EE297 Project

Neural Network

Group 3

By:

Conal Hughes   - 20365616

Stephen Gallagher   - 20470574

Gerard McIntyre   - 20332566

Amy O'Mara   - 20386631

Table of Contents

TABLE OF FIGURES

1   ABSTRACT

The aim of this project is to find the best waste/rubbish solution that ties back to one of the 17 Sustainable Development Goals (SDGs) set out by the United Nations in the 2030 Agenda for Sustainable Development. We are required to implement a neural network to address the problem of waste and to tie it back to one of these Sustainable Development Goals. This report will cover the technical details behind the implementation of one of these neural networks, the idea generation for this problem, the development of our final idea and the process and results of training one of these neural networks. To understand some of the concepts behind these, there must be a lot of research done into the structure, coding and training of a neural network. The research will primarily be using technical reports, academic papers and books with the assistance of online sources. As this is a very complex process, all information will have to be cross checked across multiple sources.
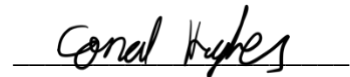
2   DECLARATION FORM

We hereby declare that this technical report titled "EE297 Final Report" submitted to Maynooth University, is a record of an original work completed by Stephen Gallagher, Amy O'Mara, Conal Hughes and Gerard McIntyre under the guidance of Dr John Dooley, and that this report is submitted as part of the partial fulfilment of the requirements for the module EE297 for the course MH306, BSC Robotics and Intelligent Devices. The contents of this report have not been submitted to any other University or Institute for the award of any degree.
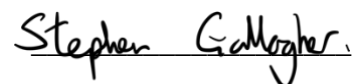
06/05/2022

Signed:

Conal Hughes

Stephen Gallagher

Gerard McIntyre

Amy O'Mara

3 INTRODUCTION

3.1 WHAT IS THE PROBLEM?

The object of this project is to design a technical solution using a neural network, to tackle a problem caused by waste. The solution should work towards solving an issue from one of the 17 Sustainable Development Goals(below) from the 2030 Agenda. The blueprint outlines the ways in which we as a community can work towards peace and prosperity for our planet for a more sustainable future. This solution should follow one of the following methods: classification, prediction, clustering, association.

3.2 ASSESSMENT REQUIREMENTS

To fulfil the assessment requirements, the following information should be presented; the exact problem being solved with the neural network and the impact of said problem, how it meets one of the 17 Sustainable Development Goals (below), give an in-depth analysis of the structure of a neural network, explain what software is being used. The data used to train the neural network, how the data was processed, where it was sourced and how the neural network was trained must also be explained. Finally, if and how the network was validated that it works or doesn't work should be documented.

3.3 MACHINE LEARNING

Machine Learning is often considered the exact same to artificial intelligence, yet machine learning is a branch of artificial intelligence. Artificial intelligence is the mimic of human behaviour. Machine Learning goes far beyond the capabilities of the simple brain of human intelligence. With the use of algorithms that change over time and get better results for what they were built for, they can process large quantities of information and data very quickly. A manufacturing factory would collect information from its operators and sensors in very large quantities which would take many humans to do the same job. Machine learning uses patterns and correlations to indicate to humans a problem that they will be able to control and fix.

3.4 THE 17 SUSTAINABLE DEVELOPMENT GOALS

[1] The 17 Sustainable Development Goals was created by Union Nations [2]. They are as follows:

1. No Poverty

2. Zero Hunger

3. Good Health and Well-Being

4. Quality Education

5. Gender Equality

6. Clean Water and Sanitation

7. Affordable And Clean Energy

8. Decent Work and Economic Growth

9. Industry, Innovation and Infrastructure

10. Reduced Inequalities

11. Sustainable Cities and Communities

12. Responsible Consumption and Production

13. Climate Action

14. Life Below Water

15. Life on Land

16. Peace, Justice and strong Institutions

17. Partnerships for the Goals

On this earth, there are over 7 billion of us. Every day, each of us produce waste in many forms. Most waste caused by the human population is not recycled, collected or disposed of correctly, causing an international global crisis. Waste management is a major problem that must be addressed by all nations. Countries treat these resources as if they will never run out, and not affect the world we live in. Growth and prosperity cannot continue without this major global priority being controlled and managed better.

Over 75% of open dump areas are on the coast which results in harmful and hazardous items and materials drifting into our oceans. Many marine animals and ecosystems have felt the damage of our pollution. For example, coral reefs are being bleached resulting in them receding, and marine mammals are washing up on beaches with their stomachs full of

plastics. SDG 1 (above) focuses on No Poverty, and that includes waste management, where the benefits of it far outweighs the cost of doing nothing at all. [3]

4   TECHNICAL BACKGROUND

4.1   WHAT IS A NEURAL NETWORK?

A neural network is an artificial representation of the human brain, often referred to as ANN, artificial neural network. The main difference between a neural network and a regular programming script is that it does not use hard coded conditions, but rather learns from past experiences and data [4]. It uses layers of neurons, where each layer interacts with the other using connections. The impact of each of these connections is determined by the activation function, weights and biases. *Figure 1(below)* is a diagram of how the output of each neuron is determined based on these parameters.

$$f\left(b + \sum_{i=1}^{n} x_i w_i\right)$$

An example of a neuron showing the input (x_1 - x_n), their corresponding weights (w_1 - w_n), a bias (b) and the activation function f applied to the weighted sum of the inputs.
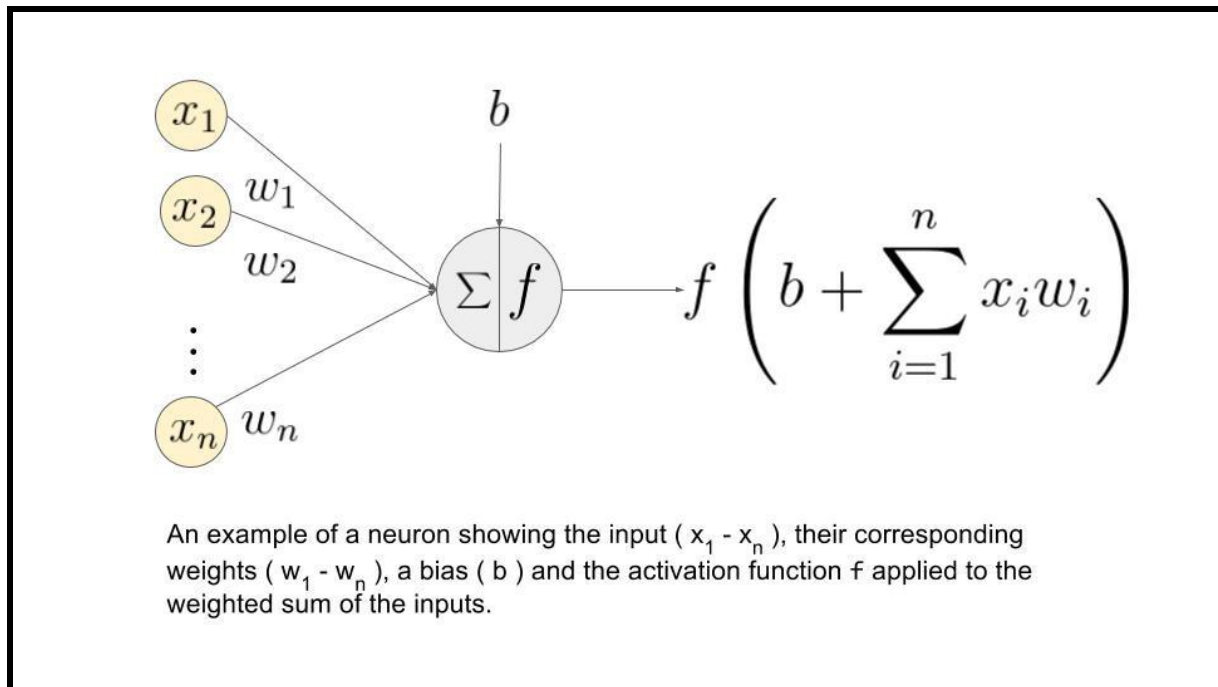
Figure 1 - Diagram of a neuron

The weighting of the connections is especially important. The following is a metaphor used to understand the effects and importance of weighting. It is based on the real-world example of surfing for simplicity:

The conditions are initialised with identical weights and are as follows:
1. Are there good waves? (Yes = 1, No = 0)
2. Is it busy? (Yes = 0, No = 1)
3. Has there been a shark attack recently? (Yes = 0, No = 1)

Consider:

Scenario 1	Score	Scenario 2	Score
Are there good waves?	1	Are there good waves?	1
Is it busy?	0	Is it busy?	1
Has there been a shark attack recently?	1	Has there been a shark attack recently?	0

The final score for both above scenarios is 2, which is a desirable score. However, the shark attack parameter should have a higher weight on the final decision than the other two parameters. This scenario proves the importance of weights and how the more important aspects should have higher weights. In a neural network, these weights are adjusted during backpropagation, and allow the neural network to learn what is and isn't important when identifying features.

Backpropagation is the process in which the neural network is trained. It does this based on the loss functions. The loss functions are calculated by getting the difference between the actual output and the desired output. [5] [6] Backpropagation uses the chain rule while moving back through the layers.

