FAKULTI TEKNOLOGI MAKLUMAT DAN KOMUNIKASI

SEMESTER 1 2019/2020

WORKSHOP II (BITU 3923)
BITI

REPORT

PROJECT TITLE
PetCare.AI

GROUP NUMBER
9

PREPARED BY:

MOHAMMAD SYAZWAN BIN ABDUL HADI	B031710443
YEAP ZHI HERN	B031710168
SHAZANA BINTI MOHAMMED AFFANDI	B031710303
NUR AZWA BINTI ZUBIR	B031810102

PREPARED FOR:
DR. NOOR FAZILLA BINTI ABD YUSOF

wor

ACKNOWLEDGEMENTS

First at all, in performing our workshop 2, we had to take the help and guideline of some respected persons, who deserve our greatest gratitude. The completion of this workshop 2 gives us much pleasure. We would like to show our gratitude to Dr. Noor Fazilla binti Abd Yusof for giving us a good guideline for this workshop 2 throughout numerous consultations. Besides that, we also want to thanks to our classmates and team member itself, have made valuable comment suggestions on this proposal which gave us an inspiration to improve our workshop. We would also like to expand our deepest gratitude to all those who have directly and indirectly guided us in completing the workshop throughout this semester.

ABSTRACT

Pets are the most adorable being ever exist in this world. We’re bless to keep them as our companion to keep us company at home. The challenge is that how the owner should take a good care of thier own pet. Besides giving them home, toys or cuddle, the most important aspect would be giving them foods. As we’re all aware, there would be no problem to give essential foods when the owner at home. The problem lies when the owner especially to those who lives alone will go far away from home and leave their pet behind. Some problems occur when owner need to travel more than a day which provides not enough food and water. So, here we are, develop a solution for these problems. The project is call as PetCare.AI. The project only focuses on giving the pet food for now. It can be called as a food dispenser but with a little twist to it. One of the features would be object detection. The owner can use the object detection so that it can only dispense food for their own pet. The pets only need to stand in-front of the dispenser so the camera would recognize the face. It will proceed to dispense a fix amount of food to feed their pet. Furthermore, the owner can also check the dispenser if their pet already eats or not.

ABSTRAK

Penjagaan haiwan peliharaan adalah menjadi sebuah kebiasaan pada hari ini. Pemilik haiwan tersebut menjadikan haiwan perliharaan mereka sebagai teman di rumah. Cabarannya ialah bagaimana mereka harus menjaga haiwan peliharaan mereka sendiri. Selain menyediakan rumah dan permainan, aspek yang paling penting ialah memberi makanan. Seperti yang kita ketahui, tidak ada masalah untuk memberi makanan ketika berada di rumah. Masalahnya terletak apabila pemilik yang tinggal bersendirian terpaksa meninggalkan haiwan mereka di rumah. Sesetengah masalah berlaku apabila pemilik perlu melakukan perjalanan lebih dari sehari yang menyediakan makanan dan air yang tidak mencukupi. Jadi, di sini kami menyediakan penyelesaian untuk masalah ini iaitu PetCare.AI. Projek ini hanya memberi tumpuan kepada memberi makanan haiwan sahaja. Salah satu ciri dalam projek ini ialah pengesanan objek. Pemilik boleh menggunakan pengesanan objek supaya hanya dapat mengeluarkan makanan untuk haiwan kesayangan yang berdaftar. Haiwan peliharaan hanya perlu berdiri di hadapan dispenser supaya kamera mengenali muka. Ia akan terus mengeluarkan sejumlah makanan untuk memberi makan haiwan itu.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

1.1 Introduction

Since long time ago, humans had bond relationship with pets. Since that day, taking care of pets had become one of the important duty or responsibilities for humans. To take care of their beloved pets, humans must bath them, feed them, and accompany them. Thus, the bonding between owners and their pets can become stronger.

But as humans, they have a lot of things to do too. They have work to do and sometimes need to travel far. They may not spend all their time to be side by side with their pets. This is biggest problem faced by pet owners especially when they need to travel far when the first worry that came to their mind will be “How am I got to feed my pet?”.

Give them a lot of food? Not going to work, because the food will be finish within a day. So, our team decided to invent a mechanism to solve this problem. A robot that can feed pets on time. The robot comes with a detection that can detect pet and then give pet the food they needed while they were away.

1.2 Business Process

In the business process view, this robot is about to be implement and will bring some huge benefits to the pet’s owners. First, the robot could solve the main problem which is pet feeder. As we know, humans are tired and need rest. But by using PetCare.AI, owners will have more time to rest and robots will feed their cats.

1.3 Problem Statement

Pet owners face a lot of problem while taking care of their pets. Especially while away from home, it is hard for the pet owners to feed their pets. Some pet owners will put the pet food outside their houses. But little do they know, this action may lead to the food given by the owners sometimes not eaten by their pets but by other stray animals such as squirrel, dog, cat, rat and raccoon. Besides, quantity given by the pet owners is not fixed every time. So, this may cause their pets to eat either too much or too little. If the pet owner gave too many foods to their pets, this may lead to the happening of food wastage. Moreover, pet owners will sometime forget to feed their pet. This is a serious problem while taking care of our pet

1.4 Objectives

1. To propose and design automatic food dispenser that can feed pet with fixed amount of food.
2. To build a sustainable solution that can detect pet for dispensing food.
3. To develop an automatic object detection food dispenser for pet.

1.5 Project Scope

In this project, we are going to use an EV3 Lego set to build a robot. This project is divided the scope into some categories as below:

i. Module to be develop

- Pet's detection
- Food dispenser

ii. Target users

- Pet's owner

iii. Specification

- Lego set
- Sensors (Color sensor, Ultrasonic sensor)

1.6 Project Significance

Nowadays, time has become an essential aspect in the daily life's and people must put up a race against the time as they not have as much of time to take care everything. Hence, they hope for an invention or new technology that can assist on feeding their pets. Therefore, an invention which is an automatic pet feeder will become important because it will be able to give the benefits for both pets and owner. This is because, the pet can get feed on time and the owner also can save their time and energy.

1.7 Hardware and Software

i. Hardware

- Laptop
- EV3 Lego Mindstorms Set
- Sensors (Ultrasonic and Color Sensor)
- Webcam

ii. Software

- Python
- Tensorflow

1.8 Conclusion

As the conclusion, this chapter gives an overview to the project and explains the purpose of this project to be developed. It presents the problem statements, primary objectives and specifies the scope of this project. Besides that, by developing this robot the pet's owners get many benefits by using this PetCare.AI. This is one of the best solutions to feeding the cats.

