Faculty of Engineering
University of the West Indies
Mona

AI Navigation System for the Blind

By

Student name: Jordan Madden
Student ID: 620118442

A Final Year Project Deliverable Submitted to the Faculty of Engineering
in partial fulfillment of the requirements for the degree of
Bachelor of Science in Engineering

Project Advisors: Dr. Kolapo Alli and Mr. Lindon Falconer

November 14, 2020

AI Navigation System for the Blind

Jordan Madden, Kolapo Alli, Lindon Falconer,
Faculty of Engineering
University of the West Indies, Mona, Jamaica
jordanmadden285@gmail.com, kolapo.alli@uwimona.edu.jm, lindon.falconer@uwimona.edu.jm

Abstract

A blind person normally uses a cane to navigate from one place to another , and while the cane is helpful, it is very limited in terms of the useful information that it provides to the user. This paper presents the work done so far in the development of a device that seeks to address the limitation of the standard cane. In accordance with this objective, I have implemented a functional object detection system based on the YOLOv3 detector from the YOLO family of object detectors and I have been able to generate the voice commands that will be used to direct the user.

Index Terms: Deep Learning, Computer Vision, Object Detection, Depth Estimation, Navigation

1 Introduction

The world as we know it was constructed/developed for the average person to be able to operate in comfortably, but blind people are far from average. As they cannot see the task of navigation, which most sighted people take for granted, becomes very dangerous. They currently rely on a cane to help them navigate from one area to another. This cane is typically used to allow the blind person to feel the area in front of them for obstacles or irregularities in their desired path. In its current form the cane is very limited in terms of the useful information that it provides to the user.

Out of necessity, many blind people have become rather proficient users of the cane. With the rising popularity of deep learning and its applications in fields such as computer vision, I believe that it is now possible to develop a device that aids the blind in their navigation from place to place. If this device is capable of detecting the obstacles around it, accurately estimating the depth of those obstacles and relaying that information back to the user, it could potentially be a major game changer to the way that the blind navigates in their environments.

1.1 Objectives

The aim of this project is to develop a device that uses deep learning and artificial intelligence to assist a blind person in navigating an environment. The device should be able to do the following:

- Identify objects/obstacles in the path
- Estimate the distance from an object
- Provide auditory and/or tactile feedback to alert the user when the path is clear/safe to traverse
- Accept voice commands (stretch objective)
- Conduct facial recognition (stretch objective)

1.2 Literature Review

Two major themes of this project are object detection and depth estimation. As such, the brunt of my research thus far has been directed towards these 2 topics. This section will document the results of that research.

Object Detection:

When considering object detectors, I found that there are 3 algorithms that are typically used for this task. They are the MobileNet-SSD[2][6], the You Only Look Once(YOLO) algorithm [3] and the Faster R-CNN algorithm [4]. Each of these approaches to object detection will be discussed.

The SSD algorithm was originally implemented by researchers at Google and the University of Michigan, Ann-Arbor [2]. This neural network discretizes the output space into a set of default bounding boxes and generates scores for the presence of an object in each box at prediction time, before adjusting the box to match the shape of the object better. This algorithm achieves a mean average precision (mAP) of 74.3 on images. The MobileNet algorithm [6] was proposed for use inside of an object detection pipeline on resource constrained devices. Instead of using traditional convolutions as seen the well-known ResNet architecture, it uses depthwise separable convolutions. This allows the network to use less parameters while achieving similar results. While it was noted that there was some sacrifice in the network accuracy, this network architecture was seen to be much more resource efficient. When the MobileNet and SSD architectures are combined, the result is an efficient deep learning object detector.

In their paper [3], Redmond et al. proposed a single stage detection algorithm that treats object detection as a regression problem rather than a classification problem (as was previously done). The algorithm takes as its input an image and learns the coordinates of the bounding boxes as well as the class label probabilities. The algorithm proposed in [3] is the third iteration in the YOLO family of object detectors known as YOLOv3. While it is noticeably larger than its predecessors, it is still a fast algorithm, capable of performing a forward pass on a 320x320 pixel input image in 22ms. This speed comes while achieving a mAP score of 28.2.