The next parameter that affects the output of each neuron is the bias. The bias adjusts the sum of inputs to the neuron and adjusts how the activation function is triggered. The bias is also adjusted during backpropagation. The output of the neuron is sent into the activation function to get the final output of the individual neuron. After conducting some research, it was determined that the ReLU and sigmoid activation functions were the most widely used. However, there are variations of the two.
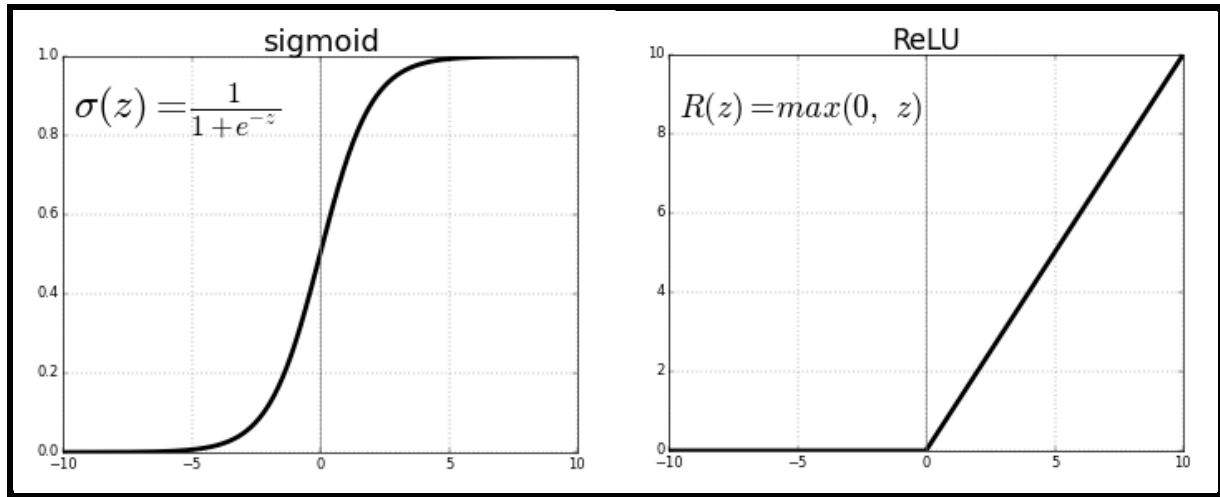
Figure 2 - Visual representation of a sigmoid and ReLU activation function

The sigmoid activation function condenses the output between two values. In *Figure 2(above),* for example, it condenses values between 0 and 1. Sigmoid ensures there are no outliers in the data. This function has a disadvantage in larger networks with more layers. It can be susceptible to "The Vanishing Gradient Problem". [7]

The vanishing gradient problem occurs during back propagation, and it results in a neural network that has improperly trained early layers. The effects the loss function has on training are related to the derivative of the activation function.

Equation 1(below) details the sigmoid formula:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Equation 1 - Sigmoid formula

With its derivative being the following:

$$f'(x) = f(x)(1 - f(x))$$

Equation 2 - Derivative of sigmoid formula

The result of the above derivative in *Equation 2* (above) will always be less than 1. It can be determined that the highest derivative will be 0.5. During backpropagation calculus, the chain rule is used, which means this 0.5 will be multiplied by itself as the neural network

12

backpropagates. This value will decrease exponentially as it moves back through the layers, meaning, the earlier layers will be affected very slightly by the training.

The ReLU activation function normalises negative values to 0 and is linear for all positive values. ReLU helps prevent the vanishing gradient problem, as it has a larger gradient than the sigmoid activation function.

A Neural Network can become "dead" if all the outputs tend to 0 and is unable to change this. ReLU can cause dead networks because any value less than zero will yield a 0 output. This means it cannot learn from negative outputs. The neurons can also output a zero if a large gradient is passed through the function. This will result in massive negative weights and biases, as the network will try to counteract this during backpropagation. From this, the neuron will always output 0 and will have no way of correcting itself.

This is also a reason as to why biases are important. A positive bias will always ensure that each neuron that uses ReLU will have a positive output to begin with. The issue arises when an entire layer within the hidden layers becomes dead, as this will cut off connections from its previous layer.

A different activation function is used on the output layer, called the SoftMax activation function. It is very similar to the sigmoid activation function but ensures that all the outputs are a probability out of 100. Equation 3(below) illustrates the function, as well as the summation of inputs that allows the even distribution of probabilities. [8] [9]

$$softmax(z_i) = \frac{exp(z_i)}{\Sigma_j exp(z_i)}$$

Equation 3 - SoftMax function

4.2 TYPES OF NEURAL NETWORKS

There are five main types of neural networks, CNN, RNN, R-CNN, TDNN and GNN. Each type has their different advantages and are used for different applications.

4.2.1 CNN

CNN stands for Convolutional Neural Network. It is useful in many different applications, in particular image related tasks. This may include image classification, image semantic segmentation, object detection in images etc. [8]Image classification is when, within an image, a major object that occupies a large portion of the image is grouped into one determined class.
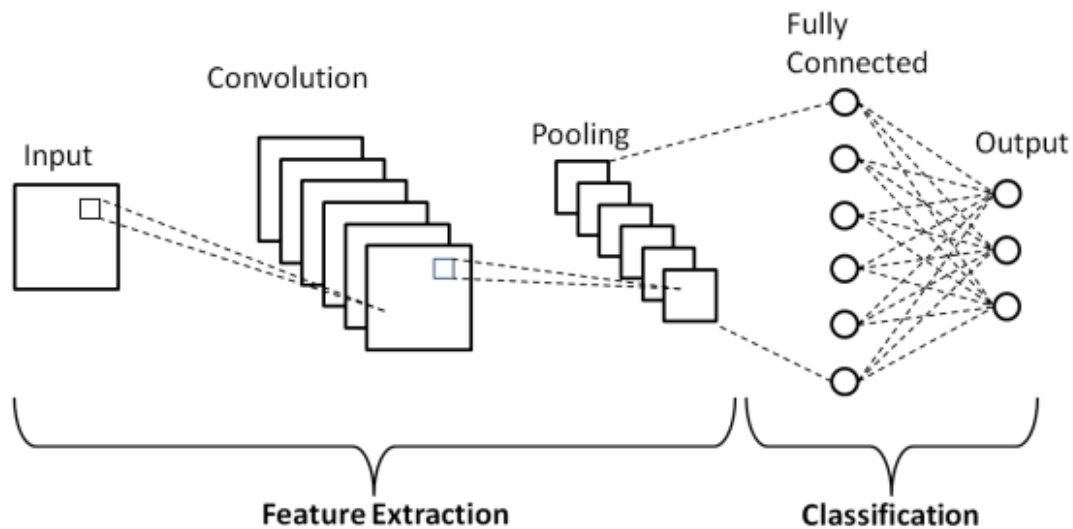
Figure 3 - CNN Architecture

The structure of a CNN is shown in *Figure 3(above)*. Input data is fed into convolutional layers. These are used to extract the defining features from the input images. The convolutional layers consist of a number of filters depending on the parameters associated with the data being interpreted. Next in the pooling layer, pooling reduces the number of pixels in the output images from the previous layer. The fully connected layer (FC) is where the unique weights and biases are stored, along with the neurons. The image passed through the previous layers is flattened and fed into this layer, this is where the classification process begins. [1]

4.2.2 RNN

Recurrent Neural Networks are neural networks that allow previous outputs to be used as inputs. RNN's are mainly used in the fields of language processing and speech recognition. It takes in previous outputs as inputs to simulate the progression of language, and as to not repeat itself. [10]

14
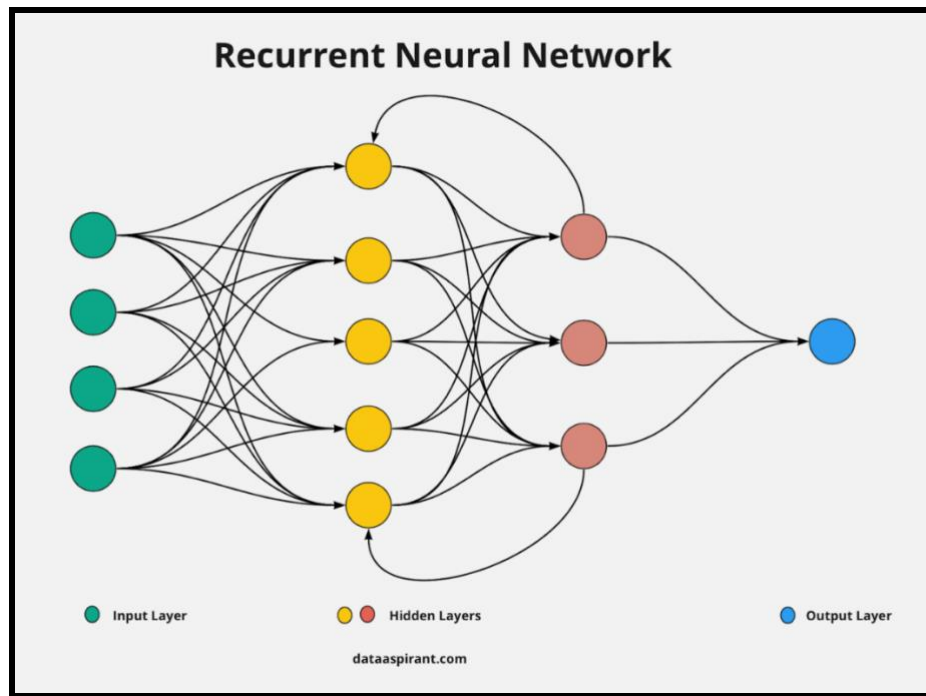
Figure 4 - RNN Architecture

As seen in *Figure 4 (above),* information is constantly being fed back within the hidden layers, this is what distinguishes RNNs from other neural networks. [11]

4.2.3 R-CNN

A Region-based convolutional neural network is a type of CNN with two main operations. Firstly, it finds where it believes an identifiable object could be, then it makes an attempt at classifying this object.
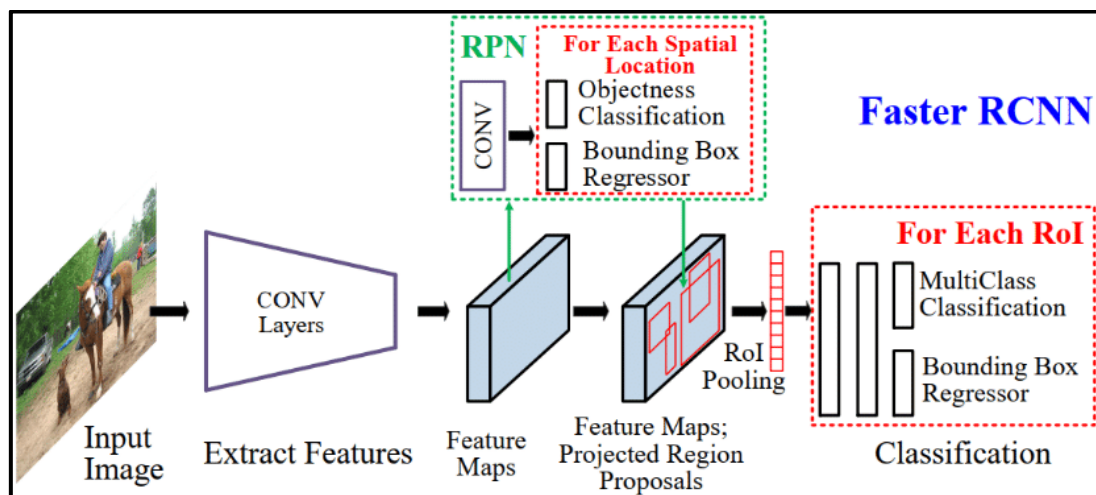
Figure 5 - R-CNN Architecture

[12]This is the type of neural network that was implemented into our design, because of the object classification in images and that the image it is viewing may consist of many objects.

