DESIGN OF PAPER PEN VENDING MACHINE

DESIGN PROJECT REPORT

Submitted by:

AKHIL M: **TRV17ME012**
ASVIN KRISHNAN: **TRV17ME021**
ASWIN S: **TRV17ME023**
VISHNU S NAIR: **TRV17ME057**

to

the APJ Abdul Kalam Technological University

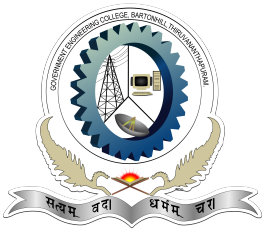
in partial fulfilment of the requirements for the award of the Degree

of

Bachelor of Technology

in

Mechanical Engineering



DEPARTMENT OF MECHANICAL ENGINEERING

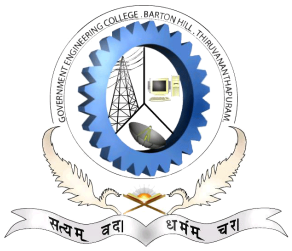
GOVT. ENGINEERING COLLEGE, BARTON HILL,

THIRUVANANTHAPURAM

NOVEMBER 2019

DEPARTMENT OF MECHANICAL ENGINEERING

GOVT. ENGINEERING COLLEGE, BARTON HILL, THIRUVANANTHAPURAM



CERTIFICATE

This is to certify that the Project report titled **Design of Paper Pen Vending Machine** submitted by **Akhil M**, Reg. No. TRV17ME012, **Asvin Krishnan**, Reg. No. TRV17ME021, **Aswin S**, Reg. No. TRV17ME023, and **Vishnu S Nair**, Reg. No. TRV17ME057 to the APJ Abdul Kalam technological University in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Mechanical Engineering, is a bonafide record of the Project work carried out by them under my guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Dr. Bijulal D
Project Supervisor

Dr. K. Jayaraj
Head of the Department

ACKNOWLEDGEMENT

We are fortunate to have, Dr. Bijulal D, as our project supervisor, who has helped us push our limits and guided us whenever we were stuck. We are highly indebted to him for providing us with this valuable opportunity to learn and present new and innovative ideas in the form of this project. Therefore, our first and foremost gratitude go to him. Next, our sincere gratitude and appreciation goes to the Principal-in-charge, Dr. Suresh K, for his wholehearted support during this project. We would also like to thank Dr. K Jayaraj, Head of the Mechanical Engineering Department for his encouragement. The constructive criticisms offered by numerous other teachers have helped in the development of this project and we are very grateful to the faculty of the Department of Mechanical Engineering.

We would also like to take this opportunity to thank our friends and all those who supported us while doing this project. Finally, we thank our parents for their moral support and encouragement.

Akhil M
Asvin Krishnan
Aswin S
Vishnu S Nair

ABSTRACT

This project addresses the issue of plastic pens as a waste in the environment. A data analysis conducted within GEC, Barton Hill, TVM, revealed that most students do not refill their pen and the effective disposal of plastic pens is not carried out. All the used pens are disposed of along with other plastic waste. Most of them are burned in the open air, which emits a large amount of CO_2. Thus, it adds up to environmental pollution. This project is an earnest effort to mitigate up to some extent, the emission of CO_2 into the atmosphere, through the implementation of Paper Pen Vending Machine (PPVM). This machine accepts three used plastic pens as a means to collect used pens and returns one seed-implanted paper pen to the user to spread environmental consciousness. Through the implementation of the PPVM within the campus, it is estimated that the emission of one tonne of CO_2, into the atmosphere, can be avoided every year.

Keywords: *Plastic Pen, Paper Pen Vending Machine*

Contents

Contents Page

ACKNOWLEDGEMENT **iii**

ABSTRACT **iv**

LIST OF TABLES **vii**

LIST OF FIGURES **viii**

1 INTRODUCTION **1**
- 1.1 Problem Definition . . . 1
- 1.2 Objectives of the Project Work . . . 2
- 1.3 Scope of the Project Work . . . 2
- 1.4 Research Methodology . . . 2
- 1.5 Limitations of the Project Work . . . 3

2 EXISTING DESIGNS AND PRODUCTS **4**
- 2.1 Conventional Vending Machine . . . 4
- 2.2 Tensorflow and Image Classification . . . 4
- 2.3 Terracycle Zero Waste Box™ . . . 5
  - 2.3.1 TerraCycle . . . 5
  - 2.3.2 The program and collected wastes . . . 5
  - 2.3.3 Processing of the waste . . . 6
- 2.4 Summary . . . 6

3 DESIGN/DATA COLLECTION **7**
- 3.1 ODK: A brief Introduction . . . 7
- 3.2 Components of ODK . . . 7
- 3.3 Workflow for data collection . . . 8
  - 3.3.1 Workflow for data collection of PPVM . . . 8
  - 3.3.2 Survey . . . 9
- 3.4 Data Analysis . . . 9
- 3.5 PPVM Design . . . 10
- 3.6 Estimation . . . 11
- 3.7 Summary . . . 11

4 DESIGN ANALYSIS **12**
4.1 Image capture 12
4.2 Image recognition and classification 12
4.3 Model Design 13
4.4 Pen Accepting and Rejecting 17
4.5 Rollout Mechanism 17
4.6 Summary 18

5 DISCUSSIONS **19**

6 CONCLUSIONS **20**

REFERENCES **21**

Appendix A **22**
A.1 Code 22
A.1.1 For creating model 22
A.1.2 For processing in PPVM 24

Appendix B **29**
B.1 Data collected 29

List of Tables

Title Page No.

3.1 Table showing the cost estimation 11

List of Figures

	Title	Page No.
2.1	A vending machine	5
3.1	ODK-Collect	8
3.2	PPVM body	10
4.1	Image recognition and classification algorithm	13
4.2	Flowchart showing the algorithm of the PPVM	14
4.3	A drawing of the proposed design (all dimensions are in inches)	15
4.4	PPVM prototype	16
4.5	The rollout mechanism	18

Chapter 1

INTRODUCTION

Plastic waste is presently one of the most important issues that mankind is facing. Currently, the most widely used method for the segregation of waste is via separate bins for different wastes. Though there are several ways to sort plastic wastes such as manual sorting, post grinding waste sorting, optical waste sorting, floatation waste sorting and many more, not all of them are effectively utilized due to lack of availability and sometimes feasibility of the process. Any new creative methods to segregate waste would help us step forward to a cleaner world.

