REVERSE VENDING MACHINE FOR PLASTIC BOTTLES RECYCLING

A Proposed Project Study

Presented to the

Department of Computer Engineering

College of Engineering and Architecture

Negros Oriental State University

Main Campus II, Bajumpandan, Dumaguete City

In Partial Fulfillment of the Requirement for the Degree of

Bachelor of Science in Computer Engineering

by

JOSHUA I. BARLOLONG

JC DEL VINCENT R. ESCALA

PROPOSAL APPROVAL SHEET

ACKNOWLEDGMENTS

This study becomes a reality with the kind support of many individuals. We would like to extend our sincerest gratitude to all of them.

Foremost, We want to offer this endeavor to our God Almighty for the wisdom, peace of mind, good health, and strength he bestowed upon us in order to finish this project.

We would like to express our gratitude to our parents for the encouragement and financial support which helped us in the completion of this paper.

Our thanks and appreciation to all people who have willingly helped out to the best of their abilities.

ABSTRACT

One of the most problematic issues in today's world is the piling up of disposable plastic bottles in our landfills that take a long time to decompose. This type of problem has already been studied and formulated with different solutions. Throughout the years, the solutions that have been implemented seems not fully effective in motivating people to recycle plastic bottles. For this purpose, the researchers have developed a reverse vending machine that gives monetary rewards to users who successfully stored accepted plastic bottles in the system. By utilizing artificial intelligence, fraudulent transactions such as the input of non-plastic objects will be prevented. The machine presents a decent percentage of distinguishing plastic bottles from other objects through AI vision. This study proposes a great potential in solving the issue of the recycling of plastic bottles when implemented properly and improved with more cost-effective materials.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Through the advancement of computers, automation of processes using Artificial Intelligence and other modern methods is not new. The only thing that needs to be done is to innovate and make use of these technologies to come up with solutions to the current problems we face, not just being efficient but also being cost-effective. Solid waste is a common and huge issue in today's world due to its harmful impacts on both animals and the environment. Plastic waste, on the other hand, is quite troublesome due to its non-biodegradability as a large component and part of urban solid waste, which will, as a result, linger in the environment for an extended period, causing a variety of problems (Dioses, 2020). Currently, there are various types of studies that aim to mitigate the problem of waste plastic bottles that have been flooding our landfills every day with no signs of stopping.

Mariya et al. (2020) stress that programs of their government such as placing recycling bins in public places are inconvenient and ineffective as the recycling process demotivates people from doing so. Another system has been implemented to boost recycling by paying cash or shopping coupons to those who bring recyclable items to the shopping mall. However, the program cannot be fully implemented as it requires a full team to manage it (p. 65). A potential solution to address this problem is the adoption of a reverse vending machine that accepts plastic bottles in exchange for credits. With this kind of approach, individuals are encouraged to deposit their plastic bottles, which could drastically help with waste management. By utilizing Artificial Intelligence's image recognition, the researchers aim to lessen the cost of building this project by minimizing the use of sensors that will be needed to identify whether the deposited item is not fraudulent. The researchers aim to cover most types of single-use plastic bottles that can be found and bought in the Philippines, which will not just help in lessening the waste management problem but also potentially revolutionize the way people dispose of their garbage leading to great possibilities for preserving our environment.

1.1 Statement of the Problem

Recycling alone is a daunting task that results in improper waste disposal of single-use plastic bottles which although convenient to people but detrimental to the environment. Verbal encouragement seems not sufficient enough to inspire people to start recycling. As a result, the researchers come up with a study that aims to provide solutions to the following questions:

1. What technologies are going to be utilized to distinguish plastic bottles from other types of garbage?
2. What countermeasures will be implemented for fraudulent items that will compromise and exploit the system?
3. How will the system compensate the users for their deposited items?
4. What types of plastic bottles does the system accept?

1.2 Objectives

In this study, the researchers aim to achieve the following:

1. To develop software that utilizes an AI vision system to distinguish plastic bottles from non-plastic bottle waste materials.
2. To develop a system using Raspberry Pi that accepts only allowed plastic bottles and rejects not allowed objects.

1.3 Significance of the Study

With the current waste management problem this study aims to provide benefits to the community through the following:

1. It provides the community with fast and convenient plastic bottle disposal.
2. It encourages people to promote recycling through rewards.
3. It manifests a versatile system that is upgradeable depending on the current demands.

1.4 Feasibility of the Study

With the current technologies, image recognition using artificial intelligence paves the way to being the main key player in simplifying projects that use many sensors to detect and prevent fraudulent actions towards potential exploitation of bugs and weaknesses of the system. Raspberry Pi provides an efficient and cheap solution in making a system that accepts and rejects predefined items utilizing AI object recognition. Being an open-source computer that has large community support, it is easy to find tutorials and documentation on every component that is being used in making the electronics system of this particular study.

1.5 Scope and Limitations

The goal of this study is to create a reverse vending machine. This machine will utilize an object recognition system that will distinguish plastic bottles and non-plastic bottle objects. It is a Raspberry Pi-based project that uses various components: servo motors to control rotating parts of the system, a load cell with an HX711 amplifier to measure the weight of the inserted objects, also to detect when an object is inserted, an LCD to display data, a button to dispense the acquired coins, LEDs and buzzer for indications.

The following constraints and features apply to this project study:

1. The system will accept most plastic bottles ranging from 200mL to 1.5L and reject bottles weighing above 3g.
2. The weight sensor will measure the weight of the objects and will reject those exceeding 3g.
3. The coin dispenser is limited only to 100 pieces of 1 peso coins.
4. The storage has no press machine to flatten plastic bottles.

CHAPTER 2

REVIEW OF RELATED LITERATURE

This chapter presented the synthesis of literature and studies relating to the recycling of plastic bottles, the development of a reverse vending machine using an Arduino microcontroller, and the usage of artificial intelligence object recognition on detecting and distinguishing plastic bottles from non-bottle waste materials.

Around 6,270 million pounds of PET bottles were sold on the market in 2018. In the United States a total of 1,816 million pounds, or 29.0% was gathered and sold through recycling initiatives with exports continuing to fall, PET reclaimers in the United States processed 1,676 million pounds of material from the United States, which is 16% higher than domestic bottle reclamation in the United States in 2017 (NAPCOR, 2018). Container deposit-refund legislation has been introduced in 23 nations over the last two decades to increase recycling rates. These rules, which encourage recycling by returning the container deposit, have been particularly successful in Finland and Norway, with return rates of all three container types exceeding 91 percent in Finland and 96 percent in Norway (“Liukkonen,” n.d.).