4.2.4 TDNN

A time delay neural network takes time into consideration during operation. They are used in speech recognition. Since speech contains patterns within a larger pattern, it needs a way of specifying which part to look at, it does this by taking time into consideration as well as the input data.
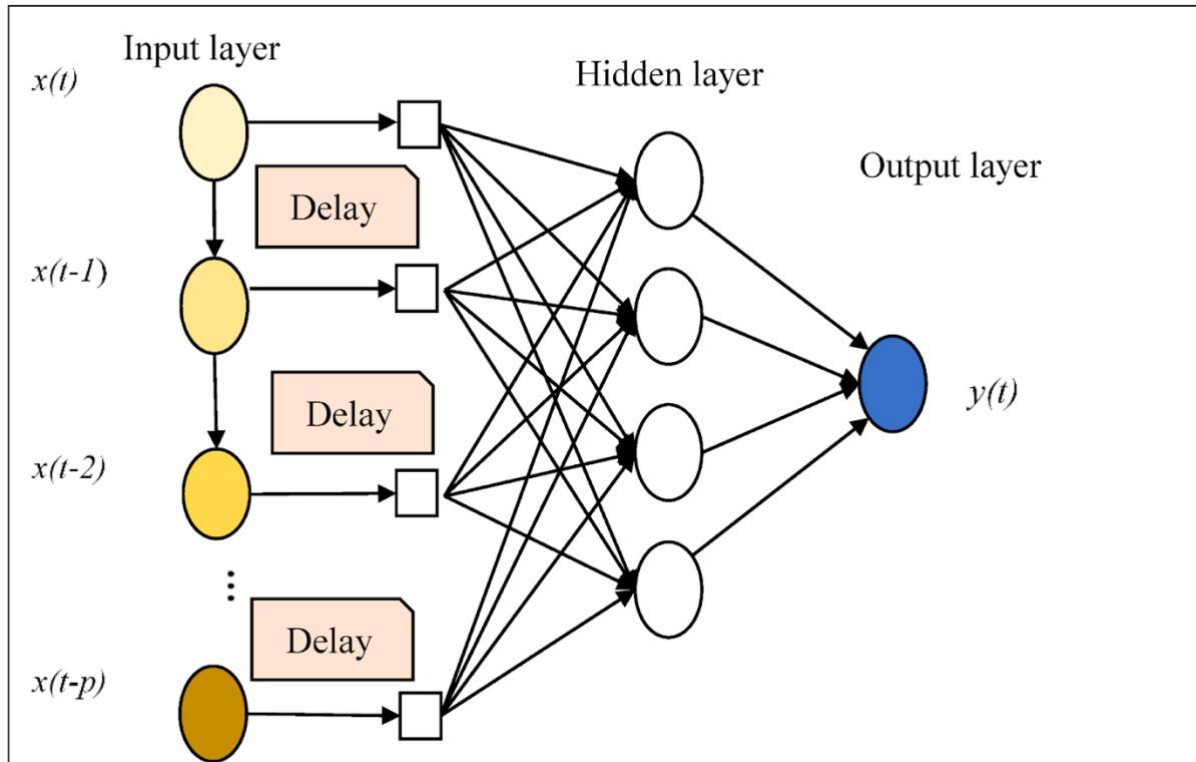
Figure 6 - TDNN Architecture

Figure 6(above) is the architecture of a TDNN neural network.

4.2.5 GNN

Graph neural networks (GNN) are a type of neural network that can be used directly on graphs. An example of a graph where a GNN could be implemented on is a roadmap. One of the team's initial ideas during the development stages was a traffic predictor that would cut down on traffic congestion. [13]

4.3   TRAINING THE NEURAL NETWORK

To train a Neural Network, a very wide range of images is used. The network needs to be able to identify many different objects under different headings, so a dataset of only a few hundred images would be useless. A large dataset of about 2000 images would need to be gathered to achieve an accurate neural network, possibly more than that.

While training the network, the dataset will need to be split into three separate headings: training, validation and testing. The training images will be the ones used to physically train the neural network and to adjust the weights of each neuron to fit our needs. The validation set makes sure that the network is recognising the right objects correctly and the test class is what the network will test itself with. It is very important to make sure that these are all different images so that it does not test itself on images that it has used to train itself with. This would result in misleading accuracy percentages.

There are many ways to train the Neural Network. Some of these methods include the gradient descent, Newton's method or the conjugate gradient method. [14]

The gradient descent method is quite common and is the simplest training method to implement. It completes its computations very quickly but takes a lot more iterations. It compares the predicted result to the actual result, calculates the gradient of the output error, then back propagates back through the neural network, changing biases and weights as needed. [15]

The Newton method is a lot different in the fact that it obtains the second derivative of the loss function and uses that to find the training direction and the training weight. This computation time is a lot longer than that of gradient descent, however it takes less iterations. Newton's method may not be as exact and will struggle to find the minimal loss function.

The conjugate gradient method is somewhere between the previous two methods. It applies the direction similarly to the Newton method and compares the gradient of the output error in a similar way to the gradient descent. What results is a method that is not as fast as the gradient descent but has less iterations. [16]

17

5 IDEA GENERATION

5.1 BRAINSTORMING

Any ideas that were thought of by the group, stemmed from one of the 17 Sustainable Development Goals(above) set out by the United Nations. [2] Five main ideas for the project were suggested, using many types of neural networks. A smart bin was discussed, which involved sorting through recycling and general waste products. This idea involved using a CNN. There was a stock predictor that measured how much of a certain item was sold in a shop to reduce excess products sold. A traffic predictor that would analyse traffic patterns and plot an optimal route. [17]That idea could incorporate the use of a GNN. We also noticed that most plastics had a number in the recycling triangle symbol, and methods of identifying these numbers were discussed. In the end we decided to go with a camera that would detect plastics in parks.

5.2 FIRST IDEA

The idea is that parks are covered in grass and green trees so if something is detected that isn't green, it probably should not be there. This idea is versatile as it can be altered for any colour. This was a progressive idea and could have been expanded in the future. Unfortunately, it was decided that it would be best to branch away from using this idea as there were too many variables that were not green but not rubbish either. This included flowers, people, footpaths, and lakes.

5.3 SECOND IDEA

It was decided to branch away from identifying colours and focus on the objects the camera would stream into our neural network model. This then would include the flowers, people, animals, footpaths and lake etc. Research was done to find the best neural network structure for identifying objects and a CNN based model suited best. The idea still stood that we would ignore certain parameters, now instead of ignoring most colours, it would ignore the objects that would be made into a dataset. It was hoped that the code could be altered to highlight anything on the stream that wasn't in the dataset. This was deemed too problematic as it was found out that identifying objects that were annotated in the dataset is easier than finding objects that weren't.

5.4 FINAL IDEA

The final idea was to reduce the amount of pollution and waste in parks, fields, and public areas by identifying rubbish in these areas. This would be done by flying a drone over an area and letting the built-in camera stream a live video to the user's device. This allows our neural network to create boundary boxes around the rubbish identified from the dataset.

Five classes were made (Glass, Paper, Cardboard, Plastic, Metal) as it was agreed that these classes are the most common types of rubbish found in these areas. This would help reduce the amount of waste destroying the environment and would keep public areas free from litter. This idea would link back to the roots of the 17 Sustainable Development Goals(above), mainly SDG 1 (No Poverty), SDG 13 (Climate Change), and SDG 15 (Life on Land). By implementing this idea, it would help reduce the environmental damage we as humans cause to the earth.

6   OUR SOLUTION

6.1   MAIN PROBLEM TO SOLVE

After deciding on our final idea, we had to solve a number of problems individually before we could bring the project together. Firstly, we had to develop a base neural network that we could design our solution on. We quickly discovered by looking at examples online and looking into the theory involved in a neural network that programming one from scratch ourselves would be beyond our capabilities. We reasoned that a pre-programmed neural network like one of the YOLO models would be the best option.

Our second problem was obtaining an annotated dataset that could be used in our neural network. This would need to be a very specific dataset containing everything that we wanted to identify, and enough of each class to identify them. We would have to find a dataset that suits our needs online, but we would likely need to either add to an insufficient dataset or to make our own one.

Thirdly, we had to then train and implement the neural network on either the given hardware or our own hardware. No one in the group had ever used a Raspberry Pi before so we weren't sure what problems that would entail so research into the given hardware was needed.

Finally, we would have to bring all the solutions to these problems together. As our idea is to be able to fly a drone over a park while our neural network is running to detect rubbish, all these problems must be solved, and the solutions should be able to run concurrently for our project to operate as desired.

Even after these problems are solved, we will need to solve further problems such as drone range, stable network connection and even GDPR problems of flying over a public area. However, our priorities are getting the neural network operating as desired before any of these extra problems are considered.