This project will focus on the contribution of plastic pens to wastes in the environment and suggest a possible solution to curb this problem.

1.1 Problem Definition

Improper disposal of waste plastic pens causes pollution to the environment. The data analysis conducted within the Government Engineering College, Barton Hill (GECBH), revealed that all the used pens are disposed of along with other plastic waste. Most of them are burned in the open air, which emits a large amount of CO_2. Some of them are dumped in landfills and some others in the ocean. It is estimated that the conventional disposal of plastic pens will emit around 1.2 tonnes of CO_2 every year, within the GECBH campus. Even though the magnitude of this problem appears to be small, its consequences are colossal. For effectuating a greener future, it is important to address this issue.

1.2 Objectives of the Project Work

The main aim of this project is to address the issue of plastic pens as a waste in the environment. The data analysis conducted within GECBH, proved the presence and intensity of this problem. To address this issue, a Paper Pen Vending Machine will be designed and demonstrated within the campus. A conventional vending machine would provide the user with goods in exchange for a credit system like money. This machine will accept used plastic pens from the user and return a seed implanted paper pen. One paper pen would be provided in exchange for three used plastic pens to maintain the feasibility. The collected plastic pens would then be sent to a recycling facility.

The objectives of this project work are:

1. Address the issue of plastic pens as a waste in the environment.
2. Design and demonstrate a Paper Pen Vending Machine which provides seed implanted paper pens to the user in exchange for three plastic pens.

1.3 Scope of the Project Work

This project focuses on used plastic pens only. The representative location for this project is Government Engineering College, Barton Hill.

1.4 Research Methodology

As part of the data analysis, a small scale survey was conducted in GECBH. The data collected mainly included the type of pen, the price of the pen and the frequency at which the pens are replaced. From these data, the total weight of the plastic pens disposed per year within the campus was estimated.

On further analysis, it was estimated that the conventional disposal of plastic pens will emit around 1.2 tonnes of CO_2 every year, within the college campus. Through the implementation of PPVM, an estimated amount of one tonne of CO_2 will be saved just within the campus. Furthermore, a feasibility study was conducted to find out how many plastic pens need to be accepted for one paper pen; the ratio was found to be 3:1.

The design aspect of this vending machine is quite simple. It accepts a used plastic pen, and check whether it is a pen or not. After recognition, it is redirected into a bin provided inside the machine. If it is not a pen, it is rejected out of the machine. For the recognition process, **"Tensor Flow"**, a library in Python is used. Using a data set consisting of images of pens, a model pattern of a pen was created. A camera takes

a picture of the waste pen and compares it with the model. After recognition, a signal is then sent to an actuator which redirects the pen. The paper pens are stacked on a ramp behind a rolling mechanism. After depositing three plastic pens, a signal is sent to the roller and a paper is provided to the user.

1.5 Limitations of the Project Work

This project focuses only on the collection of used plastic pens. Hence, other types of plastic wastes cannot be considered. The computer algorithm only checks the shape of the pen and not it's material. So it is not guaranteed that the pen deposited is indeed plastic. Another issue is that the number of paper pens that can be stored within the machine for exchanging with plastic pens is limited. The removal of collected plastic pens for shredding and the refilling of paper pens requires human effort. Also, a person in charge should be assigned for the regular maintenance of the machine.

Chapter 2

EXISTING DESIGNS AND PRODUCTS

2.1 Conventional Vending Machine

A vending machine is an automated machine that provides items such as snacks, beverages, and other items to consumers after inserting money, a credit card or a specially designed card into the machine. They are a huge convenience for the users as they are easy to operate and the products are received quickly.

But the proposed Paper Pen Vending Machine does not accept any coin or credit card payments like conventional vending machines. Instead, it accepts three waste plastic pens and provides a paper pen in return. Accepting waste plastic pens promotes the necessity for proper disposal of them and also encourages the user to use the machine as it does require any monetary element and rewards the user with, practically, a free paper pen.

2.2 Tensorflow and Image Classification

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.Google Brain Team (2017)

An article by IBM developers “Image recognition with TensorFlow and Keras” Sharma (2019), demonstrates the basic image classification algorithm. The code teaches the computer to distinguish between chihuahua and muffin which look similar. This code was utilized to do the same function with pens.

Figure 2.1: A vending machine

2.3 Terracycle Zero Waste Box™

2.3.1 TerraCycle

TerraCycle is a private U.S. recycling business headquartered in Trenton, New Jersey. It primarily runs a volunteer-based curbside collection program to collect non-recyclable pre-consumer and post-consumer waste, and then partners with corporate donors or municipalities to turn it into raw material to be used in new products. The company licenses its name to manufacturers of roughly 200 products made using its raw material. TerraCycle also manages Loop, a consumer products shopping service with reusable packaging.Wikipedia contributors (2019)

2.3.2 The program and collected wastes

TerraCycle has created the program named Zero Waste Box™as a solution for pens, pencils, and markers as waste to the environment. A box is utilized to collect

waste writing instruments and recycle this collected waste. The collected wastes consist of pens and pen caps, mechanical and wooden pencils, markers and marker caps, permanent markers, and permanent marker caps.

2.3.3 Processing of the waste

Once collected, the waste writing instruments are separated by material composition. The separated items are then cleaned, shredded, and made into new recycled products.

2.4 Summary

This chapter has discussed how the PPVM is different from conventional vending machines, what utilities were used for the software part, and how the collected pens can be recycled. From these discussions, it is possible to see the practicality of the PPVM.

Chapter 3

DESIGN/DATA COLLECTION

The survey conducted in Government Engineering College Bartonhill was done using the open source tool Open Data Kit (ODK). This chapter explores Open data Kit, workflow of ODK for ppvm.