CHAPTER II

METHODOLOGY

2.1 Robot Development Methodology

Rapid prototyping is a revolutionary and powerful technology with wide range of applications. The process of prototyping involves quick building up of a prototype or working model for the purpose of testing the various design features, ideas, concepts, functionality, output and performance. This cycle known as the spiral cycle or layered approach is iterative, meaning that products are continually improved as the cycle continues.

For our project, prototyping approach is used as a methodology. Our project uses image processing, that is object detection at which accuracy taken a very significant place. We stick to the requirement specification and design the architecture of the robot. We build a simple robot with motor. Then we build the training model of the object detection system and integrate with the robot built up earlier. Accuracy test was run to ensure everything is smooth and the accuracy is studied. The prototype is improved, and the model is improved and tested continuously.

Prototyping approach is choosing in our project for object detection with build and test the training model from time to time, continuously for obtaining the best result and accuracy.

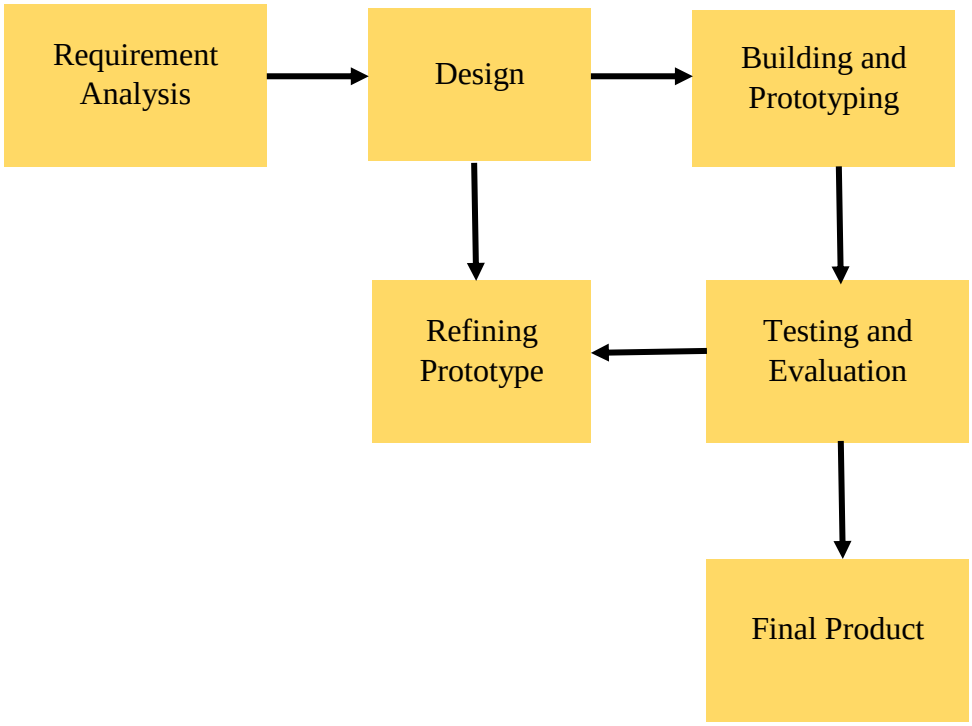


Figure 2.1.1 Prototype Model

In requirement analysis, our team discussed the problem statement and determined the requirement specification with our supervisor and between our group members. The functionality and modules are discussed.

The specification of the project requirement is then used to design the initial project architecture according to the specification which is a webcam used for object detection and motor to move the door, conveyer belt and dispenser spatula. The prototype is built according to the outcome of design analysis.

The movement of the motor is just for door, conveyer belt and dispenser spatula according to the results from object detection. The system is tested, and its performance is improved from time to time. The evaluation process is to ensure the system and the motor movement can function well. Finally, a final product which fulfil the project requirement is produced.

2.2 System Development Technique

In this project, there are few techniques are being implemented to be develop. Data from various sources have been collected and interpreted. The component and mechanism of the robot's functionality is also being analyzed and discussed. The required materials are being borrowed from the faculty according to the precise measurements of the robot body as calculated.

The prototype is built by using the Lego EV3 Mindstorm set. Other than that, tensorflow algorithm were used for the object detection. The operation of hardware and software is very important because it is the stepping stone to make the whole process of the robot to function properly. Robot requirements are dividing for two categories. A first category is software requirements. Software requirement can describe the software that used in develop the system and a second category is hardware requirement which is EV3 Lego set to build the robot body.

2.3 Summary

In conclusion, this project used the prototyping methodology which the system is done based on the requirement specification. The processes of set up the robot included requirement analysis followed by design, building and prototyping, testing and evaluation and finally a development of a final product. The built-up prototype model enables our group to be flexible while accomplishing this project based on the objectives discussed earlier in the report.

CHAPTER III

ANALYSIS

3.1 Analysis of Current Application

Figure 3.1.1 Lego Mindstorms Pet Feeder EV2

This is an existing pet feeder project made using Lego Mindstorms. It consists of using Lego Mindstorms EV3 Kit. The basic design includes the use of servo motor, EV3 intelligent brick, button(sensor), dispenser and food chamber. Compare to the predecessor, it is still having the same functionality but with a lot of improvement.

So how does it work? It is relatively simple; the button needs to be pushed for the dispenser to open for a few seconds which slide the food from the food chamber into the tray. For it to work, the pet needs to be trained to push the button on its own. This may take time as the pet learn how to do it itself. This project uses a dog to show the process.

From the above analysis of the project, there are some cons that can be considered.

i. The mechanism of the food dispenser

The mechanism it is using a 360°rotation from a motor to push the slider out. The movement seems to be sluggish which can possibly cause irregularity inside the mechanism. It moves up and down and cause vibration when it push the slider. The food also can get stuck between the slider if time incorrectly.

ii. Food Chamber

It can only hold small amount of food. If the pet uses it frequently, the food will be out in no time. Constant refilling it will waste time.

iii. Training Pet

This is one of the difficult parts since not every pet is the same. Even though dog is easy to be train, it is still time consuming and some dogs can't even do it. It is basically the same reason apply to another type of pet as well

3.2 Analysis of Proposed Robotic Application

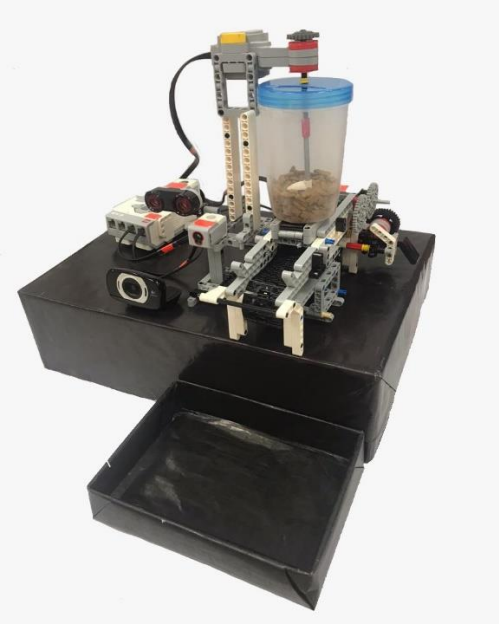


Figure 3.2.1 Lego Mindstorms PetsCare.AI

Figure 3.2.1 shows the prototype of the current PetsCare.AI project. The project will use object detection (pet) using webcam. The mechanism of the food dispenser will be based around the object detection for it to work.