6.2 SOFTWARE USED

6.2.1 ROBOFLOW AND GOOGLE COLAB

Roboflow is a website that allows users to easily create a working neural network model. Roboflow manages your images, annotations and labels, pre-processing, augmentations, and file formats. It can allow images to be uploaded that may or may not be annotated at the time. Roboflow allows the user to annotate their dataset with as many classes as the user desires, making boundary boxes to identify where the object is in the image. Roboflow is also capable of training the dataset using their own method, which is a combination of many different methods. They unfortunately do not reveal the training algorithm, and their training method and weights do not exist off the Roboflow site. It is possible to run the network from Roboflow but that is very limiting as a network connection is needed and it has to be a live video feed, the network cannot be run on a pre-recorded video.

After completing the training on Roboflow and obtaining their success rates, Roboflow's added functionality to format the images in a way that could be recognised by YOLOv5. This was exported into Google Colab which trained the Network using a git clone to obtain the YOLOv5 training files. We used our own .yaml file that held all 5 of our classes, and after the training was completed, we obtained a .pt file that we could use to either train our own network or to run it using temporary weights. We used the temporary weights option as we felt we needed to run different weights to prove or disprove any theories we might have had. The training provided us with further information about the structure of the Neural Network. There are a total of 270 layers, with 7,033,114 parameters.

6.2.2 YOLOv5

YOLOv5 is a Region-based Convolutional Neural Network. This was a recommended R-CNN for image recognition, so we decided to try and implement it ourselves. Unfortunately, there is no official YOLOv5 paper, but we managed to obtain the majority of its structural information from other sources. YOLO stands for You Only Look Once, indicating that the neural network only looks at the inputted image once. We downloaded our code from the YOLOv5 GitHub [18].

We then realised that YOLOv5 works very well with Roboflow, the website we had decided to use for our training as you could run your own .pt file as custom weights. This was perfect for us as we did not know enough python coming into this project to program a neural network. YOLOv5 is a feedforward neural network divided into different subsections, or regions, and it then predicts the probability of a bounding box in those areas. [19]Some of these regions focus on extracting smaller and smaller sections of the image for feature extraction, and some of the later regions focus on feature analysis and identification. [20]

21
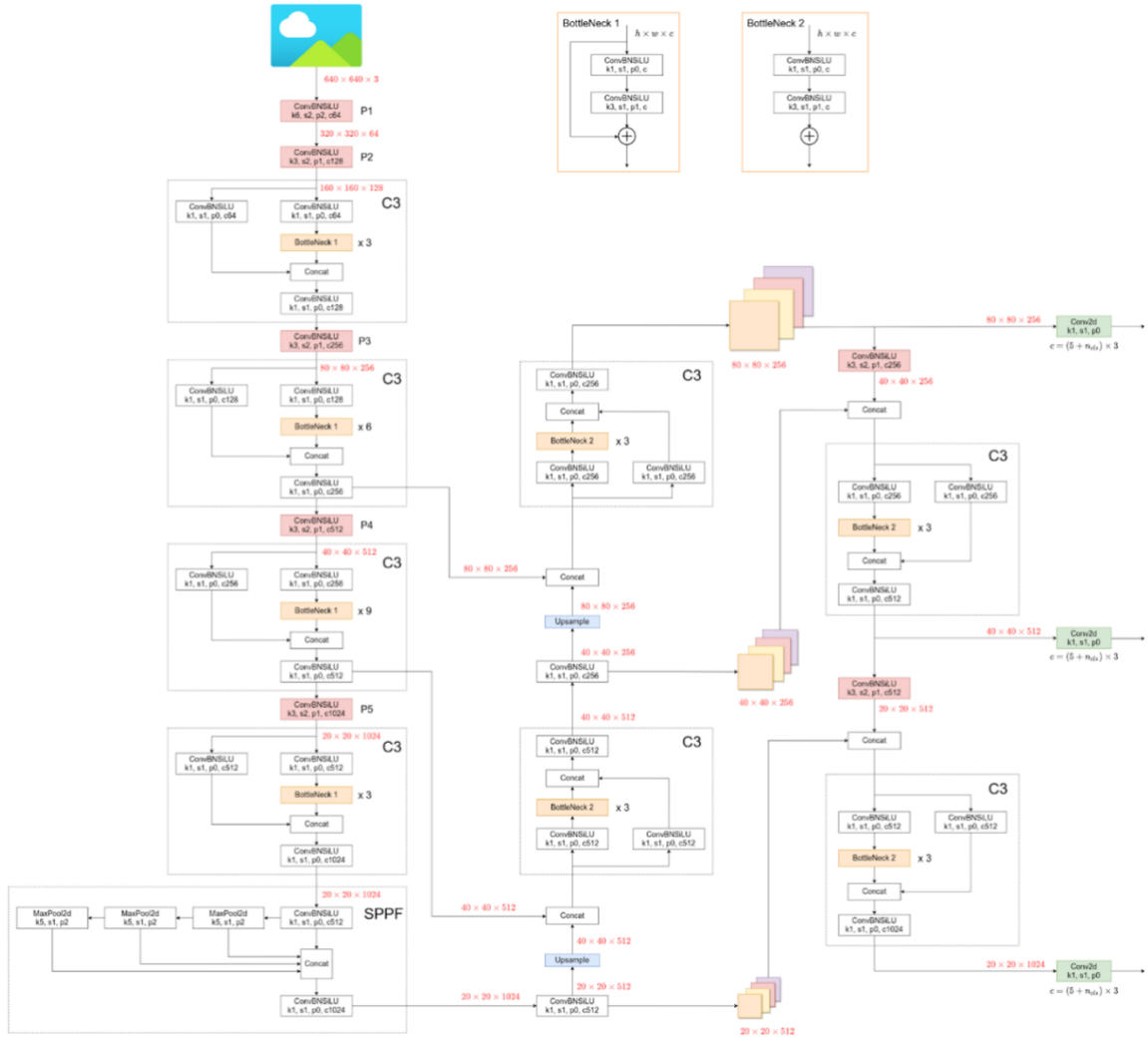
Figure 7 - YOLOv5 Structure

YOLOv5 uses the previous structure of YOLOv3 as its head but adds more layers in the backbone. It has 53 convolutional layers in the head [21], and 191 in total, however this is subject to change after training. As shown in *Figure 7(above)*, the entire network is feedforward, using a SELU activation function to allow better back propagation than other functions such as ReLU due to having a partial Edge of Chaos (EOC). [22]The pooling layers of the R-CNN can clearly be seen in the concatenation blocks in the diagram.

6.2.3 LINUX

For compiling, testing, and running our neural network, it was decided to use an Ubuntu Linux operating system rather than Windows or MacOS. This was for two main reasons. Firstly, it was a lot easier to install the required packages from a Linux terminal than it was on either windows or mac. YOLOv5 came with a requirements.txt file that could be run as a bash script from the terminal, however on another operating system all the separate packages had to be installed separately. The other reason was that a Linux system was very similar to the Raspberry Pi OS so it was assumed that if it could be implemented on Linux, then it would be a lot easier to implement it on a Raspberry Pi.

6.3 HARDWARE USED

6.3.1 LOGITECH WEBCAM

This webcam, seen in *Figure 8(below),* was used to provide the neural network with its input layer. It is a 720p 30fps webcam which suited the project very well. It has a higher resolution than the dataset image qualities (avg. 512x384), which we hope will help with identifying objects from a height rather than needing to be up close. The 30fps is perfect as any more would probably be too much for our neural network to handle and would result in massive frame drops and lag, probably reducing functionality of our solution.

Figure 8 - Logitech Webcam

6.3.2 RASPBERRY PI AND NEURAL COMPUTE STICK 2

As previously stated in section 6.2.3(above), after implementing the neural network successfully on a Linux system we assumed that it would be very similar on the Raspberry Pi, seen in *Figure 9(below).* Unfortunately, this was not the case. Our first problem occurred when we were unable to install OpenCV to gain full functionality of the Neural Compute Stick, *Figure 10(below)*. The purpose of the Neural Compute Stick is to supply extra processing power to the Pi. However, this was made impossible for us as there was a problem we didn't recognise when trying to install the software and we couldn't find a way of solving it. The problems continued when running the bash script from the Raspberry Pi terminal.

Figure 9 - RaspberryPi

Figure 10 - Neural Compute Stick 2

The python library PyTorch did not install as the installation form did not match the architecture of the Raspberry Pi. We tried installing PyTorch in many different ways from there including from the PyTorch website [23] and using .whl installation files but all resulted in the same result as before, even after choosing the .whl file that should have matched the architecture. Unfortunately, we had to abandon the idea of the Raspberry Pi implementation as PyTorch was required for YOLOv5 to work and we could not afford to spend any more time on it.

6.4   DATASET

The data set used was sourced mainly from the online website Kaggle. The set was titled 'Garbage Classification'(Vignesh) with 2532 images. This included 6 classes which were paper (584), glass (491), plastic (472), metal (400), cardboard (393) and non-descript trash (127). These photos featured the various pieces of rubbish against a white background. Items that could not be identified were removed from the data set. The sixth class 'Trash' was removed because it was not useful.

Next, two team members went around the green areas of Maynooth University to take photos of different types of rubbish to use for the data set in order to have a more accurate representation of how the rubbish would look in real life situations. In total, approximately 200 photos were added to the data set. The total size of the data set was 2665. The dataset consisted of 824 paper, 771 plastic, 498 glass, 453 metal and 430 cardboard annotations, an average of 1.11 annotations per image. Cardboard was slightly underrepresented, and the images of cardboard were generally very similar in the dataset so there was a slight concern that the neural network would struggle to identify it.

The task of annotating the data set was divided up amongst the team as it was a time-consuming job. The process took several days of continuous work, and after a while it was decided that annotating using a polyline, which was used to outline the object to be classified more precisely, was not necessary and did not aid the accuracy of the network. So instead, rectangles were used to isolate the object within the image. This significantly sped up the annotation process.

