iCan Bin
Final Report

EE297 Intelligent Systems Project

Dr. Erivelton Nepomuceno

10/02/2023

Name:	Student Number:
Markel Biain	21723801
Niall Clarke	21428996
Leah Duggan	21335851

Table of Contents

1. Introduction

The iCan Bin is a device created to improve waste collection and waste management. In most cities and towns, numerous waste collection companies have a set routine for each week where they will drive around to every house and business and collect the waste that has accumulated. However, what if some places are less populated than others and thus accumulate less waste within a week? This project uses a sensor to detect how much waste is in a bin. If the sensor senses that the bin is almost full a signal will be sent to the appropriate waste collection company which will include the location of said bin. This type of high-tech bin will also include a camera and a motor. The camera will be connected to a computer which will have all the code. This computer will use the custom neural network model the team produced to detect the type of waste that is given, this will send a signal to the Arduino specifying what it is. Then the motor would turn, tipping the shelf if the waste detected is a can. This will create a more optimal waste collection system, allowing faster routes to be planned for waste collection, thus less pollution from waste lorries. It will also avoid waste being thrown into the incorrect bin, contaminating that bin, and creating more work at the processing end which makes the waste service more expensive.

2. Objectives

2.1 Objectives Overview

This project's objective is to focus on the development of a device to sort waste automatically. This device will improve the waste collection sector and reduce the amount of contaminated waste that is currently being produced. This project relates to four of the seventeen Sustainable Development goals outlined in the 2030 Agenda for Sustainable Development.

2.2 Sustainable Development Goals relating to our project

The first goal it relates to is "Build Resilient Infrastructure, Promote Inclusive and Sustainable Industrialization and Foster Innovation". With this goal, the UN will support economic development and human well-being by developing quality, reliable, sustainable, and resilient infrastructure, with affordable and fair access for all. Additionally, it outlines its plans to upgrade infrastructure and retrofit industries to make them more sustainable, increasing resource efficiency and adoption of environmentally friendly technologies and industrial processes in accordance with each country's capabilities [1]. This relates to the project as the improvement of the waste collection sector will reduce the need for waste lorries to frequently travel around towns and cities, creating unnecessary air pollution and reducing the amount of money spent for fuel and workers to drive the lorries. By doing this, we can develop other areas of the waste sector such as developing more waste specific bins and creating cleaner air for all. This will help assist economic development and human well-being.

The second goal it relates to is "Make Cities and Human Settlements Inclusive, Safe, Resilient and Sustainable". This goal states that the UN will strengthen efforts to protect and safeguard the world's cultural and natural heritage. Additionally, it is designed to reduce the adverse per capita environmental impact of cities by focusing on air quality and waste management [1]. Like the first goal, this relates to the project as the improvement of the waste collection sector will reduce environmental impact on cities and towns. The automatic sorting of waste bins will improve waste management by reducing contamination within bins and speeding up waste processing.

The third goal it relates to is “Take Urgent Action to Combat Climate Change and its Impacts”. A major part of this goal is to build the capacity of least developed countries and small island developing states to better manage climate change, focusing on women and youth [1]. Improvements in the waste collection sector will promote effective climate change planning, not only in less developed countries, but everywhere. Knowing which bins need to be collected and where they are will help the waste collection sector to plan more efficient routes. Reducing the amount of fossil fuels being released into the air due to the bin lorries and reducing the amount of money being spent to pay for workers and pay for fuel. This means that more money can go towards improvements within the waste collection sector, further developing how to reduce its carbon footprint.

The fourth goal it relates to is “Strengthen the means of implementation and revitalize the global partnership for sustainable development”. This goal states that the UN will fully operationalize the technology bank and science and innovation capacity-building mechanism for least developed countries by 2017. It will also enhance enabling technologies [1]. This project could help thousands of different communities across the globe to tackle their waste problems. Developing this project further and bringing this type of technology into the market for developed countries will help the current climate problems in the world, but there is no point in restricting technology to one part of the world. If this project is going to help developed countries, then it is only fair that it is brought to developing countries too. Hopefully this type of technology will not only help developing countries with their waste problems but also increase their knowledge on science and engineering. The project will need to be maintained so people in these countries will have to be taught how to do so. This will not only educate people but also create more jobs in the area.

Although this project fits many of the goals within the Sustainable Development goals outlined in the 2030 Agenda for Sustainable Development, these four goals stood out the most in the inspirational process for this project.

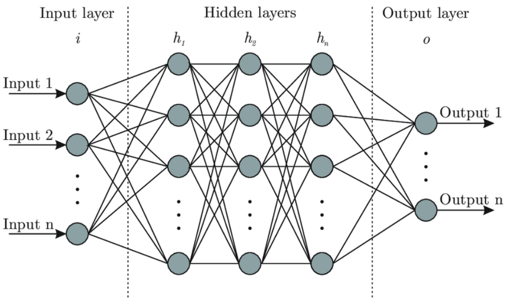
2.3 How a Neural Network Works

To give a brief history of the neural network, Geoffery Hinton conceptualized the idea of deep learning in the 1980’s. His main contribution towards machine learning was to try and adapt human thinking techniques to machine learning [2]. This led to the birth of the Neural Network, which is a deep learning technique structured to mimic how neurons in the human brain work. The idea came to Hinton as he believed the human brain to be the most powerful computational engine of all time.

The structure of Hinton's Artificial Neural Network consists of the main parts being nodes and node layers [2]. Each node is designed to work like a neuron in a brain. Each node is to perform a mathematical function, with the information being passed onto the next node [3].

This idea of having many functions that are linked and send messages to another is almost precisely how neurons in our brain work. However, if the nodes were all connected in the way our human neurons interact with each other, the processing time for Neural Networks would be quite high. Thanks to node layers however this problem is taken care of.

A Neural Network will consist of three main node layers. Input layer, hidden layers, and Output layer. A Neural Network will only have one input layer and one output layer however, it may have multiple hidden layers. Each of these layers will carry out different types of functions although all of them are connected [3]. In the figure below a diagram is shown to help conceptualize this structure. The input layer will store the given data in its nodes and send this data into the hidden layer's nodes. The hidden layers nodes will then run a series of calculations with this data. The results from these calculations will then be passed from hidden layer to hidden layer. This process will go on until all data has been passed to all hidden layers. The final hidden layer will then send the data or signals to the output layer [2].



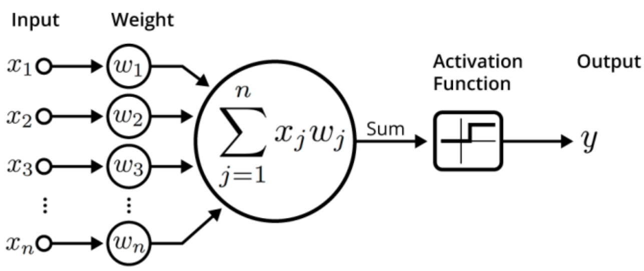


Figure 1.1, Neural Network Conceptual Structure

Figure 1.2, Activation function

This being a very simple explanation behind what happens in a neural network, without any explanation on how or why the neurons/nodes carry out mathematical functions and how they are used before output. The strings that are seen connecting the neurons on the **Figure 1.4** are called synapses. Each synapse has an associated 'weight' it carries. This impacts the overall importance of each neuron. Weights are the basis on how a model learns or is trained [2]. An example of the importance of weights can be seen below. Once a neuron receives a signal from a previous layer it sums each signal multiplied by its associated weight [2]. Once this calculation has been completed the signal is sent to an activation function.