2.1 Related Studies

2.1.1 A Simple Approach to Design Reverse Vending Machine

Gaur and Priyadarshini (2018) successfully simulated a prototype of a smart plastic recycle machine using Xilinx 14.5 ISE Simulator. The user inputs a plastic material into the system, the system enables the summation of points throughout the recycling process. The reward coins and weight of the plastic increments with every input to the system. If the user inputs an invalid type of item into the system, it cannot accept the input material and resets the system. The system worked efficiently and is cost-effective by utilizing a programmable hardware-based detection system using a capacitive proximity sensor, an infrared photoelectric sensor, and a strain gauge weight sensor.

2.1.2 Development of Reverse Vending Machine using Recycled Materials and Arduino Microcontroller

Kai Kit Wong et al. (2019) built a prototype of a reverse vending machine specifically for recycling with a reward feature. Their design is expected to aid in motivating Malaysians to recycle their waste. Their project can be one of the frameworks to overcome urban poverty issues by using the waste-to-wealth concept. The project is implemented with an Arduino microcontroller and several sensors. The microcontroller-based system for metal detection and reward system was designed in a step-by-step manner, once the system detects the correct metal for recycling, the users get an instant reward. The findings of the number of metals detected and recycled are accurate and dependable, allowing users to be appropriately compensated.

2.1.3 Reverse Vending Machine for Plastic Bottle Recycling

Mariya et al. (2020) developed a prototype design to check the work of the simulation and synthesis. The system takes an input then records the type of input and displays the output on the LCD. The depositor presses a push button to enter a unique password before inserting the bottles. They trained their AI using the Tensorflow model. The model is used for plastic bottle detection in Google Colab using a custom dataset that has been trained with LabelImg, a graphical image annotation tool. They found it to be very efficient in plastic bottle detection.

2.1.4 Bottle-SegreDuino: An Arduino Frequency-Based Bin for Tin Can and Plastic Bottle Segregation using an Inductive Proximity Effect

Dioses, Jr. (2020) conducted a study for plastic bottle and tin can segregation using Arduino Uno. The system uses an inductive proximity sensor to separate the tin cans and plastic bottles. The study proved 90& and 98% testing accuracy for plastic bottles and tin cans. They stated that the overall accuracy produced by the whole system was 94%. The project's aim has been achieved and has completed an effective implementation of plastic bottles and tin can segregator.

2.2 Synthesis

2.2.1 Similarities

The Reverse Vending Machine for Plastic Bottles Recycling system is inspired and similar in some ways to the studies mentioned above. It uses a computer to retrieve data from the sensors and control the components such as servo motors, and LCD. It also uses an AI image recognition system to distinguish objects: bottle and non-bottle.

2.2.2 Differences

The main difference in our system from the other studies is that we don't use an Arduino microcontroller as the computer that will operate the AI image recognition system that will open the software and serve the sorting algorithm into an output. Instead, the system will use a Raspberry Pi, in this method it will serve as the power source of the components except for the servos which will be powered externally. The Raspberry Pi will be the host of the software that contains the AI algorithm written in Python.

CHAPTER 3

THEORETICAL AND CONCEPTUAL FRAMEWORKS

3.1 Theoretical Framework

3.1.1 Object Detection: Single Shot Multibox Detection (SSD)

Single Shot Multibox Detector is a deep learning model that has been used to detect objects in an image or a video source. Single Shot Detector (SSD) has two components: Backbone Model and SSD Head. The Backbone Model is a pre-trained image classification network used as a feature extractor. SSD Head is a set of convolutional layers added to the backbone and the outputs are interpreted as the bounding box and classes of objects inside the spatial location of the final layer's activations.

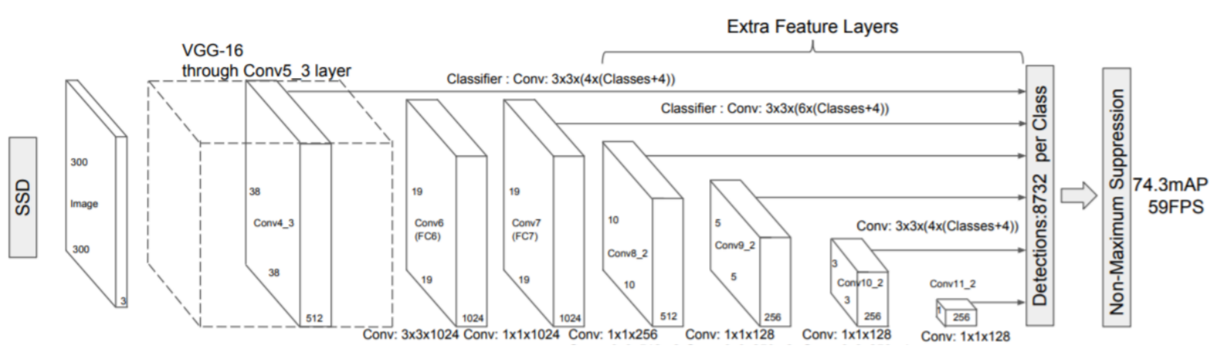

Figure 1: SSD Architecture Model, Paper: https://doi.org/10.1007/978-3-319-46448-0_2

Instead of using the traditional sliding window algorithm, Single Shot Detector divides the images into grids, and each of the grid cells is responsible for detecting objects in that region of the image. If there is no object detected, it will output as nothing or to be precise, the output will be "0" indicating that there is no object found.

