, it will track our search preferences and show the results based on the favorites when the users visit the site next time.

When we talk about Artificial Intelligence, the first image display in most people mind is robotic. Nowadays, more and more robot has been developed to perform and replace more and more jobs. This is because the robot could work exactly 24 hours every day, but humans are only able to do their jobs at most 16 hours per day because they needed to rest. The robot does not need it, and it saves the producer cost of production.

Why should a robot vision need a robot? This is to give an eye to the robot so that the robot could find the objects in its working envelope. This will reduce the need for complicated and expensive fixtures. Other than that, a robot with robot vision could perform more complex jobs, especially jobs that are dangerous such as bomb disposal jobs. The robot should become smarter to perform this kind of hazardous situation. By giving an eye to the robot, people are at ease to let the robot to do the dangerous jobs. Just like a human, we are using our eye to recognize thing and send the image that we capture by our eye to brain to perform some complex jobs. This is the same as the robot, add an eye to the robot can improve their performance in doing the complex task.

1.5 Background information

In 2018, Asia was the world's largest industrial robot market over the world. Two out of three robots that are newly deployed in 2018 were installed in Asia. It shows that there is a high demand for robotic in Asia over the world. The installation rate of robots rose by 23% per year from the year 2013 to the year 2018. The primary market for industrials robots is China, Japan, and Korea. Although it is not popularly used in Malaysia, we can see the trend of robotic in the future. The robot required by every field is increased year by year, especially the automotive industry.

Cloud technologies are surrounding people's life, and these technologies have now come to the robotics. Cloud is used to describe the software and the services on the global network. It allows the user to access the information virtually in any device. With cloud technologies, more powerful robot solutions are developed nowadays. This is because the cloud can handle computationally heavy tasks, which may require more power and cognitive collaboration.

Another thing is computer vision, which is one of the most widely used fields of machine learning. The task of computer vision is simple, which is to understand the content of the image. Google provides a lot of cloud computing services as APIs for computer vision. The Vision APIs that they develop help the user's application to understanding the content of the image by classifying the content into different categories and provided the labels.

In this project, GoPiGo will be chosen as the end devices. GoPiGo is developed by Dexter Industries as a complete kit for users to build their robot cars. It is an excellent starter kit for the beginner as it included a complete robot package for the user, such as robot body, motors, controls, camera, and others. The user can use different programming languages and the USB accessories on the Pi to turn the GoPiGo into anything from a WIFI controlled robot.

Chapter 2: Literature Review

2.1 Review of the Technologies

2.1.1 Hardware platform

2.1.1.1 GoPiGo

GoPiGo is a complete robot kit that available for Raspberry Pi. Dexter Industries develop it. Although the price of the GoPiGo is higher, it comes with all the hardware that needed to mount the Pi, and users are not required to more soldering and wirework. For the beginners, Dexter Industries had published a list of tutorials on projects for the user to try with GoPiGo, such as Obstacle Avoider and even a Canon Drone. Dexter Industries had also offered a range of accessories tools for the user to use on GoPiGo. GoPiGo is also an excellent tool that suitable to introduce the robotics to the classroom as it is easy to set up, easy to connect to wide ranges of computers, and there is packed full of tutorials available online.

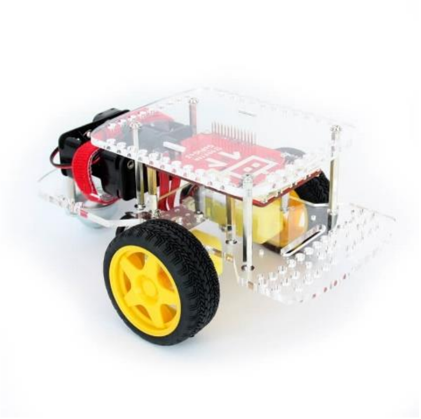


Figure 2-1-1-1 GoPiGo

2.1.1.2 Arduino Robot

Arduino is one of the largest open-source hardware platforms over the world. There are many robot kits available in the market designed, especially for Arduino. It is different from GoPiGo. Users are needed to install the driver used for Arduino in there a computer to program the Arduino Robot. Comparing the GoPiGo, the price to build an Arduino robot is cheaper than GoPiGo, but users are needed to do more soldering or wire works to make the robot.

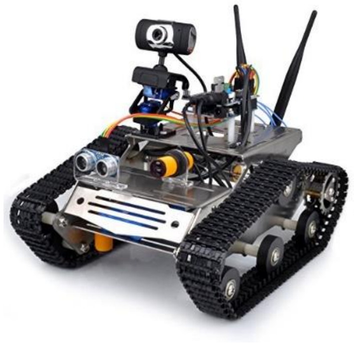


Figure 2-1-1-2 Arduino Robot

2.1.2 Firmware/OS

2.1.2.1 Dexter OS

Dexter OS is developed for GoPiGo, and it can broadcast the wifi signal that connected to any devices. Use could access it through the web browser. It is suitable for beginners as there is probably no extra thing that users needed to download and install separately, lots of tools will be available when the OS is installed. Dexter OS also consists of dozen of built-in lessons for the user to learn to program the GoPiGo. IT makes the beginner east to getting started for the first time.

2.1.2.2 Google Cloud Vision API

Google Cloud Vision API is a service that released by Google to allows the development to build image recognition and classification quickly to their applications, and it is free for users to use. It can be used to extracting the text from the images and even recognizing multiple objects in a single image. Google had trained the algorithm to identify thousands of different objects. There is also an existing project that released by Google, which using the GoPiGo to recognize the object through Google Vision API.

2.1.2.3 OpenCV

OpenCV is another API developed by Intel which can be used for image processing and computer vision application. As same as Google Vision API, OpenCV is also a free platform for the user to use. It is an open-source computer vision and machine learning library. OpenCV had provided a simple-to-use infrastructure for computer vision that can help people to build their vision application quickly (Sami, 2012). It has multi programming language interfaces and can support different type of operating system.

2.1.3 Programming Language

2.1.3.1 Python

Python is a widely used programming language. It used lesser wordy compare to other programming languages. It is a clear and concise language with an excellent built-in standard library. Besides that, the Raspberry Pi Foundation has also recommended user use Python as a language in programming because the Pi in Raspberry pi come from Programming languages. User is also allowed to use other programming languages on Raspberry Pi as they also installed by default on the Raspberry Pi.

2.1.4 Summary of the Technologies Review

Besides the technologies stated above, many technologies can be used to developed and robot with robot vision. GoPiGo is more suitable for a beginner who experiences in soldering or wirework as GoPiGo does not have is a tool that comes with many tools that required by the user, including some sensor. For a user that has little capital, they can choose Arduino Robot as the robot base because it is cheaper than GoPiGo tools, but the user is needed to do further soldering and wirework.

Both Google Vision API and OpenCV are suitable for user to use, and both of them are free platform. They are all developed for image processing and computer vision. Both platforms are now common use by people over the world but comparing to OpenCV, and Google Vision API provides more powerful image analysis than OpenCV.

2.2 Review of the Existing Systems/Applications

2.2.1 Jetson Nano Robot

JetBot robot is a robot that using Jetson Nano Developer Kit. It is a small, powerful computer that allows users to run multiple neural networks in parallel for the application like image classification, object detection and even speech processing. Jetson Nano Vision-Controlled AI Robot can be used to develop a collision avoidance robot. It is using the vision of the robot to detect the obstacle in front of the robot, and it can directly turn itself to avoid the barrier without any delay. Many people use JetBot to build the following robot. It uses the vision of the robot and using machine learning to let the robot to learn to follow the object or a person by itself without any controlling by the users.

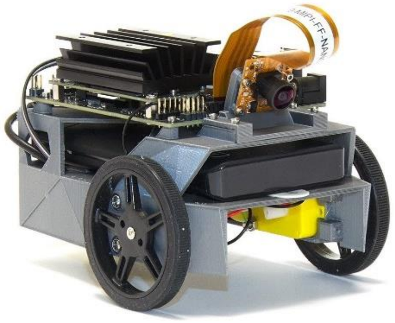


Figure 2-2-1 Jetson Nano Robot

2.2.2 Robot with Raspberry Pi and Arduino

Rene had developed and robot which using Raspberry Pi and an Arduino Uno to build and robot which can find its way back to its home. By using the Python OpenCV library, the robot can determine the environment and locate its garage. It can also move itself to a better position to make sure it can drive itself in the straight line into its garage. Rene had also managed to make the robot online on Whatsapp to ask the robot to capture the picture for the user and send it back by using Whatsapp.

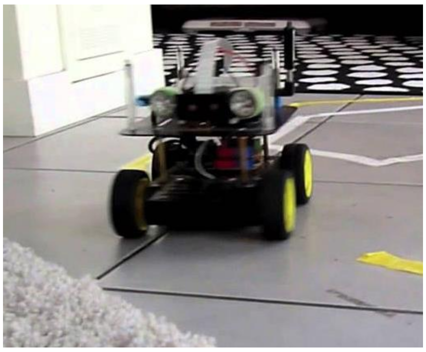


Figure 2-2-2 Robot with Raspberry Pi and Arduino by Rene

2.2.3 Vision-based Robotic work cell using Dobot Robot arm

There is a student from Assentica Robotics had developed a vision-based robot using OpenCV and other APIs. There is two robot arm in the project. The first arm will carry the item and place it on the conveyor when the package had come into the camera frame, the location of the package had been localising, and the system can move the second arm to carry the package on the conveyor.

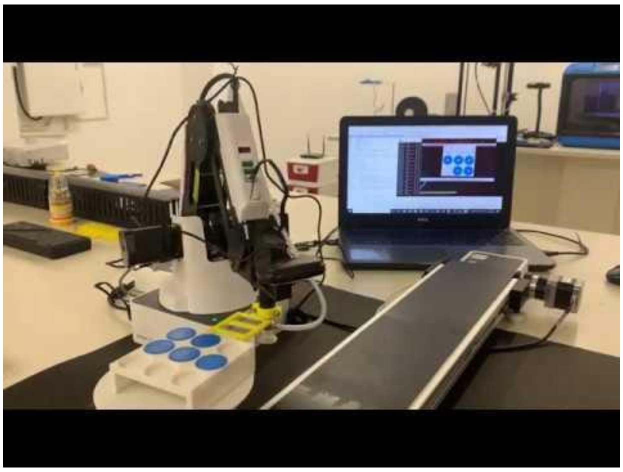


Figure 2-2-3 Vision-based Robotic work cell using Dobot Robot arm

2.2.4 GoPiGo with Google Cloud Vision

Dexter Industries had released a tutorial project that introduces the user to use GoPiGo with Google Vision API to recognise the label of the object. Other than that, the user is allowed to using the Google Vision API the detect the face that capture by the camera to recognise the emotion of the person in image whether he or she is angry or joy or others. Not only that, but the user is also to use the API to detect multiple objects in the image instead of the only label.

Chapter 3: System Methodology

3.1 System Development Models

3.1.1 Waterfall model

The waterfall model was the first model of SDLC that had been used widely at the past of the many years to ensure the success of the project. In this approach, the whole of the process will be divided into different phases. All the outcome of one step acts as the input for the next step. This approach is suitable for the project, which is short. The is the more most straightforward, easy to use and understand approach compare to others. It can work well when the requirement is well understood. When the development had completed, it will be tested with the initial need.