To be brief, an activation function is the primary basis of how neurons communicate with each other. The activation function will then calculate the output value for the neuron. Once this is done the output value/signal is sent onto the next neuron and layer in the Neural Network. Some examples of different types of activation functions are threshold functions, ReLUs, hyperbolic tangent functions and sigmoid functions [2]. To choose an activation function it must be considered what the task at hand is and what data is being used.

An example of the importance of weights can be seen below. Once a neuron receives a signal from a previous layer it sums each signal multiplied by its associated weight. Once this calculation has been completed the signal is sent to an activation function.

To then train a neural network a cost function is used. A cost function is an equation which is used to measure the error produced in a network's prediction. There are many different equations that can be used, having advantages and disadvantages to each other. One example of a cost function is MSE or mean squared error. It assesses the average squared difference between the observed and predicted values. The aim of MSE is to have the result as close to zero as possible as it would mean the predicted value is the same as the actual value [3]. Shown is the formula for MSE.

$$MSE = \frac{\Sigma(-y_i - \hat{y}i)^2}{n} \quad (1)$$

- y_i is equal to the observed value.
- $\hat{y}i$ is equal to the predicted value.
- n is equal to the number of observations.

In practice it is important to note that the MSE will never be equal to zero but ideally a model will be as close as possible.

This is a simple explanation of how a neural network thinks and is trained.

2.4 Uses of a Neural Network

Since this research involves object detection, image classification, and even natural language processing, a DCNN (Deep Convolutional Neural Networks) is employed [4]. This is crucial for the task at hand—waste management—because the Neural Network will have to recognize an object, identify its properties, and provide the appropriate output or response. There will be a variety of approaches taken, though, depending on the project. the following classifications, prediction, clustering, association, and anomaly detection techniques will be reviewed.

For classification tasks like figuring out what type of garbage an object falls under. The ideal method for dealing with this is to use a DCNN and feed photos of various types of garbage into the neural network. The next step is to send these photos through a variety of convolutional layers, each of which extracts increasingly intricate and abstract characteristics about the objects. A fully connected layer would then come after this to classify the different types of garbage.

A neural network may use unsupervised learning techniques, which simply mean that a neural network is trained with data without being provided specific labels or targets, if it were essential to group similar sorts of trash together, such as in a clustering task. Instead, the algorithm will group the given data based on similarities or patterns it finds. If used properly, it can offer an alternative viewpoint on how we approach trash. K-means clustering and self-organizing maps are two techniques that could be used to carry out this type of learning [6].

Similarly, association tasks perform in a like manner to clustering tasks. Association tasks for example could tell what a person would buy next according to what they have previously bought. To correlate this to waste management you could predict if a high amount of waste were to be produced according to the events in a local area. A simple supervised method to perform association tasks could be to use decision tree learning. This is a simple tree like model that follows a set of rules according to the input. For example, if the task was to predict the levels of waste after a concert, you could ask the AI a set path of questions like "was there more than 500 people there" and depending on the answer an estimate could be made on the amount of waste produced [7].

Finally, for anomaly detection tasks, a neural network could be used to solve the problem of illegal dumping. If abnormal amounts of waste are being produced in a particular area, the neural network could compare previous data to the current data to catch abnormal patterns. One method of solving anomaly tasks is to use auto-encoders. An auto-encoder works in brief by taking an input and slowly comparing back to the expected/original input, using lower-dimensional representations of data to do so [8].

To conclude, many waste problems can be solved by using neural networks. It is everyone's duty to correctly manage their waste as it not only affects the livelihoods of all humans but also of all living organism on earth if not managed correctly. However, with the use of neural networks, the efficiency of how waste is managed can be drastically improved providing a brighter and more eco-friendly future for the planet.

3. Background

3.1 Brief History

Technology for disposing of waste has advanced throughout time. Nowadays, automated bins are being developed, however in the Middle Ages (1000–1800), technology was not as advanced. The streets of the cities were littered with foul-smelling rubbish and other general waste, which was typically made up of human and animal excrement. Throughout the Middle Ages, there were numerous initiatives to clean up the streets. Both for their personal convenience, since the waste was making travel impossible, and due to the foul smell, which people found intolerable. This far back in time technology was not very modern so "rakers" were employed to clean the streets [17]. However, this did not last as the rich refused to pay to clean up the streets for the poor and the poor were too worried about where their next meal would come from to worry about the state of the streets.

As a result of the industrial revolution, coal was increasingly used for cooking and heating in the 1800s. As a result, residential waste now contained a significant amount of ash. Businesses and residences started allowing rubbish collection services in their locality. A few laborers were hired to sort through the rubbish after it was transported to a dust-yard. They separated soil, coal, breeze, and many other items that could be sold. Part of this debris was utilized in the production of bricks and as farm fertilizer. This collection peaked in the 1830s, but it began to drop when individuals began to pay for rubbish removal in the 1840s. However, the dust-yards remained in operation.

The trade in dust declined between 1850 and 1900. The initial understanding of the effects that improved cleanliness can have on wellbeing was brought about by a Sanitation Commission established in 1839. The 1875 Public Health Act resulted from this [17]. This act established the first waste collectors who oversaw collecting various moving containers, known today as the waste bins that are left outside. These waste bins were collected at least once a week and families and businesses were accountable for keeping their waste in a waste bin. This is the first time in recorded history that waste collectors have been employed. Due to the emergence of environmental protection, technological advancement persisted, and waste was later divided into recycling and general waste.

3.2 Current Technology

Due to a significant rise in industrialization, excessive consumption, and a lack of comprehensive and efficient waste management, waste has become a significant problem throughout the world. It is a common thought that poor waste management is, not only unpleasant to look at, but also puts surrounding people at risk. This is why there have been many developments within Intelligent Systems used to tackle waste management the most used machine-based approach to improve waste management is a prototype to estimate the amount of food waste inside a dining hall using a CNN-based image recognition model. It is believed that on-time waste collection response is important. To achieve this, like the goal for this project, a real-time alert system is used that triggers a notification to the waste collection system as soon as the container becomes full. Many approaches have been used to deal with this problem including using embedded systems combined with ultrasonic sensors or infrared sensors. In recent years experiments were performed on known DCNNs (Deep Convolutional Neural Networks). The experiments concluded that high precision can be achieved in the classification of the state of waste container using DCNNs [18]. This is why in this project the Intel Movidius Neural Network will be used.

4. Hardware and Software

4.1 Raspberry Pi 3

Developed in the U.K and released in 2012 the Raspberry Pi is mainly used to help teach the basics of programming and computer science. To a layman, the Raspberry Pi may appear just as a normal micro-controller like an Arduino. However, the Raspberry Pi is essentially a low-cost minicomputer capable of running a variety of operating systems such as Linux Ubuntu and Raspbian. It has its own CPU, RAM and its own input and output ports to be used for connecting a range of devices from motors to keyboards [9]. Although the Raspberry Pi 3 was not used, the team felt it was important to demonstrate their understanding of what would have happened if the Raspberry Pi worked for this project. It is shown in **Figure 1.3**.

Figure 2: Raspberry Pi 3

Found on Raspberrypi.com

The Raspberry Pi can be used for several purposes such as computing the neural network, controlling things such as motors or sensors and taking in the input from the webcam. The first step to achieving all these things is installing an appropriate operating system on the Raspberry Pi. As mentions previously the two main operating systems in question are the official operating system for the Raspberry Pi, Raspbian and the Linux Ubuntu [10]. Both operating Systems have their benefits and will be discussed later. Once an operating system is installed from the internet, one can then begin to use the Raspberry Pi.

The specs for the Raspberry Pi are shown in **Table 1** [11].