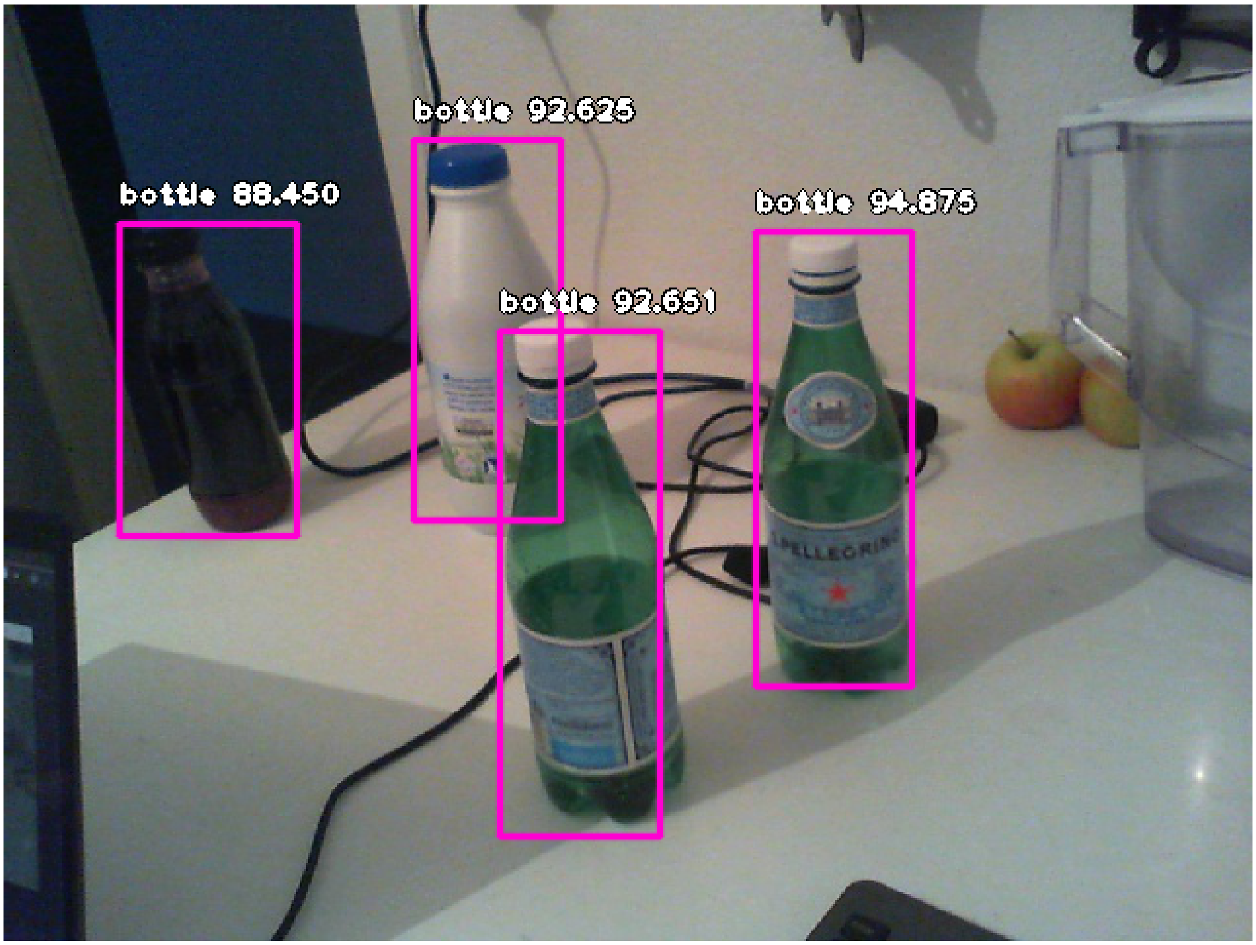

Figure 2: Anchor Boxes
(https://apiquet.com/2019/06/02/visual-recognition/)

Anchor Boxes are boxes that are assigned with multiple anchors. These are predefined and have fixed sizes and shapes within their grid cell. Based on the image above, multiple objects can be detected in an image.

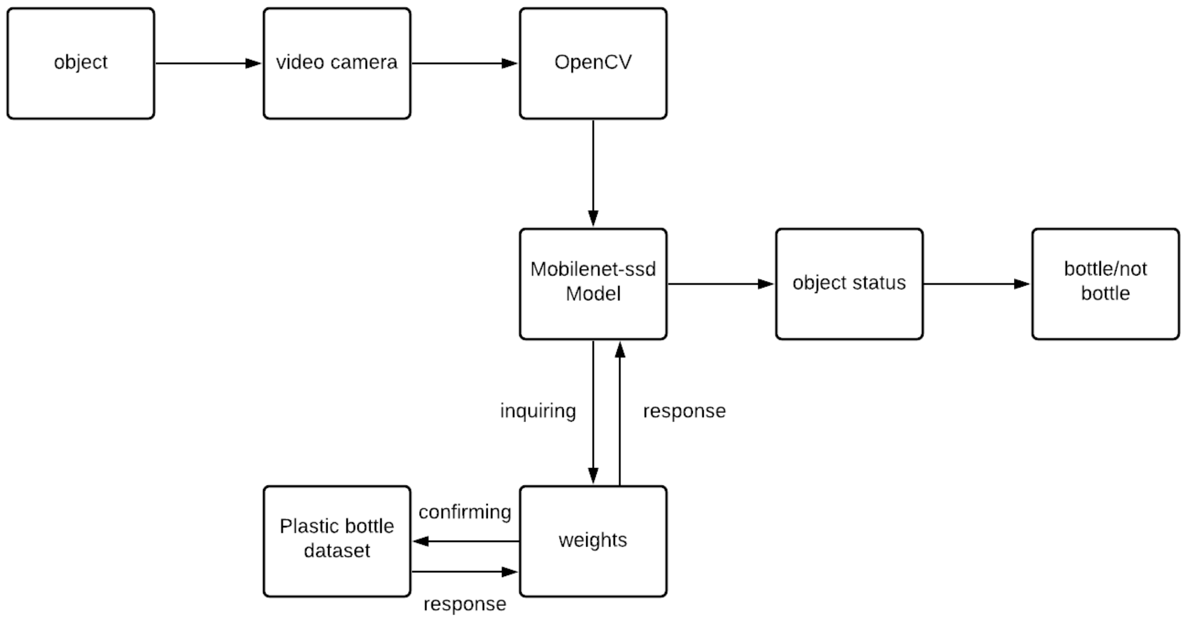

Figure 3: A Framework for Plastic Bottle Detection

The figure above shows that when an object is detected by the video camera it will be processed by the OpenCV library. The Mobilenet-SSD model which contains the configurations then identifies the type and bounding box of all instances of those classes appearing on the image. The weights which contain the different variations of instances of a class then confirm and match those detected instances to those instances in the dataset. The weights will send a response containing the instances for configuration and prediction. The model then identifies the object as positive or negative, after that it will output "bottle" or "not bottle" as a result.

3.2 Conceptual Framework

The Reverse Vending Machine for Plastic Bottles Recycling System is regulated by various existing principles, concepts, and rules that serve as the foundation for its design and production.

The Input-Process-Output (IPO) model is a functional graph that identifies the inputs, outputs, and required processes to transform the defined inputs into the desired output. The input represents the flow of data into the process from the outside. The process includes all the tasks required to transform the inputs effectively. The output is the desired outcome flowing out of the transformation process.

The input's goal is to build a core design that represents the initial steps required in the process. This approach exhibits the steps, adjustments, and other necessary conditions to achieve the system's desired output: Reverse Vending Machine for Plastic Bottles Recycling System. The methods and steps to achieve the researchers' goals are indicated in Figure 3.