The Faster R-CNN [4] network build upon the previous work by the same author by introducing a region proposal network(RPN). As indicated by its name, the RPN essentially tells the network where to look by predicting the object boundaries and objectness score at each position. The network then examines the proposed regions and classifies the objects that are located in each region. Previously, the R-CNN network was a 2 stage object detector, but the addition of the RPN into the network architecture negates the need for a 2 stage object detector and allows the Faster R-CNN to function as an end to end object detection network. The network was able to achieve a moderate speed of 5 FPS on a GPU and was seen to be extremely accurate.

Depth Estimation:

The task of depth estimation is seen as fundamental in many applications such as Augmented Reality and Scene Understanding. Prior to the work of Alhashim et al. [1], solutions to the problem of depth estimation either relied on geometric approaches where there was an object of known size in the frame of the image, or on deep learning based algorithms that produced blurry, and often incomlete approximations to the depth of the scene. Neither of these approaches were truly useful in dynammic real-time applications. To fix this, Alhashim et al. proposed a convolutional neural network architechture that produced a high quality depth map of its environment. They used a standard encoder-decoder architecture (as seen in figure 1) and applied transfer learning to achieve state of the art results on the KITTI dataset, the NYU Depth v2 dataset as well as their own Unreal 1-K dataset.

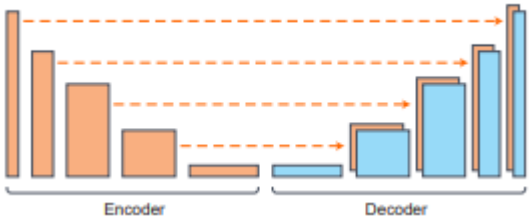

Figure 1. Illustration showing the network architecture of the neural network described in [1]

The network above uses a DenseNet-169 model with weights pre-trained on ImageNet as its encoder then its decoder consists of up-sampling layers, convolutional layers and the skip connections that are concatenated with the decoder layers. Even with this relatively simple/standard network architecture, the authors were able to obtain record breaking root mean square error and comparable average relative errors and average errors to other depth estimation techniques. The code and results for this project can be found at the authors github account here.

2 Design Criteria and Realistic Constraints

2.1 Scheduling Concerns and Time Limitation

At the moment, there are no issues regarding the schedule of the project. I have been up to date on all of the deliverables as outlined by the Gantt Chart. Thus, the timeline for the completion of the project is thus far unaffected. A few times, the meeting with my primary project supervisor had to be put off due to scheduling issues, so going forward an emphasis will be placed on meeting with him.

2.2 Realistic Design Constraints

2.2.1 Economic Constraints

This components of this project are estimated to cost $232 usd, but in the future as the system software improves, it may reach to a point where it negates the need for an expensive component and a cheaper one may be used.

2.2.2 Health and Safety Constraints

As this device would be targeted at members of the visually impaired community, their safety is paramount. Thus, the device must be rigorously tested before it is approved for human use.

3 Standards

Standards and regulations are required by law to be observed within the development process of any device. These standards provide guidelines as to the scope of the design. Without these rules in place, users may be put at risk and this is unacceptable. The following standards and regulations are those that will be adhered to in the completion of this project.

3.1 Standards

- As serial communication is a vital part of this project, the signal between the atmega328 and the raspberry pi 4 must adhere to the UART communication standard.

3.2 Regulations

- As the system is using artificial intelligence, it must adhere to the ISO/IEC AWI TR 24027 which regulates the bias in AI production systems and in AI decision making.

4 Methodology

In order to explain the approach that I have taken to the problem, I will examine the problem in 4 stages and provide a breakdown of my reasoning at each stage.

- High Level Circuit Block Diagram
- Schematic Design of the Circuit
- PCB Design of the Circuit
- Software Flowchart
- Proposed Parts List
- Enclosure Design
- Gantt Chart
- Software Design Process

4.1 High-level circuit block diagram.

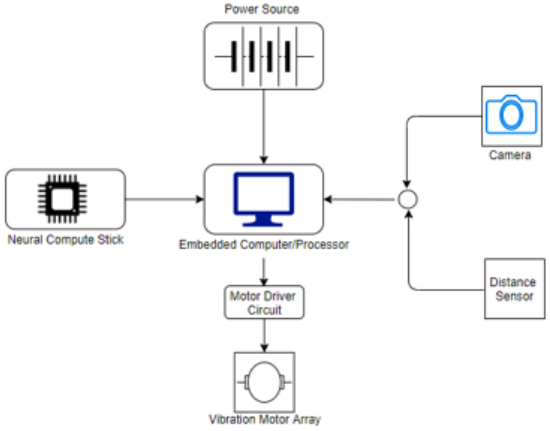

Figure 2. The illustration above shows a high-level overview of the various components in the system

The system is meant to consist of a few modules. The camera/depth sensor will send image data to the processor where the relevant deep learning algorithms will be run on the input image and the output of the system will be sent to the vibration motor array so as to direct the user (they will be worn around the waist/torso of the user with motors placed at different points around the user).

