Smart Stick Prototype for Visually Impaired People using Raspberry Pi

A DISSERTATION SUBMITTED TO MANCHESTER METROPOLITAN UNIVERSITY

FOR THE DEGREE OF MASTER OF SCIENCE

IN THE FACULTY OF SCIENCE AND ENGINEERING

2022

By

Paul Samuel Christopherson

Department of Computing and Mathematics

Contents

List of Figures

List of Tables

Abstract

Globally, at least 2.2 billion people have near or distance vision impairment. Most people with vision impairment are over the age of 50. Those who live in low- and middle-income regions are 4x more likely to have vision impairment than those in high income regions. (WHO, 2021:online)

In order to help the navigate their environment, visually impaired individuals may;

- Rely on other people
- Use a white stick
- Use guide dogs to assist them

The environment can include many obstacles such as roads, walls, bridges, pedestrians, other moving objects such as cars and changes in height such as sidewalks and edges all which could potentially pose a danger to the visually impaired person. Because many visually impaired live in third world countries problems such as war, famine and poverty create an increasing hostile environment and present risks where their lives could be at risk due to dangers and dangerous people.

The main aim of this research is to explore an assistive device that is inexpensive, mobile and life enhancing to its user. This will help the user to move and explore their environment without limits and dramatically improve the quality of life.

The assistive device in this project will make use smart technology such as sensors, cameras and voice commands to help the visually impaired identify and avoid potential obstacles/dangers as they move contributing to their decision making. This will be done using Deep learning and Tensorflow ,trained using a custom dataset of images that belong to two categories “Threat” person or “Neutral” person from the sensors and code on the Raspberry Pi. All code throughout project is in Python.

Declaration

No part of this project has been submitted in support of an application for any other degree or qualification at this or any other institute of learning. Apart from those parts of the project containing citations to the work of others, this project is my own unaided work. This work has been carried out in accordance with the Manchester Metropolitan University research ethics procedures and has received ethical approval number 46065.

Signed: S. Christopherson

Date: 21.09.2022

Acknowledgements

I give thanks to Amna Eleyan , my supervisor for giving me this research opportunity and for guiding and supporting me during the works. I give a special thanks to Manchester Metropolitan for providing me with an opportunity to study a master’s in data science , I wouldn’t be here without them. I also give thanks to my friends and family who supported me and looked after me. Most of all I give thanks to my friends , classmates and lecturers who were exemplary human beings and passionate about all thing’s tech, they inspired me to aim higher and set great standards for myself. Above all I thank God , who’s unwavering inspiration and voice guided me and continues to in ways that one can only know and witness.

Abbreviations

CNN – Convolutional Neural Network

ANN - Artificial Neural Network

Pi – Raspberry Pi

ML – Machine Learning

DL – Deep Learning

AI – Artificial Intelligence

ROC - Receiver Operating Characteristic

AUC – Area Under the Curve

FP- False positive

TP -True Positive

FN- False Negative

TN- True Negative

Chapter 1 - Introduction

1.1 Project Overview

This project is to develop a prototype assistive device that helps visually impaired people to identify obstacles in the form of people ,using Machine/Deep Learning. The research explored here was to determine whether or not the people in the visually impaired person's zone of concern (around 1.2m) were dangerous or safe

The assistive device used ultrasonic sensors, a camera for image detection, a buzzer and a Raspberry Pi . The Raspberry Pi receives and processes the ultrasonic sensor signals to calculate a distance which then generates a noise to alert the user as to how far the person is. A different frequency noise will be produced depending on how close or how far the person is allowing the visually impaired person to know how far they are based on frequency. The camera then captures the image of the person and using a trained custom built Deep learning model will then be used to determine whether or not the person is dangerous or safe. It then connects to a speaker and using Text to Speech communicates the result to aid the blind person , providing more information on whether or not the person is dangerous or not which contributes to their decision-making capability to assess whether or not the environment is safe or dangerous.

The motivation for this project was to aim to enhance the visually impaired persons lives and to protect them for hazards. It was done with the intention of protection and safety which in turn would mean greater autonomy freedom and above all peace of mind for the visually impaired and the ones they love.

1.2 Aims & Objectives

The objectives for the project are:

- Write up terms of reference.
- Conduct research into relevant papers and similar projects to form research contribution.
- Build the Prototype stick device using suitable material for the walking stick. An Arduino or Raspberry Pi will be used as the micro controller combined with sensors and camera to detect and capture object/obstacle(s). The information captured via sensors will be stored in a secure database(in an SD card or otherwise cloud) and the voice module will relay obstacle type to the blind person.
- Access and store the relevant data stored from the prototype readings to be fed into the Machine learning model.
- Pre-process the data by applying the relevant algorithms and filters or similar and discard non relevant information that has been captured.
- Sort the data into the relevant object folder type and correct size if it is an image.
- Train and build the Machine/Deep learning model to classify objects for the user and assess its performance. To be done using Python and Scikit learn in a Jupiter Notebook environment. Cloud environment to be determined.
- Implement the now classified objects in a feedback loop using a voice module to the User.
- Test and evaluate the actual objects to the predicted object classification during the experimentation and demonstrate use case scenarios.
- Write up the report