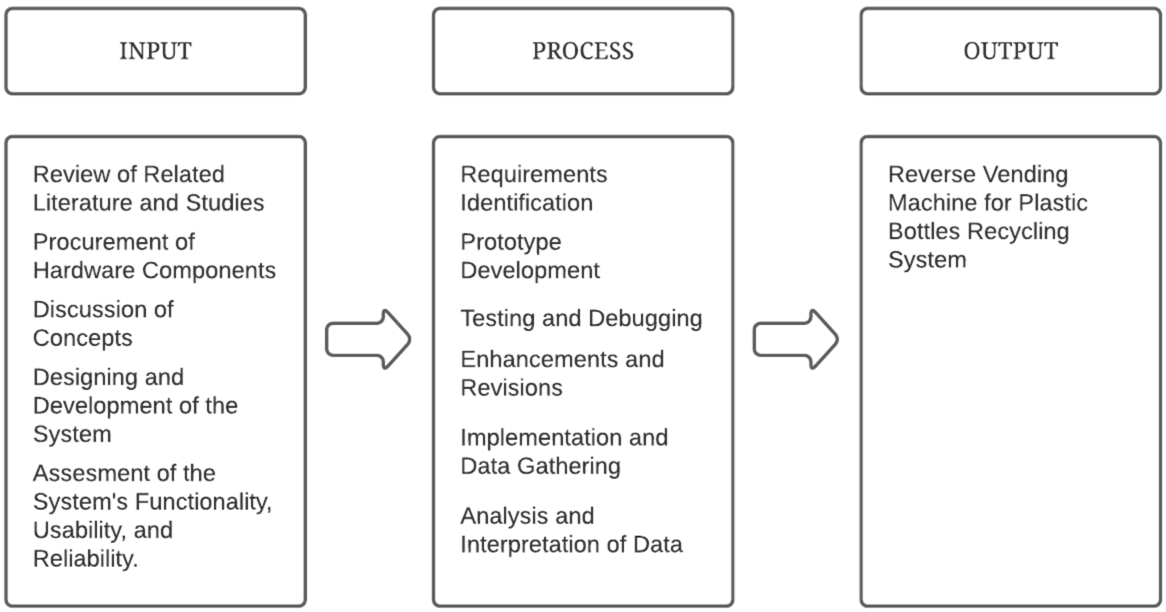

Figure 4: IPO Model

CHAPTER 4

METHODOLOGY

In this chapter, the researchers discuss the steps and methods made in creating the project and show the structure of how the system delivers the objective outcome. The researchers have researched and accumulated the required data regarding the objectives and used the necessary components and tools to provide solutions to the problems intended to be solved in this particular study. The researchers listed the equipment used in creating the machine as well as the description of why it is necessary to use such components.

The Reverse Vending Machine for Plastic Bottles Recycling is a standalone system that operates on a single Raspberry Pi computer. Python is the programming language that has been used in writing the program needed to run the AI object recognition system as well as in controlling the components used. The servo motors used in rotating parts of this system are powered externally separated from the power supply of the main computer to ensure an appropriate supply of current is being served to prevent malfunctions.

4.1 Reverse Vending Machine for Plastic Bottles Recycling System Flow Chart

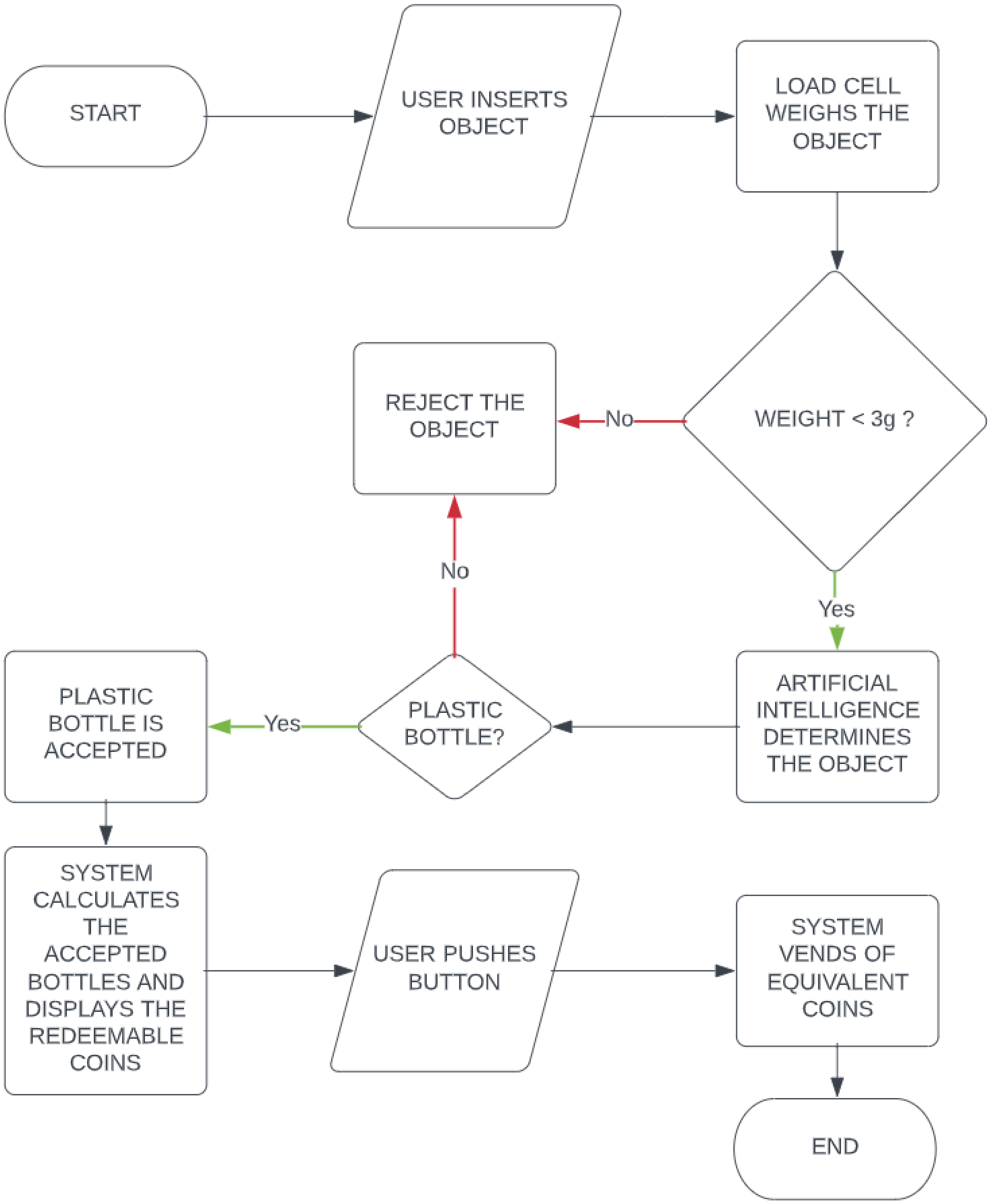

Figure 5: Reverse Vending Machine for Plastic Bottles Recycling System Flow Chart

The flow chart above shows the chronological order of the system's process with the user's interaction. The system starts when the load cell detects a force exerted on it which acts as a proximity sensor and then scans an object by determining its weight, if it is above 3 grams the system will reject the object else it will be further evaluated by the AI, if it is a plastic bottle then it will be accepted, else the object will be rejected. For each iteration, the system calculates the accepted bottles and displays the redeemable coins on the LCD. If the user pushes the button the system vends equivalent coins and then ends the transaction, then the system restarts at the beginning.