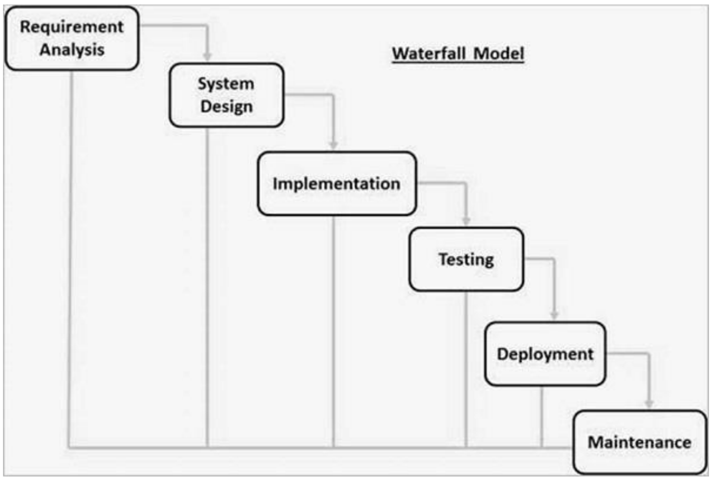


Figure 3-1-1 Waterfall approach

3.1.2 Iterative Model

Iterative model a model that system are developed through repeated cycles and in smaller portions of time. It looks like there are different mini waterfalls. It does not start with full specification of the requirement. The development will begin by specifying and implement some part of the software. For example, the developer fulfils some requirements, then test, evaluate and pinpoint further requirement. A new version of the software will be developed in each phase, and the whole process will be repeated until the complete system is ready. But this approach, the project issues can be detected and changes earlier. In this approach, some functionality can be developed in the early of the development. It required more resource because the process will be repeated frequently if the complete system had not to be developed

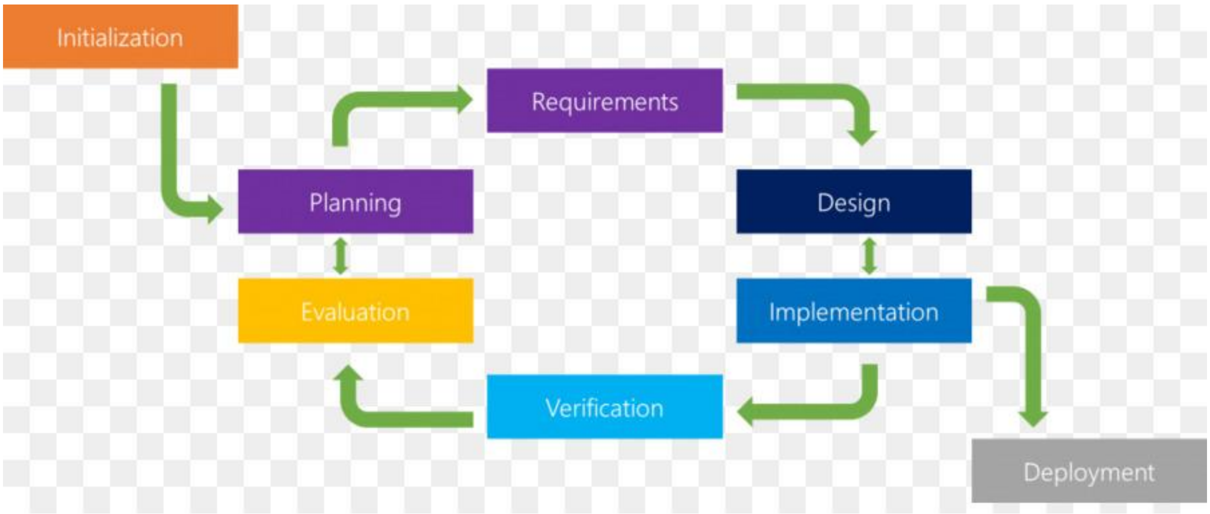


Figure 3-1-2 Iterative approach

3.1.3 Prototyping Model

In this model, requirement analysis will be started in the beginning. During this process, the user is being interviewed to know the requirement and the expectation of the system. After that, a quick but enough design will be implemented to meet the requirement of the user, but it is not yet complete design. This help to develop the prototype in the next stage. A small working model of the system is developed. After showing the system that developed to the user, the strength and the weakness of the system will be founded, and feedback will be collected from the user. If the user is not satisfied with the system, the developer needs to refine the prototype according to their feedback. This process of user evaluation and refining prototype will be repeated until all the user's requirement are met, and the final system will then be developed and been tested.

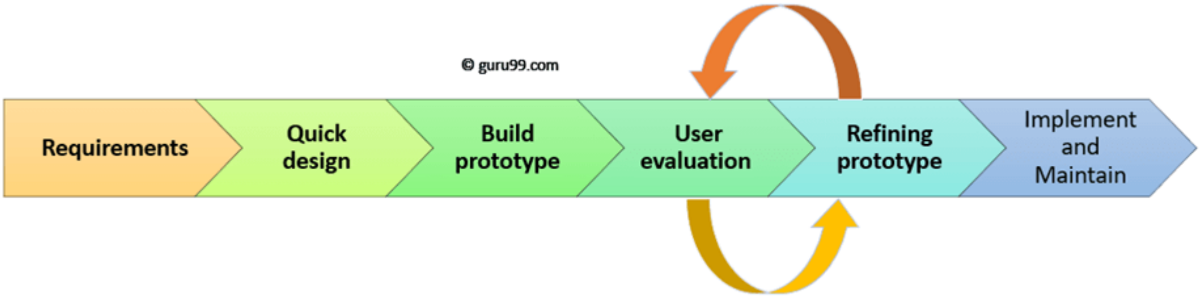


Figure 3-1-3 Prototyping Approach

3.1.4 Selected Model

Iterative model is selected for this project. The system development will be separated into a different version. More and more value will be added to the system during each cycle when a new requirement is needed for the system. By this approach, the flexibility of the changes for the system can be increased.

3.2 System Requirement (Technologies Involved)

3.2.1 Hardware

3.2.1.1 GoPiGo

In this project, GoPiGo with the Raspberry Pi 3 will be used in the project. Raspberry Pi 3 acts as a computer with its plug into computer technology. It has a 1.4Ghz 64-bit quad-core processor. It supports dual-band 2.4GHz and 5GHz wireless LAN and Bluetooth 4.2. Not only this, but it is also able to plug in a microSD card to install the operating system. This project will use 16GB microSD cars. It also contains an HDMI port to connect to the computer monitor. It was also had 4 USB 2.0 ports to connect the mouse and the keyboard. It is using the Raspbian operating system. It must be connected to a USB micro B cable to have the power supply. But in this project, the Raspberry Pi will be power up by the battery that attaches on the GoPiGo. Besides that, GoPiGo also comes with the distance sensor at the front of the robot.

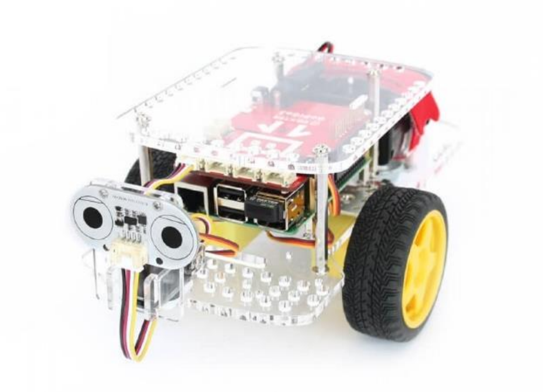


Figure 3-2-1-1 GoPiGo

3.2.1.2 Raspberry Pi Camera

To allow the robot to capture an image, a camera is needed for the project. A Raspberry Pi camera is needed to install on the Raspberry Pi to capture the image of the object. It is lightweight, and it can capture a high-quality image. With the camera, the user could explore the space by using the video that capture by the robot.

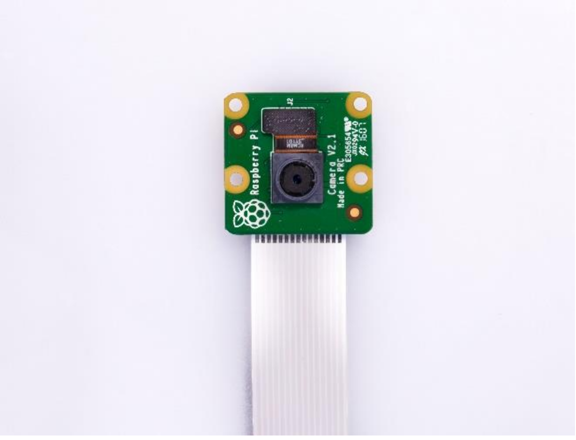


Figure 3-2-1-2 Raspberry Pi Camera

3.2.1.3 Servo Kit

In this project, two servos are needed to install on the GoPiGo. One of the servos is attached to the distance sensor and camera. This is to move the camera around the view of the environment of the robot. Another servo will the used to attach a stick at the side of the GoPiGo the carry the target item at the end of the project.

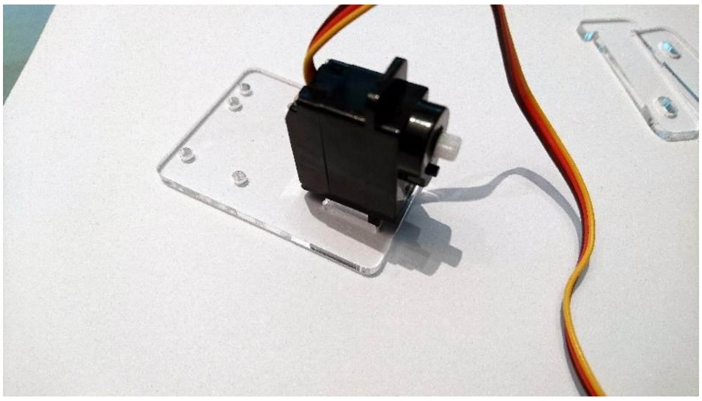


Figure 3-2-1-3 Servo Kit

3.2.2 Software

3.2.2.1 Raspbian Operating system

Operating System is needed for a computer to manages the computer hardware and software. Raspbian Operating System is a Debian-based computer operating system, and it works efficiently in the Raspberry Pi computer. It includes tools such as browsing, python programming and a GUI desktop. It is easy for install because it comes with over 35,000 packages, pre-compiled software bundled in a nice format. It is optimised for best performance on Raspberry Pi.

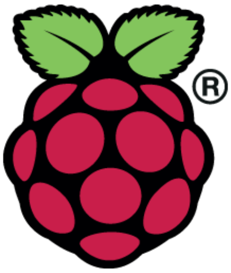


Figure 3-2-2-1 Raspbian

3.2.2.2 Python 3 Programming Language

Python is a powerful high-level, object-oriented programming language which is widely used. With Python, the concept could be express in fewer lines compared to other programming languages such c language. Because of this, it can be used to solve a complex problem. It supports automatic memory management and has a more extensive and comprehensive standard library. It is available in many operating systems.

Figure 3-2-2-2 Python

3.2.2.3 Google Cloud Vision API

Google Cloud Vision API allows us to integrate the vision detection figures within the applications. It classifies the images into thousands of the categories quickly to detect the individual object within the images uploaded. It also can find and reads the printed words that contained within the images to recognize more accurate.

Figure 3-2-2-3 Google Cloud Vision API

3.3 Functional Requirement

At the end of this project, a path finding robot will be developed. The robot can detect multiple object in the environment. Other than that, the robot is able to find the item that required by the user.

3.4 Project Milestone

Activity	Start Date	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14
FYP 1															
Self Learning on Python Programming Language	13/01/2020	🟩	🟩												
Learning of programming using Raspberry Pi	20/01/2020		🟩	🟩											
Trying basic movement coding on GoPiGo	05/02/2020				🟩										
Implement Web Browser Streaming on GoPiGo	12/02/2020					🟩	🟩	🟩	🟩						
Implement use of Google Vision API on GoPiGo	12/02/2020					🟩	🟩	🟩	🟩						
Testing of result of accuracy of the Google Vision	11/03/2020									🟩					
FYP 2															
Research on robot vision using other cloud platform	07/07/2020			🟩	🟩	🟩	🟩	🟩							
Trying implement using other cloud platform	04/08/2020								🟩	🟩					
Finalise system	18/08/2020										🟩	🟩			
Finalise Report	01/09/2020												🟩	🟩	
System Demonstration	14/09/2020														🟩