3.1 ODK: A brief Introduction

The Open Data Kit (ODK) community produces free and open-source software for collecting, managing, and using data in resource-constrained environments. It allows for the collection of data offline and submission of the data when internet connectivity is available. It allows communities to aggregate data with full control over the collected data and the servers where this data is stored. ODK Development Team (1984)

3.2 Components of ODK

- **ODK-Collect**: Android Open Source App for Data Collection even for offline use in remote areas without internet connectivity.
- **ODK Build**: Component is used for designing a questionnaire for ODK. It works as a drag-and-drop form designer for ODK XForms. It is used for data collection campaign e.g. for Health Sites
- **ODK Aggregate**: The ODK Aggregate is the backend of ODK infrastructure, receiving the data from the mobile devices. To be multiplatform it is designed as Open Source Java server, that stores, analyzes, and presents survey data. Decision support is build on the collected data.

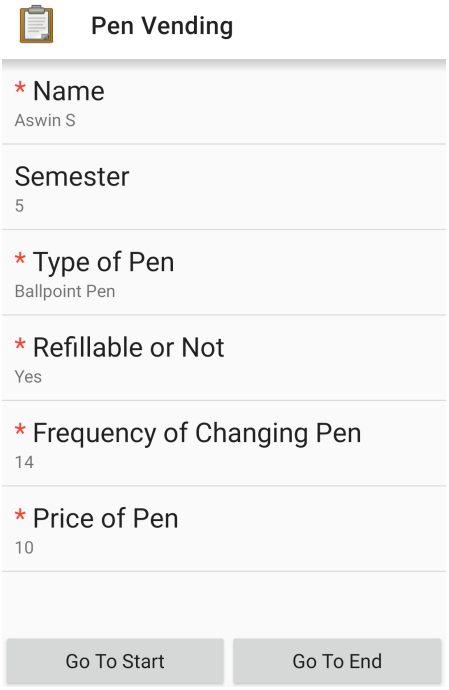


Figure 3.1: Interface of ODK-Collect

3.3 Workflow for data collection

- Download or Create a questionnaire for data collection, which is available for offline use.
- Collect the data, even if device is offline.
- Submit collected data to ODK Aggregate.
- Access aggregated results for individual decision support (optional).

3.3.1 Workflow for data collection of PPVM

- A form was created using odk build and was saved to ODK server.
- This form was exported as a .xml(eXtensible Markup Language) format file was used with ODK collect android app to carry out the survey.
- The data collected was saved within the ODK-collect app and the save was finalised at the end of the survey.
- This finalised data was then uploaded to ODK aggregate sand box server.

- The data uploaded was exported as .csv(comma seperated values) format file and opened using spreadsheet to do further estimations.

3.3.2 Survey

The survey conducted within Government Engineering College Barton Hill was carried out with ODK collect android application. The data collected consists of the following fields:

- Type of pen.
- Whether refillable or not.
- Frequency of changing of pen.
- Price of pen.

The data collected is given in Appendix B

3.4 Data Analysis

From the data collected the following estimations were deduced per person:
Average frequency of throwing a pen, f = 16 days
$\therefore$ Number of pens thrown in a day = $\frac{1}{f}$= $\frac{1}{16}$
$\implies$ Number of pens thrown in a year n_t = $\frac{1}{16} \times 365 \approx 23$
Now,
Approximate population of campus P= 1400
Total pens thrown per year in college, N_t = $n_t \times P$ = 23 $\times$ 1400 = 32,200
Average weight of plastic pen w_1= 6 gm = 0.006kg
Thus, weight of all pens thrown in a year W_1 = $n_t \times w_1$ = 32,200 $\times$ 0.006 kg = 193 kg
Amount of CO_2 by burning 1 kg of plastic pens c_1 = 6 kg
Estimated amount of CO_2 emitted per year within the campus C_1 = $c_1 \times W_1$ = 1,159 kg
Ratio of plastic pens accepted to the paper pens given = 3:1
Number of paper pens placed in the machine N_p= $\frac{N_t}{3}$ = 10,733
Weight of a paper pen $\approx$ Weight of a plastic pen w_p = 0.006 kg
Total weight of paper pens given W_p= $N_p \times w_p$ = 64.4kg
Amount of CO_2 by burning 1 kg of paper pens , c_2 = 0.6 kg
Estimated amount of CO_2 emitted by paper pens per year within the campus,
C_2 = $c_2 \times W_p$ = 38.6 kg
Estimated difference in emission of CO_2, $C_1 - C_2$ = 1120.4 kg

3.5 PPVM Design

For the prototype, it was decided that a combination of both acrylic sheet and foam board would be the best, Fig 3.2. The front of the machine had to be white, so as to provide a white background while taking the image. The sides of the machine was made with transparent acrylic sheet. A transparent box with tiltable base was attached to the interior of the machine. It was open from the top so that the pen can be entered. The ramp on which the paper pens are stored and rolled out are made with foam board. The rotating part is done using servo motors. The camera and the Raspberry Pi are attached inside.

Figure 3.2: PPVM body

3.6 Estimation

Table 3.1: Table showing the cost estimation

Sl. No.	Component	Quantity	Price
1	Acrylic Sheet (2mm)	6 sq. ft.	550
2	Foam Board (5mm)	4 sq. ft.	160
3	Raspberry Pi 4 (1 GB)	1	3400
4	Raspberry Pi camera	1	1900
5	Servo Motors (MG995)	1	700
	Total		6710

3.7 Summary

- Data was collected using ODK-Collect.
- The collected data was used to estimate the amount of CO_2 prevented from releasing to atmosphere.
- Estimated manufacturing cost: Rs. 6,710.
- Prototype made with a combination of acrylic sheet and foam board.

Chapter 4

DESIGN ANALYSIS

As mentioned in the Section 2, the proposed vending machine accepts used plastic pens instead of coins. The used pen will be deposited through the top portion of PPVM. A rubber flap will be provided which will automatically open and close when the pen is inserted. A barrier will be provided below the flap which can be used to block the entry of the pen when the machine is off. Once the pen is dropped, it falls into a transparent box. This box is attached to a base that can tilt in two directions. The object is accepted or rejected after the image processing.