4.2 High-Level Software Design (flow chart)

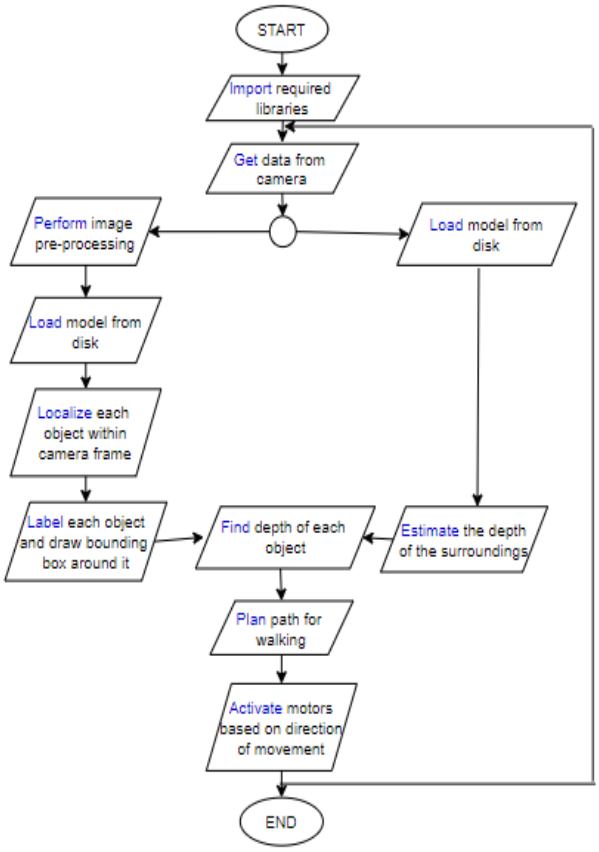

Figure 3. The flowchart above presents an overview of the proposed software for the device

In the flowchart, it can be seen that the system is using deep learning to perform two major computer vision tasks, namely object detection and depth estimation. The depth of each object will then be inferred and based on that the system will inform the user whether or not the direction that they are attempting to travel in is suitable for travel.

4.3 Schematic Diagram of Electronic Circuit

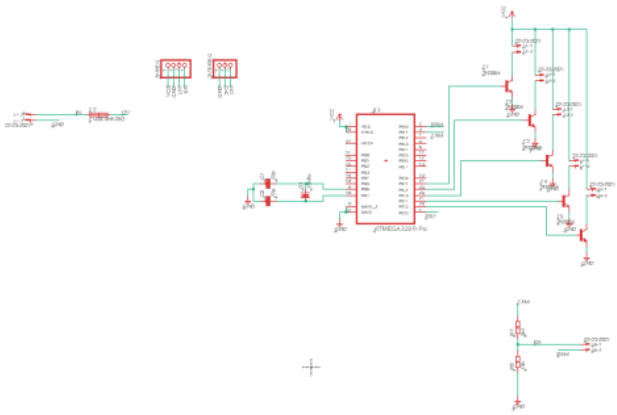

Figure 4. The diagram above presents the proposed circuit schematic for the device

The schematic in figure 4 is an extension of the high-level bock diagram. It shows a circuit that is controlled by an Atmel Atmega328p. This chip is connected to 5 terminal blocks by which can trigger the vibration motors. The chip also has serial communication connections to a raspberry pi 4 so that it can receive the data that it needs to trigger the vibration motors. There are 2 pin headers, from which the power will be supplied, and the power will run through a fuse before it is fed into a 5V voltage regulator chip and a 3.3V voltage regulator chip. The 3.3V chip will power the vibration motors while the 5V chip will power everything else on the board.