Table 3-4-1 Timeline for the project

Chapter 4: System Design

4.1 System Architecture

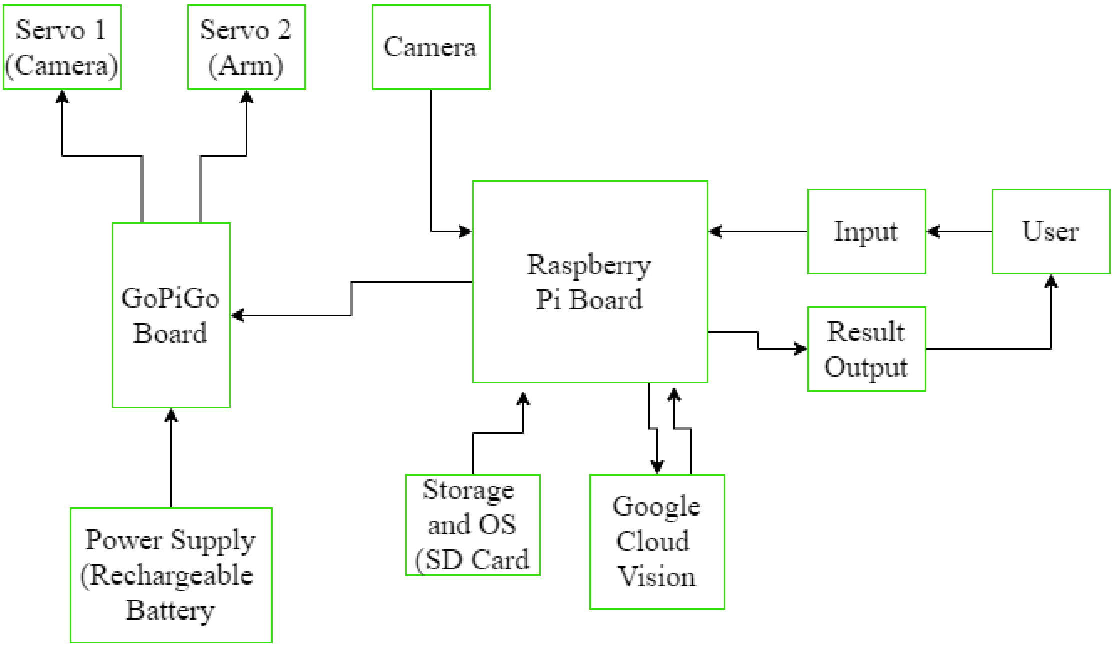


Figure 4-1 System Architecture

4.2 Functional Modules in the System

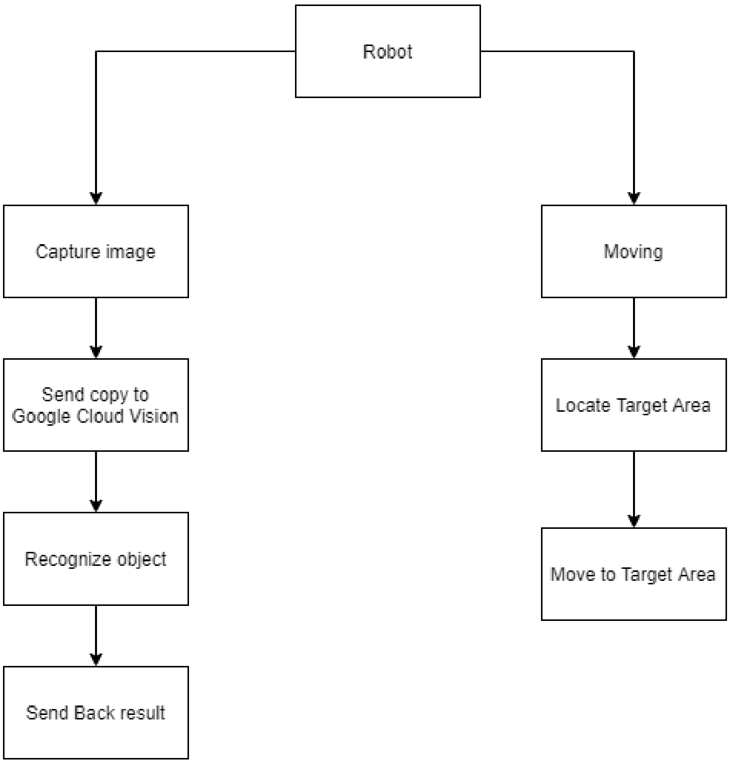


Figure 4-2 Functional Modules

4.3 System Flow

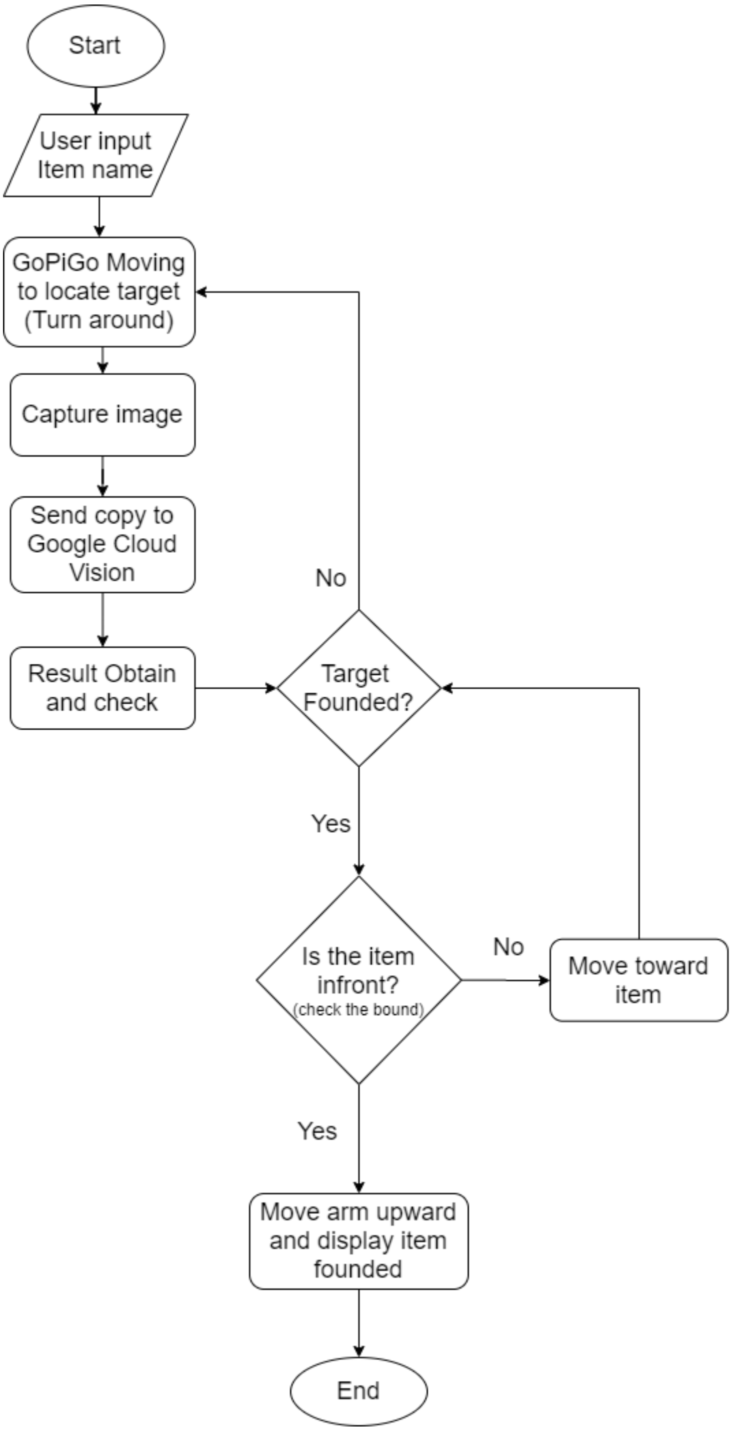

Start
User input Item name
GoPiGo Moving to locate target (Turn around)
Capture image
Send copy to Google Cloud Vision
Result Obtain and check
Target Founded?
No
Yes
Is the item infront? (check the bound)
No
Move toward item
Yes
Move arm upward and display item founded
End

Figure 4-3 System Flow

Chapter 5: System Implementation

5.1 Hardware Setup

In this project, the robot that been used is GoPiGo, all the hardware including the camera and the servo are already installed on the board.

5.2 Software Setup

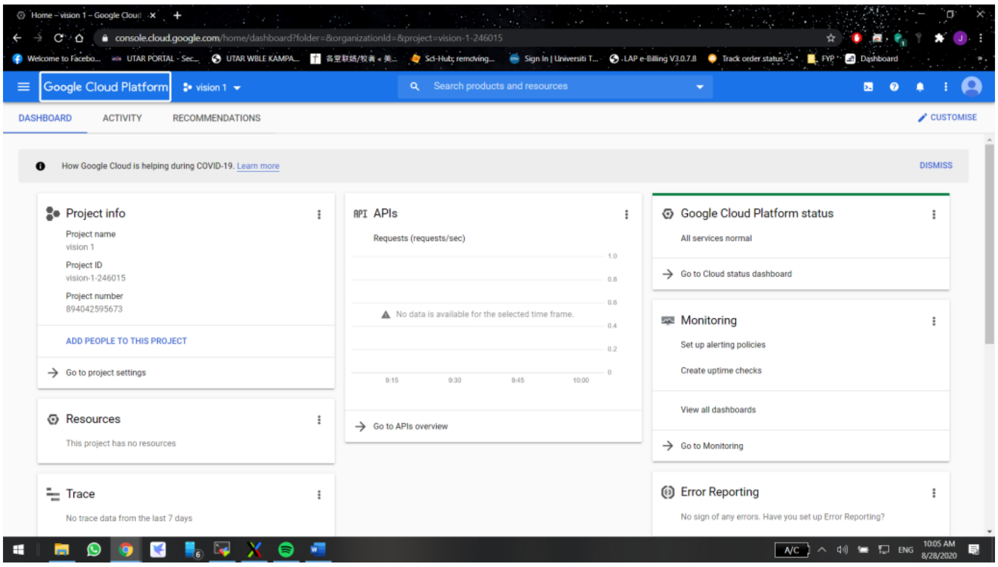
In this project, the operating system had to be installed into the SD card for the robot. The operating system that been used is Raspbian OS which can be download from Dexter Industries official website. Besides that, Etcher had to been installed to install the Raspbian into the SD card.

5.3 Setting and Configuration

In this project, there is two platform needed to be configured before running the coding for the robot to carry out the activity, which are Google Cloud part and GoPiGo part.

5.3.1 Google Cloud

1. Opening Google Cloud Platform. (Before that, must have an account which already register with google cloud platform and have set up billing account)



How Google Cloud is helping during COVID-19. Learn more

Project info

Project name
vision 1

Project ID
vision-1-246015

Project number
894042595673

ADD PEOPLE TO THIS PROJECT

Go to project settings

Resources

This project has no resources

Trace

No trace data from the last 7 days

APIs

Requests (requests/sec)

No data is available for the selected time frame.

Go to APIs overview

Google Cloud Platform status

All services normal

Go to Cloud status dashboard

Monitoring

Set up alerting policies

Create uptime checks

View all dashboards

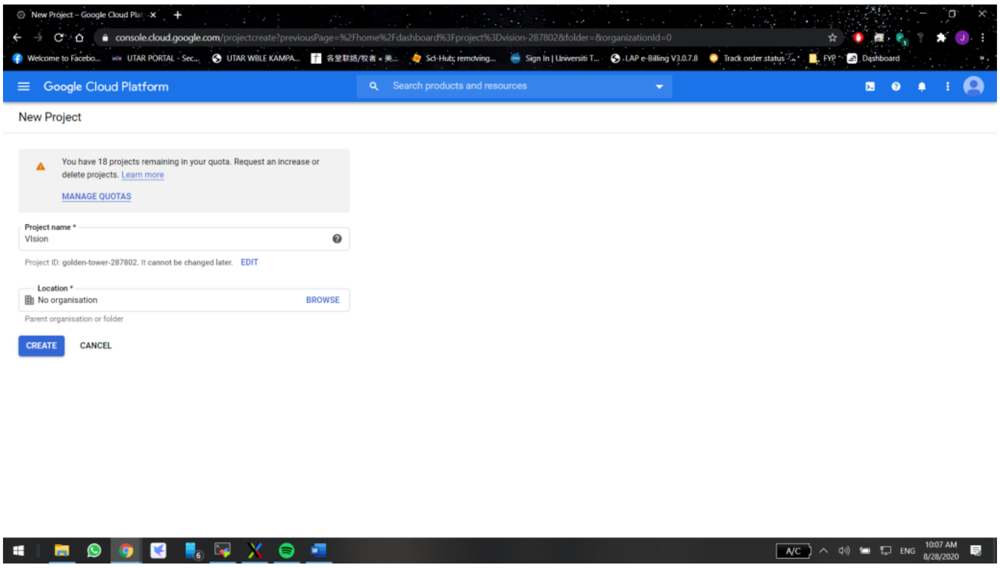
Go to Monitoring

Error Reporting

No sign of any errors. Have you set up Error Reporting?

Figure 5-3-1-1 Google Cloud Configuration 1

2. Create a new project. In this example, named it as Vision.


New Project – Google Cloud Plat
console.cloud.google.com/projectcreate?previousPage=%2Fhome%2Fdashboard%3Fproject%3Dvision-287802&folder=&organizationId=0
Welcome to Facebo...
UTAR PORTAL - Sec...
UTAR WBLE KAMPA...
Sci-Hub; removing...
Sign In | Universiti T...
LAP e-Billing V3.0.7.8
Track order status
FYP
Dashboard
Google Cloud Platform
Search products and resources
A/C
ENG
10:07 AM
8/28/2020

New Project

You have 18 projects remaining in your quota. Request an increase or delete projects. Learn more

MANAGE QUOTAS

Project name *
Vision

Project ID: golden-tower-287802. It cannot be changed later. EDIT

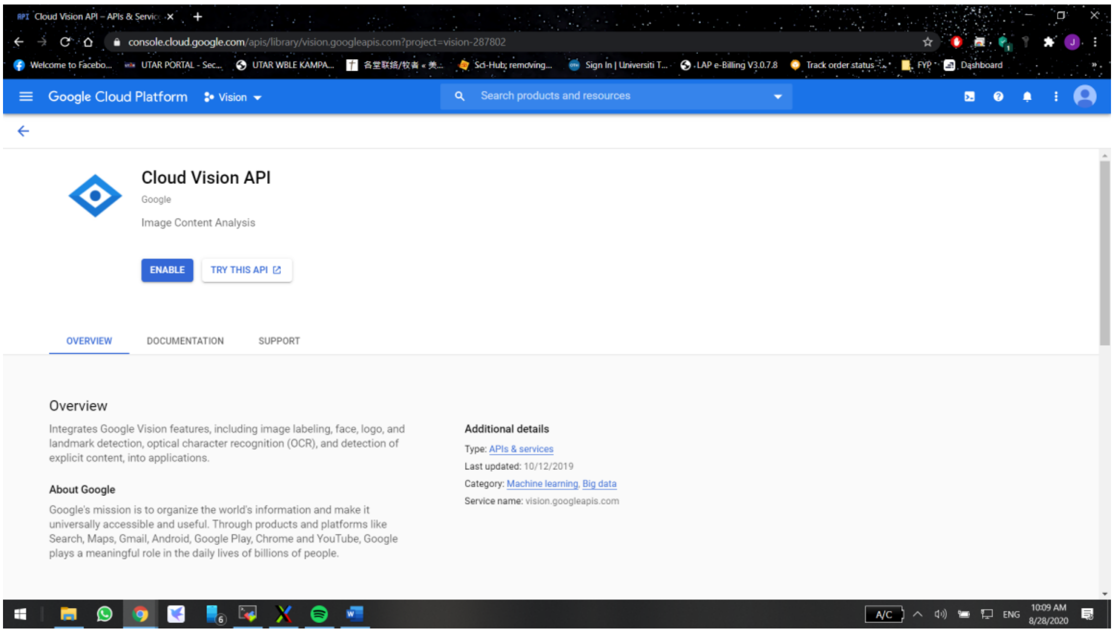
Location *
No organisation BROWSE

Parent organisation or folder

CREATE CANCEL

Figure 5-3-1-2 Google Cloud Configuration 2

3. Enable the Google Cloud Vision API



console.cloud.google.com/apis/library/vision.googleapis.com?project=vision-287802

Cloud Vision API

Google

Image Content Analysis

Overview

Integrates Google Vision features, including image labeling, face, logo, and landmark detection, optical character recognition (OCR), and detection of explicit content, into applications.