i. Mechanism of the robot

Our robot is made from EV3 except for the food container and food bowl. There are three part for our robot which is top part, middle part and bottom part. The top part is food container. We made using external part which is a water bottle. We used EV3 large motor for pushing foods and medium motor for the door. Next, the middle part is where the foods will be drop. The bottom part will have food bowl to catch the food that drops from middle part and the pets will eat form the bowl. Lastly, we used webcam for object detection part.

ii. Movement of the robot

1. The pet owners will store the foods into food container.
2. When the pets come to the robot, the ultrasonic sensor will send signal to color sensor.
3. The color sensor will produce variety color.
4. The webcam will detect the pets.
5. EV3 medium motor will push the slider for 1 seconds and some foods will drop from food container onto the conveyer belt.
6. Conveyer belt will bring the food into foods bowl.

iii. Functionality of the robot

- Food container: store foods
- EV3 medium motor: control food cover
- EV3 large motor: control conveyer belt and dispenser spatula
- Food cover: cover and drop foods
- EV3 small elements: build the food cover and based of the robot
- Bowl: store the drop foods and place where pet eat
- Webcam: recognize the pet’s face

Compare from the previous analysis of Lego Mindstorms Pet Feeder V2, PetsCare.AI improves it in various sections such as:

- Does not need to train pet. It will automatically recognize the pet face and drop the food.
- The food holder can hold better amount of food.
- The movement of the food can easily flow to the food tray.
- The slider does not move sluggishly since it only moves in a one-way perpendicular direction. This can prevent food stuck and easy to manage the slider.