6.5   TRAINING THE NEURAL NETWORK

As mentioned above, the network was trained on Roboflow. The results were then exported to Google Colab to be converted into a usable .pt file for our own network. The training process used YOLOv5's training algorithm, a gradient descent training algorithm variant called Stochastic gradient descent. This means that the neural network trains itself using the gradient descent method on each individual image rather than grouping them [24]. This training method suited a deep neural network as there were many layers to propagate through, however the number of images in the dataset would affect the training time

immensely as each image is processed individually. By using gradient descent, the time taken to calculate and adjust weights in each layer was much less than using any other method.

The Neural Network was trained over the course of 150 epochs, where a single epoch is one full cycle of training, where every image in the dataset has been used.

In terms of the activation function, SELU was used, shown in *Figure 11(below)*. SELU is similar to RELU in that it is linear for all positive values, but it reduces the chances of dead neurons occurring by having a value other than 0 on the negative. For all negative values, x if given by an exponential function. [25]
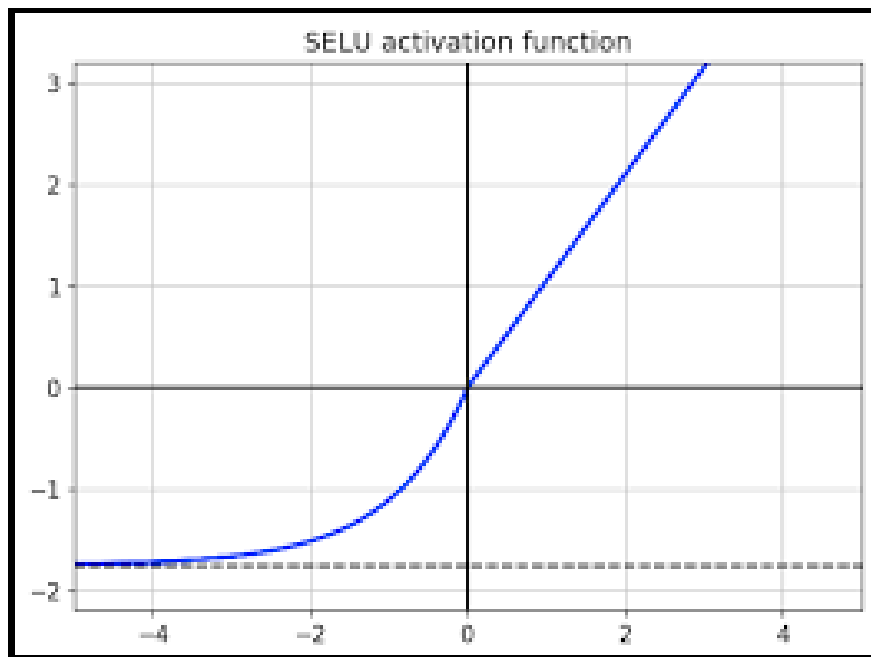
Figure 11 11 - Visual representation of the SELU activation function

6.6   VALIDATING THE NEURAL NETWORK

The Neural Network was validated in two main ways. The first way was using Roboflow's in-built validation and test method. After completing its training and validation cycles, it then tested itself using a portion of the dataset that was kept aside and not used for the training process. This returned a percentage accuracy for the neural network. This percentage was very different to the actual field accuracy when we ran the network from a laptop and used the webcam to search for rubbish in parks. There were two versions of the neural network, one purely using images found online and the other we added our own pictures along with more online images. There were around 1000 more images in the second version. To test the statistical significance of using this many more images, a 5% allowance was set and the two accuracies were calculated in an online statistical significance calculator [26]. The conversion rate was set to the percentage accuracy obtained from Roboflow's tests. [27]

25

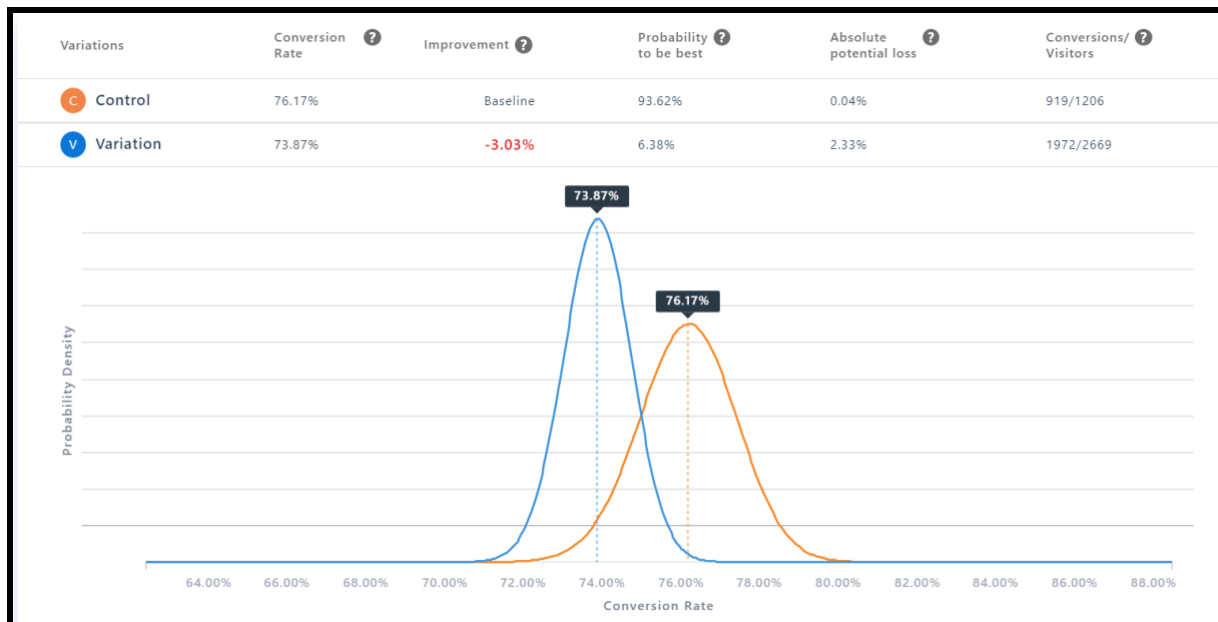| Variations	Conversion Rate	Improvement	Probability to be best	Absolute potential loss	Conversions/ Visitors
C Control	76.17%	Baseline	93.62%	0.04%	919/1206
V Variation	73.87%	-3.03%	6.38%	2.33%	1972/2669

Figure 12 12 - Statistical significance of less images in the Roboflow dataset

The results were not very good from the data shown in *Figure 12(above)*. A p-value of 0.06 was achieved which significantly disproved our hypothesis that adding our own images would increase the accuracy of the neural network.

This led us onto the second part of our testing and validation. It was proposed that this reduction in accuracy was possibly due to the fact that our own images were more likely to be something seen by our implementation and were very different to the rest of the images in the dataset. Therefore, it is not a surprise that Roboflow struggled a little bit more with less familiar backgrounds and lighting. It was decided to run both networks on a laptop and use the webcam to walk around parks trying to simulate a drone flight. The results were exactly as expected.

The original neural network performed very badly and struggled to identify anything. This is probably due to the fact that the online dataset was primarily up-close images on white backgrounds. From the images that it was able to identify, it was estimated to have an average of 22% certainty. The larger dataset was used with our own images and the results were much better. While it still struggled to identify objects from above head height, it was much better at recognising different types of rubbish at a much higher degree of certainty. The average degree of certainty was estimated to be around 50%. There are some examples of this in Appendix 9.1(below). The statistical significance of this was calculated using the same method as before.
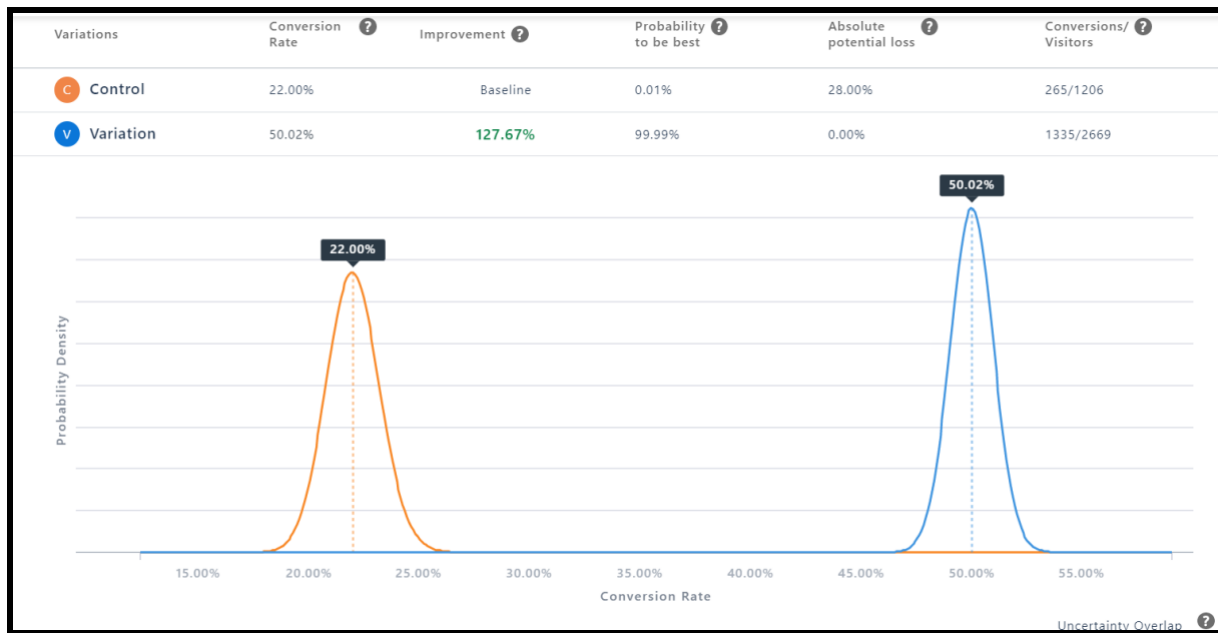
Variations	Conversion Rate	Improvement	Probability to be best	Absolute potential loss	Conversions/ Visitors
C Control	22.00%	Baseline	0.01%	28.00%	265/1206
V Variation	50.02%	127.67%	99.99%	0.00%	1335/2669

Figure 1313 - Statistical significance of more items in the dataset including our own images.