1.3 Timeline and Tools

Activity Schedule

Table 1 - Activity Schedule for project

TASK	START DATE	END DATE	DURATION (DAYS)
TOR AND ETHICS	13/06/2022	29/06/2022	13
LITERATURE REVIEW	22/06/2022	05/08/2022	33
BUILD PROTOTYPE STICK	11/07/2022	22/07/2022	5
STORAGE , ACCESS, AND PREPARATION OF DATA.	25/07/2022	05/08/2022	10
MACHINE LEARNING/DEEP LEARNING MODEL	08/08/2022	26/08/2022	15
TEST&EVALUATION	22/08/2022	26/08/2022	5
WRITE UP REPORT	29/08/2022	23/09/2022	20

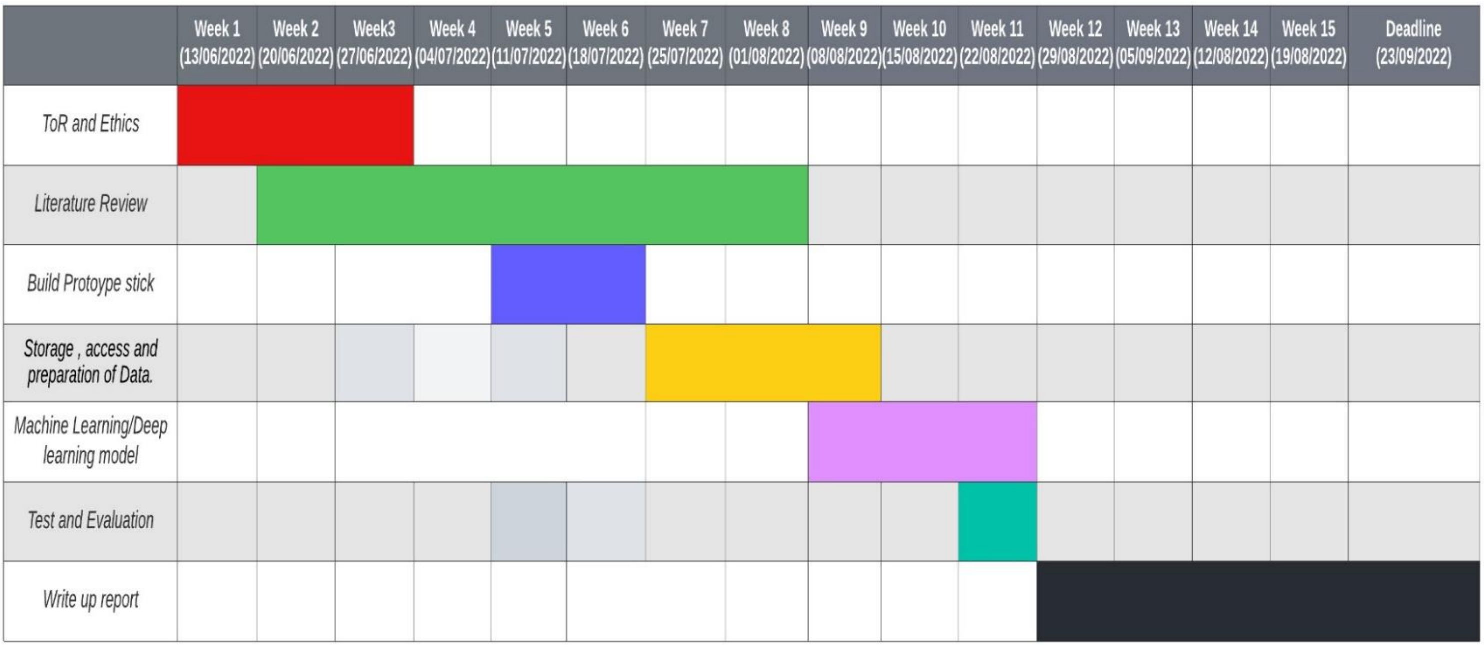


Figure 1 – Gantt Chart for the project

Tensorflow lite was used for Deep learning and images were collected from Google's image search platform. The custom-built dataset was created using Google Chrome extension application "Download All Images" and 2 folders were made to store approximately 500 images of "Neutral" type and 500 images of "Threat" type . Neutral images consisted of human beings of different race and gender with no weapons and effort was exuded in order to include humans holding drills and other household tools. Threat images consisted of an assortment of human beings carrying explosive devices , guns from small to large and knives. Those in prison uniforms were also included. All images were then trained using the online cloud environment Google Collaboratory .