3.3 Structure Chart of Proposed Robotic Application

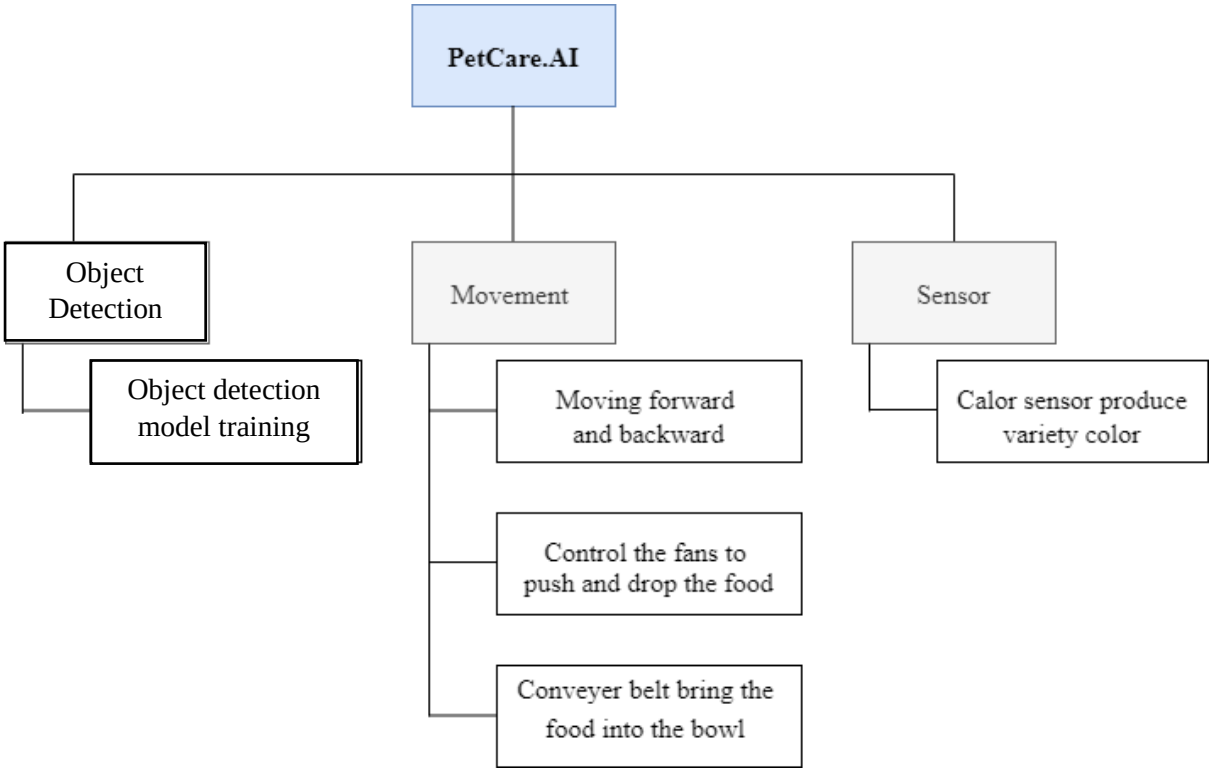


Figure 3.3.1 Structure Chart

3.4 Work Breakdown

i. Roles and Responsibility

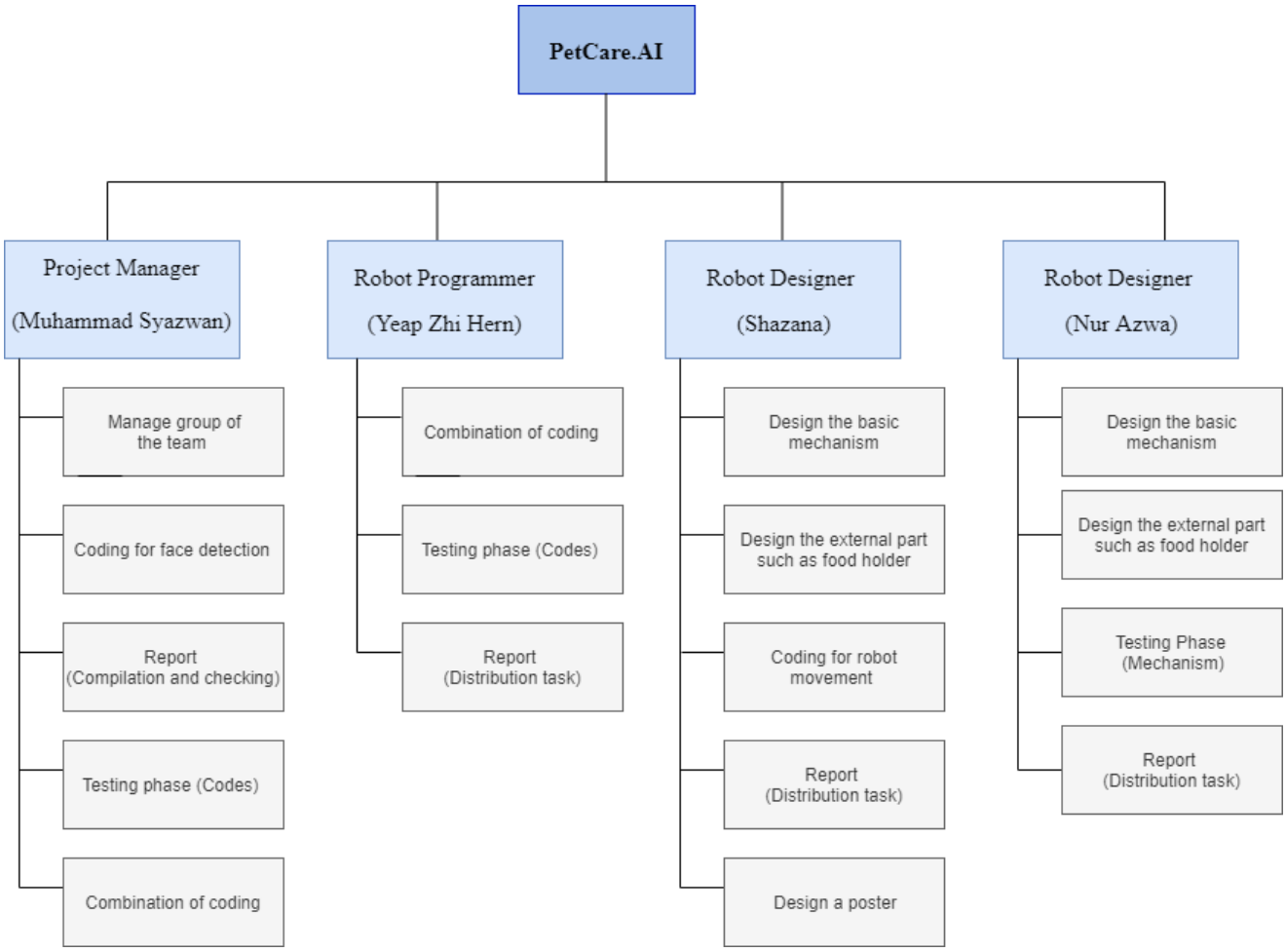


Figure 3.4.1 Roles and responsibilities

ii. Organization Structure

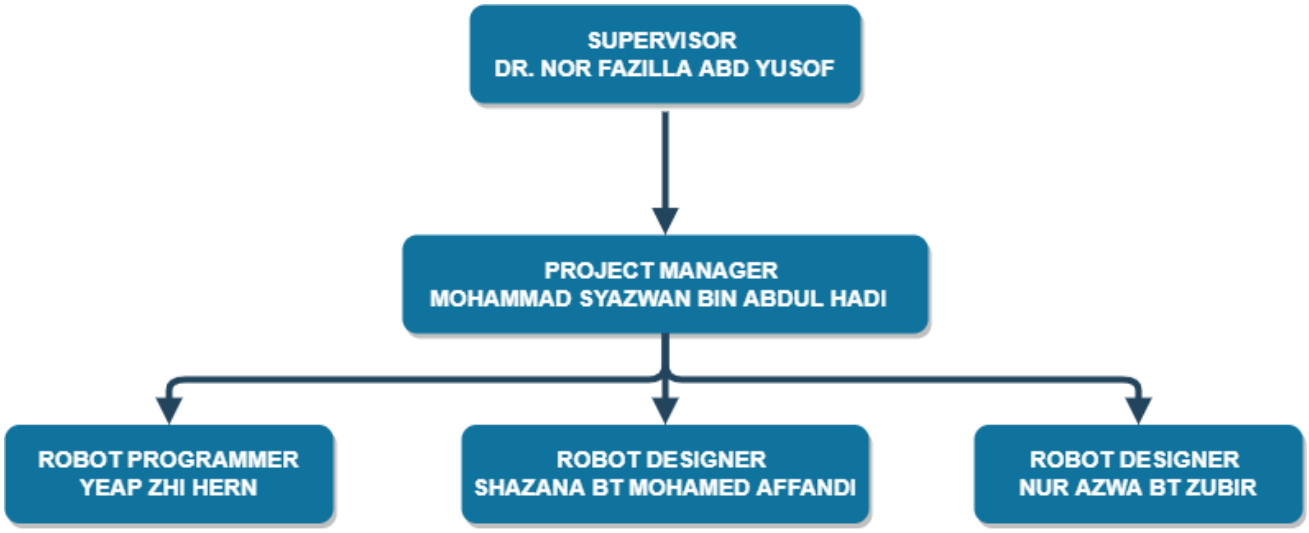


Figure 3.4.2 Organizational Chart

iii. Gant Chart

Table 3.4.3 Gant chart

ACTIVITY / WEEKS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Briefing of Workshop 1	■														
Assigning Students to Supervisors	■														
Discussion / Verification of title and sypnosis. Proposal preparation		■													
Sudent submits proposal to Supervisor & Committee		■	■												
Methodology and analysis phase				■											
Module Design					■	■									
Module Implementation							■	■							
Module Integration									■	■					
Testing Project											■				
System Demostration												■			
Poster and CD Submission													■	■	
Presentation Week															■

3.5 Summary

As a conclusion, the development of robot design and the implementation of code instruction of the PetCare.AI must function properly in order to develop the robot and complete the project successfully. This robot will be developed using Python programming language.

CHAPTER IV

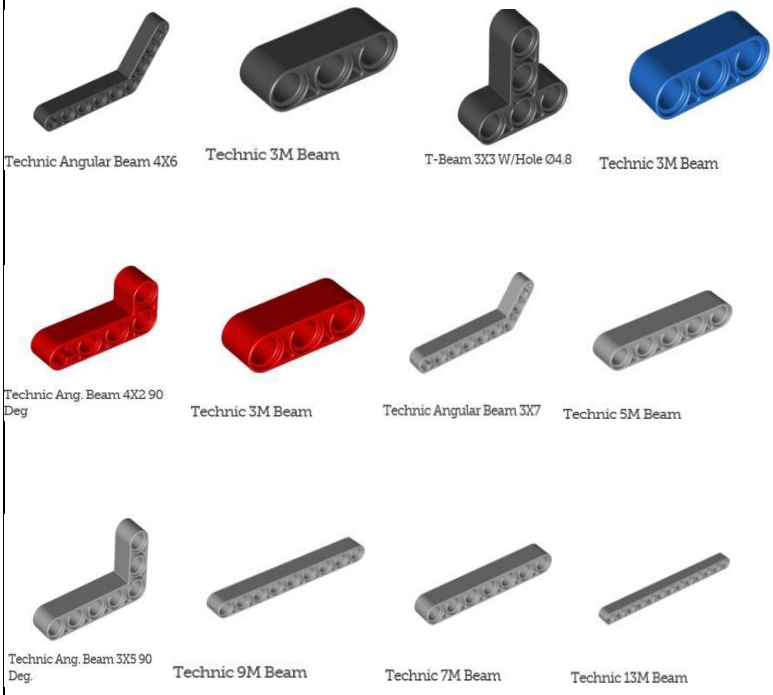
DESIGN

4.1 Introduction

PetCare.AI is using EV3 Lego Mindstorm components to complete the architecture and design. Other material such as boxes and container to build food holder and bowls.