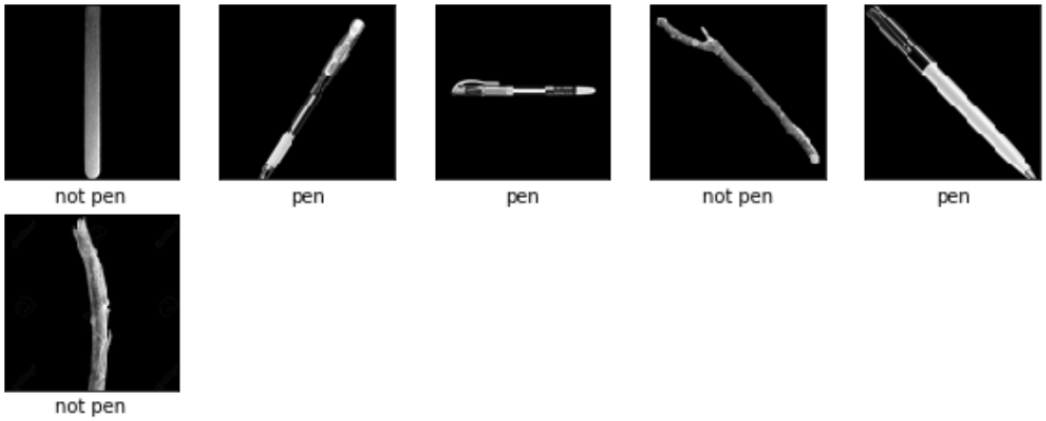
4.1 Image capture

A camera attached inside the machine scans the transparent box. It takes an image and sends it to the processor of the machine. For the prototype, a Raspberry Pi 3b was used. The processor uses pattern recognition and image classification to determine whether the object inserted is a pen or not. If it is a pen, the processor sends a signal to the tiltable base and the pen is redirected to the storage bin inside the machine. If it is not a pen, then the base ejects the object out of the machine. After insertion of three pens into the machine, a signal is sent to the rotating delivery system that will roll out one pen and deliver it to the user.

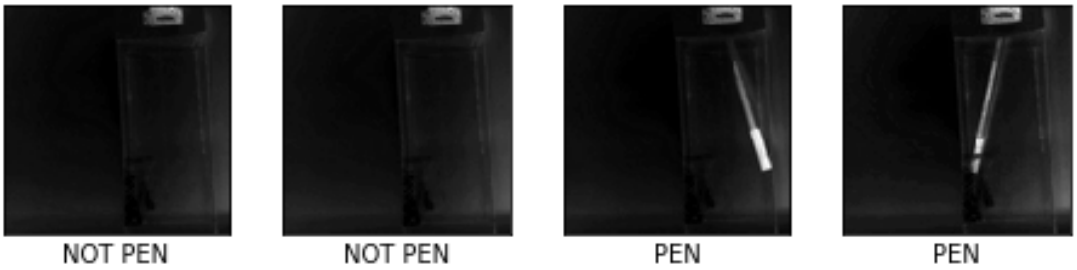
4.2 Image recognition and classification

The image capture and manipulation was done with the help of a python library called Open-source Computer Vision or OpenCV Intel (1999). For the recognition process, “TensorFlow” was used, as mentioned in Section 2.2. Using a training dataset consisting of images of pens, a model pattern of a pen was created. This model is then

compared with a trial dataset to determine the accuracy of the model. If the accuracy is low, the model is re-trained. The camera takes a picture of the waste pen and adds it to the test dataset. Once recognized, the image is properly labeled, either “pen” or “not pen”, and then adds this to the training dataset. Then the model is retrained. Thus, by implementing machine learning, PPVM will become more and more accurate with each use. See Appendix A.1 for the detailed code. An example of the image classification during the testing stage is shown in Fig 4.1(a). This algorithm optimized and implemented in the prototype, gave the results as shown in Fig 4.1(b). A flowchart showing this algorithm is shown in Fig 4.2.



(a) Image classification during tests



(b) Image classification implemented in prototype

Figure 4.1: Image recognition and classification algorithm

4.3 Model Design

Careful calculations were done for making the prototype. The PPVM needed to be as small as possible and the space needed to be utilised as much as possible. The model drawing of the PPVM is shown in Fig 4.3 with all dimensions in inches. The prototype made is shown in Fig 4.4.