All code for this project including that which was used for running operations on interfaces on the Raspberry Pi, Deep learning model and Google Colab environment were written using Python . Where relevant Tensorflow and Keras libraries were used and Espeak TTS engine to communicate results via the Anker portable speaker. Storage used was Pi SD card which contained within its folder structures on the Pi's OS.

Main hardware used was Raspberry Pi 4, HC-SR04 (Ultrasonic Sensor) , Piezo Passive Buzzer Sounder and NextPro Raspberry Pi Camera Module 5M 1080p. A more extensive list provided for Chapter - System Design of Project.

1.4 Project evaluation

The aims & objectives will be evaluated with regards to their completion and effectiveness.

These are:

- Terms of Reference – A detailed plan about the project to be carried out and its required objectives and activities.

- Build and completion of the Prototype Smart Stick and its associated Database that will feed into a Machine/Deep learning model as required.

- Data Analysis and Evaluation - Results from training and testing from all the machine/deep learning algorithms and the upgrades required to make the Prototype Smart Stick more effective. Evaluate the product by comparing it with the existing ones in the literature.

- Final Report - Completed report detailing the previous objectives and results obtained from these steps to evaluate the effectiveness of the Smart Stick, the accuracy of each of the machine/deep learning algorithms classification.

1.5 Structure of report

Report content is split into chapters , each one is outlined below:

• Chapter 1 – Provides necessary introduction, project overview , aims and objectives , evaluation and timescales for project delivery.

• Chapter 2 - Provides the necessary background and information for the project such as concepts , hardware used, techniques and includes related works with a final conclusion that declares research contribution and intention with regards to research area and the aims and objectives.

• Chapter 3 - covers system design and architecture and the implementation of the project.

• Chapter 4 - analyses the results produced and evaluates them in relation to the overall intention of this dissertation.

• Chapter 5 - describes relevant ideas that could not be pursued in this report and that could form the basis of future research.

• Chapter 6 - provides an overall conclusion of the report and evaluates the project in relation to its aims and objectives from the evaluation criteria that was set.

Chapter 2 – Literature review and related works

2.1 Introduction

This chapter provides background information in order to further understand the project and its software, hardware and concepts deployed. This chapter will also detail the related works and a conclusion in which it provided the project with a sense of direction and motivated its contribution in the field of visually assistive devices.
Topics discussed were:

- Machine learning
- Deep learning
- Artificial Neural Networks
- Gradient Descent
- Cost function
- Tensors
- Convolutional Neural Networks
- Padding
- Pooling
- MobileNet
- MobileNetV2
- Activation functions
- Confusion Matrix
- Recall
- Precision
- F1 scores
- AUC-ROC
- Google Colab
- Tensorflow
- Raspberry Pi
- IOT
- Ultrasonic sensor
- Buzzer
- Camera

2.2 Machine Learning

Machine Learning is the ability of a machine to learn using large datasets instead of hard encoded rules.

There are 2 categories in which the machine learns , Supervised and Unsupervised learning.

Supervised learning occurs when there are labelled datasets that have inputs and expected outputs. If an output generated by the artificial intelligence is wrong then it we will re-adjust its calculations iteratively over a dataset until there are no more errors. An example is the prediction of the weather , using historical data. The training data has inputs such as pressure, humidity , wind speed and the output will be temperature.

Unsupervised learning is the task of machine learning using data sets with no specified structure. One example is behaviour-predicting AI for an e-commerce website. It doesn't have labelled data for inputs and outputs, so It creates its own classification based on the input of the data. It could tell you which kind users are likely to buy different products. (Free Code Camp, 2017:online)

The proposed project is a supervised learning type in which training had taken place using a dataset of 1000 photos approximately and 50% were labelled Threat and 50% were labelled Neutral. The training took place using a subset of Machine learning called Deep learning and the project used image classification to determine whether or not the person detected was a threat or a neutral. Images which were taken via the camera which was attached to the Raspberry Pi.

2.3 Deep learning

Deep learning is a subset of Machine Learning, it uses artificial neural networks which mimic the way human beings own neural networks in their brain think and learn. Deep learning does this by learning and improving on its own algorithms. (Venkatesh, 2021:Online)

Deep learning architecture and models were deployed in this project in order to carry pout image classification.

What first inspired deep learning is the structure and function of neural networks of the brain that activate during the learning process for humans called artificial neural networks.

2.3.1 Artificial Neural Networks – (ANN)

ANN's consist of node layers , like the brain it is made of different interconnected layers of nodes. The layers are called , Input layer → Hidden layer(s) →Output layer. The input layer receives the input data, which in the case of proposed project is the image that was captured from the camera.

Each node/artificial neuron connects to another in the subsequent layers and has an associated weight and threshold. If the output of an individual layer is above a certain threshold , it becomes activated and transmits data to the next layer of the network. Otherwise , no data may be passed along the next layer of the network and the node/artificial neuron does not activate.