Raspberry Pi Specifications Sheet
Quad Core 1.2GHz Broadcom BCM2837
1GB RAM
BCM43438 Wireless LAN and Bluetooth Low Energy (BLE) on Board
100 Base Ethernet
40-Pin Extended GPIO
4 USB 2 Ports
4 Pole Stereo Output and Composite Video Port
Full Size HDMI
CSSI Camera Port for Connecting a Raspberry Pi Camera
DSI Display Port for Connecting a Raspberry Pi Touchscreen Display
Micro SD Port for Loading an Operating Systm and Storing Data
Upgraded Switched Micro USB Power Source up to 2.5A

Table 1, Raspberry Pi Specifications Sheet

4.2 Python

Python is a programming language developed in the 1980's by Guido van Rossum and since has become one of the most popular programming languages. This is due to it having use in web development, machine learning, data analysis and artificial intelligence [12].

What makes python so popular is that it designed to be an easy-to-use language, having many abilities such as object-oriented programming to name one. It has large number of libraries for users to choose from, increasing the languages usefulness. It also can run on all three main operating systems such as Linux, Windows and macOS [13].

One disadvantage of using this language is that, in Maynooth University, Python programming has not been formally taught to students but, this is believed to be a small problem due to its simplistic design it will be manageable to learn.

Its huge library selection catered towards object detection. Some of these potentially useful libraries are TensorFlow and PyTorch. Python is also operational on a Linux OS meaning it can be used on the provided Raspberry Pi. Due to these advantages, it has been the chosen language for programming the intel neural network [14].

4.3 TensorFlow

The Google Brain team created the open-source software package TensorFlow for creating and refining machine learning models. It offers a versatile and effective platform for creating and implementing deep learning algorithms, such as neural networks, convolutional neural networks, and recurrent neural networks. One of the most well-known machine learning libraries in the world is TensorFlow because it enables developers to create sophisticated machine learning models and execute them across a variety of platforms, including desktop, server, and mobile devices. Users can create and train models using TensorFlow for several applications, such as speech recognition, image classification, and natural language processing. TensorFlow also supports distributed computation, enabling the use of several GPUs or even computer clusters to train models on massive datasets.

TensorFlow is an end-to-end platform for machine learning. It supports the following [21]:

- Multidimensional array based numeric computation
- GPU and distributed processing
- Automatic Differentiation
- Model construction, training and export
- And more

The following figure shows the hierarchy of TensorFlow toolkits:

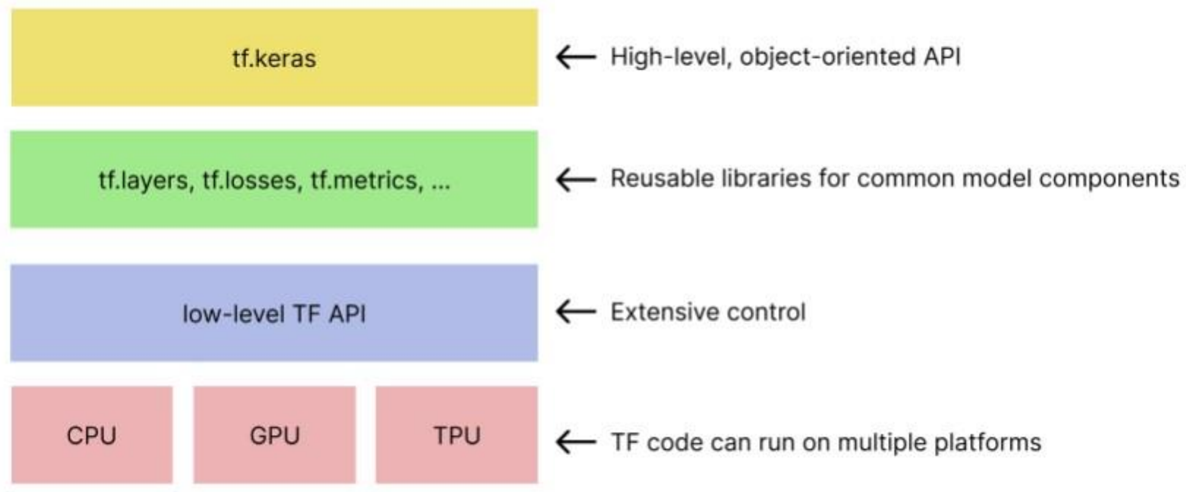


Figure 3.1: Hierarchy of TensorFlow toolkit [20]

TensorFlow was selected for the project because it takes care of a lot of the complexity involved in creating a neural network, allowing the user to concentrate on testing and refining the model. On its own GitHub page, TensorFlow offers several open-source setup and operation options.

The excellent visualization tools provided by TensorFlow (TensorBoard), simple debugging (good error messages, error pointers, running subparts of graphs for problem detection, etc.), scalability (depending on the configuration selected, the final model can be implemented to everything from a high-end computer to a smartphone), and pipelining (parallel design and GPU support) are some additional benefits.

TensorFlow has drawbacks like everything else, some of which include environment that is mostly focused on Linux (Windows compatibility can be patchy at times, but there are workarounds, such as using the anaconda environment), slow training, resource-intensive, and limited to NVidia GPUs for GPU training.

TensorFlow is an effective and favoured open-source library for creating and improving machine learning models. It offers a flexible and effective platform for creating deep learning algorithms, such as neural networks, CNNs, and RNNs (Recurrent Neural Network), and supports a variety of applications, including speech recognition, image classification, and natural language processing. Models can be trained utilizing several GPUs or computer clusters on huge datasets because to TensorFlow's support for distributed computing. In general, TensorFlow is a crucial tool for data scientists and machine learning engineers that want to create and implement cutting-edge machine learning models.

4.4 Arduino

To give a very brief history, Arduino Board itself was developed by the creators of the Arduino Platform, Massimo Banzi and David Mellis, developed, and released in 2003.

The Arduino board rose in popularity due to its cost effectiveness and easy to understand style. Due to this it has become an extremely popular device between individuals beginning programming and established professionals.

For this project, the Arduino Uno will be used to control the flow of information between the Neural Network models and the actual bin itself due to the reasons listed above.

Arduino C is the programming language associated with the Arduino board. It is a simplified version of the established C++ language. The language allows users to interact at an extreme hardware level with the Arduino board while maintaining an easy-to-follow syntax, due to this it has become popular with beginners and professionals.

Although the syntax of Arduino C is based off C++ there are still some key differences. Arduino C requires a setup function and loop function. The setup function initializes the board, and the loop function runs the code for the board. Below is a sample of Arduino code to demonstrate how it works. It is important to note that this code is not related to our project and is only included to further explain the Arduino language.

```
// Define constants
const int buttonPin = 2;  // Button pin
const int motorPin = 3;   // Motor pin

// Define variables
int buttonState = 0;      // Variable for reading button state

void setup() {
  // Set up the button and motor pins
  pinMode(buttonPin, INPUT);
  pinMode(motorPin, OUTPUT);

  // Set up serial communication
  Serial.begin(9600);
}

void loop() {
  // Read the button state
  buttonState = digitalRead(buttonPin);

  // If the button is pressed, turn the motor and print to serial
  if (buttonState == HIGH) {
    digitalWrite(motorPin, HIGH);
    Serial.println("Motor turned");
  } else {
    digitalWrite(motorPin, LOW);
  }
}
```

Figure 3.2: Arduino Demonstration.

This program turns a motor every time a button is pressed, printing a serial message “Motor turned”. As you can see the language itself is similar to C++ however, functions such as digitalWrite() and Serial.println() show that the language is a separate entity.

For this project, Arduino is extremely important to the physical functionality of the project. A code has been written for an Arduino Uno (a version of the Arduino) that interacts with Jupyter and the python code.