4.2 Technical Aspect Ratio

The Reverse Vending Machine for Plastic Bottles Recycling System is a project that uses materials and equipment that is fit for the objective and functionality.

4.2.1 Raspberry Pi 3 Model B

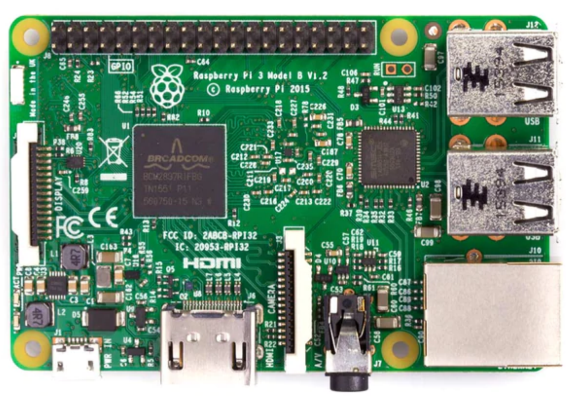

Figure 6: Raspberry Pi 3 Model B
(https://thepihut.com/products/raspberry-pi-3-model-b)

The Raspberry Pi is a single-board computer made by the Raspberry Pi Foundation and is commonly used to build hardware projects, home automation, IoT projects, and even industrial applications. It is a computer that operates in the open-source ecosystem and can run in various distributions of the Linux kernel. It provides a set of GPIO pins that allows control of electronic components.

The Raspberry Pi 3 Model B features a quad-core 1.2Ghz 64bit CPU, 1GB RAM, 40-pin extended GPIO, 4 USB 2 ports, micro SD port for loading the operating system and storing data, and a full-size HDMI port. This specification is enough to run most of the projects that use multiple components and provides low cost, fast processing, added connectivity, and a consistent board format.

4.2.2 Servo Motors

4.2.2.1 MG995

Figure 7: MG995 Servo
(https://www.smart-prototyping.com/Servo-Motor-M-Metal-Gear-MG995)

The MG995 is a popular servo motor for its performance and price. The motor is commonly used in robotics and drones. It has three terminals: signal pin (orange) which states the axis position that is given through the PWM signal, VCC (red) which is the positive power supply that operates between 4.8-7.2V, and ground (brown) which is connected to the ground of circuit or power supply.

This servo provides precise rotation over a 180-degree angle range, it is suited for designing projects in which the wear and tear are high. Having a metal gear, this servo has a long life span and provides a satisfying torque which is enough to overcome resistance.

4.2.2.2 SG90

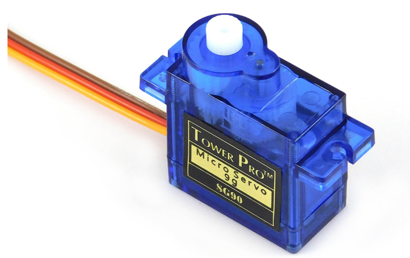

Figure 8: SG90 Servo
(https://www.smart-prototyping.com/Servo-Motor-SG90)

The SG90 is a tiny, lightweight micro servo motor with high output power. This servo can rotate at approximately 180 degrees at 90 degrees in each direction. It works like the standard kinds of servo but is smaller, and good for mini rotating mechanisms.

4.2.3 Load Cell

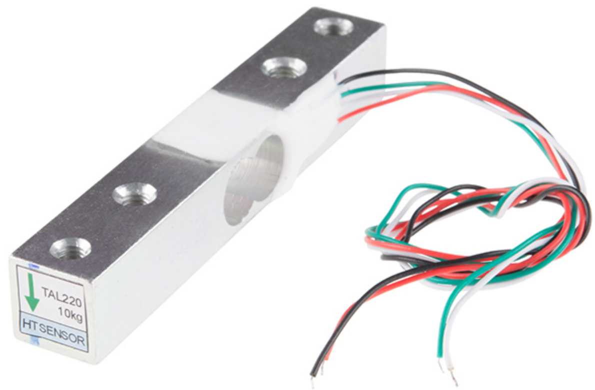

Figure 9: Load Cell
(https://learn.sparkfun.com/tutorials/getting-started-with-load-cells/all)

A load cell also called a “load transducer” is a sensor that translates load or force into an electrical signal. This sensor can be used as a force measurement, also it is used in various measuring instruments commonly in electronic balances, testing machines, and industrial scales.

4.2.4 HX711 Load Cell Amplifier

Figure 10: HX711 Load Cell Amplifier
(https://www.makerlab-electronics.com/product/load-cell-amplifier-hx711/)

The HX711 module allows the load cell measurements to be read by converting the measured changes in the resistance value of the load cell through the conversion circuit and transforming it into an electrical output. The measurements can be calibrated and be able to get accurate weight measurements. This module is very handy in creating an industrial scale and simple presence detection.

4.2.5 I2C LCD 1602 Module

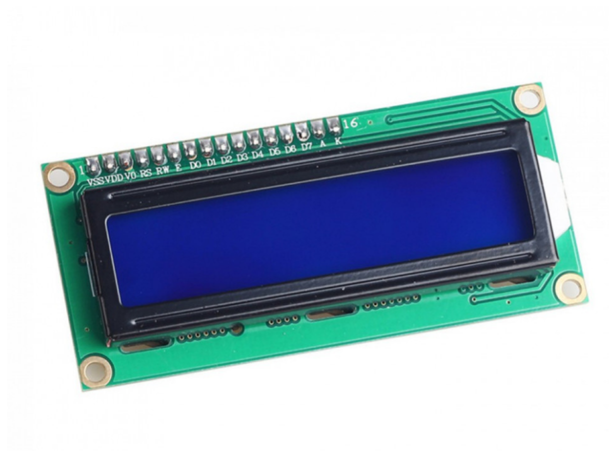

Figure 11: I2C LCD 1602 Module
(https://circuit.rocks/lcd-i2c-1602-display-module-blue-backlight.html)

An LCD module is a module for displaying text or characters. A 16x2 LCD features a LED backlight and can display 32 ASCII characters in two rows up to 16 characters on each row. The I2C LCD Module is an LCD that uses the 12C protocol to communicate with microcontrollers. A normal 16x2 LCD uses 9 pins in order for it to work while the I2C LCD uses only 4 connections: Power, Ground, SDA, and SCL.

CHAPTER 5

RESULTS AND DISCUSSION

With the data gathered from the existing studies and projects, the researchers developed a Reverse Vending Machine for Plastic Bottles Recycling System using a Raspberry Pi computer and the required components to solve the proposed problem. In this chapter, The researchers discuss the status of the developed system as well as the encountered flaws that hinder some of its functionalities.

5.1 The System

The researchers developed a system for a project that complies with the presented problem. The system consists of two main prerequisites, (1) the AI object recognition algorithm that distinguishes plastic bottles from non-plastic bottle objects and the (2) the sorting mechanism which is operated with servo motors.