About Google

Google's mission is to organize the world's information and make it universally accessible and useful. Through products and platforms like Search, Maps, Gmail, Android, Google Play, Chrome and YouTube, Google plays a meaningful role in the daily lives of billions of people.

Additional details

Type: APIs & services

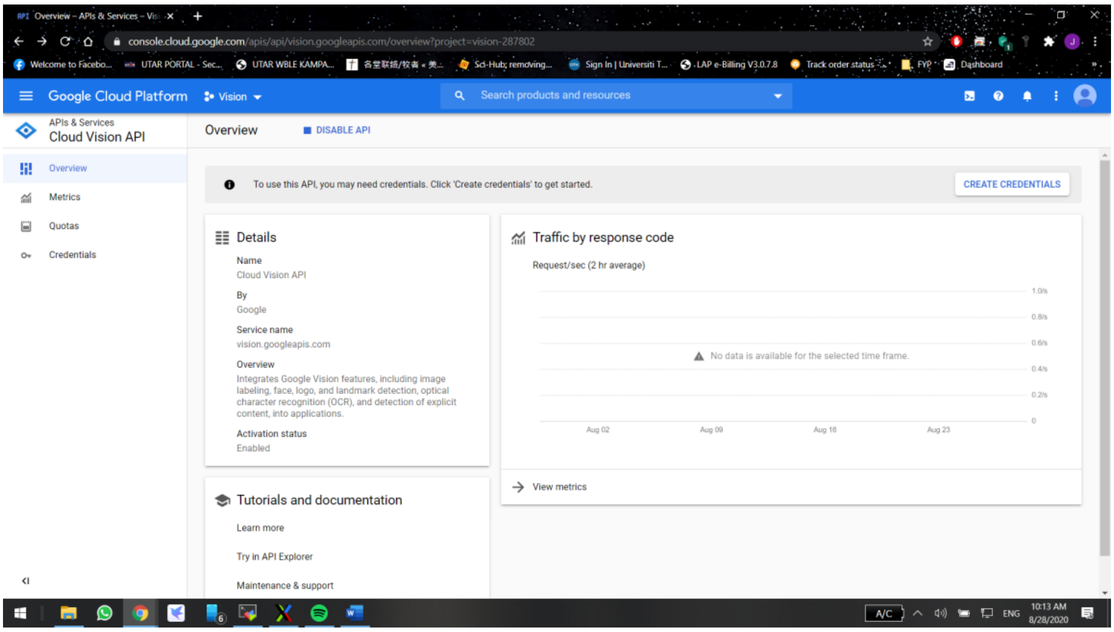
Last updated: 10/12/2019

Category: Machine learning, Big data

Service name: vision.googleapis.com

Figure 5-3-1-3 Google Cloud Configuration 3

4. Getting the JSON key which used to handle the authentication to use the Google Cloud Account. Firstly, we create the credentials for Google Cloud Vision API.



console.cloud.google.com/apis/api/vision.googleapis.com/overview?project=vision-287802

To use this API, you may need credentials. Click 'Create credentials' to get started.

Details

Name: Cloud Vision API

By: Google

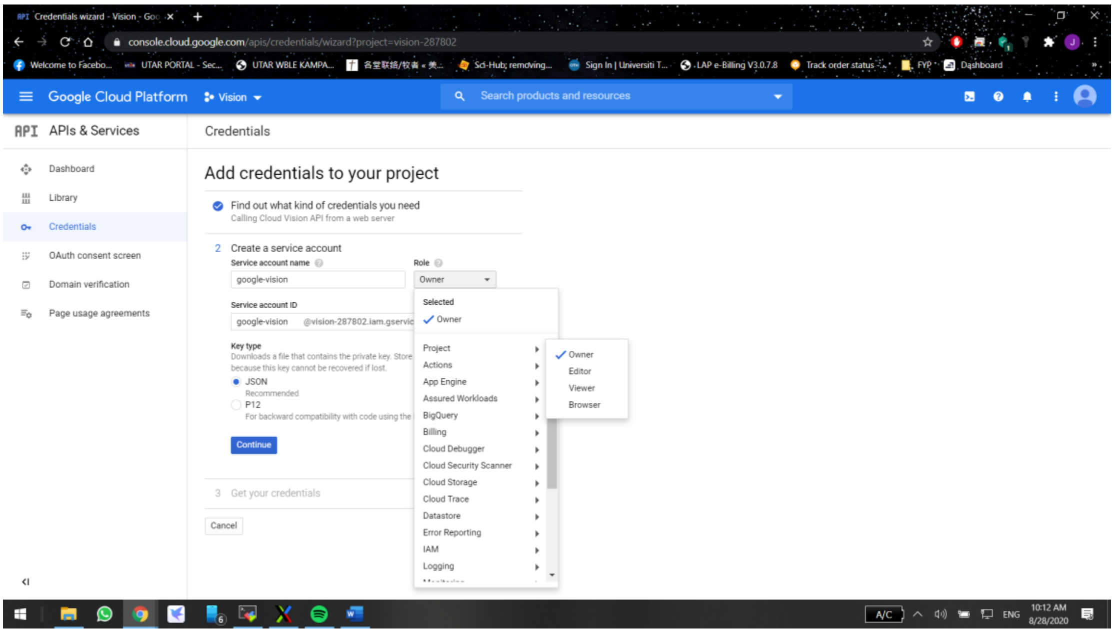
Service name: vision.googleapis.com

Overview: Integrates Google Vision features, including image labeling, face, logo, and landmark detection, optical character recognition (OCR), and detection of explicit content, into applications.

Activation status: Enabled

Figure 5-3-1-4 Google Cloud Configuration 4

5. Enter the service account name. Change the role to Project/Owner to get full control. Change the key type to JSON. After that click continue to download the file.



Add credentials to your project

Find out what kind of credentials you need
Calling Cloud Vision API from a web server

2 Create a service account

Service account name: google-vision

Role: Owner

Service account ID: google-vision @vision-287802.iam.gservic

Key type
Downloads a file that contains the private key. Store because this key cannot be recovered if lost.

● JSON
Recommended

○ P12
For backward compatibility with code using the

3 Get your credentials

Figure 5-3-1-5 Google Cloud Configuration 5

6. Rename the json file so that it is easy for implement in later part. This example below, rename it as vision.json.

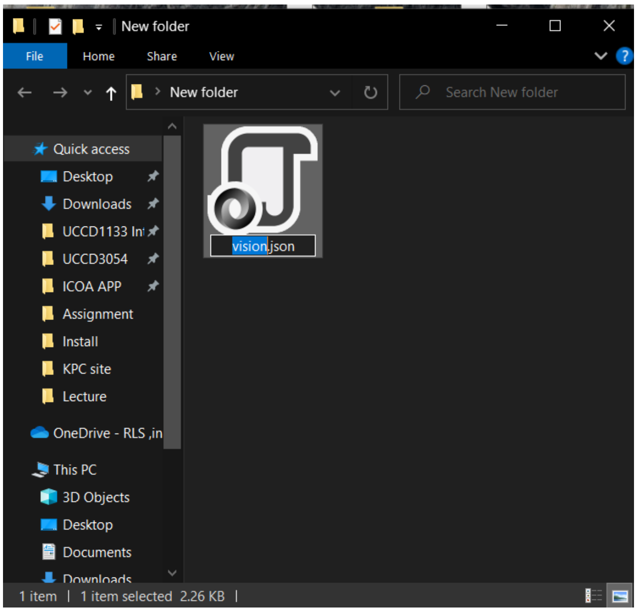
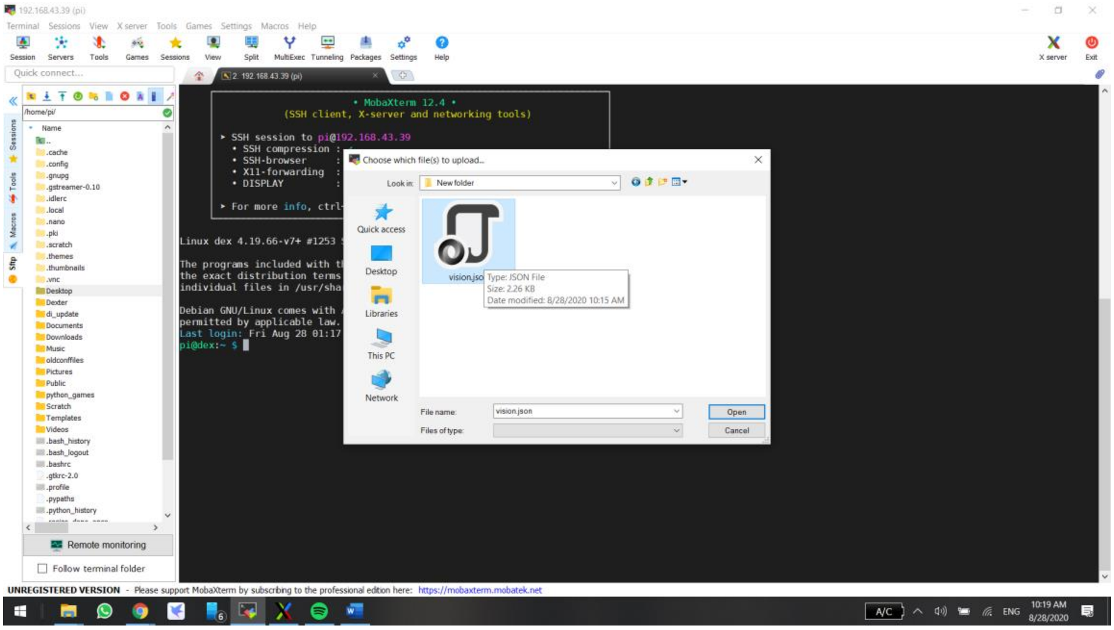


Figure 5-3-1-6 Google Cloud Configuration 6

7. Upload the file to GoPiGo, can use FTP prohram such as FileZilla or Samba. In this example, MobaXterm is used. In the later part, MobaXterm also been used as wireless display to view the screen for GoPiGo instead of using cabling.



```
                    • MobaXterm 12.4 •
        (SSH client, X-server and networking tools)

 ➤ SSH session to pi@192.168.43.39
   • SSH compression :
   • SSH-browser     :
   • X11-forwarding  :
   • DISPLAY         :

 ➤ For more info, ctrl-

Linux dex 4.19.66-v7+ #1253

The programs included with t
the exact distribution terms
individual files in /usr/sha

Debian GNU/Linux comes with
permitted by applicable law.
Last login: Fri Aug 28 01:17
pi@dex:~ $
```

Choose which file(s) to upload...