4.4 PCB Design of Electronic Circuit

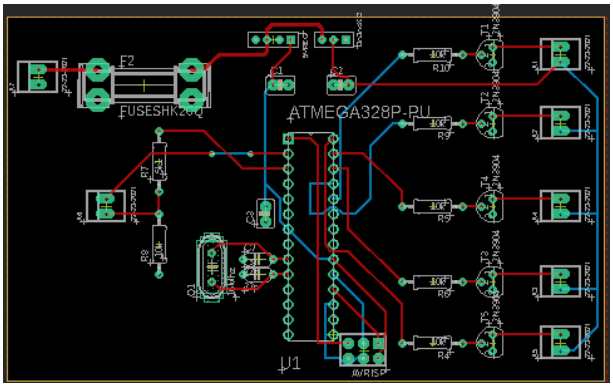

Figure 5. The diagram above presents the proposed printed circuit board(PCB) for the device

The PCB design in figure 5 is an extension of the design proposed in figure 4. It shows the electrical connections between each of the components.

NB. The ground connections are not shown above as they are all connected to a common plane.

4.5 Parts List

Component	Unit Price ($ USD)	Amount	Price ($ USD)
Atmel Atmega328P	3.99	1	3.99
Intel Realsense D415 Camera	159	1	159
Vibration Motors	-	15	13.99
2N2222A BJT	-	15	14.99
5V Voltage Regulator	8	1	8
3.3V Voltage Regulator	-	6	6.75
1A Fuse 5x20mm	7.49	1	7.46
Fuse Holder 5x20mm	3.99	1	3.99
Latching Push Button	7.99	1	7.99
3.5mm Terminal Block	0.17	10	1.67
16 MHz Crystal Oscillator	3.99	1	3.99
Total Price ($ USD)		**231.82**	

Figure 6. The table above shows the proposed parts list for the device

4.6 Enclosure Design

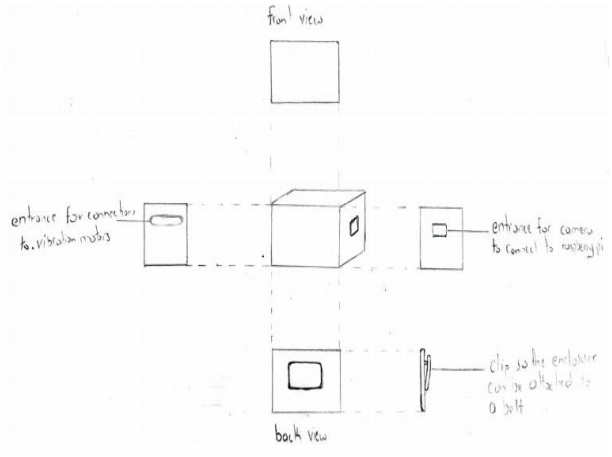

Figure 7. The drawing above shows an initial sketch of the enclosure for the device

This is supposed to be a portable system that a blind person will be able to walk around with. The idea behind the enclosure sketch as shown above is that it will contain the processor (Raspberry Pi 4) as well as the circuit shown in figure 4 and the power supply. The box will be able to be clipped onto the belt of the user and wires will run from the box to the external camera and vibration motors.

4.7 Gantt Chart

A Gantt Chart is a project management tool that allows you to keep track of the progress that you have made in the project by setting timelines for the completion of each task. This is a year-long project and as such, the associated Gantt Chart is too large to fit inside this document however, a link to the Gantt Chart can be found in the appendix. This section will provide an overview of the current Gantt Chart.

Task	Due Date
Get Depth Data from the Camera	31/12/20
Generate Depth Map of the Environment	2/2/21
Integrate Speech Synthesis	10/12/20
Develop embedded software	31/12/20
Order PCB	25/1/21
Test circuit on breadboard	25/2/21
Solder components on PCB	10/3/21
3D print enclosure	15/4/21
Poster draft	21/5/21
Final Report draft	2/6/21
Final Report	3/6/21
Project Demonstration	4/6/21

4.8 Software Design Process

The previous sub-sections have focused on my implementation of the various deliverables that were required by all students for this course (ECNG3020). This sub-section will seek to explain the way that I have gone about completing this project thus far and why I have chosen the methods that I have.

As mentioned in section 1.2 (Literature Review), two major themes in this project are object detection and depth estimation. For this reason, the majority of my work thus far has been directed towards object detection. I plan to use a Raspberry Pi 4 as the processor for these algorithms, thus it was very important to me that the algorithms that I choose be both fast and resource efficient. For that reason, I looked into working with an implementation of the YOLOv3 algorithm and the MobileNet-SSD algorithm. I first looked into the MobileNet-SSD algorithm as it was developed specifically for embedded devices. It achieves increased efficiency by using Depthwise-Separable Convolutions as its primary feature extraction operation instead of normal Convolutions. A convolution is an operation by which a portion of an input image is multiplied (matrix multiplication) by a kernel/filter and the output is the product of that multiplication. The operation of a traditional convolution is shown below in figure 8.