i. Loose components of Lego Mindstorm

Table 4.1.1 Components used of EV3 Lego

Categories	Image
Beams. These parts are used for the main frame of the robot.	

	 Technic 11M Beam  Technic 9M Beam
Beams, Special. Act as a connecter between the frames. Some special parts can connect frames in different ways.	 Double Cross Block  Beam 1X2 W/Cross And Hole  Cross Block 90°  Beam 3 M. W/4 Snaps  Angular Beam 90Degr. W.4 Snaps  Cross Block 3X2
Cables. Used to connect the motor and sensor to the main EV3 bricks.	 Cable 250 Mm  Cable 350 Mm
Connectors Act as a connecter between the frames.	 Cross Axle 4M  Cross Axle 6M  Cross Axle 8M  Cross Axle 12M  Connector Peg W. Friction  Conn.Bush W.Fric./Crossale  Connector Peg W. Friction 3M  2M Fric. Snap W/Cross Hole

	 2M Cross Axle W. Groove  1/2 Bush  3M Connector Peg  Bush For Cross Axle  Cross Axle 5M  Cross Axle 7M  Connector Peg  Cross Axle 3M  Cross Axle 9M  3-Branch Cross Axle W/Cross H.
Wheels and Conveyer Belt. Moves the robots.	 Double Conical Wheel Z12 1M  Gear Wheel 40T 
Motors. Act as motor that control the robots.	 2x 
Brick Main controller of the whole EV3.	 1x

Sensor	

4.2 Robot Architecture

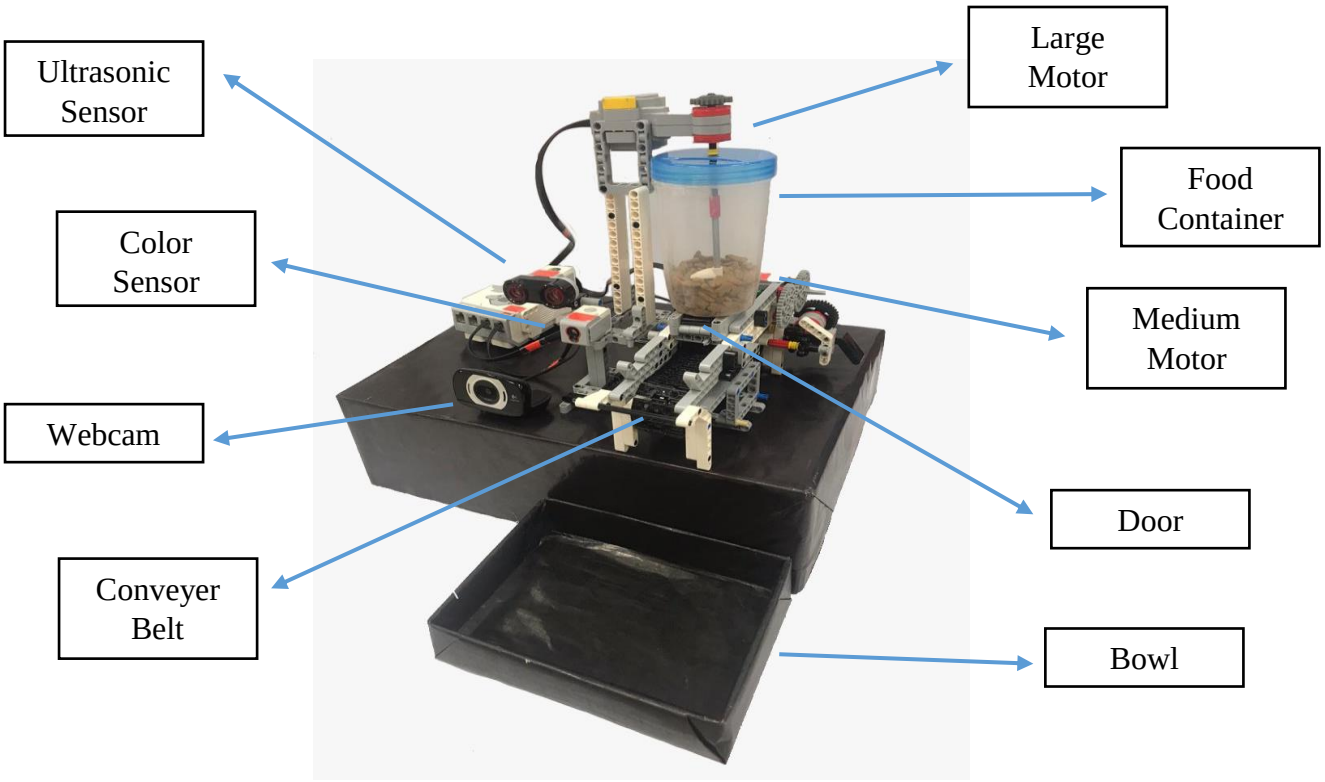


Figure 4.2.1 Final design of PetCare.AI

The core part of our robot is a webcam, the gate that allows food to flow out and the scanner locate at the top of the container. There is two sensor we used in our robot which is ultrasonic and color sensor. When cat come in front of ultrasonic sensor, it will send signal to color sensor to produce variety color. The camera will be set beside the robot. This is the place where our pet's face will be scanned. After the camera detects the correct face, it will inform the brick and the brick will send a signal to the door to open.