Look in: New folder

vision.json — Type: JSON File, Size: 2.26 KB, Date modified: 8/28/2020 10:15 AM

File name: vision.json

Files of type:

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

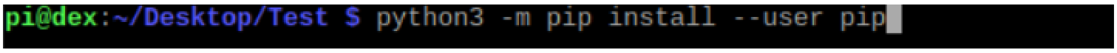
Figure 5-3-1-7 Google Cloud Configuration 7

5.3.2 GoPiGo

As mention before, MobaXterm is used to connect to the GoPiGo for wireless display for the robot. At the first time using the GoPiGo, there is two command must to be run in order to get the latest operating system. To get the latest update of operating system and hardware profiles by following commands (Make sure the robot had internet connection):

```
sudo apt-get update

sudo apt-get upgrade
```

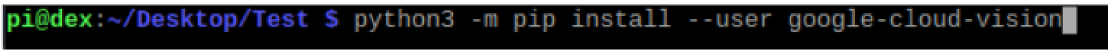
1. After updating, there are a few of libraries is need to be install. Firstly, the libraries for Python3.



```
pi@dex:~/Desktop/Test $ python3 -m pip install --user pip
```

Figure 5-3-2-1 GoPiGo Configuration 1

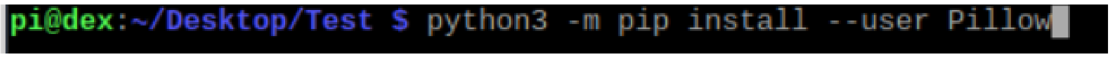
2. Install Google API Python Client.



```
pi@dex:~/Desktop/Test $ python3 -m pip install --user google-cloud-vision
```

Figure 5-3-2-2 GoPiGo Configuration 2

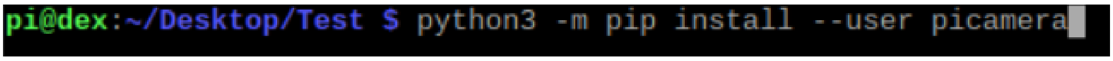
3. Install Python Imaging Library



```
pi@dex:~/Desktop/Test $ python3 -m pip install --user Pillow
```

Figure 5-3-2-3 GoPiGo Configuration 3

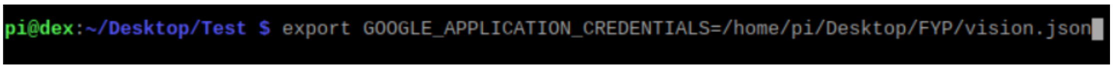
4. Install latest version of Picamera



```
pi@dex:~/Desktop/Test $ python3 -m pip install --user picamera
```

Figure 5-3-2-4 GoPiGo Configuration 4

5. Make the JSON file available to any application that run (must be run one more time when a new terminal had been used). This is to get authentication for google cloud. The path is depend on the location of the json file that downloaded before.



```
pi@dex:~/Desktop/Test $ export GOOGLE_APPLICATION_CREDENTIALS=/home/pi/Desktop/FYP/vision.json
```