As shown in the graphic, *Figure 13(above),* there is a substantial increase in accuracy. The p-value is 0, and there is a 127.67% improvement in accuracy. This confirmed our hypothesis that our dataset made it harder for Roboflow to identify objects from the online dataset but identified a lot more objects during field testing. Considering there between 430 and 800 images per class, these results were deemed a success.

6.7   FUTURE PLANS

Given another few weeks for this project, there were 3 main areas we would have liked to develop further in this project. Not being able to implement the neural network onto a Raspberry Pi was disappointing, however given another week or two, the team is confident that it could have been implemented. We likely would have split into two teams to tackle the other two areas of interest.

The second development would have been to be able to stream live drone footage to either the Raspberry Pi or a laptop with the neural network running so that it would no longer be necessary to walk around a park. While it would have been possible to record the footage and then use that video file as the input to the neural network, it would have been preferred to implement a live stream.

The other development would have been to expand the dataset even further. A huge difference was evident in terms of certainty after adding our own images to the dataset, we are curious how accurate the neural network could get if there were two thousand realistic images for each class. This would have taken a lot of time to do so unfortunately, given some more time, the team would have expanded the dataset to contain entirely realistic images.

In terms of implementing the neural network onto the Raspberry Pi, the team is reasonably confident that the neural network created by would have been small enough to run comfortably on the hardware provided.

A similar project was found online [27], which also uses the small YOLOv5 architecture. The project used a Raspberry Pi 4, which is a newer generation than the one provided to the team. However, the project implemented a "base" Raspberry Pi 4, which has the same amount of RAM (1GB) as the Raspberry Pi 3. Another reason that the team is confident in the implementation is that this example model seems to classify more objects. It can be seen identifying people, buses, cars, handbags, etc. The model in question has been trained to identify 80 objects, whereas ours has been trained for 5. Despite this large number of objects, the Raspberry Pi 4, still managed an average of 1.6 FPS on the testing footage. It is expected that a neural network that classified significantly less objects would be much easier to run, resulting in superior performance.

7   CONCLUSIONS

The neural network operated nearly exactly as intended. While the drone option that the team intended on implementing did not come into effect, the neural network itself performed very well under both lab and field tests. Tests conducted to see whether adding more relevant images to the dataset affected its accuracy were very conclusive and were very similar to what the team expected from them. While the end goal was not reached, we were very close to the final implementation and given a few more weeks the team is confident we would have had it working. However, the team had a fully operational neural network working on a laptop that identified all five of the objects that it was designed to.

8   BIBLIOGRAPHY

[1] M. Gurucharan, "Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network," upGrad, 7 December 2020. [Online]. Available: https://www.upgrad.com/blog/basic-cnn-architecture/. [Accessed 7 March 2022].

[2] U. Nations, "THE 17 GOALS | Sustainable Development," United Nations, [Online]. Available: https://sdgs.un.org/goals. [Accessed 03 March 2022].

[3] WasteAid, "Waste and the Sustainable Development Goals," WasteAid, 28 June 2021. [Online]. Available: https://wasteaid.org/waste-sustainable-development-goals/). [Accessed 15 April 2022].

[4] R. Python, "Python AI: How to Build a Neural Network & Make Predictions," Real Python, 27 February 2021. [Online]. Available: https://realpython.com/python-ai-neural-network. [Accessed 5 April 2022].

[5] 3Blue1Brown, "Chapter 3, Deep learning," YouTube, 14 November 2017. [Online]. Available: https://www.youtube.com/watch?v=Ilg3gGewQ5U. [Accessed 02 March 2022].

[6] 3Blue1Brown, "Backpropagation calculus | Chapter 4, Deep learning," YouTube, 14 November 2017. [Online]. Available: https://www.youtube.com/watch?v=tIeHLnjs5U8. [Accessed 5 March 2022].

[7] U. o. T. CSC 321 Winter 2017: - Department of Computer Science, "CSC 321 Winter 2017: - Department of Computer Science, University of ...," University of Toronto, Toronto, 2022.

[8] J. Wu, "Introduction to Convolutional Neural Networks," LAMDA Group, Nanjing University, China, 2017.

[9] S. Saxena, "Introduction to Softmax for Neural Network," Analytics Vidhya, 5 April 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/04/introduction-to-softmax-for-neural-network/. [Accessed 3 March 2022].

[10] S. Amidi and A. Amidi, "Recurrent Neural Networks cheatsheet," Stanford, [Online]. Available: https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks. [Accessed 2 April 2022].

[11] S. Polamuri, "RECURRENT NEURAL NETWORK," Data Aspirant, 5 November 2020. [Online]. Available: https://dataaspirant.com/3-recurrent-neural-network/. [Accessed 4 March 2022].

[12] S. GHOURY, C. SUNGUR and A. DURDU, "Real-Time Diseases Detection of Grape and Grape Leaves using Faster R-CNN and SSD MobileNet Architectures," Konya Technical University, Konya/Turkey, 2019.

[13] A. Menzli, " Graph Neural Network and Some of GNN Applications: Everything You Need to Know," Neptune Blog, 6 December 2021. [Online]. Available: https://neptune.ai/blog/graph-neural-network-and-some-of-gnn-applications. [Accessed 15 March 2022].

[14] A. Quesada, "5 algorithms to train a neural network," Neural Designer, 2022. [Online]. Available:
https://www.neuraldesigner.com/blog/5_algorithms_to_train_a_neural_network.
[Accessed 22 March 2022].

[15] A. Ghaffari, H. Abdollahi , I. S. Bozchalooi, A. Dadgar and M. Rafiee-Tehrani, "Performance comparison of neural network training algorithms in modeling of bimodal drug delivery," Elsevier, Tehran, 2006.

[16] O. Osadcha and Z. Marszaek, "Comparison of steepest descent method and conjugate gradient method," Silesian University of Technology, Gliwice, 2017.

[17] O. Lange and L. Perez, "Traffic prediction with advanced Graph Neural Networks," Deep Mind, 3 September 2020. [Online]. Available: https://www.deepmind.com/blog/traffic-prediction-with-advanced-graph-neural-networks. [Accessed 17 April 2022].

[18] G. Jocher, "YOLOv5 in PyTorch > ONNX > CoreML > TFLite," GitHub, 2022. [Online]. Available: https://github.com/ultralytics/yolov5. [Accessed 5 April 2022].

[19] A. Garg, "How to Use Yolo v5 Object Detection Algorithm for Custom Object Detection," Analytics, 2021.

[20] M. Rajput, "YOLO V5—Explained and Demystified," Towards AI, 1 July 2020. [Online]. Available: https://towardsai.net/p/computer-vision/yolo-v5%E2%80%8A-%E2%80%8Aexplained-and-demystified. [Accessed 27 March 2022].

[21] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," University of Washington, Washington, 2018.

[22] S. Hayou, S. H. Doucet and J. Rousseau, "On the Impact of the Activation Function on Deep Neural Networks Training," 2019.

[23] PyTorch, "PyTorch," PyTorch, [Online]. Available: https://pytorch.org/. [Accessed 20 March 2022].

[24] S. Ruder, "An overview of gradient descent optimization algorithms," Aylien Ltd, Dublin, 2017.

[25] Z. Huang, T. Ng, L. Liu, H. Mason, X. Zhuang and D. Liu, "SNDCNN: SELF-NORMALIZING DEEP CNNs WITH SCALED EXPONENTIAL LINEAR UNITS FOR SPEECH RECOGNITION," Apple Inc, Cupertino, 2020.

[26] VWO, "A/B Split Test Significance Calculator," VWO, [Online]. Available: https://vwo.com/tools/ab-test-significance-calculator/. [Accessed 28 April 2022].

[27] Qengineering, "YoloV5 for a bare Raspberry Pi 4," GitHub, 11 January 2022. [Online]. Available: https://github.com/Qengineering/YoloV5-ncnn-Raspberry-Pi-4. [Accessed 25 March 2022].

9 APPENDIX

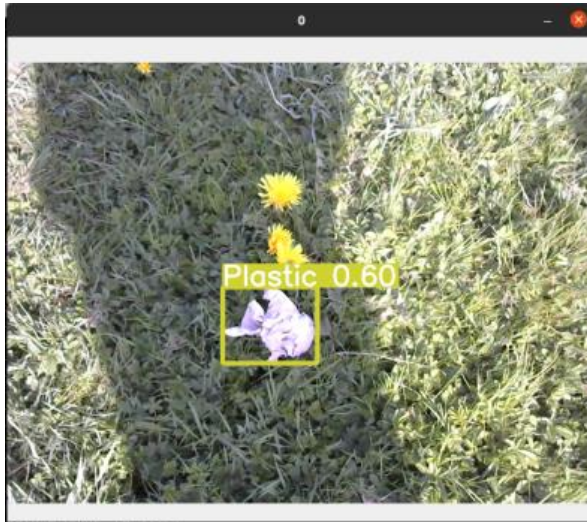## 9.1 OBJECT DETECTION IMAGES

Figure 14 14 - Plastic detected
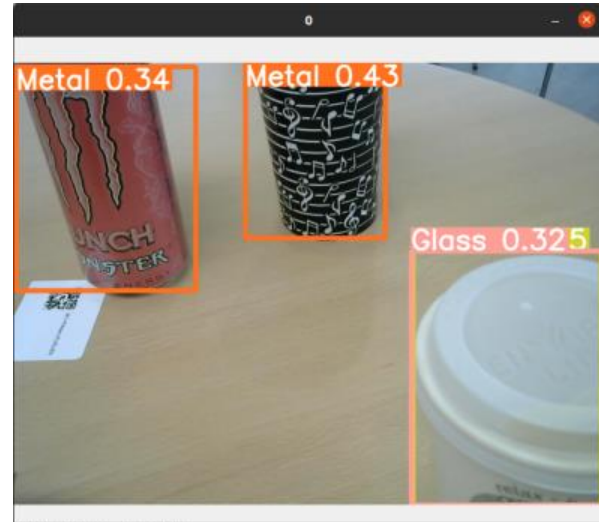

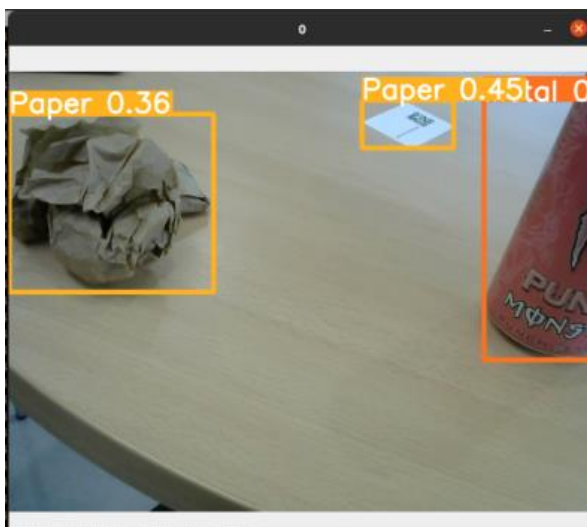*Figure 15 15 - Metal and Plastic detected, Glass incorrect*