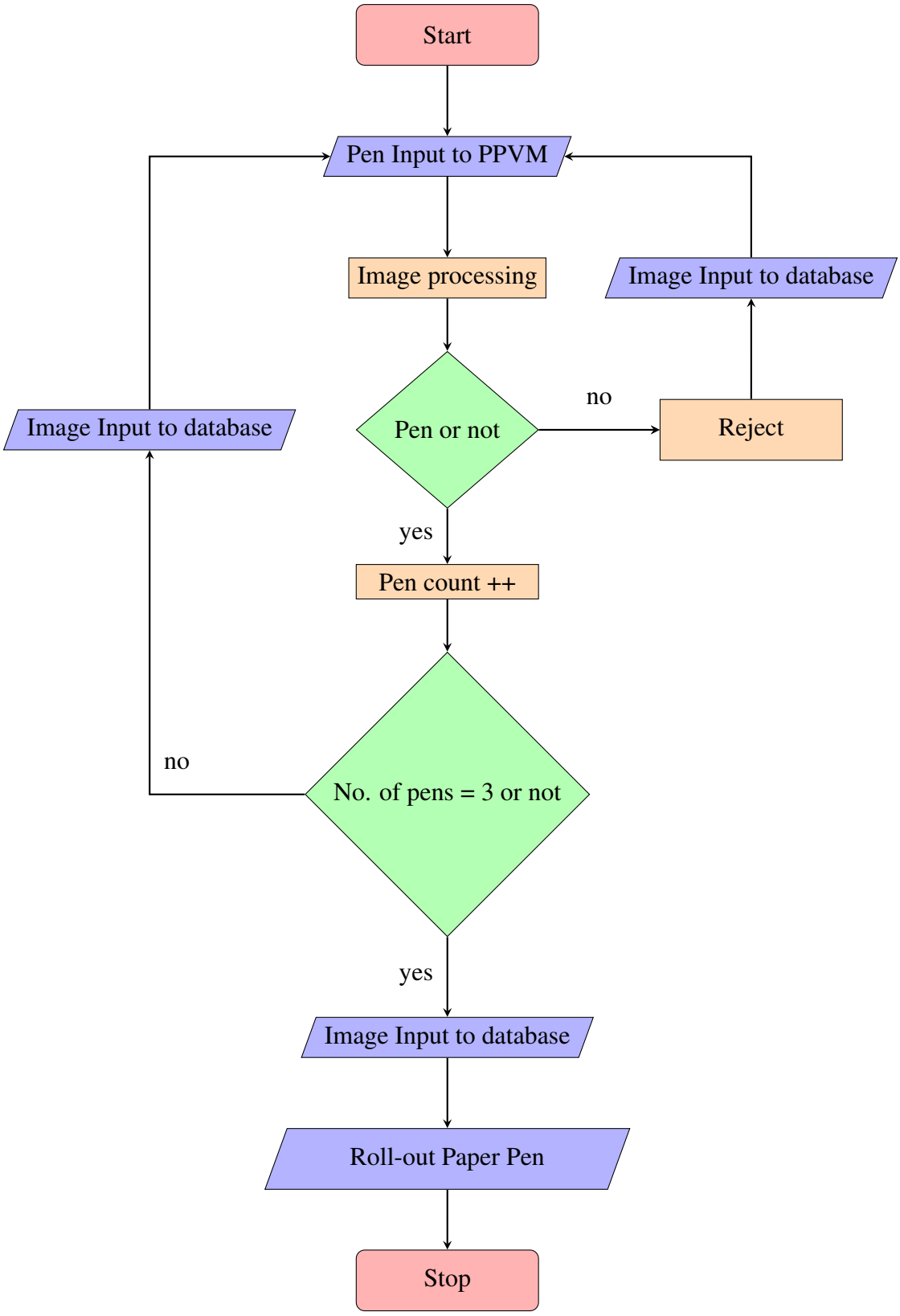


Figure 4.2: Flowchart showing the algorithm of the PPVM

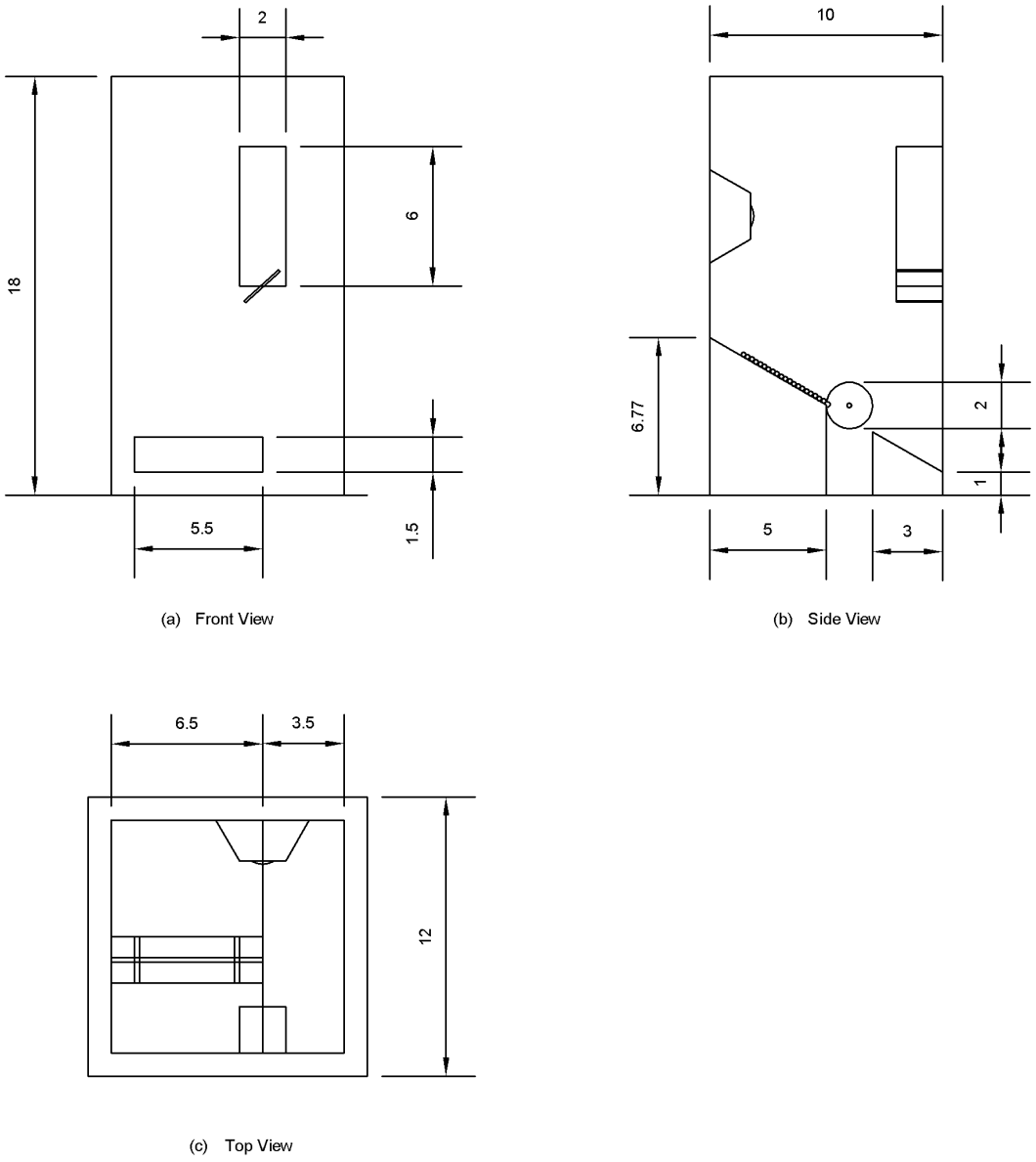


Figure 4.3: A drawing of the proposed design (all dimensions are in inches)

(a) Isometric view

(b) Top view

(c) Side view

Figure 4.4: PPVM prototype

4.4 Pen Accepting and Rejecting

The tiltable base redirects the pen depending on whether it is a pen or not. The base is connected to a servo motor. The servo motor will turn the base left if it accepts and turns right if it rejects the deposited item. The rejected item will go out through the same ramp which is used for delivering paper pen.