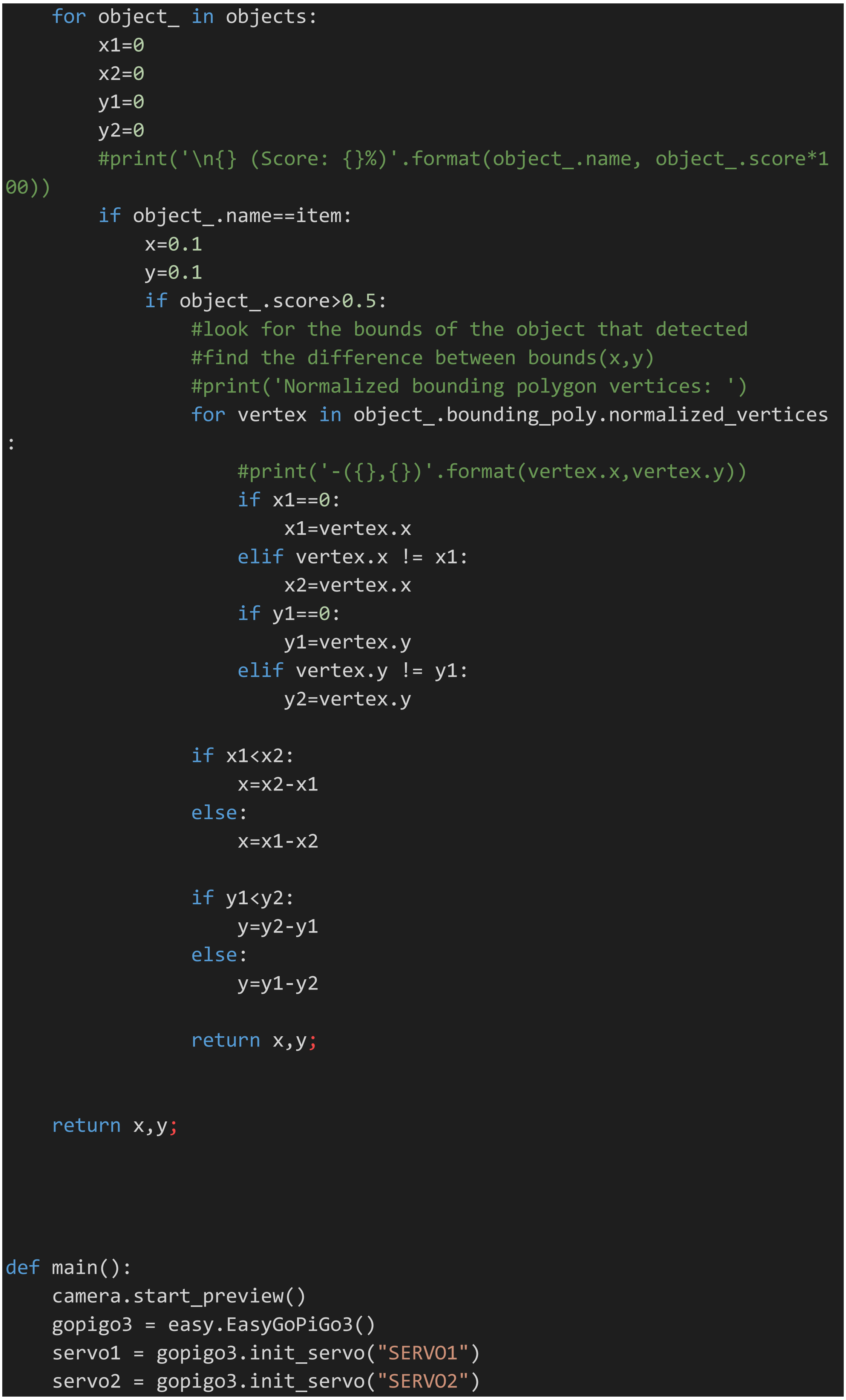
Figure 5-3-2-5 GoPiGo Configuration 5

5.4 System Operation

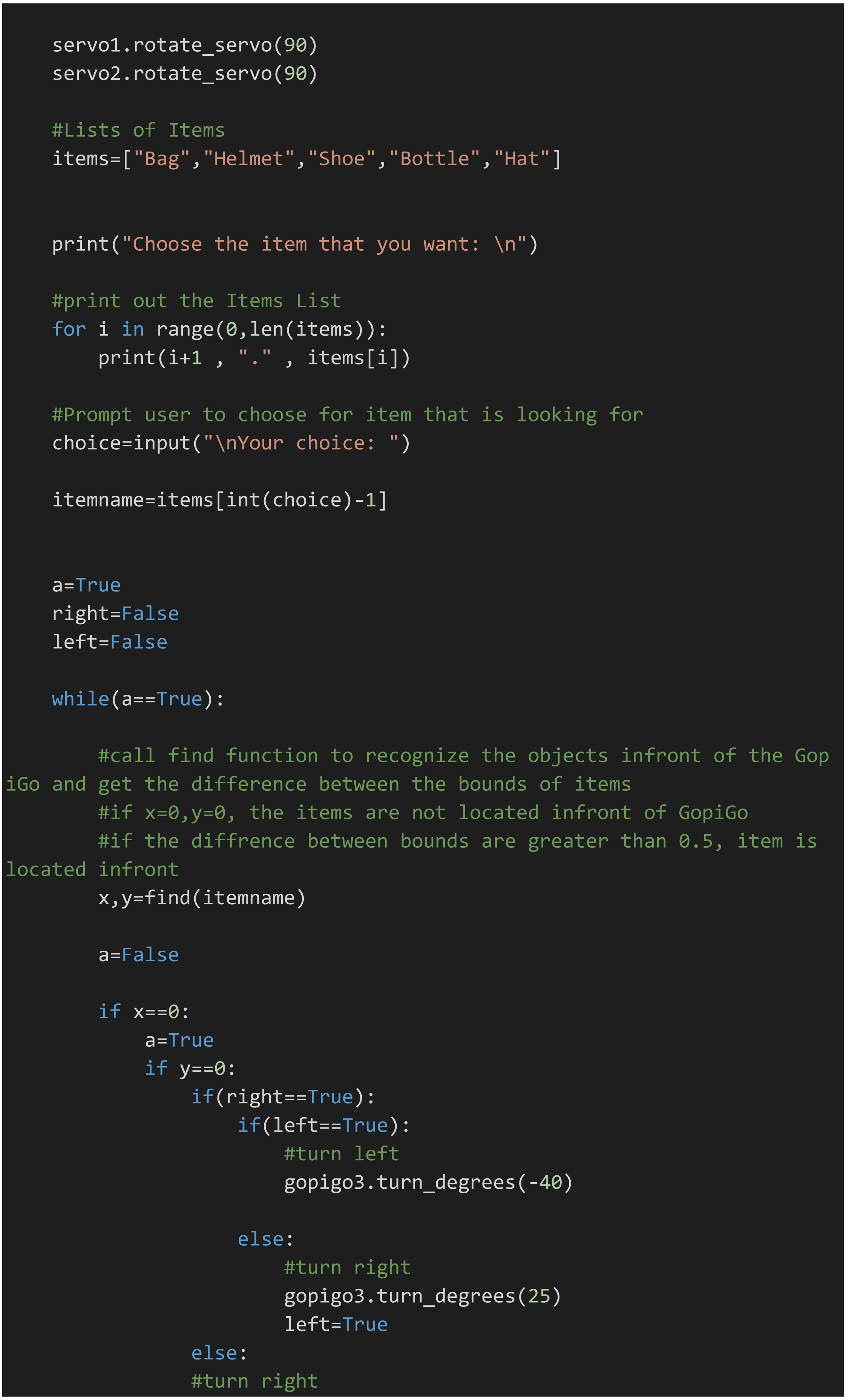
5.4.1 Coding

vision.py

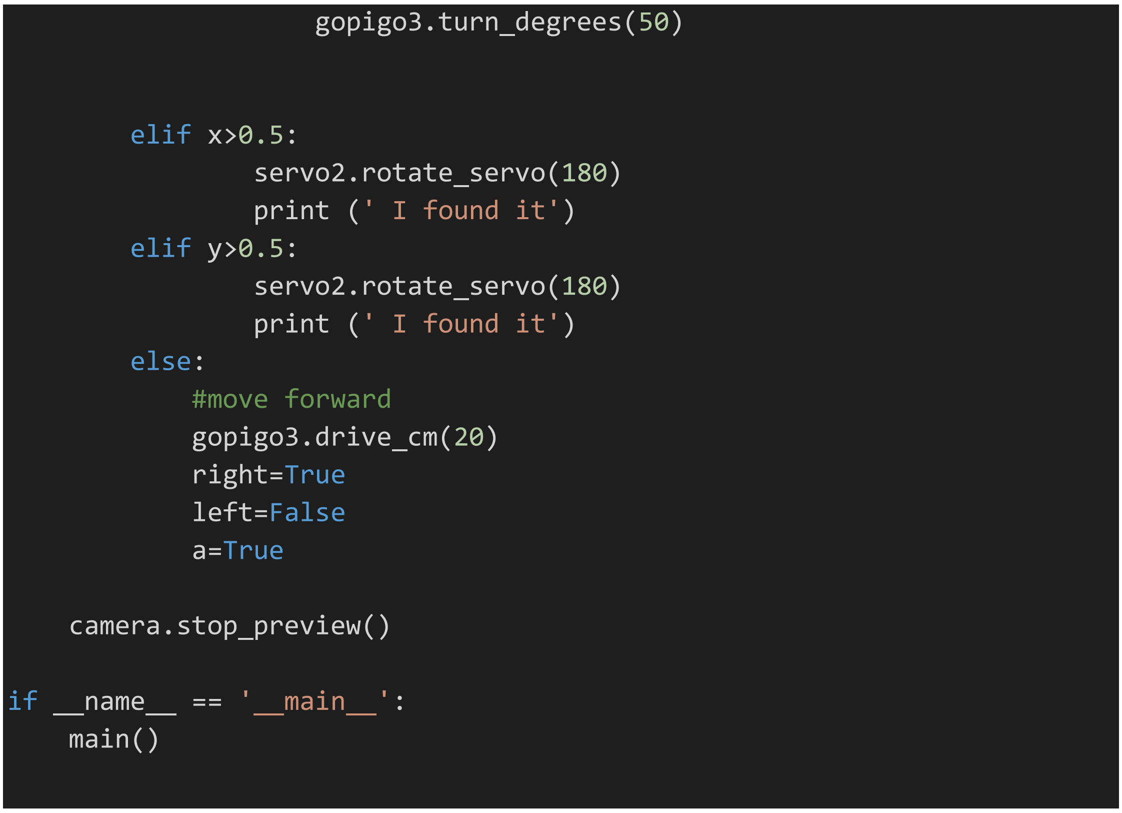
```
from picamera import PiCamera
import easygopigo3 as easy
from google.cloud import vision
import time

client = vision.ImageAnnotatorClient()
camera = PiCamera()

#capture currebt vision
def takephoto():

    time.sleep(1)
    camera.capture('/home/pi/Desktop/FYP/image.jpg')



def find(item):

    takephoto() # First take a picture

    #open the image that capture just now
    with open('/home/pi/Desktop/FYP/image.jpg', 'rb') as image_file:
        content = image_file.read()

    #upload the image to google vision api for recognization
    image = vision.types.Image(content=content)
    objects = client.object_localization(
            image=image).localized_object_annotations

    x=0
    y=0
    #print('Number of objects found: {}'.format(len(objects)))
```



```
    for object_ in objects:
        x1=0
        x2=0
        y1=0
        y2=0
        #print('\n{} (Score: {}%)'.format(object_.name, object_.score*1
00))
        if object_.name==item:
            x=0.1
            y=0.1
            if object_.score>0.5:
                #look for the bounds of the object that detected
                #find the difference between bounds(x,y)
                #print('Normalized bounding polygon vertices: ')
                for vertex in object_.bounding_poly.normalized_vertices
:
                    #print('-({},{})'.format(vertex.x,vertex.y))
                    if x1==0:
                        x1=vertex.x
                    elif vertex.x != x1:
                        x2=vertex.x
                    if y1==0:
                        y1=vertex.y
                    elif vertex.y != y1:
                        y2=vertex.y

                if x1<x2:
                    x=x2-x1
                else:
                    x=x1-x2

                if y1<y2:
                    y=y2-y1
                else:
                    y=y1-y2

                return x,y;


    return x,y;




def main():
    camera.start_preview()
    gopigo3 = easy.EasyGoPiGo3()
    servo1 = gopigo3.init_servo("SERVO1")
    servo2 = gopigo3.init_servo("SERVO2")
```



```
    servo1.rotate_servo(90)
    servo2.rotate_servo(90)

    #Lists of Items
    items=["Bag","Helmet","Shoe","Bottle","Hat"]


    print("Choose the item that you want: \n")

    #print out the Items List
    for i in range(0,len(items)):
        print(i+1 , "." , items[i])

    #Prompt user to choose for item that is looking for
    choice=input("\nYour choice: ")

    itemname=items[int(choice)-1]


    a=True
    right=False
    left=False

    while(a==True):

        #call find function to recognize the objects infront of the Gop
iGo and get the difference between the bounds of items
        #if x=0,y=0, the items are not located infront of GopiGo
        #if the diffrence between bounds are greater than 0.5, item is
located infront
        x,y=find(itemname)

        a=False

        if x==0:
            a=True
            if y==0:
                if(right==True):
                    if(left==True):
                        #turn left
                        gopigo3.turn_degrees(-40)

                    else:
                        #turn right
                        gopigo3.turn_degrees(25)
                        left=True
                else:
                #turn right
```



```
                    gopigo3.turn_degrees(50)


        elif x>0.5:
                servo2.rotate_servo(180)
                print (' I found it')
        elif y>0.5:
                servo2.rotate_servo(180)
                print (' I found it')
        else:
            #move forward
            gopigo3.drive_cm(20)
            right=True
            left=False
            a=True

    camera.stop_preview()

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

5.4.2 Code Running

1. Place the item and the GoPiGo. In my example, there are six item around the GoPiGo. The items are bag, helmet, shoe, two bottles, and hat.

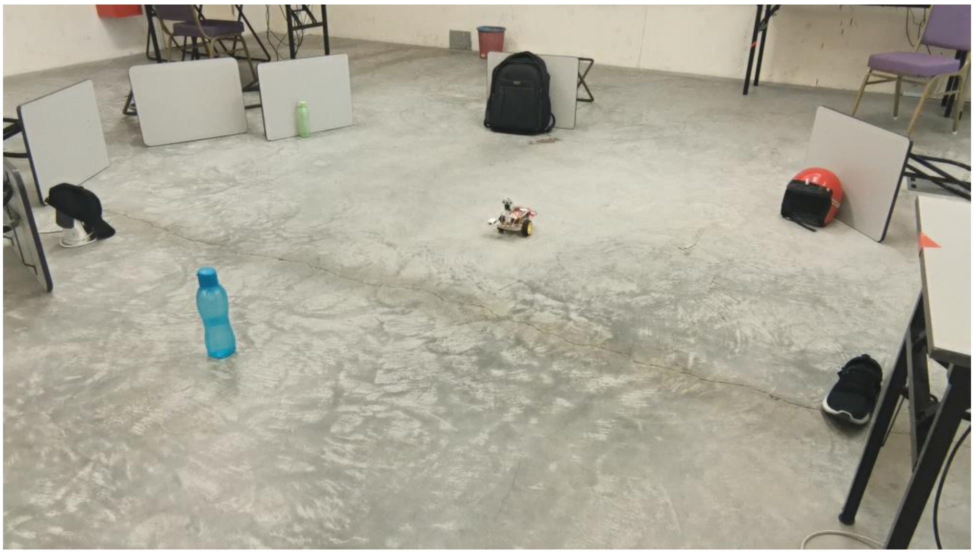
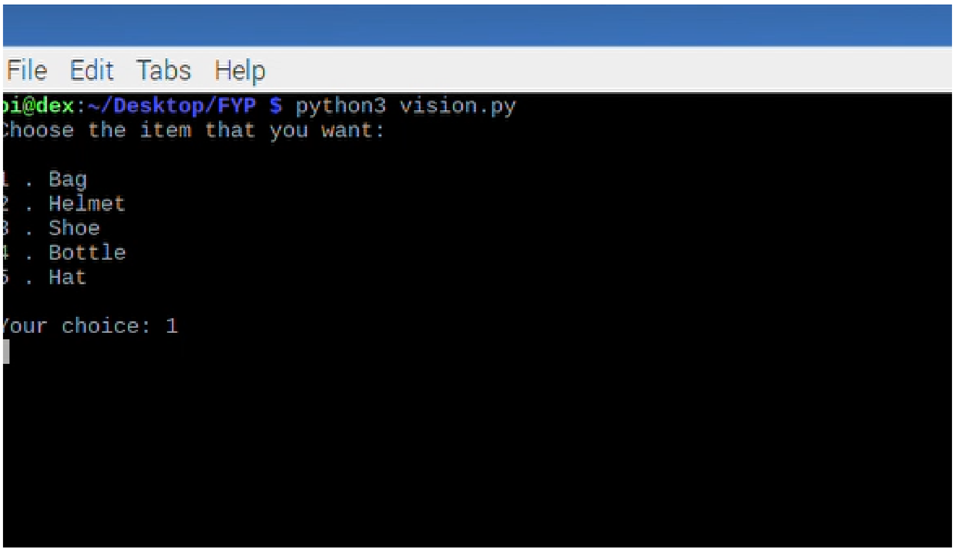


Figure 5-4-2-1 System Operation 1

2. Run the code. It will display the list of the item that can be choose. In this example, we choose 1 for bag.



```
pi@dex:~/Desktop/FYP $ python3 vision.py
Choose the item that you want:

1 . Bag
2 . Helmet
3 . Shoe
4 . Bottle
5 . Hat

Your choice: 1
```

Figure 5-4-2-2 System Operation 2

3. The robot will start turn itself until it find the bag.

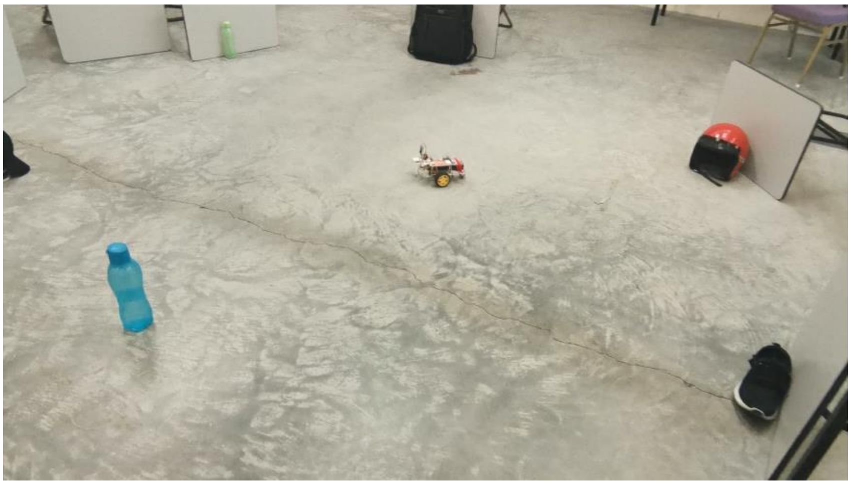


Figure 5-4-2-3 System Operation 3

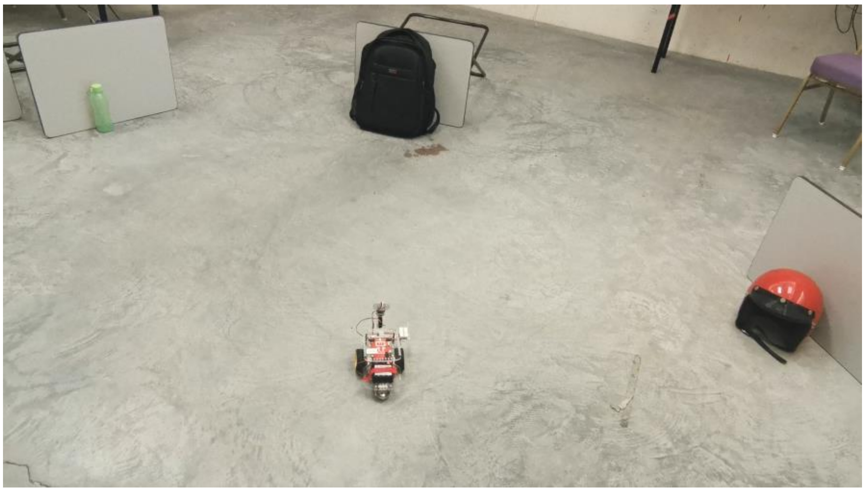


Figure 5-4-2-4 System Operation 4

4. When it found the bag, it will check whether the bag is near, if not, it will move toward the bag.

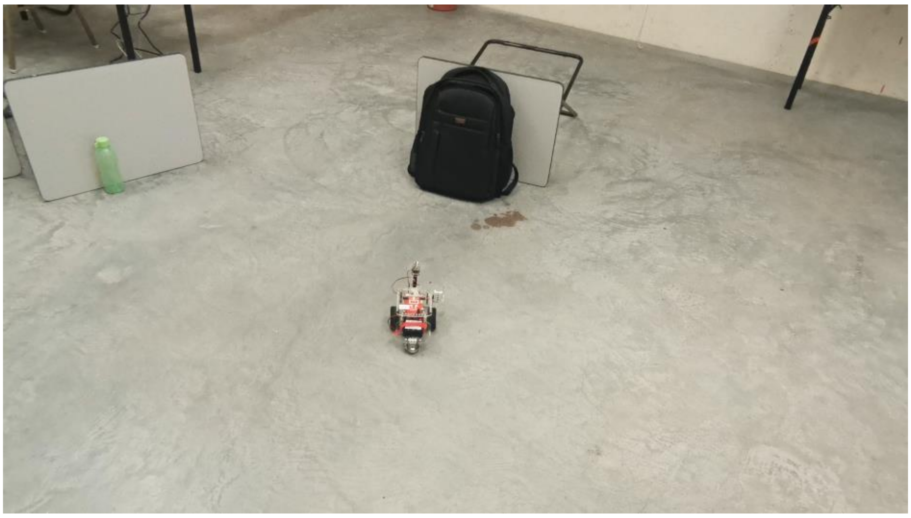
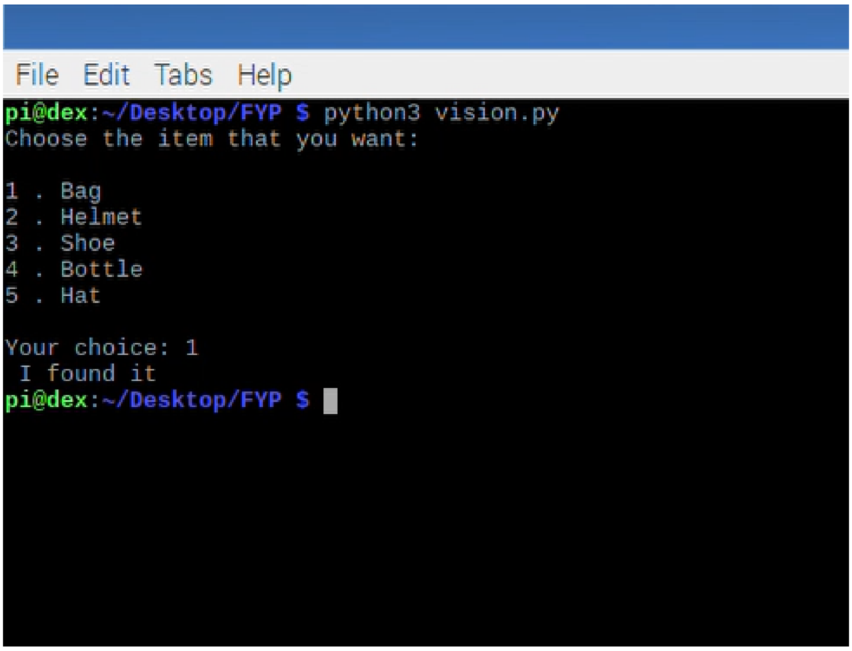


Figure 5-4-2-5 System Operation 5

5. After it ensure that the bag is in front it, it will turn the servo2 upward and display “I found it” on screen.