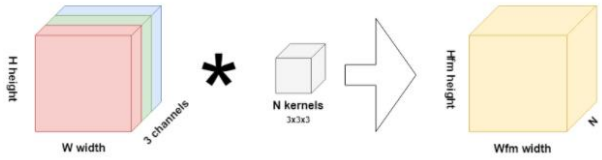

Figure 8. Illustration showing the behavior of the normal convolution operation

These convolutional operations are very resource intensive as the kernel is convolved with each of the RGB input channels of the input image. To make this operation more efficient, the authors of [2] proposed the use of Depthwise-Separable Convolutions. The operation of these are shown below.

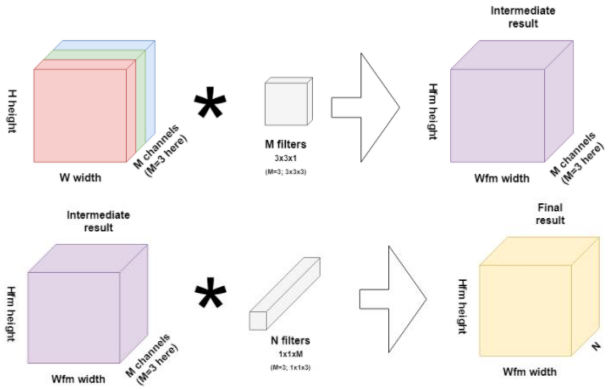

Figure 9. Illustration showing the behavior of the depthwise-separable convolution operation

The Depthwise-Separable Convolution splits the convolution into a depthwise operation and a pointwise operation and this allows similar results to be achieved with significantly less operations.

I implemented an object detection system with the help of Adrian Rosebrock and his PyImageSearch blog [5]. I used a pretrained model and found that it worked well, however it was not able to recognize enough objects for it to be useful in a real-world setting. Thus, I began to look at using the YOLOv3 object detction algorithm. This algorithm was reported to be faster than the Mobilenet-SSD algorithm and it achieves this in part by using many 1x1 convolutions to reduce the number of parameters. For direct location prediction, the algorithm uses a logistic activation function. This activation function is shown in figure 10 below.

$$\phi(z) = \frac{1}{1 + e^{-z}}$$

Figure 10. Sigmoid activation function used in logistic regression.

I was able to implement an object detector based on YOLOv3 with the help of Adrian Rosebrock and his PyImageSearch blog [7].

5 Results

As was mentioned previously, I made use of a pretrained YOLOv3 object detector whose weights were finalized after being trained on the COCO dataset. COCO (Common Objects in Context) is a large dataset that consists of 80 labels which include objects such as:

- People
- Vehicles(Cars, trucks etc)
- Bicycles
- Stop signs
- Fire hydrants
- Animals (Cats, dogs, birds etc)
- Furniture

Among others…

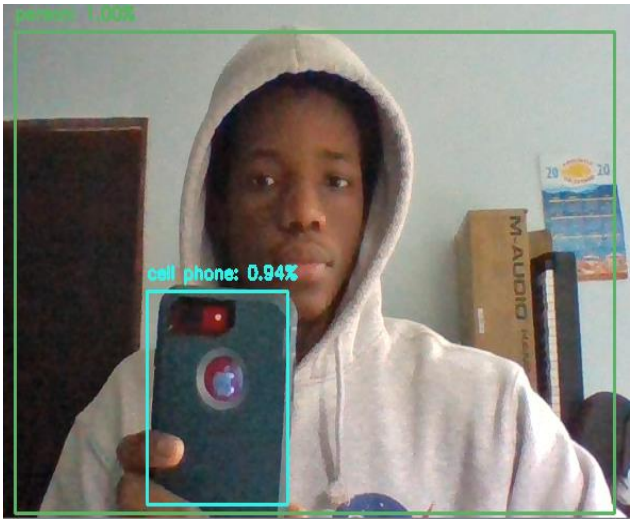

Figure 8. Image showing the results of the YOLOv3 object detector running on the webcam of my laptop

As seen above in figure 6, the model is able to detect persons (myself) and cell phones, while predicting them with excellent accuracy. As the webcam captures data, frame by frame, the model performed a forward pass on the frame and draws a bounding box around each object that it detects. The model is said to perform a single forward pass in 22ms in [3], however it was noted that the performance of the model was rather slow.