4.5 Rollout Mechanism

For delivering the paper pen, a pair of circular discs are used. These discs are attached to a thin aluminum shaft which is connected to a servo motor. A small groove is cut on the outer portion of the discs, to fit just one pen. The paper pens are stacked on a ramp behind the discs. When a pen is being rolled out, the outer part of the disc prevents the stored pens from falling. The rolled-out pens fall on a ramp in front of the discs and are delivered to the user as shown in Fig 4.5

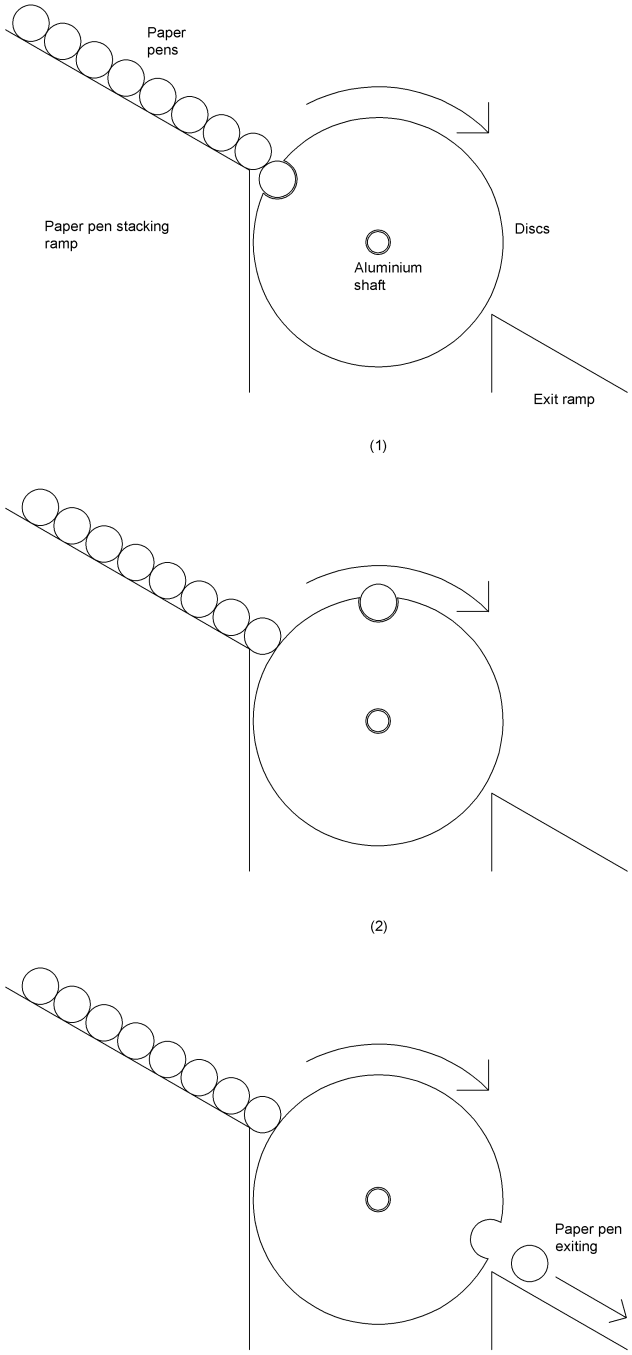


Figure 4.5: The rollout mechanism

4.6 Summary

In this chapter, the main discussion was on the design of the model and its functioning. The algorithms and the procedure of operation was explained in detail. A model drawing of the prototype was also shown.

Chapter 5

DISCUSSIONS

During the initial phase of the project, the problem statement had to be analyzed through data collection and analysis. Thus, small scale surveys were conducted in GECBH using standard methods. After a while, a better way to conduct these surveys was found and hence ODK was adopted as the prime tool.

While discussing the software part of the machine, the initial thought was to go with image processing. But this method was proven to be more time-consuming. A better way to tackle the issue was pattern recognition. It was soon understood that there was abundant documentation available on the internet regarding pattern recognition and image classification. Thus, TensorFlow - a utility by Google - was chosen for this task.

The body of the machine needed to be sturdy and strong enough to withstand the components and the rotating servo. For the prototype, it was decided that a combination of foam board and acrylic sheet would be the best. The bigger parts would be joined together using clamps and the smaller ones with glue.

To make the machine autonomous, a microprocessor needed to be attached to the machine. This processor should also be able to act as a microcontroller and the best candidate for this was Raspberry Pi.

Chapter 6

CONCLUSIONS

The prototype of the PPVM was successfully completed. Initially the accuracy of detection was not excellent. The accuracy was improved by increasing the number of epochs while training the model. It is also expected that with further usage of this machine, it will become more and more accurate, as mode samples will be added to the dataset.

Through this project, the following concepts were learned

- Basics of TensorFlow and Machine Learning
- Basics of Raspberry Pi
- Basics of OpenCV
- Design and function of a vending machine

Many issues were faced while working on Raspberry Pi. It was difficult to install TensorFlow and OpenCV in Pi and took many hours. The HDMI port of Pi stopped working after a while. Further work on Pi was done by enabling remote access using a laptop. Even this method was not perfect and it failed to connect with three different laptops. While dealing with each problem, new concepts were learned about programming, designing, and networking.

The implementation of this machine is expected to mitigate the emission of CO_2, as mentioned in Chapter 3. For this, a stable and durable model needs to be constructed. More studies need to be conducted for perfecting and implementing this machine.