```
pi@dex:~/Desktop/FYP $ python3 vision.py
Choose the item that you want:

1 . Bag
2 . Helmet
3 . Shoe
4 . Bottle
5 . Hat

Your choice: 1
 I found it
pi@dex:~/Desktop/FYP $ 
```

Figure 5-4-2-6 System Operation 6

Chapter 6: System Evaluation and Discussion

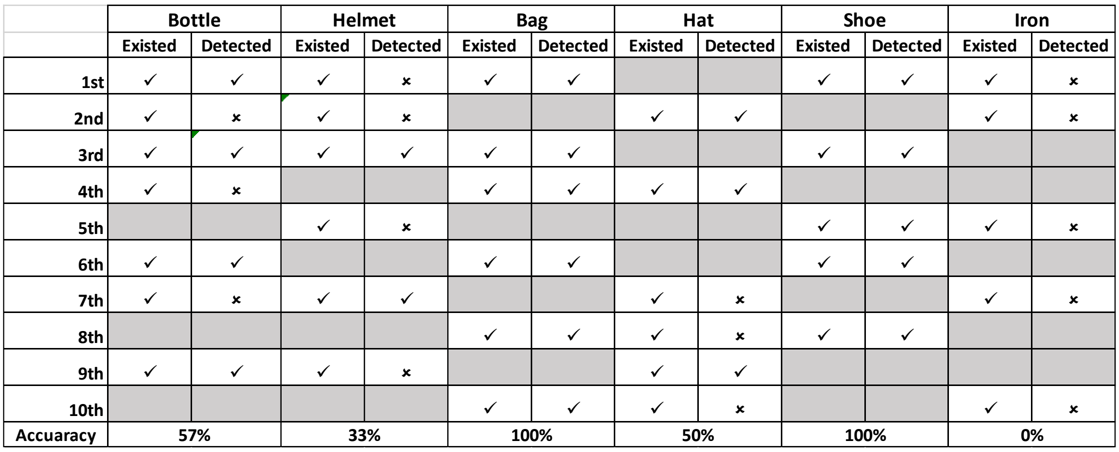
6.1 System Testing and Performance Metrics

Firstly, the accuracy of the object recognition of the google cloud will be tested. In this project, a list of items is been used to test the accuracy. Besides that, delay time for the GoPiGo to send and receive the data to and from the Google Cloud is been measured

6.2 Testing Setup and Result

6.2.1 Result of the accuracy of the object recognition through google cloud vision

The items that had been used to tested are bottle, helmet, bag, hat, shoe and Iron. Different combination of the items is placed together to test the accuracy for 10 rounds. After testing, the accuracy is around 40 to 50% as the item's shape are not similar as the item of the library. For example, the shoe could be recognised correctly for every test, but helmet is only 33%. Other than that, it could not recognize the iron in every test. The common item such as chair and table could also be recognised when the background of the image contains these items. Below is the testing result of the object recognition for every tests.



	Bottle		Helmet		Bag		Hat		Shoe		Iron	
	Existed	Detected	Existed	Detected	Existed	Detected	Existed	Detected	Existed	Detected	Existed	Detected
1st	✓	✓	✓	×	✓	✓			✓	✓	✓	×
2nd	✓	×	✓	×			✓	✓			✓	×
3rd	✓	✓	✓	✓	✓	✓			✓	✓		
4th	✓	×			✓	✓	✓	✓				
5th			✓	×					✓	✓	✓	×
6th	✓	✓			✓	✓			✓	✓		
7th	✓	×	✓	✓			✓	×			✓	×
8th					✓	✓	✓	×	✓	✓		
9th	✓	✓	✓	×			✓	✓				
10th					✓	✓	✓	×			✓	×
Accuaracy	57%		33%		100%		50%		100%		0%	

Figure 6-2-1-1 Object Recognition Testing Result

6.2.2 Delay time between Google Cloud and GoPiGo

The delay time had been tested by around five times. It depends on the internet connection. In this project, because of the needed of mobility, wireless internet connection is needed, the average delay time are two to five seconds. Even the GoPiGo is plug in the RJ45 ethernet cable, it also takes 2 seconds for the GoPiGo to get the result.

6.3 Project Challenges

The first challenge of the project is the accuracy of the recognition result. In the example in Chapter 5, when robot is been requested to find the bag, if the bag is too far, it can't correctly recognise the bag, it will result in "Luggage and Bags" from google cloud for a few time. Other than that, if there is some item that the google cloud cannot recognize, it will result in "Packaged goods". In order to have more accuracy result, the item must not been place too far.

Other than that, it is aim to let the robot to carry the item after it found the item at the beginning of the project, but it is hard for the GoPiGo to carry the item especially big item. There are two reason, first is GoPiGo only can connect to two servo, one had been used for the camera, and another one are placed beside the robot, but the movement of the servo are only upward or downward which is zero degree until 180 degree. It is hard to control the robot to carry the item by servo. Second, if the item is too big and heavy, it is impossible for the GoPiGo to carry as the GoPiGo is too light, it may cause rollover. For example, if the user is asking the GoPiGo to carry the helmet, It may cause rollover.

6.5 Objectives Evaluation

The main objective of the project to let the robot to perform some task after the recognition. In the end of the project, the robot are able to find the path itself toward the item that requested by the user. The sub objective which is to let the robot to carry the item back towards the user or starting point are not implemented successfully.

Chapter 7: Conclusion and Recommendation

7.1 Conclusion

In conclusion, a robot vision through the cloud will be developed. There is a lot of problems that will face by robot vision. The two significant issues are the background of the environment of the robot and the ability of the robot the distinguish the background and the object. As the robot had become the trend of the world, the robot vision is needed for the robot to carry out some works more efficiency.

7.2 Recommendation

It is recommended to use the OpenCV for recognition as it can do real time capture and recognizing. In this project, the robot is needed to capture the image and send to the google cloud for recognition then carry out the activity after getting the result from the google cloud. It may also cause some delay when send and receive the result to and from the google cloud depend on the internet connection.

References

Assemtica Robotics (2019) Vision based Robotic workcell using OpenCV, Custom APIs and Python using Dobot Robot arm.[online video] Available at: https://www.youtube.com/watch?v=UZPBamdOj2M [Accessed 10/4/2020]

ExplainingComputers (2019) NVIDIA JetBot: Jetson Nano Vision-Controlled AI Robot.[online video] Available at: https://www.youtube.com/watch?v=wKMWjIKaU68 [Accessed 10/4/2020]

Half, R. (2019, December 14). 6 basic SDLC methodologies: Which one is best? Retrieved from Robert Half: https://www.roberthalf.com.au/blog/employers/6-basic-sdlc-methodologies-which-one-best

Industries, D. (2017, November 28). GoPiGo: The Delightful Raspberry Pi Robot. Retrieved from Kickstarter: https://www.kickstarter.com/projects/john-cole/gopigo-the-delightful-raspberry-pi-robot

Kerley, N. (2018, May 2). 5 Challenges in Developing Sharp Robotic Vision. Retrieved from Machine Design: https://www.machinedesign.com/mechanical-motion-systems/article/21836692/5-challenges-in-developing-sharp-robotic-vision

Mohamad, M., Saman, M. Y., & Hitam, M. S. (2015). A Review on OpenCV. doi:10.13140/RG.2.1.2269.8721

Owen-Hill, A. (2017, November 20). Top 10 Challenges for Robot Vision. Retrieved from Robotiq Blog: https://blog.robotiq.com/top-10-challenges-for-robot-vision

Rene Bakx (2015) Robot with Raspberry Pi finds back its charging station using Python OpenCV. [online video] Available at: https://www.youtube.com/watch?v=vyJfpf-QpD4 [Accessed 10/4/2020]

Sami, M. (2012, March 15). Software Development Life Cycle Models and Methodologies. Retrieved from Mohamed Sami Blog: https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies

APPENDIX A POSTER

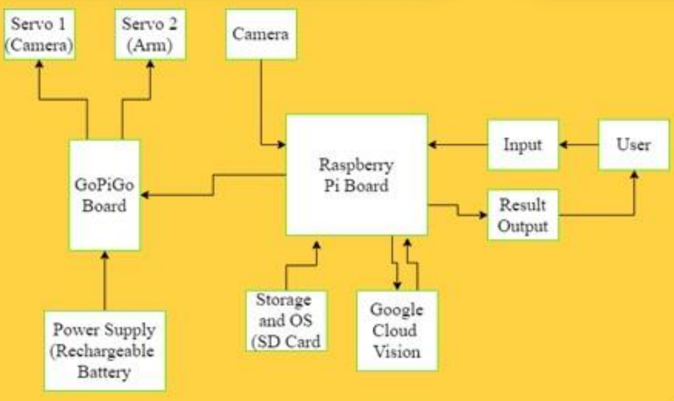
ROBOT VISION THROUGH CLOUD FOR EMBEDDED DEVICE

Why robot is needed ?

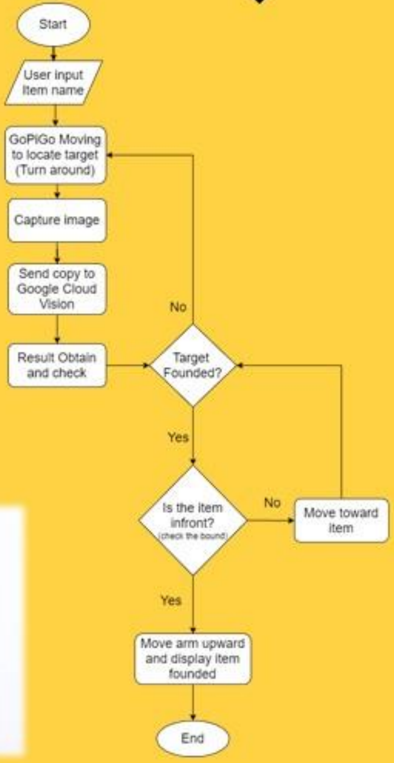
Does a robot need eyes ?

What is it used for ?

Is it necessary?



How it work?


Start
User input Item name
GoPiGo Moving to locate target (Turn around)
Capture image
Send copy to Google Cloud Vision
Result Obtain and check
Target Founded?
No
Yes
Is the item infront? (check the bound)
No
Move toward item
Yes
Move arm upward and display item founded
End

There is a lot of problem that needed to be overcome with robot vision. In this project, a path finding robot will be developed. The robot are able to find the item that required by the user by itself

APPENDIX B FINAL YEAR PROJECT WEEKLY REPORT

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4S1	**Study week no.: Week 2**
Student Name & ID: Jeffrey Wong Haw Yin	
Supervisor: Dr. Goh Hock Guan	