The system uses a 3D printed model coin dispenser that can store up to a hundred 1-peso coins and uses a single SG90 servo which dispenses a single coin in each iteration.

5.2 The AI Object Recognition Algorithm

The AI object recognition algorithm distinguishes plastic bottles from non-plastic bottle objects at 40% to 60% accuracy. The researchers suspect that the 1.2GHz processing power and 1GB memory of the computer creates a bottleneck in the performance of the algorithm thus providing slow and inaccurate outputs during detection. Due to this limitation, it is better not to add any more input components such as proximity sensors to stabilize the performance of the software.

5.3 The Servo Motors

The 3 servo motors that are being used for the rotating parts of the system require at least 5 volts and 2 amperes of power supply in order to run simultaneously. Due to the limitations of the MG995 servo motors' strength, there has a high possibility that heavy objects inserted into the system might destroy the servos' gears.

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

Conclusion

With the existing principles and studies, the researchers developed a Reverse Vending Machine that accepts allowed plastic bottles and rejects non-plastic bottle objects. The software is not functioning according to the expected performance due to the limited processing power the Raspberry Pi model B has. However, the machine can still function as intended but is expected to have low accuracy and slow response. The researchers cannot add more input components such as coins and storage level indicators to the machine because it will only contribute to the low performance of the software. Overall, the researchers achieved the target aim of building a working reverse vending machine for plastic bottles recycling.

Recommendations

The study is open to further developments and innovations. The researchers then recommend:

- To upgrade its response time and performance, future researchers must use a Raspberry Pi 4 computer or better.
- To implement a feature that determines the level of bottles in the storage and the number of coins in the coin dispenser.
- To reinforce the machine, stronger materials can be used such as metal, and stronger servo motors are highly recommended.
- To update its software, a customized trained dataset and better algorithm for the AI is a must in order to improve the reliability of the system.

REFERENCES

Dioses, Jr, J. L. (2020). Bottle-SegreDuino: An Arduino Frequency-Based Bin for Tin Can and Plastic Bottle Segregation using an Inductive Proximity Effect. *International Journal of Advanced Trends in Computer Science and Engineering*, *9*(4), 5451–5454. https://doi.org/10.30534/ijatcse/2020/184942020

Mariya, D., Usman, J., Mathew, E. N., Ph, H., & Azeez, A. (2020). *Reverse Vending Machine for Plastic Bottle Recycling*. *8*(2), 7.

Liukkonen, J. (n.d.). Machine vision system for a reverse vending machine. 76.

NAPCOR. (2018). *Report on Postconsumer PET Container Recycling Activity in 2018*. https://napcor.com/wp-content/uploads/2021/07/Postconsumer-PET-Recycling-Activity-in-2018.pdf

Gaur, A., & Priyadarshini, R. (2018). *A Simple Approach to Design Reverse Vending Machine*.

Wong, K. K., Samah, N. A. A., & Sahimi, M. S. (2019). *Development of Reverse Vending Machine using Recycled Materials and Arduino Microcontroller*. 10.

GLOSSARY

Artificial Intelligence (AI) – is the simulation of human intelligence embedded in machines that are programmed to think like humans and automate actions.

Central Processing Unit (CPU) – is the electronic circuitry that executes instructions comprising a computer program.

Coin – a flat, typically round piece of metal with an official stamp, used as fiat money.

General-purpose Input/Output (GPIO) –an uncommitted digital signal pin on an integrated circuit or electronic circuit board which may be used as an input or output, or both, and is controllable by software

High-Definition Multimedia Interface (HDMI) – standard for simultaneously transmitting digital video and audio from a source.

Inter-Integrated Circuit (I2C)– is a bus interface connection protocol incorporated into devices for serial communication.

Light Emitting Diode (LED) – a semiconductor device that emits infrared or visible light when charged with an electric current.

Liquid Crystal Display (LCD) – is a flat-panel display that uses the light-modulating properties of liquid crystals combined with polarizers. Liquid crystals do not emit light directly.

Open CV – is an open-source library that can be used to perform tasks like face detection, objection tracking, and landmark detection.

PET – Polyethylene terephthalate, a type of clear, strong, lightweight, and recyclable plastic.

Python – an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991.

Random Access Memory (RAM) – is a volatile memory that temporarily stores data.

Raspberry PI – is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries.

Single Shot Multibox Detection (SSD) – is a deep learning model that has been used to detect objects in an image or a video source.