References

Google Brain Team. (2017). Open-source software library [Computer software manual]. Retrieved from https://www.tensorflow.org/ ([Online; accesed 19-November-2019])

Intel. (1999). Open-source computer vision [Computer software manual]. Retrieved from https://opencv.org/ ([Online; accessed 23-November-2019])

ODK Development Team. (1984). Open data kit [Computer software manual]. Retrieved from https://opendatakit.org/ ([Online; accessed 23-November-2019])

Sharma, P. (2019). Image recognition with tensorflow and keras. Retrieved from https://developer.ibm.com/articles/image-recognition-challenge-with-tensorflow-and-keras-pt1/

Wikipedia contributors. (2019). *Terracycle — Wikipedia, the free encyclopedia.* Retrieved from https://en.wikipedia.org/w/index.php?title=TerraCycle&oldid=926740878 ([Online; accessed 23-November-2019])

Appendix A

A.1 Code

A.1.1 For creating model

```
# TensorFlow and tf.keras
import tensorflow as tf
from tensorflow import keras
import h5py
# Helper libraries
import numpy as np
import matplotlib.pyplot as plt
import glob, os
import re
# Directories of datasets
test_dir = '/home/pi/tensorflow1/Pen/PPVM/pen_dataset/test_set/'
trial_dir = '/home/pi/tensorflow1/Pen/PPVM/pen_dataset/trial_set/'
train_dir = '/home/pi/tensorflow1/Pen/PPVM/pen_dataset/train_set/'
# Pillow
import PIL
from PIL import Image
# Use Pillow library to convert an input jpeg to a 8 bit grey scale
    image array for processing.
def jpeg_to_8_bit_greyscale(path, maxsize):
        img = Image.open(path).convert('L')   # Convert image to 8-
    bit grayscale
        # Make aspect ratio as 1:1, by applying image crop.
        # Needs to locate the subject and then crop or scale
    accordingly.
        WIDTH, HEIGHT = img.size
        if WIDTH != HEIGHT:
                m_min_d = min(WIDTH, HEIGHT)
                img = img.crop((0, 0, m_min_d, m_min_d))
```

```
        # Scale the image to the requested maxsize by Anti-alias sampling.
        img.thumbnail(maxsize, PIL.Image.ANTIALIAS)
        return np.asarray(img)
# Load the images
def load_image_dataset(path_dir, maxsize):
        images = []
        labels = []
        os.chdir(path_dir)
        for file in glob.glob("*.jpg"):
                img = jpeg_to_8_bit_greyscale(file, maxsize)
                if re.match('pen.*', file):
                        images.append(img)
                        labels.append(0)
                elif re.match('not_pen.*', file):
                        images.append(img)
                        labels.append(1)
        return (np.asarray(images), np.asarray(labels))
maxsize = 100, 100
(train_images, train_labels) = load_image_dataset(train_dir, maxsize)
(trial_images, trial_labels) = load_image_dataset(trial_dir, maxsize)
class_names = ['PEN', 'NOT PEN']
train_images.shape
(26, 100, 100)
print(train_labels)
trial_images.shape
(14, 100, 100)
print(trial_labels)
# Display the images
def display_images(images, labels):
        plt.figure(figsize=(10,10))
        grid_size = min(25, len(images))
        for i in range(grid_size):
                plt.subplot(5, 5, i+1)
                plt.xticks([])
                plt.yticks([])
                plt.grid(False)
                plt.imshow(images[i], cmap=plt.cm.binary)
                plt.xlabel(class_names[labels[i]])
display_images(train_images, train_labels)
plt.show()
train_images = train_images / 255.0
trial_images = trial_images / 255.0
# Setting up the layers
```

```
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(100, 100)),
        keras.layers.Dense(128, activation=tf.nn.sigmoid),
        keras.layers.Dense(16, activation=tf.nn.sigmoid),
    keras.layers.Dense(2, activation=tf.nn.softmax)
])
sgd = keras.optimizers.SGD(lr=0.01, decay=1e-5, momentum=0.7,
   nesterov=True)
model.compile(optimizer=sgd,
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
loops=10000
# Trying to create a model
model.fit(train_images, train_labels, epochs=loops)
# Measure accuracy using trial dataset
trial_loss, trial_acc = model.evaluate(trial_images, trial_labels)
print('Trial accuracy:', trial_acc)
predictions = model.predict(trial_images)
print(predictions)
display_images(trial_images, np.argmax(predictions, axis = 1))
plt.show()
# Re-train if model is not accurate enough
while (trial_acc<0.90):
    loops = loops + 200
    model.fit(train_images, train_labels, epochs=loops)
    trial_loss, trial_acc = model.evaluate(trial_images, trial_labels
    )
    print('Trial accuracy:', trial_acc)
    predictions = model.predict(trial_images)
    print(predictions)
    display_images(trial_images, np.argmax(predictions, axis = 1))
    plt.show()
#saving the model
model.save('/home/pi/tensorflow1/Pen/PPVM/save_model/my_model.h5')
```