Project Title: Robot Vision Through Cloud For Embedded Device	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

In these two weeks, I had researched the for other platform for robot vision like Microsoft Azure and OpenCV library

2. WORK TO BE DONE

Work to be done for the next two weeks are continue view the way in using another cloud platform for robot vision

3. PROBLEMS ENCOUNTERED

There is no problem encountered in these two weeks

4. SELF EVALUATION OF THE PROGRESS

All the scheduled tasks are in good progress.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4S1	**Study week no.: Week 4**
Student Name & ID: Jeffrey Wong Haw Yin	
Supervisor: Dr. Goh Hock Guan	
Project Title: Robot Vision Through Cloud For Embedded Device	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

In these two weeks, I had found the way to using OpenCV but cannot try using GoPiGo because of MCO.

2. WORK TO BE DONE

Planned to do look for virtual machine for testing.

3. PROBLEMS ENCOUNTERED

There is no problem encountered on these two weeks

4. SELF EVALUATION OF THE PROGRESS

All the scheduled tasks are in good progress.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4S1	**Study week no.: Week 6**
Student Name & ID: Jeffrey Wong Haw Yin	
Supervisor: Dr. Goh Hock Guan	
Project Title: Robot Vision Through Cloud For Embedded Device	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

In these two weeks, trying to using virtual machine to implement the coding.

2. WORK TO BE DONE

Start to implement using GoPiGo in coming week

3. PROBLEMS ENCOUNTERED

Facing problem when wanted to use camera in virtual machine

4. SELF EVALUATION OF THE PROGRESS

All the scheduled tasks are in good progress.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4S1	**Study week no.: Week 8**
Student Name & ID: Jeffrey Wong Haw Yin	
Supervisor: Dr. Goh Hock Guan	
Project Title: Robot Vision Through Cloud For Embedded Device	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

I was stared to try for OpenCV library for object recognition

2. WORK TO BE DONE

If OpenCV run successfully, plan to do autonomy for robot

3. PROBLEMS ENCOUNTERED

Face problem when using OpenCV, TensorFlow is needed, but installing progress failed every time.

4. SELF EVALUATION OF THE PROGRESS

Trying to keep up with planning progress

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4S1	**Study week no.: Week 10**
Student Name & ID: Jeffrey Wong Haw Yin	
Supervisor: Dr. Goh Hock Guan	
Project Title: Robot Vision Through Cloud For Embedded Device	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Failed to install OpenCV and realized that OpenCV is not as a cloud platform, switching back to Google Cloud platform and do autonomy.

2. WORK TO BE DONE

Planning to finish remaining chapter of the report
Prepare demo video presentation for FYP presentation

3. PROBLEMS ENCOUNTERED

There is no problem encountered in these two weeks

4. SELF EVALUATION OF THE PROGRESS

All the scheduled tasks are in good progress.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4S1	**Study week no.: Week 12**
Student Name & ID: Jeffrey Wong Haw Yin	
Supervisor: Dr. Goh Hock Guan	
Project Title: Robot Vision Through Cloud For Embedded Device	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Finish the report and submit to Turnitin before submission.
Video demo had been done.

2. WORK TO BE DONE

Preparing for the presentation

3. PROBLEMS ENCOUNTERED

There is no problem encountered on these two weeks

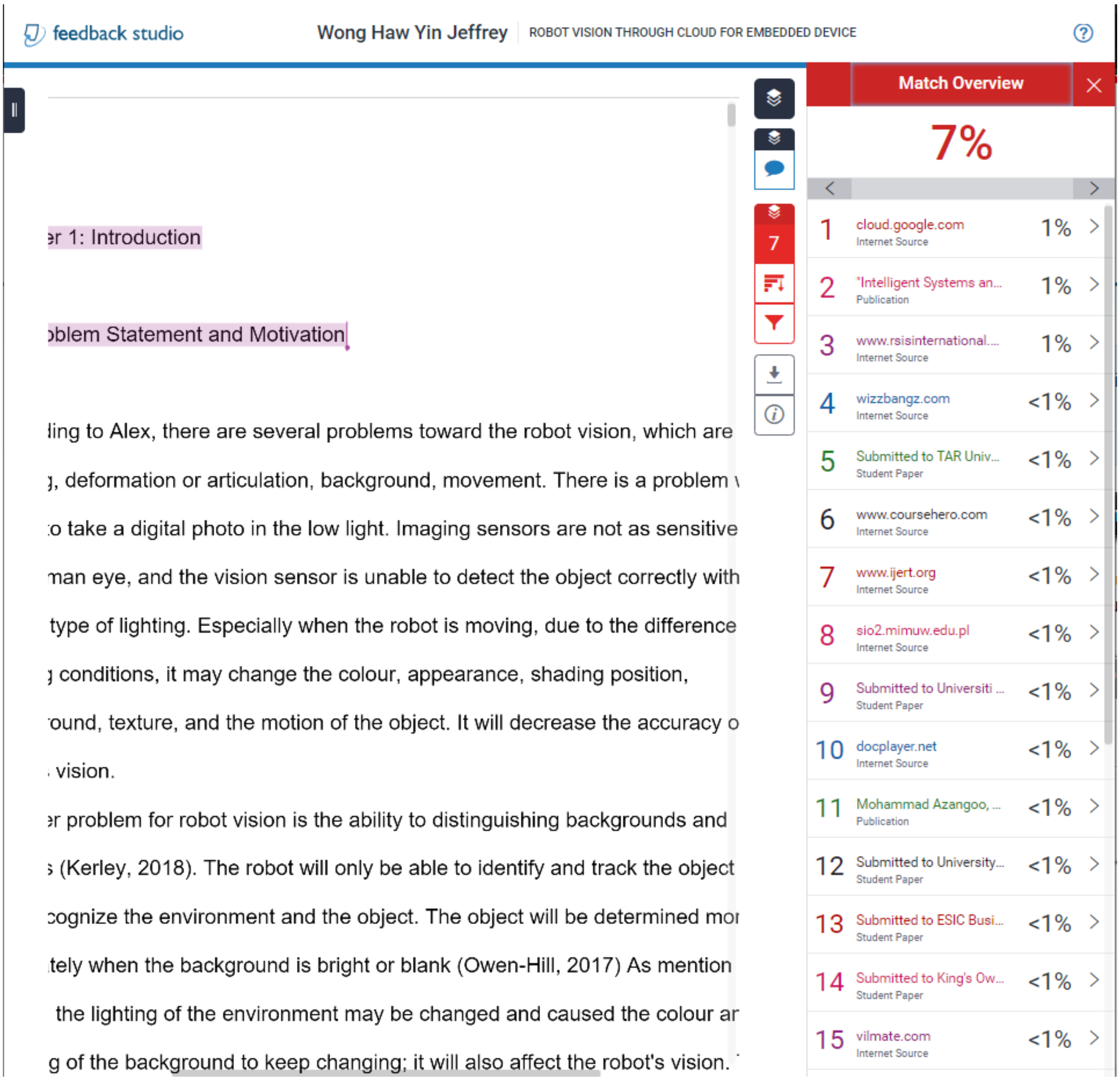
4. SELF EVALUATION OF THE PROGRESS

All the scheduled tasks are in good progress.

Supervisor's signature

Student's signature

PLAGIARISM CHECK RESULT



Wong Haw Yin Jeffrey | ROBOT VISION THROUGH CLOUD FOR EMBEDDED DEVICE

er 1: Introduction

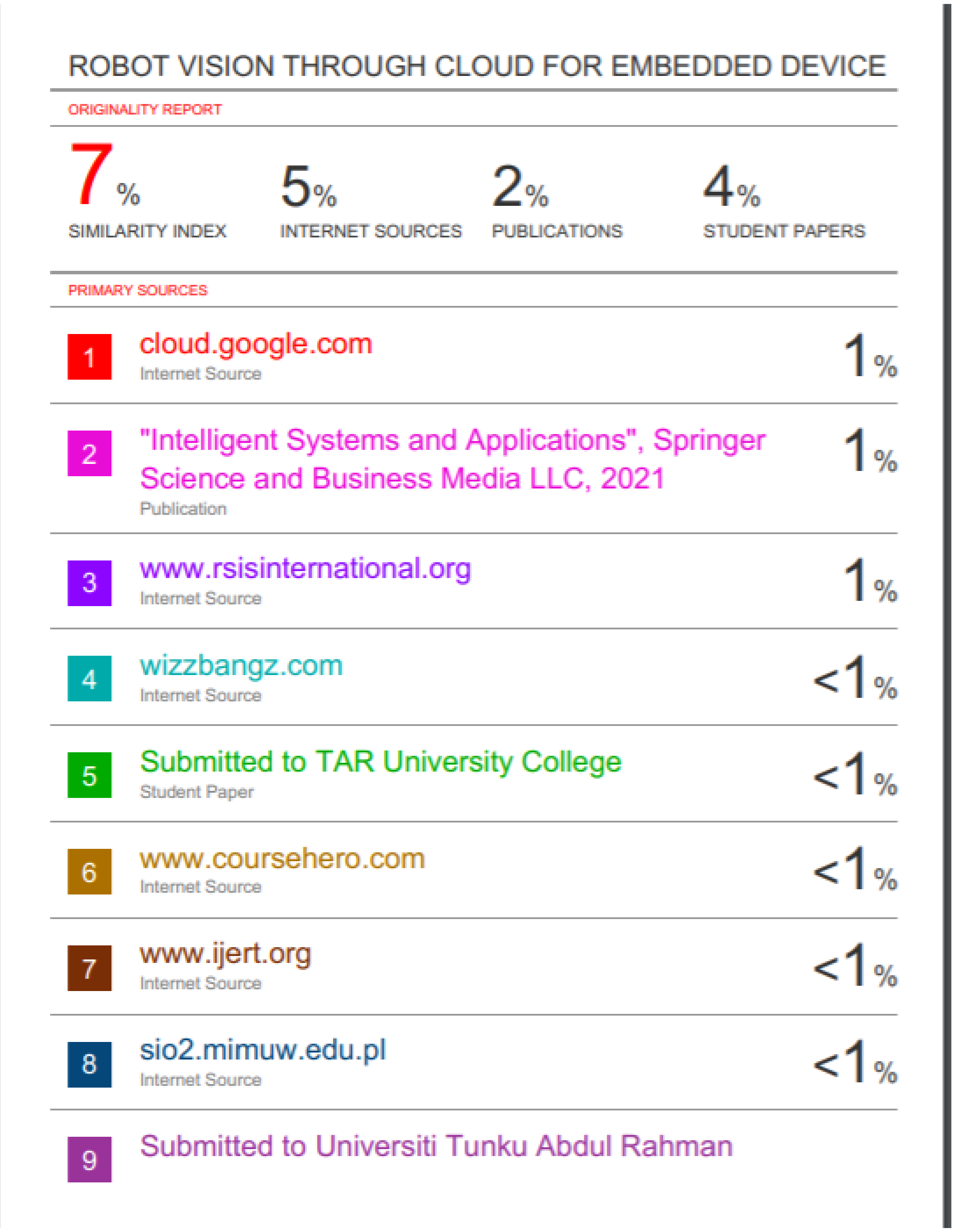
oblem Statement and Motivation

ling to Alex, there are several problems toward the robot vision, which are
g, deformation or articulation, background, movement. There is a problem
o take a digital photo in the low light. Imaging sensors are not as sensitive
man eye, and the vision sensor is unable to detect the object correctly with
type of lighting. Especially when the robot is moving, due to the difference
g conditions, it may change the colour, appearance, shading position,
ound, texture, and the motion of the object. It will decrease the accuracy o
vision.
er problem for robot vision is the ability to distinguishing backgrounds and
s (Kerley, 2018). The robot will only be able to identify and track the object
cognize the environment and the object. The object will be determined mor
tely when the background is bright or blank (Owen-Hill, 2017) As mention
the lighting of the environment may be changed and caused the colour ar
g of the background to keep changing; it will also affect the robot's vision.

Match Overview

7%

1	cloud.google.com Internet Source	1%
2	"Intelligent Systems an... Publication	1%
3	www.rsisinternational.... Internet Source	1%
4	wizzbangz.com Internet Source	<1%
5	Submitted to TAR Univ... Student Paper	<1%
6	www.coursehero.com Internet Source	<1%
7	www.ijert.org Internet Source	<1%
8	sio2.mimuw.edu.pl Internet Source	<1%
9	Submitted to Universiti ... Student Paper	<1%
10	docplayer.net Internet Source	<1%
11	Mohammad Azangoo, ... Publication	<1%
12	Submitted to University... Student Paper	<1%
13	Submitted to ESIC Busi... Student Paper	<1%
14	Submitted to King's Ow... Student Paper	<1%
15	vilmate.com Internet Source	<1%



ROBOT VISION THROUGH CLOUD FOR EMBEDDED DEVICE

ORIGINALITY REPORT

7%	5%	2%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	cloud.google.com Internet Source	1%
2	"Intelligent Systems and Applications", Springer Science and Business Media LLC, 2021 Publication	1%
3	www.rsisinternational.org Internet Source	1%
4	wizzbangz.com Internet Source	<1%
5	Submitted to TAR University College Student Paper	<1%
6	www.coursehero.com Internet Source	<1%
7	www.ijert.org Internet Source	<1%
8	sio2.mimuw.edu.pl Internet Source	<1%
9	Submitted to Universiti Tunku Abdul Rahman	

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	JEFFREY WONG HAW YIN
ID Number(s)	16ACB03498
Programme / Course	BIS (Hons) Communications And Networking
Title of Final Year Project	Robot Vision Through Cloud For Embedded Device

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 7 **%** **Similarity by source** Internet Sources: 5 % Publications: 2 % Student Papers: 4 %	
Number of individual sources listed of more than 3% similarity: 0	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

____
Signature of Supervisor

Name: Dr. Goh Hock Guan

Date: 10/9/2020

____
Signature of Co-Supervisor

Name: ______________________

Date: ______________________

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	16ACB03498
Student Name	Jeffrey Wong Haw Yin
Supervisor Name	Dr. Goh Hock Guan

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Front Cover
√	Signed Report Status Declaration Form
√	Title Page
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report. ________________________ (Signature of Student) Date: 9th September 2020	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction. ________________________ (Signature of Supervisor) Date: 10/9/2020