The hidden layers preform mathematical computation on the input data. One of the challenges is to decide the number of hidden layers as well as the number of neurons for each layer.
Output layer returns the output data which in the case of the proposed project was the label that was generated in order to correctly classify what the image showed , i.e Threat or Neutral .
Because this was a binary classification problem the activation of just 1 node can actually be sufficient enough to determine whether the image is a "Neutral" or "Non-Neutral" which means "Threat". This was used as the final dense layer and the activation function used was the sigmoid for the node.

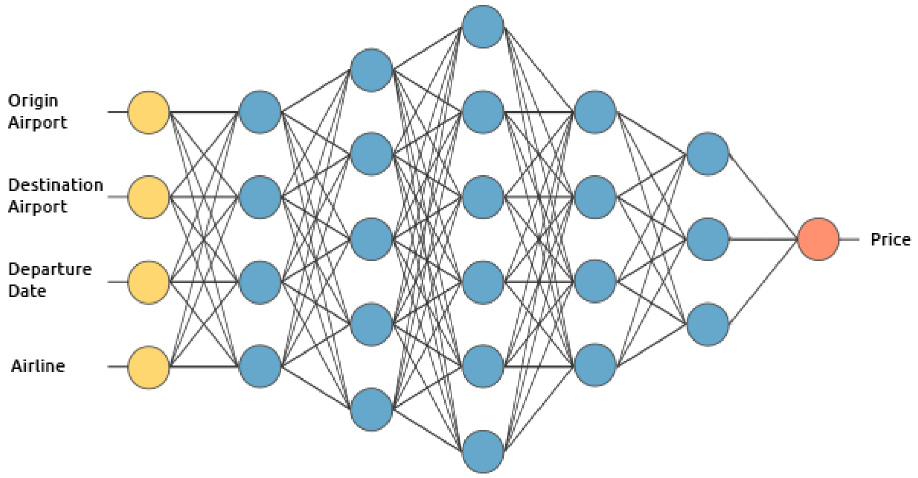


Figure 2- example of a neural network (freecodecamp image, 2017:Online)

Each connection between a neuron has a weight to decide how much of an impact it will have on a network . The sum of all connections going into a particular neuron determine whether or not the neuron activates provided that set threshold has been reached. That particular node is said to have activated and can then pass on data to the next layer of nodes/neural network.

Bias helps determine the threshold level. The magnitude of the weights determines the importance of a given variable, The greater the magnitude of a given input the more significant its contribution to the output .

Once one pass of the input data that is passed through the hidden layers and then exited through the output layer is complete , it is said that one epoch has occurred. The initial weights are set randomly and adjusted and updated as the data passes through. During the training phase for a deep learning model , the outputs that were produced during training are compared with the true values of the output from the historical data. This is how the machine can begin to tweak the weights of connections between neurons in order to get higher accuracy per each pass. The difference between the AI's outputs and the real outputs are best described using something called a cost function which mathematically illustrates the difference best and how to modify and train the network in order to get a greater accuracy.

2.3.2 Gradient Descent

Gradient Descent is a technique that allows one to find the minimum point of the cost function , because by finding the minimum of the cost function the correctness of a fit for the model is optimised. Through the training process the algorithm adjusts its weights and the parameters of the model adjust to converge at a minimum , it determines the direction to take as it updates these parameters to reduce errors or "minimise the cost function".

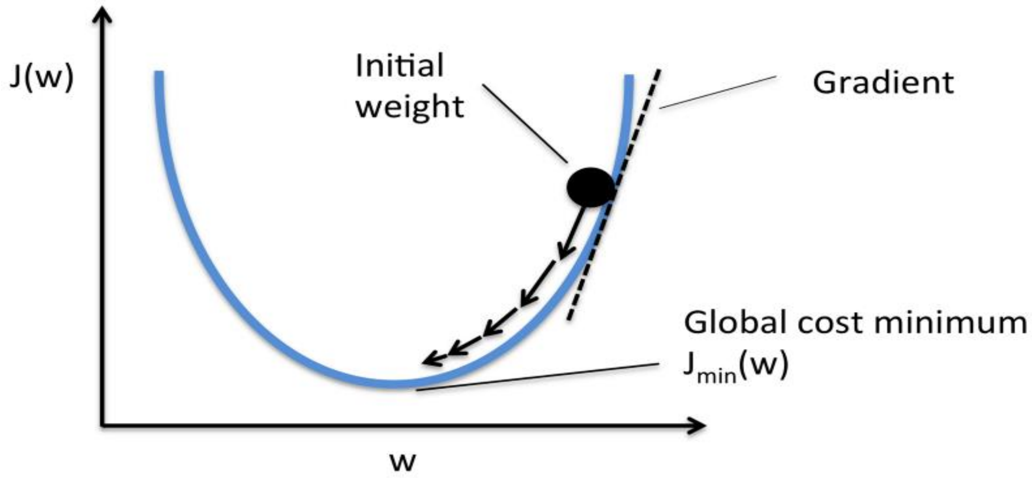


Figure 3- Gradient Descent (Freecodecamp - gradient descent, 2017:online)

A lot of computation power will be used in order to find the minimum because the dataset must be iterated through many times. Because of the high computation powered required a Tensorflow lite model was deployed because of high accuracy yet smaller computation needed to carry out image classification on mobile devices.