Figure 16 16 - Paper and Metal detected

Figure 17 17 - Paper and Plastic detected
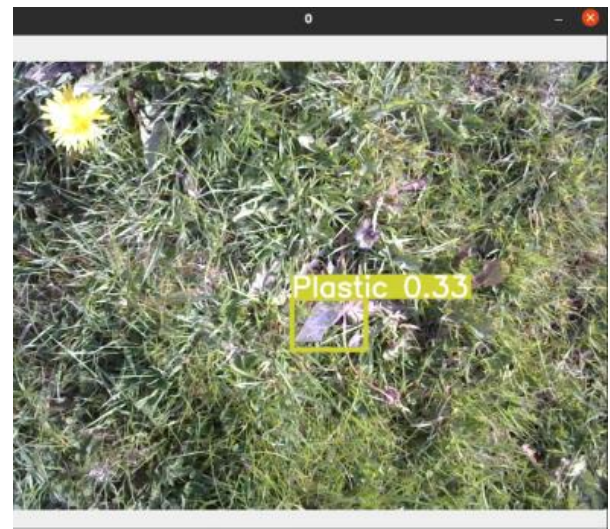
Figure 18 18 - Glass detected

Figure 19 19 - Plastic detected
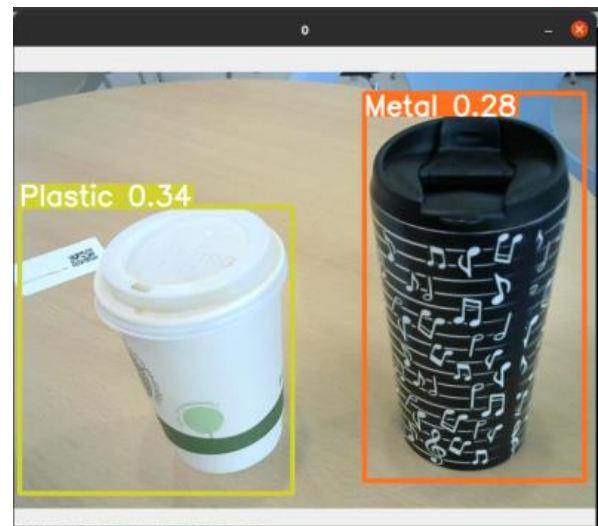

Figure 20 20 - Plastic detected
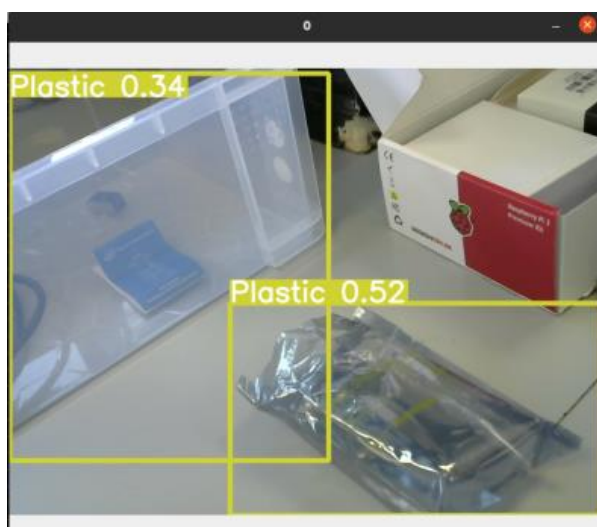

Figure 21 21 - Plastic and Metal detected

Figure 22 22 - Plastic detected

Figure 23 23 - Paper detected
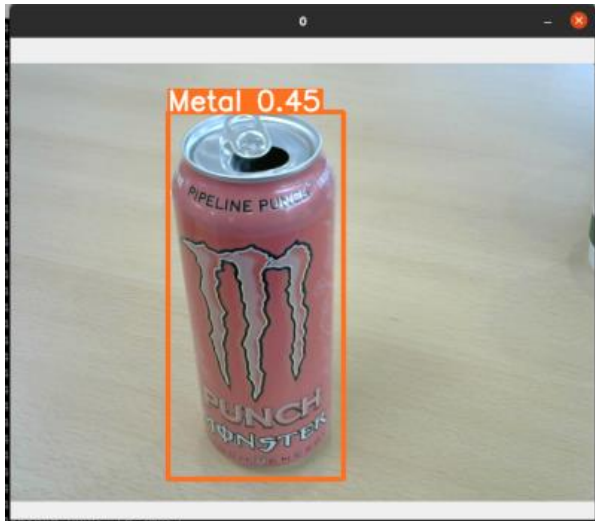
Figure 24 24 - Metal detected
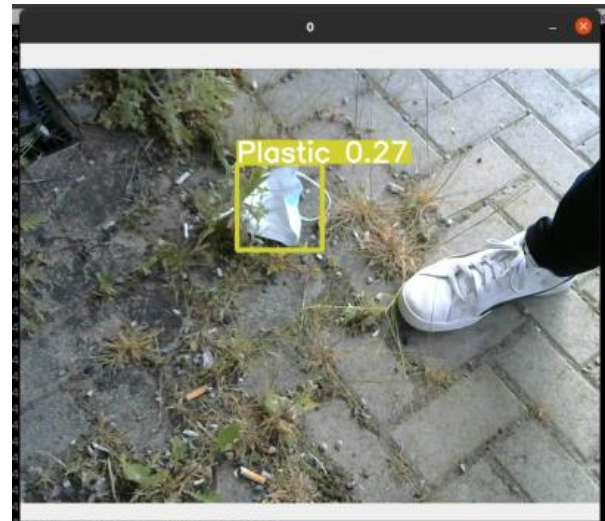
Figure 25 25 - Plastic detected

9.2 RISK ASSESSMENT

NAME AND STUDENT NUMBER: A. Conal Hughes -20365616 B. Stephen Gallagher - 20470574 C. Gerard McIntyre - 20332566 D. Amy O'Mara - 20386631	PROJECT NAME: EE297 Project Group 3
SUPERVISOR: Dr. John Dooley	PROJECT LOCATION: E2.01B E2.08 E1.01A Maynooth Library

BRIEF DESCRIPTION OF PROJECT: Your team must implement a technical solution using neural networks to address a problem caused by waste.

Hazards, Risk [High(H) Medium (M) Low (L)], and Control Measures

HAZARD	Risk	Controls
Eye strain	M	• Reduce time spent in front of screens. • Take regular screen time breaks
Inadequate breaks	L	• Take breaks when needed, rather than work long hours and reduce work performance.
Electrical Wiring	H	• Tidy wires around labs • Make sure sockets are switched off if not in use

Identified risks should be discussed with your supervisor and a safe system of work agreed. A more in-depth risk assessment may be required after initial review. **Do not proceed until form is signed off.**

Further Controls Required

None
Note: that this is not currently and exhaustive list of potential risks. As risks are identified they will be added to this document and assessed.

SIGNATURE OF STUDENT: _____Conal Hughes_____   DATE: ___04/03/2022___

SIGNATURE OF STUDENT: _____Stephen Gallagher._____   DATE: ___04/03/2022___

SIGNATURE OF STUDENT: _____   DATE: ___04/03/2022___

SIGNATURE OF STUDENT: _____   DATE: ___04/03/2022___

SIGNATURE OF SUPERVISOR: _____   DATE: _____

DEPT HEALTH AND SAFETY OFFICER: _____   DATE: _____

9.3 REFLECTIVE JOURNAL 1

Gerard:

The team that I am in, which is Team 3(Amy, Stephen, and Conal) I felt worked together very well from the start as we all were friendly and knew each other beforehand. There were little to no quarrels, and we all knew our strengths and weaknesses towards the project. We started learning about neural networks and how they behave. We found this difficult as this was new territory to us all, but we also didn't know how to code in python either. We started learning how this new information can help us develop our own ideas. I enjoyed being in our team as learning the information was manageable because we explained and helped each other understand neural networks and the coding.

I found it difficult to connect our idea in terms of the neural network and how the code is implemented from theory to code. Since we had not studied python before this was even more difficult. I found doing the presentation stressful trying to remember all the new information learned but it was a good way to cement our idea and to make sure all team members have the same knowledge. For the presentation, I worked on making the slides and inputting the information and points to be made. If I was to start this project again, I don't think I would change a lot except just focusing on reading and understanding the material as early as possible. Overall, I am happy and content with our group's progression towards this module.

Conal:

For the first few weeks of this project, we struggled a lot. Most of the group were unavailable due to illness all at different times so it was very difficult to build any cohesion in the group. However, once everyone was back and well, the project started to come together very well. It was very difficult to concentrate in a lab setting, but we discovered if we met in one of the library group study rooms, or an empty room with access to a whiteboard, we could concentrate much better and that having something central to write and draw on helped enormously. Discovering these details was central to our progression from then on.