4.3 Movement Module

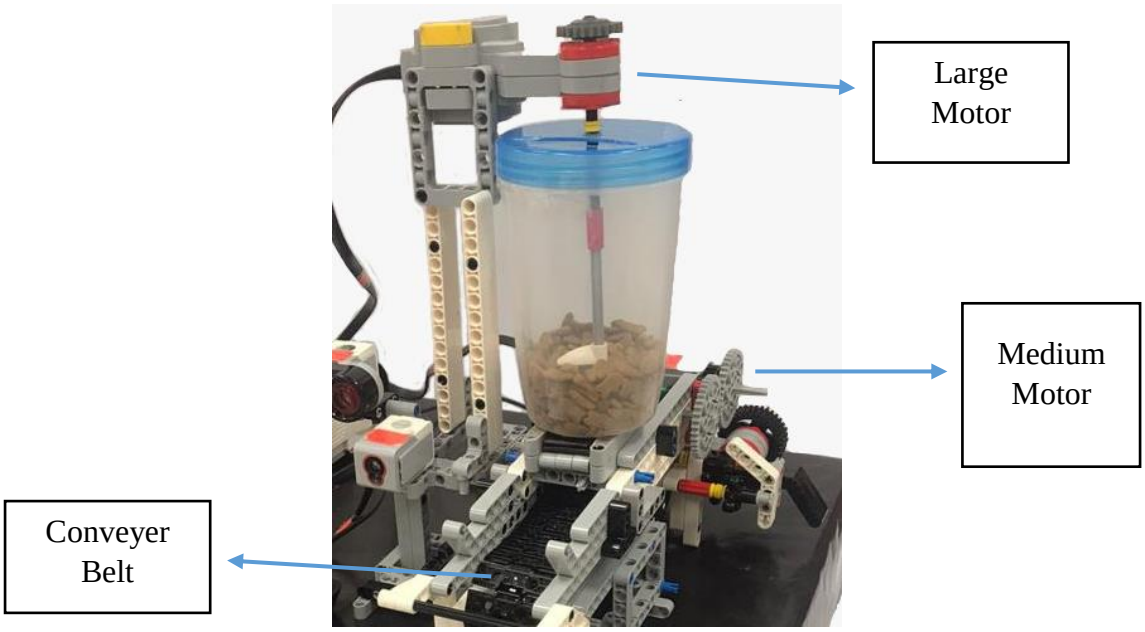


Figure 4.3.1 Movement Module Components

The figure above shows the PetCare.AI model that connected with large and medium motor. The motors are connected to EV3 Bricks. Medium motor control the door to open and close. Large motor control the dispenser spatula and conveyer belt. Dispenser spatula will push and drop the food onto the conveyer belt. Then, conveyer belt will bring the food into the bowl.

4.4 Sensors (Detection and Response) Module

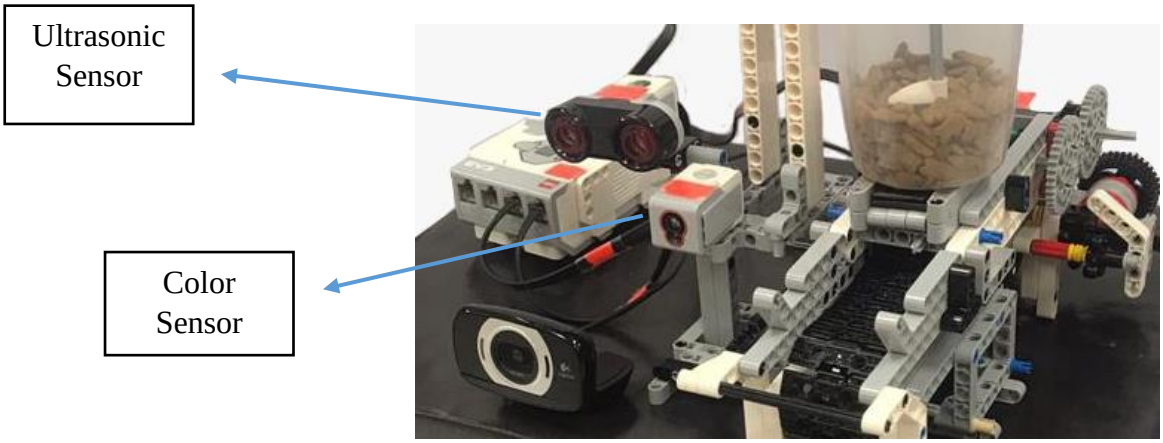


Figure 4.4.1 Sensors Components

As shown on the diagram above, we used ultrasonic sensor in our project for detect object in front of robot and send signal to color sensors. Then, color sensor will produce variety color to attract cat to go to the robot.

4.5 Intelligent Module

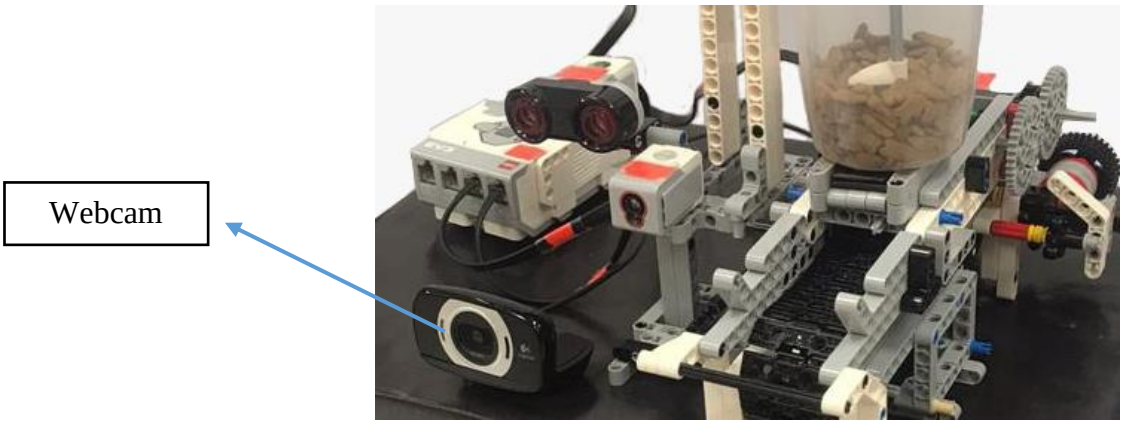


Figure 4.5.1 Intelligent Module Components

We have implemented image processing using the webcam which webcam will detect cat and send signal to the robot. As for the object detection, TensorFlow –GPU algorithm is used in training the model. Python programming language is used when implementing the object detection model.

4.6 Summary

As conclusion, the PetCare.AI is designed by using the components of EV3 Lego Mindstorm set which consist of bricks, motors and sensors.

CHAPTER V

IMPLEMENTATION

5.1 Introduction

EV3 PetCare.AI is a robot which is programmed to feed cats after the camera detect there are cats around. A model is trained and embed into the EV3 program. The model will help the robot to detect whether the animal that approach the robot. Once the camera captures a live time picture of a creature, the picture will then send to the robot to identify the creature. If the robot detects the creature as a Cat, then the robot will give out cat food. The architecture of the robot is developed and changed throughout the project to fulfil the requirements.