The program itself is simple and works as follows.

```
#include <Servo.h>

Servo myservo;

void setup(){
  Serial.begin(9600);
  myservo.attach(3);
  pinMode(2, INPUT);
}

void loop(){
  while(digitalRead(2) != HIGH){}

  if(Serial.available() > 0){


    String input = Serial.readStringUnitl('\n');
    if(input == "Can"){
      myservo.write(0);
      Serial.println("Can");

    }
    else if (input == "Bottle") {

      myservo.write(100);
      Serial.println("Bottle");
    }
    delay(2000);
    Serial.println(input);
  }
}
```

Figure 3.3: Arduino Code

What the above code does is create a connection between the Arduino and output of the machine learning model. The Arduino is given a string, either “Can” or “Bottle. Depending on what string the Arduino is given the Arduino will print the received string into a serial monitor and will then move the servo motor in the given manner. If a can is detected the servo will move to a 0-degree position, whereas if a bottle was detected the servo will move to a 100-degree position.

5. Implementation

5.1 Implementation Overview

The implementation of this convolutional neural network for it to be able to detect cans required many different steps. The first was gathering images. The process used to do this is called web scraping. This technique can be helpful for acquiring a lot of data, but it needs a lot of pre-processing to make sure the information is accurate and appropriately tagged. The next step was preparing the environment. In this case, Jupyter lab was the environment used. Jupyter lab allowed us to import everything that was necessary including TensorFlow and matplotlib.

5.2 Gathering Images

To gather images for this project, a process called web scraping was used. Using software or scripts, web scraping is the process of automatically obtaining data from websites. Typically, the data is taken from HTML pages, which can be analysed and parsed to take out specific data. Gathering images can take quite a lot of time. So, to speed up the process, google chrome have an extension which can be downloaded called "Download All Images". This means all that must be done is search up "Aluminium cans" on google, run the extension and all images on that pages will be downloaded into a zip folder. Problems may occur here as dodgy images may be included into the zip folder. These could be corrupted pictures, pictures that aren't of aluminium cans or have the incorrect extension applied. However, two dependencies can be imported into the code to allow for dodgy images to be removed. These are "cv2" and "imghdr". "imghdr" checks that the correct extension has been applied to the images gathered. The extensions desired for this project were of the type's "jpeg", "jpg", "bmp" and "png". Images under 10KB were also manually removed as these images were very small and so not beneficial to the project.

5.3 Loading Data

Pip must first be installed so importations such as NumPy may occur. NumPy is a Python library for doing fast linear algebra [22]. Keras has a data pipeline built into it. The code line "*data = tf.keras.utils.image_dataset_from_directory('data')*" is how it is accessed. This builds an image dataset directory automatically. This means that for this project, time was saved as the labels and classes were built by this directory. It also does much of the pre-processing. This can be seen by using the code "*tf.keras.utils.image_dataset_from_directory*". This code shows that Keras labels the images, creates a label mode for the images, creates class names, detects colours, creates a batch size, creates image size, shuffles the data, creates a validation split, creates subsets, creates an interpolation, and crops the images to an aspect ratio. Resizing of the images is an important step in preparing image data for neural networks as the input layer of the neural network is fixed in size. This means if the input images are not the same size, they can't be processed by the network. A validation split is also an important step within neural networks as it is a technique used to evaluate the performance of a model during the training process. In general, Keras saves the team a lot of time compared to if it was manually done.

```
In [17]: tf.data.Dataset??

In [18]: import numpy as np
         from matplotlib import pyplot as plt

In [33]: data = tf.keras.utils.image_dataset_from_directory('data')
         Found 219 files belonging to 2 classes.
```

Figure 4.1: Code to import helpful functions and pipelines

This isn't a dataset which is pre-loaded into memory it is actually a generator. This means if the team desired to look at the first instance of this data an error will occur. This is easily fixed by converting the data into a NumPy iterator using the code "*data_iterator = data.as_numpy_iterator()*". This allows access to the generator from the data pipeline by looping through it. Now, consecutive batches can be obtained by using the "*.next()*" function. For this project, detected cans is the main priority. However, for this code to run smoothly it is easier to mark it up against another set of data. This other set of data was chosen to be bottles. This means that the code will give back either a 1, if it's a can, and a 0, if it's a bottle. This can be seen in Figure _. To check if this is running correctly, "*data_iterator = data.as_numpy_iterator()*" can be run again to create a new batch of data.

```
In [34]: data_iterator = data.as_numpy_iterator()

In [35]: batch = data_iterator.next()

In [36]: fig, ax = plt.subplots(ncols=4, figsize=(20,20))
         for idx, img in enumerate(batch[0][:4]):
             ax[idx].imshow(img.astype(int))
             ax[idx].title.set_text(batch[1][idx])
```

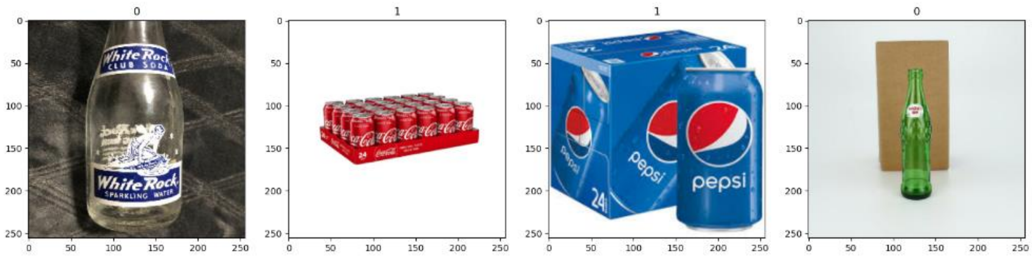


Figure 4.2: Code to create a new batch of data

5.4 Pre-processing Data

Pre-processing data scales the image values to between 0 and 1 rather than 0 and 255. This helps the deep learning model to generalize faster and produces better results. Within the pre-processing, the data is also split into training, testing and validation partitions to ensure that data isn't overfit into the neural network. Inside of the data pipeline there is a function called "map()". This function allows us to apply a particular type of transformation as the data is being pre-processed through the data pipeline. This means when the data is pre-fetched it's going to do that transformation as well. So, this speeds up how the data can be accessed from the disk.

```
In [39]: batch[0].shape
Out[39]: (32, 256, 256, 3)

In [40]: data = data.map(lambda x,y: (x/255, y))

In [43]: scaled_iterator = data.as_numpy_iterator()

In [44]: batch = scaled_iterator.next()

In [45]: data.as_numpy_iterator().next()
Out[45]: (array([[[[0.5132889 , 0.5132889 , 0.4662301 ],
                   [0.50021446, 0.50021446, 0.45315564],
                   [0.4819087 , 0.4819087 , 0.4348499 ],
                   ...,
                   [0.42369792, 0.44941023, 0.38504902],
                   [0.8835631 , 0.9070925 , 0.86613053],
                   [0.9861979 , 0.9983992 , 0.9705116 ]],

                  [[0.49982384, 0.49982384, 0.45276502],
                   [0.48649955, 0.48649955, 0.43944073],
                   [0.46536666, 0.46536666, 0.4183078 ],
                   ...,
                   [0.57776165, 0.6030576 , 0.5399456 ],
                   [0.9008065 , 0.9243359 , 0.8833739 ],
                   [0.84997797, 0.868662  , 0.8342917 ]],

                  [[0.4852558 , 0.4852558 , 0.438197  ],
                   [0.47161943, 0.47161943, 0.4245606 ],
                   [0.44758174, 0.44758174, 0.40052292],
```