2.3.3 Tensor

A Tensor is a data structure used by machine learning algorithms in order to learn.

Some examples of common tensor representations are:
Vectors: 1D — (features)
Sequences: 2D — (timesteps, features)
Images: 3D — (height, width, channels)
Videos: 4D — (frames, height, width, channels)

Overall machine learning algorithms deal with subsets of data at a time called batches of data. In images as an example for batches a 4D tensor was used because the training takes place in Batches of total images out of 1000, i.e 32 images at a time. The 4D being because we determine the sample size or as it is more commonly known batch size.

This would be : Samples(i.e Batch size) , Height of Image , Width of Image , Channels (i.e RGB)
In general, the primary attributes that define a tensor are rank , shape and type.

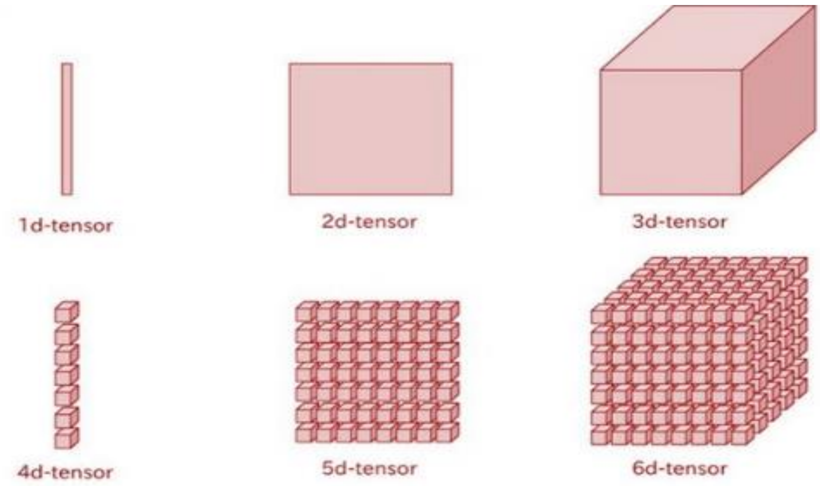


Figure 4- Tensor data structure (Data Driven Investor, 2021:Online)

2.4 Convolutional Neural Network (CNN)

CNNs are one of the main ways to carry out image recognition and image classification using Deep learning. A CNN allows learning to take place directly from the data without needing to manually extract features. CNN model is the main way that image classification occurred during the proposed project and due to efficiency, accuracy and space requirements a CNN model of MobileNetV2 was chosen.

CNN image classifications take the input image , process it and then classifies it to belonging to a certain category of a given list. The proposed project has binary classification in which the categories for the output are either Neutral or Threat.
Deep learning models cannot consume unprocessed images like humans , the images must first be converted x,y coordinates and Red, Green and Blue values (RGB). These are then stored in Tensors of the correct dimension usually the 3rd unless it's a batch in which case the 4th is necessary to include the batch size number.

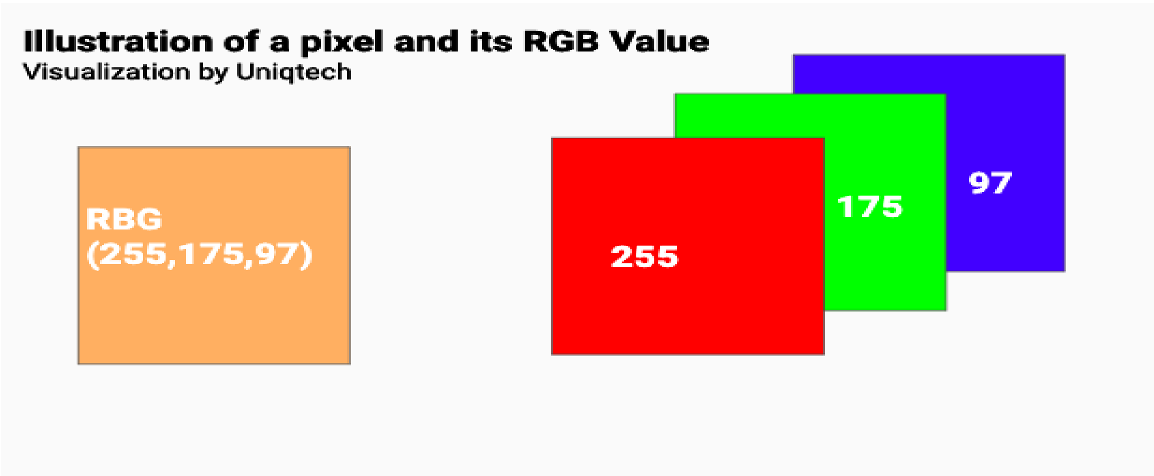


Figure 5- Illustration of a pixel and its RGB Value (Uniqtech, 2019:online)