5.2 Movement

i. Medium motor (door)

The motor will move to open the door according to the results of image detection. A medium motor and another component of EV3 LEGO set are used to build the door with moveable motor. The movement of the motor is determined by the result of the cat detection model. The movement of the motor is just to control the gate by moving it forward (to close the door) and backward (to open the door).

ii. Large motor (blender)

This motor is later added into the robot after we found that there is a problem with the flowing of food from the container. Due to the existence of pressure in the food container, the food will have stuck at the opening of the gate if the food reaches a certain amount. So, we decided to add in a blender to average out the pressure so that the food can come out from the opening and drop on the conveyer belt successfully. So, the main function of the motor is just to decrease the pressure and allow the food to flow out from the container without blocking the opening.

iii. Large Motor (conveyer belt)

This motor is added to the robot to help the movement of the conveyer belt. Once the medium motor is used and the gate is open, the food will fall down and landed on the conveyer belt. We decided to use a larger motor is because we need a motor with a greater power to ensure the conveyer belt function well. EV3 large motor has enough power to move heavy object compare to medium motor but the only draw-back of using large motor is it consumes more energy from the EV3 brick.

5.3 Convolutional Neural Network Object Detection

i. Object Detection API

A webcam is used to detect and capture cat's face that passed by the webcam. The system will detect the face and classify the cat. The object detection is implemented using TensorFlow –GPU to identify and draw boxes around specific objects in pictures or in a webcam feed.

ii. Medium Motor

The medium motor will response to the results obtained from the system embedded with the face detection. When a cat is detected, the motor will move to open the door. The door will open only for a while and then will close back up to avoid over flow of cat food.

iii. Large Motor 1 and 2

The first main large motor (motor 1) move to rotate the gear to allow the movement of the conveyer belt. The second motor (motor 2) has a blender-like function. It is set with the container to prevent the food from not flowing out from the container. Both large motors will move at the same time when the medium motor move.

5.4 Completing Task via Intelligence

i. Object Detection

To make the object detection technique works, we provide the computer a few datasets to learn and then come out with a classifier. The perfect outcome is the system can differentiate the object is a cat or not. There are 435 training model and 50 testing model collected. The training and testing objects are labelled using tools called LabelImg and save as .xml file containing the label data for each image. These .xml files will be used to generate TFRecords, which are one of the inputs to the TensorFlow trainer. With the labelled images, TFRecords is generated to serve as input data to the TensorFlow training model. It used xml_to_csv.py and generate_tfrecord.py scripts. Before configuring the training, label map is created, and the training configuration file is edited. The webcam will detect the object, analyses it is using the trained model. The webcam will send the video captured to the system, the system will split it into frame and then start the detection process. In return, the system will come out with what the webcam captured but with square around the face detected.

5.5 Summary

As conclusion, the intelligent application was successfully implemented and developed. The modules of functions are triggered in response with the environment at which the webcam perceive the external environment for image detecting and capturing. The system will run when image was detected.

CHAPTER VI

TESTING

6.1 Testing Method

For this chapter, we are going to test the performance of our PetCare.AI whether it satisfies the specified requirements. The testing part will cover the testing for face detection and motor movement. We try to identify any gaps, error or missing requirement in contrary to the actual requirements.

There is two type of testing which is Unit Testing and Integration Testing. The testing helps in classify the bug. Other than that, it helps in stipulates information for improvement and better performance of the system.

i. Unit Testing

Each part of the program is tested to ensure every part are functioning well and satisfies the specifies requirements. In our project, the testing is run on object detection and the motor movement. The object detection is tested in the program to make sure the image specification is correct. The motor is tested to ensure it can open the door, move the conveyer belt and dispenser spatula.

ii. Integration Testing

For integration testing, the various part is to combine and run as an independent project. It tested to ensure the whole project can run smoothly and functioning well. The system containing face detection program and motor movement. It is connected to EV3 brick via bluetooth.

6.2 Test Result Analysis

i. Object Detection

Table 6.2.1 Testing for Object Detection

Subject	Actual	First Test	Second Test	Third Test
Cat present	The door will open Foods will move out to the bowl.	False	False	True
Cat does not present	The door will not open. Foods will not move out	False	False	True

ii. Medium Motor Movement

Table 6.2.2 Testing for Medium Motor Movement

Subject	Actual	First Test	Second Test	Third Test
Cat present	The door will open	False	True	True
Cat does not present	The door will not open	False	True	True

iii. Large Motor Movement

Table 6.2.3 Testing for Large Motor Movement (Dispenser Spatula)

Subject	Actual	First Test	Second Test	Third Test
Cat present	The dispenser spatula will push and drop the foods onto the conveyer belt	False	True	True
Cat does not present	The dispenser spatula not move	True	False	True

Table 6.2.4 Testing for Large Motor Movement (Conveyer Belt)

Subject	Actual	First Test	Second Test	Third Test
Cat present	The conveyer belt will move the foods into the bowl	False	True	True
Cat does not present	The conveyer belt not move	False	False	True

6.3 Robot Constraints

i. Object Detection

The object detection is done by training the model and use the model to detect cat. Since we only trained cat, the detection is not clearly accurate as it accidently detects another object as a cat. Furthermore, the webcam cannot detect object if it not places well in specific part. Other than that, the object detection is depends on lighting of some place. Dark place can prevent the webcam to do object detection.

6.4 Summary

In conclusion, we were successfully tested the coding, webcam and motors. However, there are some limitations in robot functions according to the system constraints. The limitations absolutely affect the performance in completing task via Intelligence.

CHAPTER VII

CONCLUSION

7.1 Robot Advantages and Commercial Values

Today society, Petcare.AI would be a very helpful invention to solve many main problems. This robot can help pet's owner to give the food to their pet. There are so many benefits of using this robot such as pet's owner no need to ask friends or family for favors, to feed their pet if they're out for long hours a day. Other than that, it also can save time. If pet's owner has a busy schedule or are engaged in demanding undertakings, they should really consider having such a device for their loving pet companion. The reason behind this suggestion is the time they get to save with these feeders. By having these advantages, it could be easy weight management.

7.2 Robot Weaknesses

The developed Petcare.AI has a few slight weaknesses which can be easily fixed in order to prepare the robot for deployment into the real world. The constraint of hardware and software is the main weakness. Because the EV3 Product Sorting Machine is a prototype, it was built from EV3 Lego Set which would be unlikely used in the real world. Besides that, the movement of the food to drop down is disturbed sometimes due to the hardware imperfection. Therefore, changing it into a stronger and better-quality material like metal would solve this problem. The usage of intelligence in PetCare.AI also part of weakness which it detects another object as a cat.