Figure 4.3: Code to scale images between 0 and 1

Now the data can be split into a training and testing partition. This means that when the data is validated, it ensures that the model hasn't overfit the neural network. In order to do this, the training data sizes need to firstly be established. In this case, the training size will be 70% of the data size, validation size will be 20% of the data size and test size will be 10% of the data size. The training data is used to train the deep learning model. The validation data is used to evaluate the model while it is training. It's good to have a training and validation partition during the building of neural networks because even though the model will not have seen the validation partition before it starts the training, it can still be used to fine tune the deep learning model. The test partition is used post-training to do the evaluation. However, at this stage of the code all that is being done is allocating the data to each of the partitions.

```
In [54]: train_size = int(len(data)*.7)
         val_size = int(len(data)*.2)+1
         test_size = int(len(data)*.1)+1
```

Figure 4.4: Code to allocating data to each of the partitions

The last step in pre-processing the data includes using the "*take()*" and "*skip()*" functions available inside the TensorFlow dataset pipeline. "*take()*" defines how much data to take within that particular partition. For the validation data, the "*skip()*" function skips the batches within the training partition and then it takes the data within the validation partition. Then for the test partition it takes everything that is left over. So, it skips the training data and validation data and takes the test data. This helps to establish the train, validation, and test partitions.

```
In [56]: train = data.take(train_size)
         val = data.skip(train_size).take(val_size)
         test = data.skip(train_size+val_size).take(test_size)
```

Figure 4.5: Code to establish the training, validation and testing partitions

6. LIT Review

The objective of this project is to design a neural network model with the intention of benefiting waste management. To complete this task a lot of research was directed towards previous projects of similar design, in team B's case this was an object detection model targeting cans and glass-bottles.

https://www.nature.com/articles/nbt1386

This article goes through a brief history of the Neural Network mentioning some of the earliest and most primitive instances of Neural Networks such that in 1943 when McCulloch and Pitts imagined a neuron sort of as a switch that was triggered depending on the weight inputted to the switch. It also comprehensibly compares the concepts of how the human brain's neurons compare to Neural Networks. It is brought to our attention that although our current Neural Networks are powerful and can solve a wide range of problems, they still can't compare to the complexity of the human brain's neurons. The idea of how a Neural Network works is mentioned glossing over the fundamentals of the concept of layers, activation functions, weights, and more. Learning these concepts lends itself to our gr group as it helps us understand the purpose, significance, and potential of Neural Networks. At the end of the article, we begin to understand the potential of Neural Networks. The article highlights how it can benefit cancer research and more, not just object detection, which was something unknown at the start of our research.

https://ieeexplore.ieee.org/abstract/document/8704146

This case study had an influence over our team's decision to create a sorting bin. Although the case study uses LDR’s and proximity sensors to determine what type of waste is present, the overall idea of a sorting bin is there, just without a DCNN for detecting waste. What we saw we could improve upon was the design of the bin and the reliance on multiple servo motors. For our design we believe that only one servo is necessary.

https://www.mdpi.com/2076-3417/12/5/2281

This case study explores the idea of object detection and classification models designed to contribute to rubbish management in smart cities. The main technique for the object detection system is an arithmetic optimization algorithm which uses an improved RefineDet mode for object detection. The accuracy of the model was determined using benchmark datasets, which produced promising results of up to 98.61% accuracy. Through reading this case study it inspired our group heavily, in terms of choosing a waste management problem to solve. Their approach focused on recognizing more than two types of waste, rather than our model only being able to recognize either cans or glass bottles. This was due to our capacity to produce such a model, although through doing the project we would feel up for the challenge of producing a more optimized model that can detect multiple types of waste in the future.

A popular report was made by The Linely Group, a techological reaserch company that specialises in semiconductors, called "Intel Neural Compute Stick 2 (Intel NCS2): A Plug and Play Development Kit for AI Inferencing at the Edge". In the report after providing an overview of the device and testing its performance, it is concluded that the stick is a powerful and versatile AI inference solution. AnandTech also wrote a review which is more in depth and here they explain the design, performance and software stack. At the end the conclusion was that this gadget is a significant improvement over the previous generation (NCS1) and it is well suited for developers who want to experiment with AI. Another interesting report is one published by The Next Platform called: “Intel's Neural Compute Stick 2 aims to bring AI to the masses”. They go over its potential applications and how anybody can purchase the device to build an intelligent project. The conclusion they came to is that the Neural Compute Stick is the cheap and easy way to smart upgrade a system. "Intel Neural Compute Stick 2: A Smart Upgrade for Edge Devices" by RTInsights mostly goes over the features and benefits tech device has, more precisely its low power consumption, high performance and support for popular deep learning frameworks. Concluding the NCS2 to be a promising solution for developers looking to bring AI to edge devices(IoT device).

7. The Design

7.1 The Design Overview

The design for this bin is based off a basic rubbish bin. It was a main concern for this bin to be appealing to the eye so keeping it like the common bin was an agreed idea. Technology can be scary to many people and, obviously, the aim of this project is to promote proper waste management and disposal of waste. If the bin looked too futuristic or advanced people may not be as inclined to use it. It was important for this bin to look familiar but also promote a sense of curiosity which will make people want to use the bin and hence, dispose of their waste.

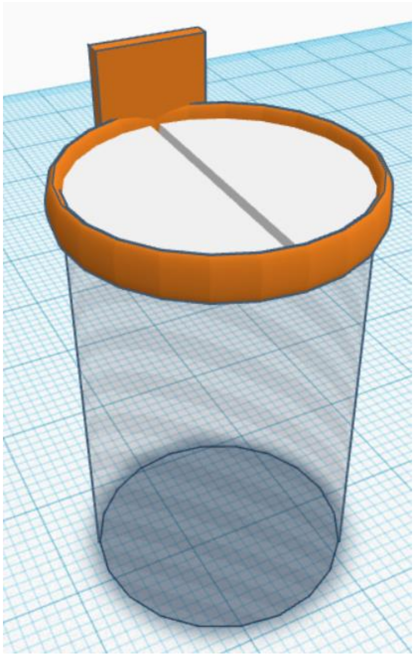


Figure 5.1, The Design: Made using CAD

7.2 The Stand

The stand is a simple concept which will be placed directly at the back, top of the bin. This can be seen in **Figure 2.2**. This stand will display a message that will read "Thank you for disposing your cans". At the bottom of the stand the camera will be placed. This camera will be able to recognise what has been placed in front of it on the shelf and detect whether it is a can.

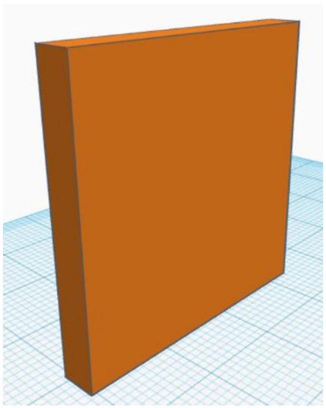


Figure 5.2, The Stand: Made using CAD

7.3 The Shelf

The shelf will be made of two platforms for the waste to sit on. These can be seen in **Figure 2.3**. Two motors will be placed underneath the shelf to open these two platforms. As discussed in the previous subsection, the camera within the stand will detect whether the waste which has been placed in front of it is a can. If it is a can, a message will be sent to the motors, telling them to rotate and hence, turning the platforms so the waste can fall into the bin. If the waste placed in front of the camera is not a can, then the motors will not rotate and a buzzer will sound, letting the person know that they may not throw their waste in there and so the person must dispose of their waste elsewhere.