A.1.2 For processing in PPVM

```
accept=0
reject=180
rollout=180
reset=0
#for delays
from time import sleep
#for controlling pins in raspberry pi
import RPi.GPIO as GPIO
```

```
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
#servo pins
pan = 33
tilt = 12
GPIO.setup(tilt, GPIO.OUT)
GPIO.setup(pan, GPIO.OUT)
#Servo function
def setServoAngle(servo, angle):
  #  assert angle >=30 and angle <= 150
    pwm = GPIO.PWM(servo, 50)
    pwm.start(8)
    dutyCycle = angle / 18. + 3.
    pwm.ChangeDutyCycle(dutyCycle)
    sleep(0.3)
    pwm.stop()
#switch function if switch is being used
def switch1():
    GPIO.setup(11, GPIO.IN, pull_up_down=GPIO.PUD_UP)
    while True:
        input_state = GPIO.input(11)
        if input_state == False:
           print('Button Pressed')
           time.sleep(5)
#capturing image using camera
def imgcap():
    # Importing all necessary libraries
    import cv2
    import time
    # Read the video from specified path
    cam = cv2.VideoCapture(0)
    time.sleep(2)
    ret,frame = cam.read()
    name = '/home/pi/tensorflow1/Pen/PPVM/pen_dataset/test_set/0.jpg'
    print ('Creating...' + name)
    cv2.imwrite(name, frame)
    gamma_correction(name)
    # Release all space and windows once done
    cam.release()
    cv2.destroyAllWindows()
#for gamma correction
def gamma_correction(img):
    import numpy as np
    import cv2
```

```
    def adjust_gamma(image,gamma=1.0):
        invGamma = 1.0/gamma
        table = np.array([((i/255.0) ** invGamma) *255
                          for i in np.arange(0,256)]).astype("uint8")
        return cv2.LUT(image, table)
    original = cv2.imread(img)
    adjusted = adjust_gamma(original, gamma=2.5)
    cv2.imwrite(img,adjusted)
#executing tf algorithm
def execute(l):
    # TensorFlow and tf.keras
    import tensorflow as tf
    from tensorflow import keras
    # Helper libraries
    import numpy as np
    import matplotlib.pyplot as plt
    import glob, os
    # Directories of datasets
    test_dir = '/home/pi/tensorflow1/Pen/PPVM/pen_dataset/test_set/'
    train_dir = '/home/pi/tensorflow1/Pen/PPVM/pen_dataset/train_set/
    '
    # Pillow
    import PIL
    from PIL import Image
    # Use Pillow library to convert an input jpeg to a 8 bit grey
    scale image array for processing.
    def jpeg_to_8_bit_greyscale(path, maxsize):
            img = Image.open(path).convert('L')   # Convert image to
    8-bit grayscale
            # Make aspect ratio as 1:1, by applying image crop.
            # Needs to locate the subject and then crop or scale
    accordingly.
            WIDTH, HEIGHT = img.size
            if WIDTH != HEIGHT:
                    m_min_d = min(WIDTH, HEIGHT)
                    img = img.crop((0, 0, m_min_d, m_min_d))
            # Scale the image to the requested maxsize by Anti-alias
    sampling.
            img.thumbnail(maxsize, PIL.Image.ANTIALIAS)
            return np.asarray(img)
    # Display the images
    def display_images(images, labels):
            plt.figure(figsize=(10,10))
            grid_size = min(25, len(images))
```

```
92              for i in range(grid_size):
93                      plt.subplot(5, 5, i+1)
94                      plt.xticks([])
95                      plt.yticks([])
96                      plt.grid(False)
97                      plt.imshow(images[i], cmap=plt.cm.binary)
98                      plt.xlabel(class_names[labels[i]])
99      # Load test image
100     def load_image_dataset_test(path_dir, maxsize):
101             images = []
102             os.chdir(path_dir)
103             for file in glob.glob("*.jpg"):
104                     img = jpeg_to_8_bit_greyscale(file, maxsize)
105                     images.append(img)
106             return (np.asarray(images))
107     # Rename as Pen and add to training dataset
108     def Rename_pen():
109         for filename in os.listdir(test_dir):
110             dst = "pen" + str(l) + ".jpg"
111             src = test_dir + filename
112             dst = train_dir + dst
113             os.rename(src,dst)
114     # Rename as Not Pen and add to training dataset
115     def Rename_not_pen():
116         for filename in os.listdir(test_dir):
117             dst = "not_pen" + str(l) + ".jpg"
118             src = test_dir + filename
119             dst = train_dir + dst
120             os.rename(src,dst)
121     maxsize = 100, 100
122     class_names = ['PEN', 'NOT PEN']
123     test_images = load_image_dataset_test(test_dir, maxsize)
124     new_model = tf.keras.models.load_model('/home/pi/tensorflow1/Pen/
    PPVM/save_model/my_model.h5')
125     # Predict the images in test dataset
126     predictions = np.round(new_model.predict(test_images))
127     print(predictions)
128     # Display the prediction during debugging
129     #display_images(test_images, np.argmax(predictions, axis = 1))
130     #plt.show()
131     # Rename and move to training dataset for improving accuracy
132     if (predictions[0,0]==1):
133         Rename_pen()
134         l+=1
```

```
        setServoAngle(tilt, accept)
        return l
    elif (predictions[0,0]==0):
        Rename_not_pen()
        #import Servo_right
        setServoAngle(tilt, reject)
        return l
while(True):
    #switch1() if we are using switch
    setServoAngle(pan, 0)
    setServoAngle(tilt, 90)
    sleep(1)
    while(not(l==3)):
        imgcap()
        l=execute(l)
        sleep(2)
        print(l)
    setServoAngle(pan, rollout)
    sleep(2)
    setServoAngle(pan, reset)
    sleep(2)
```

Appendix B

B.1 Data collected

Sl.No	NAME	Type of Pen	Price	Frequency of changing Pen
1	Sreeram	Plastic Ball Point	3	30
2	Adarsh	Plastic Ball Point	10	15
3	Ansar	Plastic Ball Point	3	20
4	Nihal	Plastic Ball Point	3	15
5	Vinayak	Plastic Ball Point	10	20
6	Aromal	Plastic Ball Point	3	30
7	Dr.Rajesh N. R	Plastic Ball Point	10	20
8	Gayathri	Plastic Ball Point	3	5
9	Chethas	Plastic Ball Point	10	3
10	Vishnu	Plastic Ball Point	3	10
11	Augustine	Plastic Ball Point	5	15
12	Sandeep	Plastic Ball Point	10	5
13	Akhil	Plastic Ball Point	5	7
14	Aswin	Plastic Ball Point	10	14
15	Josh	Plastic Ball Point	10	5
16	Joel	Plastic Ball Point	10	21
17	Yadu	Plastic Ball Point	10	30
18	Jaykrishnan	Plastic Ball Point	3	14
19	Prahlad	Plastic Ball Point	3	14
20	Anagha	Plastic Ball Point	10	7
21	Mahadev	Plastic Ball Point	10	21
22	Sreeraj	Plastic Ball Point	5	14
Average			6.277273	15.227