APPENDICES

APPENDIX A

PROGRAM CODES

```
import os
import argparse
import cv2
import numpy as np
import sys
import time
from threading import Thread
import importlib.util
import RPi.GPIO as GPIO
import pigpio
import drivers

LED_PIN = 18
LED_PIN2 = 12
GPIO.setmode(GPIO.BCM)
GPIO.setup(LED_PIN, GPIO.OUT)
GPIO.setup(LED_PIN2, GPIO.OUT)
BUTTON_PIN = 27
GPIO.setup(BUTTON_PIN, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)

display = drivers.Lcd()

class VideoStream:
    def __init__(self,resolution=(640,480),framerate=30):
```

```
        # Initialize the PiCamera and the camera image stream
        self.stream = cv2.VideoCapture(0)
        ret = self.stream.set(cv2.CAP_PROP_FOURCC, cv2.VideoWriter_fourcc(*'MJPG'))
        ret = self.stream.set(3,resolution[0])
        ret = self.stream.set(4,resolution[1])

        # Read first frame from the stream
        (self.grabbed, self.frame) = self.stream.read()

            # Variable to control when the camera is stopped
        self.stopped = False

    def start(self):
            # Start the thread that reads frames from the video stream
        Thread(target=self.update,args=()).start()
        return self

    def update(self):
        # Keep looping indefinitely until the thread is stopped
        while True:
            # If the camera is stopped, stop the thread
            if self.stopped:
                # Close camera resources
                self.stream.release()
                return

            # Otherwise, grab the next frame from the stream
```

```
        (self.grabbed, self.frame) = self.stream.read()

    def read(self):
        # Return the most recent frame
        return self.frame

    def stop(self):
        # Indicate that the camera and thread should be stopped
        self.stopped = True

EMULATE_HX711=False
referenceUnit = -7030.0

if not EMULATE_HX711:
    import RPi.GPIO as GPIO
    from hx711 import HX711
else:
    from emulated_hx711 import HX711

hx = HX711(4, 17)
hx.set_reading_format("MSB", "MSB")
hx.set_reference_unit(referenceUnit)
hx.reset()
hx.tare()
print("Tare done! Add weight now...")
entrance_servo = Servo(24)
servo = 24
```

```
#coin = 9
coin2=9
#entrance_servo.value = -0.8
#time.sleep(2)
pwm = pigpio.pi()
pwm.set_mode(servo, pigpio.OUTPUT)

coin = pigpio.pi()
coin.set_mode(coin2, pigpio.OUTPUT)

pwm.set_PWM_frequency( servo, 50 )
coin.set_PWM_frequency( coin2, 50 )

#GPIO.setmode(GPIO.BCM)
GPIO.setup(10, GPIO.OUT)
#GPIO.setup(9, GPIO.OUT)
#GPIO.setup(24, GPIO.OUT)
#GPIO.setup(10, GPIO.OUT)

sorter=GPIO.PWM(10, 50)
#entrance=GPIO.PWM(24, 50)
#coin=GPIO.PWM(9, 50)

sorter.start(0)
#entrance.start(0)
#coin.start(0)
```

```
# HX711
def weight_sensor():
    val = hx.get_weight(5)
    absolute=abs(round(val*10)/10)
    #print(absolute, val)
    hx.power_down()
    hx.power_up()
    return absolute

# ENTRANCE SERVO
def open_servo():
    entrance_servo.value = 0.5
    time.sleep(1)

def close_servo():
    entrance_servo.value = -1
    time.sleep(1)

# SORTER SERVO
def sorterr(angle):
    duty = angle / 18 + 2
    GPIO.output(10, True)
    sorter.ChangeDutyCycle(duty)
    time.sleep(1)
    GPIO.output(10, False)
    sorter.ChangeDutyCycle(0)
```

```
def coinn(angle):
    duty = angle / 18 + 2
    GPIO.output(9, True)
    coin.ChangeDutyCycle(duty)
    time.sleep(1)
    #GPIO.output(9, False)
    #coin.ChangeDutyCycle(0)

def entrancee(angle):
    pwm.set_servo_pulsewidth( servo, angle ) ;
    time.sleep(1)

def coins(angle):
    coin.set_servo_pulsewidth( coin2, angle ) ;
    time.sleep(1)

display.lcd_display_string("Welcome", 1)
entrancee(1100)
sorterr(75)
#coinn(80)
count = 5
coin_count = 5
time.sleep(1)
display.lcd_clear()

# Define and parse input arguments
parser = argparse.ArgumentParser()
```

```
parser.add_argument('--modeldir', help='Folder the .tflite file is located in',
                    required=True)
parser.add_argument('--graph', help='Name of the .tflite file, if different than detect.tflite',
                    default='detect.tflite')
parser.add_argument('--labels', help='Name of the labelmap file, if different than labelmap.txt',
                    default='labelmap.txt')
parser.add_argument('--threshold', help='Minimum confidence threshold for displaying detected
objects',
                    default=0.5)
parser.add_argument('--resolution', help='Desired webcam resolution in WxH. If the webcam
does not support the resolution entered, errors may occur.',
                    default='1280x720')
parser.add_argument('--edgetpu', help='Use Coral Edge TPU Accelerator to speed up detection',
                    action='store_true')

args = parser.parse_args()

MODEL_NAME = args.modeldir
GRAPH_NAME = args.graph
LABELMAP_NAME = args.labels
min_conf_threshold = float(args.threshold)
resW, resH = args.resolution.split('x')
imW, imH = int(resW), int(resH)
use_TPU = args.edgetpu

# Import TensorFlow libraries
```

```
# If tflite_runtime is installed, import interpreter from tflite_runtime, else import from regular
tensorflow
# If using Coral Edge TPU, import the load_delegate library
pkg = importlib.util.find_spec('tflite_runtime')
if pkg:
    from tflite_runtime.interpreter import Interpreter
    if use_TPU:
        from tflite_runtime.interpreter import load_delegate
else:
    from tensorflow.lite.python.interpreter import Interpreter
    if use_TPU:
        from tensorflow.lite.python.interpreter import load_delegate

# If using Edge TPU, assign filename for Edge TPU model
if use_TPU:
    # If user has specified the name of the .tflite file, use that name, otherwise use default
'edgetpu.tflite'
    if (GRAPH_NAME == 'detect.tflite'):
        GRAPH_NAME = 'edgetpu.tflite'

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

# Path to .tflite file, which contains the model that is used for object detection
PATH_TO_CKPT = os.path.join(CWD_PATH,MODEL_NAME,GRAPH_NAME)

# Path to label map file
```

```
PATH_TO_LABELS = os.path.join(CWD_PATH,MODEL_NAME,LABELMAP_NAME)

# Load the label map
with open(PATH_TO_LABELS, 'r') as f:
    labels = [line.strip() for line in f.readlines()]
if labels[0] == '???':
    del(labels[0])
if use_TPU:
    interpreter = Interpreter(model_path=PATH_TO_CKPT,
                              experimental_delegates=[load_delegate('libedgetpu.so.1.0')])
    print(PATH_TO_CKPT)
else:
    interpreter = Interpreter(model_path=PATH_TO_CKPT)

interpreter.allocate_tensors()

input_details = interpreter.get_input_details()
output_details = interpreter.get_output_details()
height = input_details[0]['shape'][1]
width = input_details[0]['shape'][2]
floating_model = (input_details[0]['dtype'] == np.float32)
input_mean = 127.5
input_std = 127.5
outname = output_details[0]['name']

if ('StatefulPartitionedCall' in outname): # This is a TF2 model
    boxes_idx, classes_idx, scores_idx = 1, 3, 0
```

```
else: # This is a TF1 model
    boxes_idx, classes_idx, scores_idx = 0, 1, 2

frame_rate_calc = 1
freq = cv2.getTickFrequency()
videostream = VideoStream(resolution=(imW,imH),framerate=30).start()

while True:
    display.lcd_display_string("Insert Bottle", 1)
    display.lcd_display_string("Bottle Count:"+str(count), 2)
    time.sleep(1)
    # Start timer (for calculating frame rate)
    t1 = cv2.getTickCount()

    # Grab frame from video stream
    frame1 = videostream.read()

    # Acquire frame and resize to expected shape [1xHxWx3]
    frame = frame1.copy()
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    frame_resized = cv2.resize(frame_rgb, (width, height))
    input_data = np.expand_dims(frame_resized, axis=0)

    # Normalize pixel values if using a floating model (i.e. if model is non-quantized)
    if floating_model:
        input_data = (np.float32(input_data) - input_mean) / input_std
```

```
    # Perform the actual detection by running the model with the image as input
    interpreter.set_tensor(input_details[0]['index'],input_data)
    interpreter.invoke()

    # Retrieve detection results
    boxes = interpreter.get_tensor(output_details[boxes_idx]['index'])[0] # Bounding box coordinates of detected objects
    classes = interpreter.get_tensor(output_details[classes_idx]['index'])[0] # Class index of detected objects
    scores = interpreter.get_tensor(output_details[scores_idx]['index'])[0] # Confidence of detected objects

    # Loop over all detections and draw detection box if confidence is above minimum threshold
    for i in range(len(scores)):
        if ((scores[i] > min_conf_threshold) and (scores[i] <= 1.0)):

            # Get bounding box coordinates and draw box
            # Interpreter can return coordinates that are outside of image dimensions, need to force them to be within image using max() and min()
            ymin = int(max(1,(boxes[i][0] * imH)))
            xmin = int(max(1,(boxes[i][1] * imW)))
            ymax = int(min(imH,(boxes[i][2] * imH)))
            xmax = int(min(imW,(boxes[i][3] * imW)))

            #cv2.rectangle(frame, (xmin,ymin), (xmax,ymax), (10, 255, 0), 2)

            # Draw label
```

object_name = labels[int(classes[i])] # Look up object name from "labels" array using class index

weight = weight_sensor()

print(weight)

if object_name == 'bottle' and weight>0.0 and weight<3.0:

GPIO.output(LED_PIN, GPIO.HIGH)

GPIO.output(LED_PIN2, GPIO.LOW)

entrancee(1900)

entrancee(1200)

#cv2.rectangle(frame, (xmin,ymin), (xmax,ymax), (10, 255, 0), 2)

label = '%s: %d%%' % (object_name, int(scores[i]*100)) # Example: 'person: 72%'

labelSize, baseLine = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.7, 2) # Get font size

label_ymin = max(ymin, labelSize[1] + 10) # Make sure not to draw label too close to top of window

#cv2.rectangle(frame, (xmin, label_ymin-labelSize[1]-10), (xmin+labelSize[0], label_ymin+baseLine-10), (255, 255, 255), cv2.FILLED) # Draw white box to put label text in

#cv2.putText(frame, label, (xmin, label_ymin-7), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 0), 2) # Draw label text

#display.lcd_clear()

#display.lcd_display_string("Bottle",1)

sorterr(20)

sorterr(75)

GPIO.output(LED_PIN, GPIO.LOW)

count+=1

```
        display.lcd_display_string(        str(count), 2)



      elif object_name != 'bottle' and weight>0.0 or weight>3.0:
        GPIO.output(LED_PIN2, GPIO.HIGH)
        GPIO.output(LED_PIN, GPIO.LOW)
        entrancee(1900)
        entrancee(1200)
        #cv2.rectangle(frame, (xmin,ymin), (xmax,ymax), (0, 0, 255), 2)
        label = '%s: %d%%' % (object_name, int(scores[i]*100)) # Example: 'person: 72%'
        labelSize, baseLine = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.7,
2) # Get font size
        label_ymin = max(ymin, labelSize[1] + 10) # Make sure not to draw label too close to
top of window
        #cv2.rectangle(frame, (xmin, label_ymin-labelSize[1]-10), (xmin+labelSize[0],
label_ymin+baseLine-10), (255, 255, 255), cv2.FILLED) # Draw white box to put label text in
        #cv2.putText(frame, label, (xmin, label_ymin-7), cv2.FONT_HERSHEY_SIMPLEX,
0.7, (0, 0, 0), 2) # Draw label text
        #display.lcd_clear()
        #display.lcd_display_string("NOT Bottle",1)
        sorterr(150)
        sorterr(75)
        GPIO.output(LED_PIN2, GPIO.LOW)

  # Draw framerate in corner of frame
```

```
    #cv2.putText(frame,'FPS:
{0:.2f}'.format(frame_rate_calc),(30,50),cv2.FONT_HERSHEY_SIMPLEX,1,(255,255,0),2,cv2.
LINE_AA)

    # All the results have been drawn on the frame, so it's time to display it.
    cv2.imshow('Plastic Bottle Detector', frame)

    # Calculate framerate
    t2 = cv2.getTickCount()
    time1 = (t2-t1)/freq
    frame_rate_calc= 1/time1

    # Press 'q' to quit
    if cv2.waitKey(1) == ord('q'):
        break
    elif GPIO.input(BUTTON_PIN) and count == 0:
        display.lcd_clear()
        display.lcd_display_string("Insert Bottle", 1)
        display.lcd_display_string("First", 2)
        time.sleep(2)
        display.lcd_clear()

    elif GPIO.input(BUTTON_PIN) and count!=0:
       # print(count)
       display.lcd_clear()
       display.lcd_display_string("Thank You", 1)
       display.lcd_display_string("For Recycling", 2)
```

```
    for i in range(0,count):
        print(i)
        coins(500)
        coins(1500)
    count = 0
    time.sleep(2)
    display.lcd_clear()

cv2.destroyAllWindows()
videostream.stop()
```

APPENDIX B

PROJECT COST

No.	Quantity	Item	Unit Price (Php)	Total (Php)
1.	1pc	Rasperry Pi 3 Model B Set	4,000.00	4,000.00
2.	2pc	MG995 Servo Motor	270.00	540.00
3.	1pc	SG90 Servo Motor	80.00	80.00
4.	1pc	3D Printed Coin Dispenser	200.00	200.00
5.	1pc	Load Cell	130.00	130.00
6.	1pc	HX711 Amplifier	50.00	50.00
7.	1pc	I2C LCD 1602 Module	120.00	120.00
8.	3pck	Screws	30.00	90.00
9.	2kls	Nails	60.00	60.00
10.	20pcs	L bracket	12.00	260.00
11.	1bndl.	Jumper Wires	75.00	75.00
12.	12pcs	8x2x1 Wood	130.00	1560.00
13.	4pcs	8x2x0.5 Wood	75.00	300.00
14.	1pc	8x4x2 Plywood	350.00	350.00
			Grand Total	7815.00

CURRICULUM VITAE

Name: Joshua Igot Barlolong

Address: Goleo, Sindangan Zamboanga del Norte

Phone Number: 0927-341-8762

E-mail: joshuabarlolong@gmail.com

EDUCATIONAL BACKGROUND

20016 - 2012: Sindangan Pilot Demonstration School

2012 - 2018: Sindangan National High School

2018 - 2022: Negros Oriental State University

EXTRA-CURRICULAR ACTIVITIES

2018 - 2022: Percussionist - NORSU KABILIN Marching Band

2018 - 2022: Member - Philippine Institute of Computer Engineering Students(PhICES)

2020 - 2022: Chairman - Philippine Institute of Computer Engineering Students(PhICES) - Youth Esports Program

2021 - 2022: Second Representative - Philippine Institute of Computer Engineering Students(PhICES) - LYDO

2021 - 2022: Graphics Unit Head - The Touch Publication

CURRICULUM VITAE

Name: JC Del Vincent Retada Escala

Address: Purok 3 Jantianon, Amlan, Negros Oriental

Phone Number: 0955-290-3236

E-mail: jcescala700@gmail.com

EDUCATIONAL BACKGROUND

20016 - 2012: Jantianon Elementary School

2012 - 2016: Negros Oriental High School

2016-2018: St. Paul University Dumaguete

2018 - 2022: Negros Oriental State University

EXTRA-CURRICULAR ACTIVITIES

2018 - 2022: Member - Philippine Institute of Computer Engineering Students(PhICES)