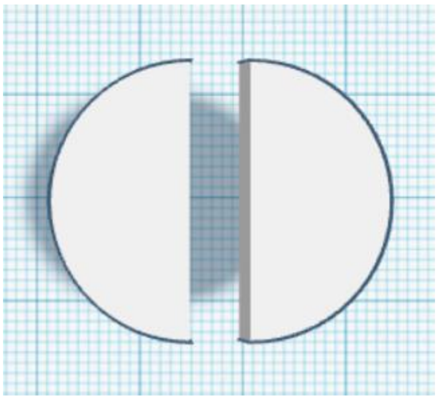


Figure 5.3, The Shelf: Made using CAD

7.4 The Main Bin

The bin structure which can be seen in **Figure 2.4**, is like that of a common rubbish bin. Within this structure there will be two ultrasonic sensors placed on either side of the bin. The ultrasonic sensors will detect whether the bin is full. The reason two sensors are used is because the waste may build up on just one side of the bin. If there was only one ultrasonic sensor then it may think the bin is full when, it's not. The two sensors can detect either side of the bin and then both of their readings can be checked to see if they're similar. If they are similar and if the bin is nearly full then that bin can be added to the route planned for the waste collectors.

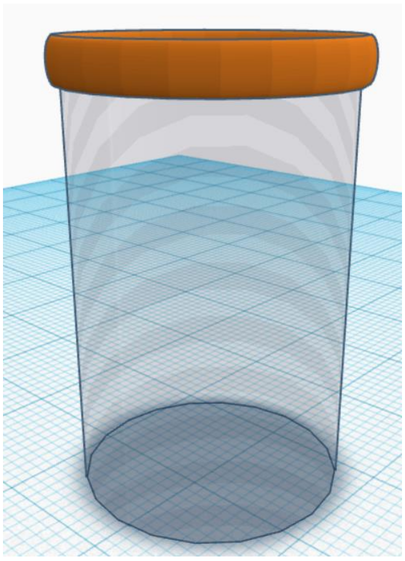


Figure 5.4, The Main Bin: Made using CAD

7.5 Improvements

When this project becomes further developed and once the team have fully understood how to use the camera and Neural Network the project can be even more advanced. A new design could include halving the bin and making one side for cans and the other for bottles. The camera will detect whether the item is a can or a bottle and then send a message to one of the motors depending on which type of waste it is. In this design the shelf would need to be sloped downwards to reduce the issue of waste being stuck on one side of the shelf. This would allow for more waste to be properly disposed of and sorted before it even gets to a waste collectors lorry.

8. Sustainability

8.1 Sustainably Developing the Design

Since this is a waste management project, it is greatly important to the team that the project is not only easy to use and interesting but also sustainably developed. The team do not have many extra equipment, such as a 3D printer. The team only have the resources given. This is why the project will be demonstrated using an old, unused bin. Every other part of the project such as the stand and the shelf will be equipment found at home that is no longer in use, making the structure of the bin completely recycled.

8.2 Sustainably Developing the Code

Although it is not mentioned much, sustainably developing code can be a major factor when it comes to a project working or not working. A major problem within software development is a term called, technical debt [19]. This refers to when a decision made in the early stages of the project end up causing long-term problems. These problems then require a large amount of time and effort to fix. If these problems are not fixed the whole project can become unsustainable and the problems too large to fix. So, the whole project must be restarted.

A few guidelines can be followed to make code more sustainable. These guidelines are not just unique to Python, many are used in a variety of coding languages such as Java, C++ and C. Firstly, it is suggested that classes are capitalised, lowercase is used for functions and variables, and that uppercased letters are used for constants. It is important to document your code as well. This can be done using comments. Comments not only help the programmer to understand what they have coded later but also helps other programmers working on the project, or others who may wish to replicate the project, understand what is happening within the code. Commenting code is not only sustainable but also considered a good coding practice. It also makes the code more efficient. Examples of these sustainable coding practices would be:

```python
class Car: #example of a class

    MY_CONST = 3.14 #example of a constant
    wheels = 4 #example of a variable
```

Figure 6, Example of Sustainable Python Coding.

Made using Visual Studio Code

9. Work Plan and Gantt Chart

Week1: 01 Feb	Week2: 08 Feb	Week3: 15 Feb	Week4: 22 Feb	Week5: 01 Mar	Week6: 08 Mar	15 Mar	Week7: 22 Mar	Week8: 29 Mar	Week9: 05 Apr	12 Apr	Week10: 19 Apr	Week11: 26 Apr	Week12: 03 May
Workshop1 - Overview	Workshop2 - Report+Presenting	Come up with ideas	Interim Presentation due	Understanding the Raspberry pi	Feedback Interim Report	Study Week	Start on Final Report	Develop shelf to only open when it detects a can	Further develop shelf	Easter Break	Continue working on Final Report	Finish up with Final Report	Final Interview due
Understanding the Task at hand	Assigned our groups	Start on Interim Report	Write up LIT Review	Downloading Ubuntu	Understanding our feedback		Start on Final Presentation	Further develop Bluetooth sync	Further develop Bluetooth sync		Continue working on Final Presentation	Finish up with Final Presentation	Final Presentation due
Research the 17 UN goals	Collaborating on what we researched last week	Write up Introduction	Create Gantt Chart	Understanding Ubuntu	Uploading more images to Neural Network		Build shelf to open and close	Further develop code for Raspberry Pi	Further develop code for Raspberry Pi		Test runs for project	Practice Presentation	Final Report due
	Understanding the kits	Outline the UN goals our project relates to	Write up CRediT Author Statement	Uploading images to our Neural Network	Develop CAD Model for project		Bluetooth to sync with our sensor	Continue working on Final Report	Upload more images to Neural Network		Fix any issues or bugs that arise during test runs	Practice Interviewing	
	Create Work Plan	Start on Interim Presentation	Combine all of our References into one document	Using Ubuntu to identify the images uploaded	Develop Schematics for project		Connect LEDs to sync with sensor	Continue working on Final Presentation	Continue working on Final Report			Continue doing test runs for project	
		Write up a Brief History of Technology in Waste	Interim Report due				Connect buzzer for Camera	Advance project to deal with bottles as well as cans	Continue working on Final Presentation			Continue fixing any issues that arise	
		Write up about Current State-of-the-Art Technologies	Understanding the Neural Network				Start code for Raspberry Pi	Upload more images to Neural Network	Connect built sections of project together				
		Start on Interim Presentation											

Figure 7.1, Work Plan: Made using Microsoft Word

The group have set up a group chat and a Teams channel. The group chat is used for updating each other on what we have accomplished and for scheduling meetings. So far, the group chat has been extremely helpful with communication within the team. It is important for any team to have effective communication skills to work together properly. The Teams channel is for sharing work more easily. In this channel the Final Report can be found so every member of the team can work on it at once. This also allows us to easily see what others have updated and changed. It also allowed for ease of sharing websites and images that the group found to improve the understanding of the project within the group. The group's Interim Presentation can also be found, allowing each member of the team to practice their set slides before the presentation.