In order to direct the user, the device needs to have some way to communicate its commands to the user. The system should be able to communicate via tactile feedback/voice commands. In order to test this functionality, I implemented a voice command system. To do this, I used the Google Text to Speech API to convert a text command to an English command that is stored in a .mp3 file. This .mp3 file can then be accessed and played back through a speaker in order to give the user auditory commands.

6 Discussion

Regarding the object detector that I had implemented, I had noted that its performance was a bit poorer than what was reported in the paper. I suspect that the drop in performance could be due to that fact that the computer that the model was being run does not have any high-performance hardware (core i5 and 4Gb of RAM on the Windows operating system with no GPU). I chose to use a pre-trained detector as it was already able to detect a vast majority of objects that a person would see in their everyday life while walking.

A potential issue with this detector is that it does not account for objects such as streetlights. To overcome this issue, I believe that if I try to re-train an object detector model on my own data, it will generalize better to the environment that it should operate in. I intend to use a model from the TensorFlow Object Detection API to test my hypothesis.

The next major aspect of the project that needs to be completed is the depth estimation. There is a camera (Intel Realsense D415) that can return a depth map (RGBD image) of the environment along with the a regular RGB image, however it is very expensive. Thus, before I use this camera in the system I would like to see if this problem can be solved

with deep learning. Initially I will test the model proposed by Alhashim et al. to see how well it generalizes to an image in an unknown dataset. If I am dissatisfied with the results of that test, then I plan to build upon the work done by them and retrain the model that they used on my own dataset. The issue with this approach is that I will need both a picture of the environment as well as the correspnding depth map of said picture. In order to generate this depth map, I plan to use the previously mentioned depth camera to acquire both the regular picture and the depth map at the same time. If this method does not work well, then I will fall back on using the Realsense camera to extract the depth of the objects in the environment.

Finally, if I finish the previous objective with enough time to spare and successfully integrate the object detection as well as the depth estimation into the device I will attempt to add a facial recogntion feature so that the device can alert the user when someone who they know has been detected (Images of the person would have to be previously provided in order for the model to recognize the person).

7 Conclusions

In conclusion, the objective of this project is to develop a system that uses deep learning and computer vision to develop a system that can aid the blind in their navigation from place to place. In order to do this, I need to have a working object detection system and a wrking depth estimation system. Thus far I have implemented a working object detection system and in this paper I have outlined my plans for moving forward with this project.

8 Acknowledgement

I would like to express my sincerest gratitude to Dr. Kolapo Alli who has made himself available thus far for any queires that I have had and Mr Lindon Falconer who has also guided me as to my approach to this problem. I would also like to thank my parents for the support, financially and otherwise, that they have provided thus far.

9 References

[1] Alhashim, Ibraheem, and Peter Wonka. “High Quality Monocular Depth Estimation via Transfer Learning.” arXiv, March 10, 2019.

[2] Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. “SSD: Single Shot MultiBox Detector.” Computer Vision – ECCV 2016 Lecture Notes in Computer Science, December 29, 2016, 21–37. https://doi.org/10.1007/978-3-319-46448-0_2.

[3] Redmond, Joseph. “YOLOv3: An Incremental Improvement.” arXiv, April 8, 2018.

[4] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” IEEE Transactions on Pattern Analysis and Machine Intelligence 39, no. 6 (2017): 1137–49. https://doi.org/10.1109/tpami.2016.2577031.

[5] Rosebrock, Adrian. “Object Detection with Deep Learning and OpenCV.” P, September 11, 2017. https://www.pyimagesearch.com/2017/09/11/object-detection-with-deep-learning-and-opencv/.

[6] Sandler, Mark, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. “MobileNetV2: Inverted Residuals and Linear Bottlenecks.” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018. https://doi.org/10.1109/cvpr.2018.00474.

[7] Rosebrock, Adrian. “YOLO Object Detection with OpenCV.” PyImageSearch.com, April 18, 2020. https://www.pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/.

10 Appendix

As was mentioned previously, this is a year-long project and as such, the associated Gantt Chart is too large to fit inside this document however, a link to the Gantt Chart can be found here.