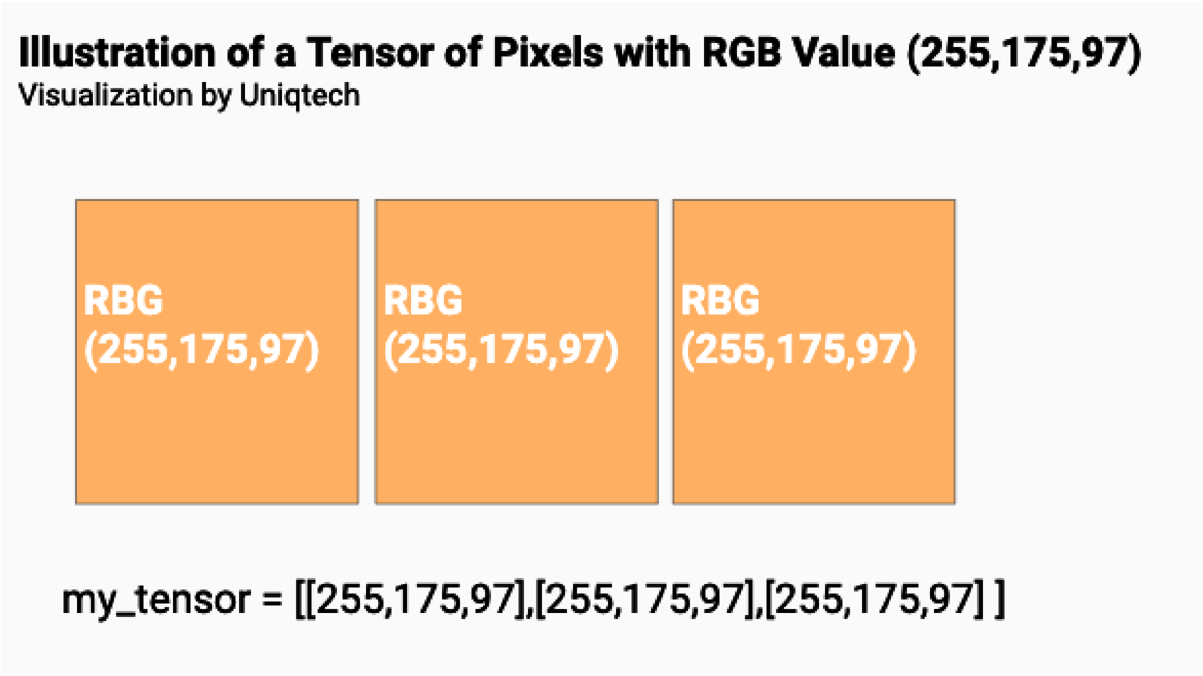


Figure 6 - Illustration of a Tensor (Uniqtech, 2019:online)

The AI sees an array of pixels of 4D tensor during training. This format is usually height x width x dimension of pixel.

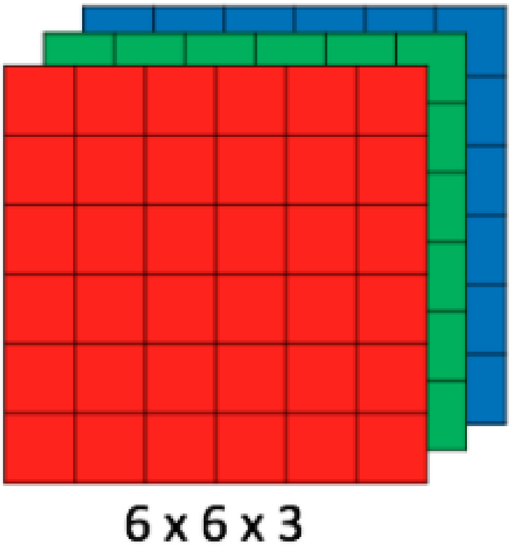


Figure 7- Array of RGB Matrix((Prabhu, 2018:Online)