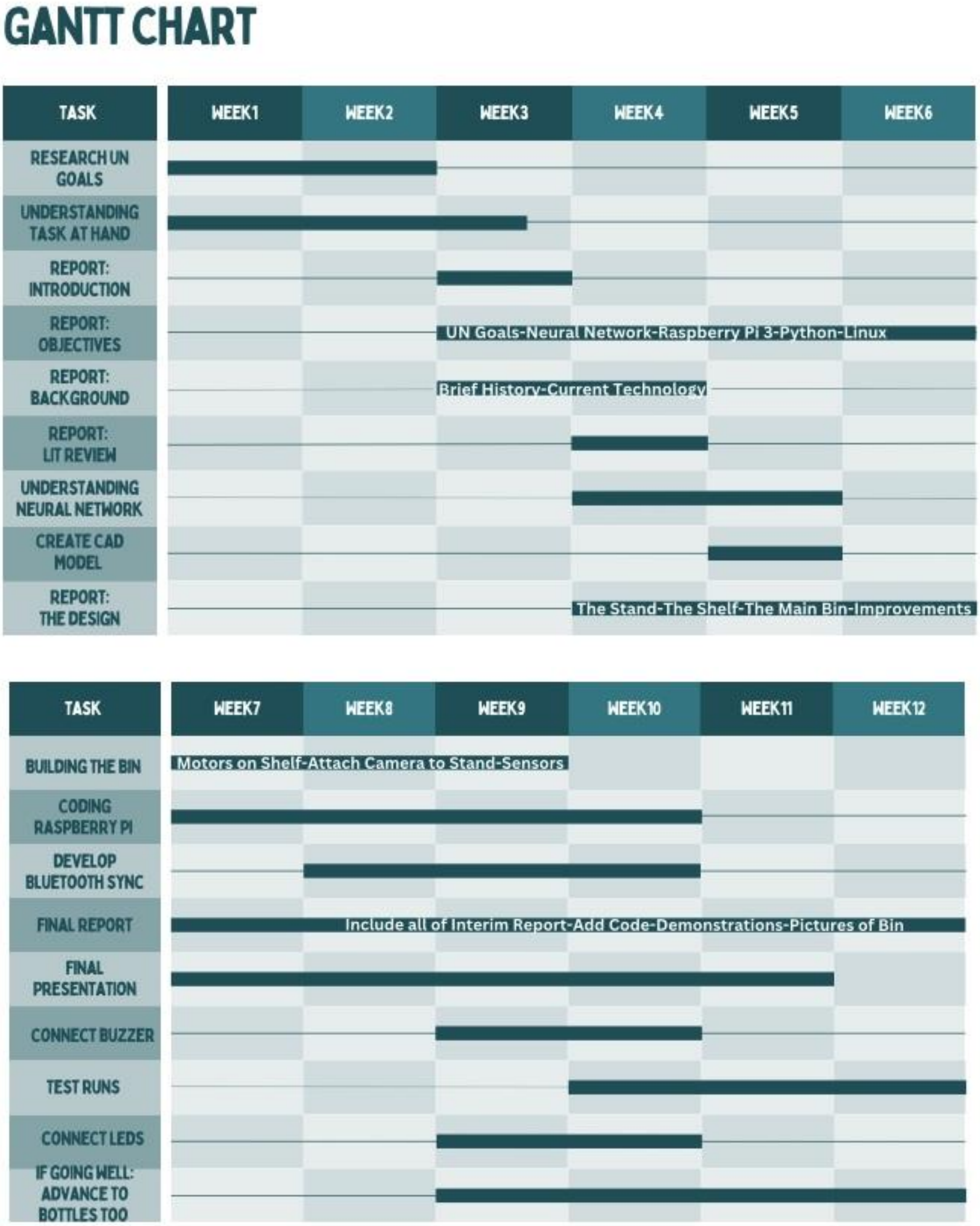


Figure 7.2, Gantt Chart: Made using Canva

10.Results

Although the device has not yet been tested using real life objects, the team are confident that the neural network will do as is desired. The accuracy graph below can be summarized as having a 98% accuracy in detecting cans through images fed into the network. The overall val accuracy is also roughly around 98% but leaning more towards 97%. Val accuracy records the average percentage of correctly classified images while the custom model is being evaluated.

As can also be seen, the graph fluctuates. Meaning the neural network isn't entirely accurate and there is room for improvement. However, since the overall accuracy is over 90%, these improvements can be easily fixed with time.

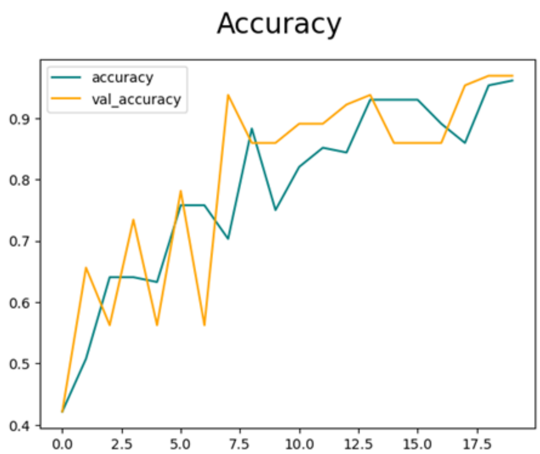


Figure 8: Accuracy graph

Building a neural network requires a lot of time and effort but compared to alternative computer vision algorithms, neural networks are far more successful and have a higher capability for complex and varied data.

11.CRediT Author Statement

Leah attended 22/22 lab sessions provided within our scheduled timetable. These were 3-hour labs on Wednesdays 9-12 and Thursdays 1-2, then 3-5. In these labs she made a start on researching the UN's 17 Sustainable Development goals outlined in the 2030 Agenda for Sustainable Development. She recognised that five of the goals related to waste management but only four of these five goals related to the project agreed upon by all members of the team. Leah worked a lot on the report. She wrote sections *1. Introduction*, *2.1 Objectives Overview*, *2.2 Sustainable Development Goals Relating to our Project, 3.1 Brief History, 4.3 TensorFlow 5. Implementation, 7. The Design, 8. Sustainability, 10. Results* and she also designed a work plan and Gantt Chart which can be seen in **Figure 3.1** and **Figure 3.2**. Leah attempted to set up Linux Ubuntu on her computer but due to issues with her laptop this was not possible. Her laptop also made her unable to work on the code as jupyter lab would overwork her laptop causing it to crash.

Niall attended 21/22 lab sessions provided within our scheduled timetable. In these labs he researched the Intel Neural Network and learned how it functions and what it can do within our project. He also researched the Raspberry Pi. In relation to this research, Niall wrote sections *2.3 How a Neural Network Works, 2.4 Uses of a Neural Network*, *4.1 Raspberry Pi 3, 4.2 Python, 4.4 Arduino* and he also wrote *3.2 Current Technology* which describes the Neural Network today. Niall worked a lot on the power point. This was one of his main functions within the team as he enjoys using the PowerPoint app and had more skills than everyone else in the team at using it.

Markel attended 20/22 lab sessions provided within our scheduled timetable. In these labs he researched the Raspberry Pi and learned how we could use it within the project. Markel brought the kit home so he could learn more on how to use each device. It was here that he noted there was no SD card within our Raspberry Pi but luckily our lecturer had a spare for us to use, so he installed the Raspberry OS. In relation to the report, Markel wrote *4. LIT Review* and continued to work with the equipment we were given so the team could better understand how each item worked and thus, draft a better report including what we had learned. Markel also wrote the code used in the Arduino for moving the motor to open and close the bin.