We felt that giving roles would put too much pressure on team members, so the work was divided evenly and assigned to people we thought had the best prior understanding of the topic. Personally, I researched into the structure and training of a neural network, how it worked, what to look out for etc. This was difficult as we had never used one before, I only had a very passable knowledge of them. I enjoyed working on this topic, it is a very interesting field, and I am looking forward to implementing our own neural network. We have a very progressible idea, so while we hope to be able to achieve our end goal, it might not be possible so we will always have previous stages to fall back on. I think we did very well to be at the stage we're at, given the circumstances and I look forward to building on our work in the future.

Amy:

I am part of team 3 for the EE297 project, and I have enjoyed working with the group so far. I was in charge of taking the meeting minutes. In terms of delegating other roles, we shared much of the research and write up, which upon reflection possibly delayed our progress. With no concrete roles given out it was difficult to lay down goals for the future and ensure that certain tasks were completed in a timely fashion. It also made it difficult to take care of multiple tasks at once, because everyone was working on the same things. When we began our investigation into the project, I was initially overwhelmed at the prospect of finding an interesting solution to a waste problem and learning how to implement a neural network to solve it. There seemed to be such a large number of options that it was hard to settle on one idea. Although I feel that our idea is strong. It is outside of the box enough to stand out but hopefully not convoluted enough to be unachievable.

We definitely found it difficult having team meeting in busy and loud environments where the chance for distraction was great. We decided instead to have meetings outside of the lab times and book rooms in the library. This worked very well for us, and we were much more productive there. My hope for the future is that we execute our idea successfully. I am proud of where our team is currently and feel that our progress is promising. We have shown ourselves that when we put our heads down and focus, we can produce quality work and I am looking forward to what comes next.

Stephen:

Overall, I have been very happy working with Conal, Gerard and Amy. Since we formed the team ourselves, we already knew each other quite well, resulting in team meetings with strong communication from the start.

My teammates are all very well rounded in terms of ability which meant that we could all chip in when learning new topics. This also meant no one was left behind. We also made sure everyone was up to speed by teaching each other the information we had learned as we learned it. Teaching teammates also helped solidify our understanding of the content. I personally did research into Neural Networks, particularly the different activation functions.

In terms of content, I struggled with learning about neural networks, as we had no prior experience. It is also a very expansive subject, with many small nuances to learn about. Despite the difficulty, I did enjoy learning about neural networks, as well as learning a new programming language, Python. I also enjoyed brainstorming with the group. We did this by throwing ideas at each other, no matter how ridiculous, and trying to make something plausible from it. In terms of what I would do differently next time, I would allocate research subjects more effectively, as on occasion we would go off and research subjects at the same time, when only one person needed to.

9.4 Reflective Journal 2

Gerard:

The team that I am in, which is Team 3(Amy, Stephen, and Conal) I felt worked together very well from the start as we all were friendly and knew each other beforehand. There were little to no quarrels, and we all knew our strengths and weaknesses regarding the project. We changed our project from identifying any objects in a park that was not the colour green, to identifying different common types of waste. The categories included Metal, Plastic, Paper, Cardboard, and Glass. We focused on getting the pictures for our dataset and found an annotating software called Roboflow. In Roboflow we could annotate our pictures into the different groups with boundary boxes.

We collected over 2500+ pictures and divided the annotating job between the four of us. Roboflow gave us an accuracy of 73% for the neural network. I enjoyed training the dataset and seeing all the annotating work when we achieved a high accuracy score. It was satisfying to see that because we focused on the start of the theory of the neural network, we could now see the practical side and appreciate it. We then uploaded our code with the weights given by Roboflow and uploaded it to Conal's laptop. We now could do a live stream from his camera and boundary boxes appeared when it identified waste in the stream. We will now try and implement the code into raspberry pi and finish our report for this project. Overall, I am happy and content with our group's progression toward this module.

Conal:

Unfortunately, we realised that our initial idea was too complex and specific to complete in the designated time frame, so we had to change our idea. We decided to implement a neural network to recognise common types of rubbish on the ground in parks or on campus. We found a dataset and annotated all 2500+ images ourselves. We also took our own pictures of rubbish around campus and annotated that too and added it to the dataset. We trained our neural network and achieved an accuracy of 73% on Roboflow. I enjoyed learning about how neural networks work and what is required to train it, and I'm sure I'd enjoy it more now that I know the fundamentals.

We found programming it very difficult as there was no group member that had ever programmed in python before, so to combat this we used an example of a CNN online and adapted it to suit our needs. We used the weights obtained from Roboflow and ran the network from a Linux terminal and it seemed to work. Further testing is needed, and we now need to implement it onto a Raspberry Pi, but it seems that our project is working as required. We felt our teamwork in completing the annotation and figuring out the software was really good and made learning about the project a lot easier. If we were doing this again, we would try and write more of our own code if we knew more python. Overall, however I think this has been a great success.

Amy:

Our initial idea that we were working on was a neural network that identified green spaces and "safe" objects, in order to detect if there was some rubbish in the area. We worked on this for a few weeks before deciding that it was not going to be an achievable solution and there were more straightforward ways to achieve the same end result. Instead, we begin working on a neural network that recognises common types of rubbish in parks and green spaces. We made a data set of plastic, cardboard, glass, paper and metal and fed this into an online software called Roboflow. After training our network that contained over 2500 images, we achieved an accuracy of 73%.

At any stage that decisions had to be made about the idea we were using, we made sure that everyone on the team was on board and could give their opinion, this led to no conflicts within our group and things ran smoothly. We are now going to be finetuning our network and recording all the work we've done in our report.

Stephen:

Since our last reflective journal, we decided that our initial idea of detecting "safe" objects was too complex. Instead, we decided to implement a rubbish detecting Neural Network. We found Roboflow, a piece of software that we used to annotate over 2500 images. We got a set of weights from Roboflow and used them in a CNN we found online. This yielded a working neural network that could not only detect rubbish, but also differentiate which kind it was, e.g., plastic, paper, glass, cardboard. Planning ahead, we would like to implement this working Neural Network onto the Raspberry Pi provided. The workload in terms of annotating was spread evenly amongst the team, and we feel we worked well together. In the future we would like to become more comfortable using python and be able to code the CNN from scratch.

9.5   REFLECTIVE JOURNAL 3

Gerard:

I feel that the project in an overall sense was successful and enjoyable to complete. We completed the task of getting a physical working neural network model to help solve one of the 17 sustainable development goals. During the final weeks leading up to the end of the semester, we focused on the report on how the theory and the technical background of a neural network connected to our working model. The writing of the report is going well with us finishing and editing parts of it still. It's interesting to see the final report nearly finished and how our theory from the beginning is being used to understand the results that are being fed back to us by our own model.

Given more time we could have tried to complete our stream from a drone and see if it's camera would identify any "waste" objects using our neural network model from a height/range of 2 metres. This could have been done by using the Wi-Fi on the drone to live stream the video back to a laptop with our neural network model on it. If I was to change one thing about our project, it would be to create a larger dataset as the confidence level of the neural network would have increased with the larger dataset. Everyone worked really well together and helped each other in areas that were not familiar to us. I learned a lot doing this project from increasing my communication skills with my teammates, and to increasing my knowledge about neural networks.

Conal:

Overall, I think that our project was quite successful. While we didn't achieve our end goal of full drone implementation, it would have been the next step in our process, given another week or two. We got our Neural Network running perfectly on a laptop and very nearly on the RaspberryPi, identifying target objects from a range of about 2 metres while in motion. This couldn't have been achieved without the fantastic cohesion amongst our team. I felt that we communicated very well, and everyone made every attempt possible to be present for team meetings and tests. We tried to include everyone in decision making and made sure that no one was above anyone else in any way. We tried to keep everyone up to date with what we were doing so that one person didn't feel like they were doing everything and so that no one fell behind.

We felt that it was important that we worked as a team in everything that we did. This proved to be very successful as we completed nearly everything we had hoped to accomplish, and no one fell behind. I enjoyed working on this project and working with this group. I felt that we learned a lot about a topic that we had never encountered before, and that we completed what was asked of us to the best of our abilities. Given another week or two, we would have hoped to have implemented this same program on a drone and streamed live footage to a laptop with the YOLOv5 program running. Another future amendment would be to expand our dataset, as although it was operational, it was relatively small and had a lot of room for improvement in object detection confidence.

Amy:

In terms of team 3's final progress I am more than pleased with where we have gotten. As a group, we overcame obstacles and issues, we understood and recognised when we should re-evaluate our ideas when they weren't working out. We have gotten a working final solution that we are proud of. I have enjoyed working with my team, we communicated well, everyone was willing to put in the time and effort to complete the various deadlines, and we all made sure that we were onboard with the technical aspects of the project so that nobody was confused or left behind in the understanding.

Given some more time, and possibly without the added stress of other modules and exams, we would have liked to have implemented our idea into drone footage and expand the range of our neural network, because as of now, while it can detect objects consistently well, we can only test it while using a laptop and webcam. But we may still achieve this in our own time outside of college. I feel that I have learned a lot from this project, not just knowledge of neural networks and the intricacies of which I've never come across before, but I have also expanded my teamwork, technical writing, presenting and public speaking skills.

Stephen:

Overall, I am very happy with the work done throughout this module. I enjoyed working with my team members and felt as if we complimented each other's strengths well. We overcame obstacles well, specifically the change of plan, from a neural network that detects "safe objects" to one that detects 5 different types of rubbish. Throughout the project, the workload was spread amongst the team well and everyone was kept up to speed with everyone else's progress. I am very happy with the amount I learned, especially considering we knew very little before this project. Given more time, we would have attempted to get the neural network running on the Raspberry Pi, as we were very close to getting it to work. We would also have tried either implementing the neural network onto a drone, or simply feeding drone footage into it to see how the movement and height affected its performance.