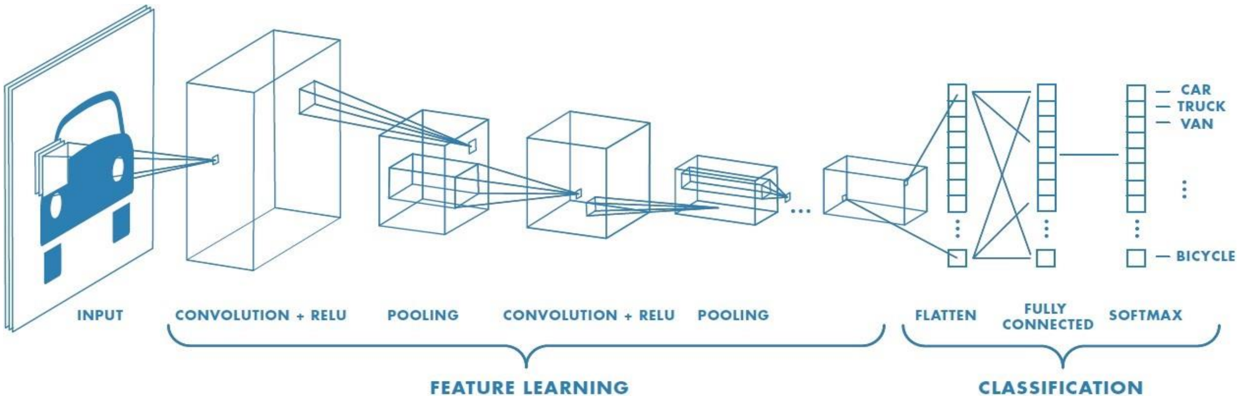


Figure 8- Neural network with many convolutional layers((Prabhu, 2018:Online)

Convolution takes place within the first input layer to extract features from the data. The mathematical process taking place uses the image matrix and a filter or a kernel in whereby convolution preserves the relationship between pixels.
The filter or kernel is of a certain value in that it can extract features such as edges and curves from the input image of data. A feature map of the image matrix is what is created that accentuates these sought out for features of the image. This is what occurs during the feature learning phase.

Deep learning CNN models train and test , with each input image (captured image) passing through a series of convolution layers. Pooling , fully connected layers (FC) and apply Softmax ,Sigmoid or Relu function to classify an object with probabilistic values between 0 and 1.
Below is an example of this convolution taking place:

- An image matrix (volume) of dimension **$(h \times w \times d)$**
- A filter **$(f_h \times f_w \times d)$**
- Outputs a volume dimension **$(h - f_h + 1) \times (w - f_w + 1) \times 1$**

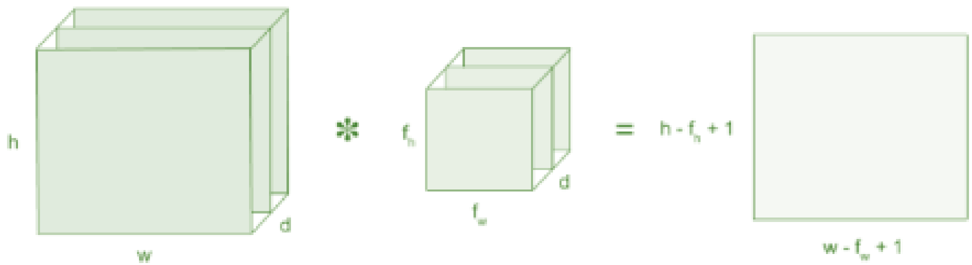
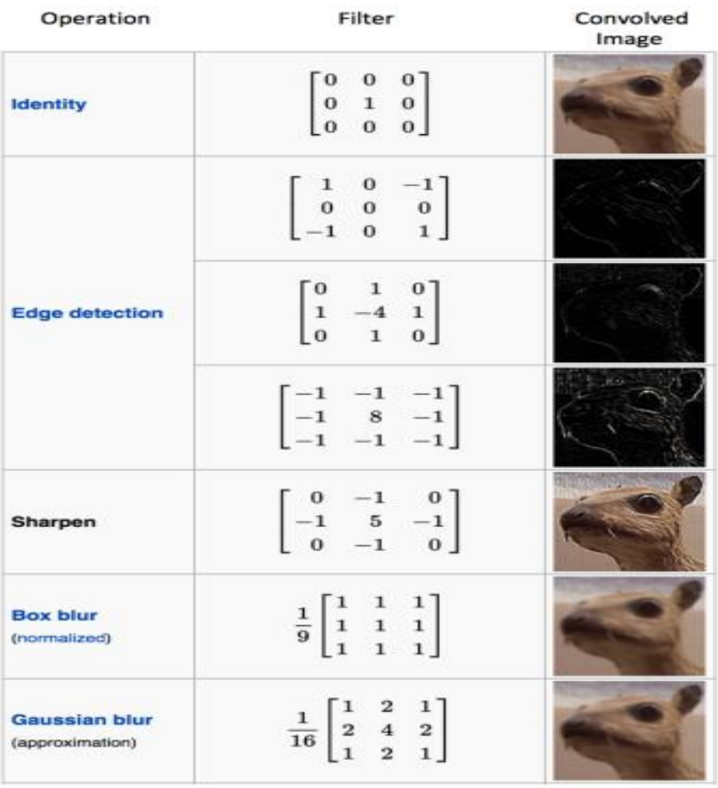


Figure 9 - Image matrix multiplies kernel or filter matrix((Prabhu, Image matrix multiplies kernel or filter matrix, 2018:Online)



Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Figure 10- Some common filters (Prabhu, Some common filters, 2018:Online)

Sometimes through matrix multiplication a filter does not fit the input image so 2 options are presented , one called padding whereby the picture has 0's added to its matrix so that It fits or to drop the part of the image where the filter did not fit.
An image matrix usually has 3 channels , Red , Green and Blue and the values within are between 0 – 255.