12. References

[1] *Transforming our world: The 2030 agenda for sustainable development | department of economic and social affairs* (no date) *United Nations*. United Nations. Available at: https://sdgs.un.org/2030agenda

[2] McCullum, N. (2021) *Deep Learning Neural Networks explained in plain English, freeCodeCamp.org. freeCodeCamp.org*. Available at: https://www.freecodecamp.org/news/deep-learning-neural-networks-explained-in-plain-english/

[3] Cansiz, S. (2020) *The math behind training neural networks, Medium. Towards Data Science.* Available at: https://towardsdatascience.com/adventure-of-the-neurons-theory-behind-the-neural-networks-5d19c594ca16

[4] Frost, J. (2021) Mean squared error (MSE), Statistics By Jim. Available at: https://statisticsbyjim.com/regression/mean-squared-error-mse/

[6] Saeed, M. (2023) *An introduction to recurrent neural networks and the math that powers them, MachineLearningMastery.com*. Available at: https://machinelearningmastery.com/an-introduction-to-recurrent-neural-networks-and-the-math-that-powers-them/#:~:text=A%20recurrent%20neural%20network%20(RNN,are%20independent%20of%20each%20other

[7] (LEDU), E.E. (2018) *Understanding K-means clustering in machine learning, Medium*. Towards Data Science. Available at: https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1

[8] *Supervised learning and other machine learning tasks* (no date) *SuperAnnotate*. Available at: https://www.superannotate.com/blog/supervised-learning-and-other-machine-learning-tasks

[9] *Autoencoders in Deep learning: Tutorial & use cases [2023]* (no date) *V7*. Available at: https://www.v7labs.com/blog/autoencoders-guide#:~:text=An%20autoencoder%20is%20an%20unsupervised,even%20generation%20of%20image%20data

[10] *What is a Raspberry Pi?* (2015) *Raspberry Pi*. Available at: https://www.raspberrypi.org/help/what-%20is-a-raspberry-

pi/#:~:text=The%20Raspberry%20Pi%20is%20a,languages%20like%20Scratch%20and%20Python

[11] Shukla, S. (2022) *Raspberry pi operating systems*, *Scaler Topics*. Scaler Topics. Available at: https://www.scaler.com/topics/operating-system-of-raspberry-pi/

[12] Raspberry Pi (no date) *Buy A raspberry pi 3 model B*, *Raspberry Pi*. Available at: https://www.raspberrypi.com/products/raspberry-pi-3-model-b/

[13] *Python® – the language of today and Tomorrow* (no date) *About Python*. Available at: https://pythoninstitute.org/about-python#:~:text=Python%20was%20created%20by%20Guido,called%20Monty%20Python's%20Flying%20Circus.

[14] *Introduction to tensorflow | machine learning | google developers* (no date) *Google*. Google. Available at: https://developers.google.com/machine-learning/crash-course/first-steps-with-tensorflow/toolkit

[16] Mäntylä, K. (no date) *Linus Torvalds*, *Encyclopædia Britannica*. *Encyclopædia Britannica, inc.* Available at: https://www.britannica.com/biography/Linus-Torvalds#/media/1/600294/117654

[17] *(PDF) development drivers for waste management - researchgate* (2007). Available at: https://www.researchgate.net/publication/6226122_Development_drivers_for_waste_management

[18] *A deep convolutional neural network for classifying waste ... - IEEExplore* (no date). Available at: https://ieeexplore.ieee.org/document/9071746/

[19] *Writing sustainable code* (no date) *Writing Sustainable Code – Managing Academic Software Development.* Available at: https://southampton-rsg.github.io/swc-project-novice/05-quality/index.html

[20] *Introduction to tensorflow| machine learning |google developers* (no date) *Google*. Available at: https://developers.google.com/machine-learning/crash-course/first-steps-with-tensorflow/toolkit

[21] *Tensorflow Basics: Tensorflow Core* (no date) *TensorFlow*. Available at: https://www.tensorflow.org/guide/basics

[22] Nielsen, M.C. (2023) *Neural networks and deep learning: A textbook*.

Can be found at: http://neuralnetworksanddeeplearning.com

13.Equation Legend

Equation Number	Explanation
(1)	Formula for MSE (Mean Squared Error)

14.Figure Legend

15.Code

```
import tensorflow as tf

import os

gpus = tf.config.experimental.list_physical_devices('GPU')

for gpu in gpus:

	tf.config.experimental.set_memory_growth(gpu, True)

tf.config.list_physical_devices('CPU')

import cv2

import imghdr

data_dir = 'data'

image_exts = ['jpeg', 'jpg', 'bmp', 'png']

for image_class in os.listdir(data_dir):

	for image in os.listdir(os.path.join(data_dir, image_class)):
```

```
            image_path = os.path.join(data_dir, image_class, image)
            try:
                img = cv2.imread(image_path)
                tip = imghdr.what(image_path)
                if tip not in image_exts:
                print('Image not in ext list{}'.format(image_path))
                os.remove(image_path)
            except Exception as e:
                print('Issue with image{}'.format(image_path))
os.listdir(data_dir)
tf.data.Dataset??
import numpy as np
from matplotlib import pyplot as plt
data = tf.keras.utils.image_dataset_from_directory('data')
data_iterator = data.as_numpy_iterator()
batch = data_iterator.next()
fig, ax = plt.subplot(ncols=4, figsize=(20,20))
for idx, img in enumerate(batch[0][:4}):
    ax[idx].imshow(img.astype(int))
    ax[idx].title.set_text(batch[1][idx])
batch[0].shape
data = data.map(lambda x,y: (x/255,y))
scaled_iterator = data.as_numpy_iterator()
batch = scaled_iterator.next()
data.as_numpy_iterator().next()
batch[0].min()
train_size = int(len(data)*.7)
val_size = int(len(data)*.2)+1
test_size = int(len(data)*.1)+1
train = data.take(train_size)
val = data.skip(train_size).take(val_size)
```

```
test = data.skip(train_size+val_size).take(test_size)
len(test)
from tensorflow.keras.models import Sequential
from tensforflow.keras.layers import Conv2D, MaxPooling2D, Dense, Flatten, Dropout
model = Sequential()
model.add(Conv2D(16, (3,3), 1, activation ] ‘relu’, input_shape = (256,256,3)))
model.add(MaxPooling2D())
model.add(Conv2D(32, (3,3), 1, activation = ‘relu’))
model. add(MaxPooling2D())
model.add(Conv2D(16, (3,3), 1, activation = ‘relu’))
model.add(MaxPooling2D())
model.add(Flatten())
model.compile(‘adam’, loss=tf.losses.BinaryCrossentropy(), metrics = [‘accuracy’])
model.summary()
logdir = ‘logs’
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=logdir)
hist = model.fit(trains, epochs = 10, validation_data = val, callbacks = [tensorboard_callback])
fig = plt.figure()
plt.plot(hist.history[‘loss’], color=’teal’, label = ‘loss’)
plot.plot(hist.history[‘val_loss’], color= ‘orange’, label = ‘val_loss’)
fig.suptitle(‘Loss’, fontsize = 20)
plt.legend(loc = ‘upper left’)
plt.show()
fig = plt.figure()
plt.plot(hist.history[‘accuracy’], color = ‘teal’, label = ‘accuracy’)
plt.plot(hist.history[‘val_accuracy’], color = ‘orange’, label = ‘val_accuracy’)
fig.suptitle(‘Accuracy’, fontsize = 20)
plt.legend(loc = ‘upper left’)
plt.show()
from tensorflow.keras.metrics import Precision, Recall, BinaryAccuracy
pre = Precision()
```

```
re = Recall()
acc = BinaryAccuracy()
for batch in test.as_numpy_iterator():
X, y = batch
yhat = model.predict(X)
pre.update_state(y, yhat)
re.update_state(y, yhat)
acc.update_state(y, yhat)
print(f'Precision:{pre.result().numpy()}, Recall:{re.result().numpy()}, Accuracy:{acc.result()}')
import cv2
img = cv2.imread('bottle.jpg')
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.show()
resize = tf.image.resize(img, (256,256))
plt.imshow(resize.numpy().astype(int))
plt.show
yhat = model.predict(np.expand_dims(resize/255, 0))
if yhat > 0.5:
	print(f'Predicted class is CAN')
else:
	print(f'Predicted class is BOTTLE')
from tensorflow.keras.models import load_model
model.save(os.path.join('models', 'BottleCanModel.h5'))
new_model
yhatnew = new_model.predict(np.expand_dims(resize/255, 0))
if yhatnew > 0.5:
	print(f'Predicted class is CAN')
else:
	print(f'Predicted class is BOTTLE')
```