7.3 Suggestions for Robot Improvement

Regarding PetCare.AI, the physical design can be improved since we are restricted amount of EV3 parts. We had to build it with parts lent by FTMK lab and we also rent from a store to complete this robot. Besides, coding for the object detection also be improved to be better. At the initial process of our project, we did many adjustments to make sure that robot can be humanoid figure. Afterwards, we adjust the robot that it can adapt with the coding. Furthermore, if there are more EV3 Lego part, the design can be better, and the robot function will be increase.

7.4 Summary

The system that has been developed will provides benefits to pet's owner besides decrease the happening of food wastage. The system still needs to be improved with better software and hardware for higher effectiveness and performance of the system.

Besides, the development of this whole project also able to increase our skills in performing brainstorming, leadership and task arrangement with team member contributions. The knowledge gains from the lectures and theory can be practically uses for this project and sharpen our skills and expertise.

REFERENCES

EdjeElectronics (2019, June 22). “How to train a TensorFlow Object Detection Classifier for multiple object detection on Windows” Reference from https://github.com/EdjeElectronics/TensorFlow-Object-Detection-API-Tutorial-Train-Multiple-Objects-Windows-10

LEGO Group (2013). “Element List” Reference from http://www.depts.ttu.edu/coe/stem/gear/ev3/documents/EV3-Parts-List.pdf

TensorFlow (2019, September 26) “tf.image.extract_patches” Reference from https://www.tensorflow.org/api_docs/python/tf/image/extract_patches

Bennett, Coleman & Co. Ltd (2019) “Bennett, Coleman & Co. Ltd” Reference from https://economictimes.indiatimes.com/definition/prototype-model

Gordon McMillan (2019, December 16) “Socket Programming HOWTO” Reference from https://docs.python.org/3/howto/sockets.html

Visual Paradigm (2019) “What is Work Breakdown Structure?” Reference from https://www.visual-paradigm.com/guide/project-management/what-is-work-breakdown-structure/

Juan Sebastian, Jovani Alberto & Juan Fernando (2011, March 31) “DESIGN CYCLE OF A ROBOT FOR LEARNING AND THE DEVELOPMENT OF CREATIVITY IN ENGENEERING” Reference from http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0012-73532011000600006

APPENDICES

Object Detection Coding

```
# Import packages
import os
import cv2
import numpy as np
import tensorflow as tf
import sys
from time import sleep

#import socket to pass value from tensorflow to ev3
import socket
import pickle
import threading

# This is needed since the notebook is stored in the object_detection folder.
sys.path.append("..")

# Import utilites
from utils import label_map_util
from utils import visualization_utils as vis_util

# Name of the directory containing the object detection module we're using
MODEL_NAME = 'inference_graph'

# 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,'training','labelmap.pbtxt')


# Number of classes the object detector can identify
NUM_CLASSES = 1


## Load the label map.
# Label maps map indices to category names, so that when our convolution
# network predicts `1`, we know that this corresponds to `cat`.
# Here we use internal utility functions, but anything that returns a
# dictionary mapping integers to appropriate string labels would be fine
label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
categories = label_map_util.convert_label_map_to_categories(label_map,
max_num_classes = NUM_CLASSES, use_display_name=True)
category_index = label_map_util.create_category_index(categories)


# Load the Tensorflow model into memory.
detection_graph = tf.Graph()
with detection_graph.as_default():
   od_graph_def = tf.GraphDef()
   with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
      serialized_graph = fid.read()
      od_graph_def.ParseFromString(serialized_graph)
      tf.import_graph_def(od_graph_def, name='')


   sess = tf.Session(graph=detection_graph)


# Define input and output tensors (i.e. data) for the object detection classifier
# Input tensor is the image
image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')
```

```
# Output tensors are the detection boxes, scores, and classes
# Each box represents a part of the image where a particular object was detected
detection_boxes = detection_graph.get_tensor_by_name('detection_boxes:0')

# Each score represents level of confidence for each of the objects.
# The score is shown on the result image, together with the class label.
detection_scores = detection_graph.get_tensor_by_name('detection_scores:0')
detection_classes = detection_graph.get_tensor_by_name('detection_classes:0')

# Number of objects detected
num_detections = detection_graph.get_tensor_by_name('num_detections:0')

# Initialize webcam feed
print("[INFO] starting video stream...")
video = cv2.VideoCapture(0)
ret = video.set(3,1280)
ret = video.set(4,720)

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("169.254.70.137", 3000))

while(True):
   ret, frame = video.read()
   frame_expanded = np.expand_dims(frame, axis=0)

   # Perform the actual detection by running the model with the image as input
   (boxes, scores, classes, num) = sess.run (
      [detection_boxes, detection_scores, detection_classes, num_detections],
      feed_dict={image_tensor: frame_expanded})
```

```
    # Draw the results of the detection (aka 'visulaize the results')
    vis_util.visualize_boxes_and_labels_on_image_array(
        frame,
        np.squeeze(boxes),
        np.squeeze(classes).astype(np.int32),
        np.squeeze(scores),
        category_index,
        use_normalized_coordinates=True,
        line_thickness=7,
        min_score_thresh=0.60)

threshold = 0.5

    objects = []
    for index, value in enumerate(classes[0]):
        object_dict = {}
        if scores[0, index] > threshold:
            object_dict[(category_index.get(value)).get('name').encode('utf8')] = scores[0,index]
            objects.append(object_dict)
    print (objects)

if objects:
        exist = 1
        if exist != '0':
            s.send(pickle.dumps(exist))
            print(exist)
if cv2.waitKey(1) == ord('q'):
        break
print("[INFO] cleaning up...")
video.release()
cv2.destroyAllWindows()
```

Movement of Robot Coding

```
#!/usr/bin/env python3
import socket
import pickle

from ev3dev.ev3 import *
from ev3dev2.motor import SpeedPercent, MoveTank
from time import sleep

#socket created
print("Waiting to be connected.....")
s = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.bind((socket.gethostname(),3000))
s.listen(1)
conn,addr = s.accept()
true=True
addr = str(addr)
print('Connecting by : %s ' %addr )

cl = ColorSensor('in3')
us = UltrasonicSensor('in4')

us.mode='US-DIST-CM'

blender = LargeMotor('outA')
belt = LargeMotor('outC')
door = MediumMotor('outD')

while True:
    data = pickle.loads(conn.recv(1024))
```

```
if data == 1:
    distance = us.value()/10  # convert mm to cm
    if distance < 100:
        Sound.beep()
        cl.mode='COL-COLOR'
        Sound.beep()
        door.run_timed(time_sp=3000, speed_sp=-300)
        sleep(1)
        blender.run_timed(time_sp=9000, speed_sp=180)
        belt.run_timed(time_sp=12000, speed_sp=360)
        sleep(10)
        door.run_timed(time_sp=3000, speed_sp=300)
        cl.mode='COL-REFLECT'
        break
    else:
        cl.mode='COL-REFLECT'
```