2.4.1 Pooling

A Pooling layer reduce the number of parameters when the images are too large , down sampling reduces the dimensionality of each map however important information is retained.

2.4.2 MobileNetV2

For this project , Deep learning architecture consisting of the CNN of Mobile Net v2 has been deployed. The reasoning behind this selection is that Mobile Net v2 is because of how much space it takes up in comparison with some of the other traditionally used Deep learning Architectures.
To understand MobileNetV2 it is necessary to understand its predecessor MobilenetV1 .The Mobile Net has far fewer parameters and the size of the model itself has as few as 1.3 million parameters. It only requires 16 – 18 MB of space compared to the VGG16

model which requires over 500MB of space. It is important to make this distinction because the proposed project uses a raspberry Pi which has smaller memory and lower computing speed than traditional hardware such as computers.

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
GoogleNet	69.8%	1550	6.8
VGG 16	71.5%	15300	138

Table 1. MobileNet parameter and accuracy comparison against GoogleNet and VGG 16 (Source: Table from the original paper)

Figure 11 - Mobile Net parameter and accuracy comparison (Sarkar, 2021:online)

The reason why Mobile Net has similar accuracy to the traditional models but requires far less computing power is because of Depth wise Convolution .
Depth wise Convolution are 2D convolutions where each channel (BGR) is kept separate and then convolutions are applied in comparison with Convolution where 2D convolutions are preformed over multiple input channels, which results in a decreased overall sum of channels.

Modern state of the art neural networks requires high computational power in order to produce high accuracy which are often beyond the capacity of mobile and embedded devices.

MobileNetV2 was developed by Google as a successor to MobileNetV1 and also uses Depth wise separable convolution. It was developed specifically to run on mobile devices such as Phones, Tablets and Raspberry Pi's whereby there is significantly less computing power and memory than a computer.

In both MobileNetV1 and MobileNetV2 , 2 convolutions called Depthwise convolution and Point-wise convolution take place.

Depth wise separable convolutions are convolutions that take place across the 3 RGB channels of an image. The key difference being is that each channel is multiplied by a tensor of depth 1 rather than a bigger calculation taking place whereby the whole image matrix is multiplied by a tensor of depth 3.
When each convolutional layer is multiplied by a tensor of depth 1 for each channel , Point—wise convolution then is applied to get the total result. This makes the

computational power needed less and the efficiency improved without affecting the accuracy.

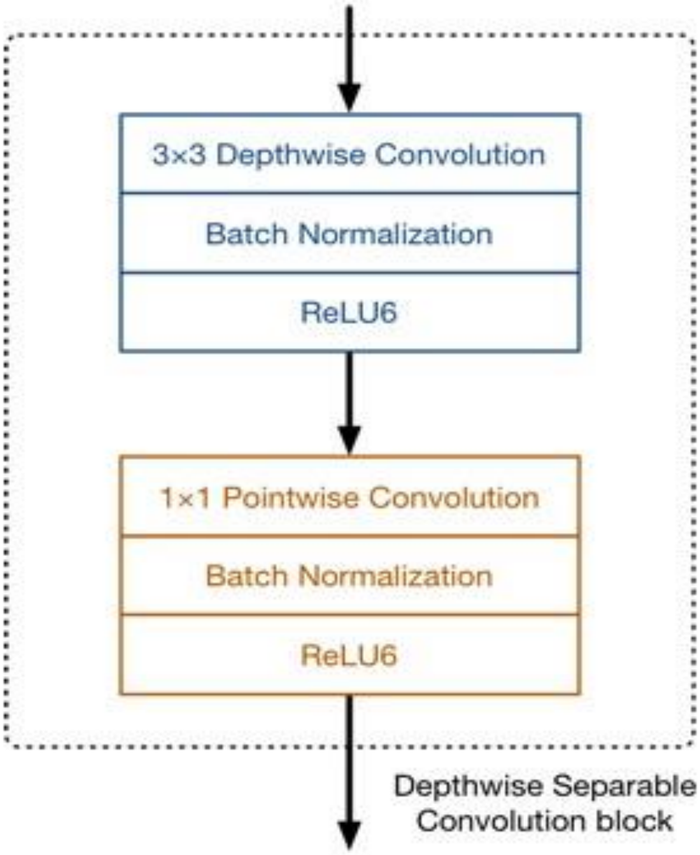


Figure 12 – Depthwise Separable Convolution block (Hollemans, 2018:online)

MobileNetV2 differs with MobileNetV1 in as such that:
"The expansion layer acts as a decompressor (like unzip) that first restores the data to its full form, then the Depthwise layer performs whatever filtering is important at this stage of the network, and finally the projection layer compresses the data to make it small again.

The trick that makes this all work, of course, is that the expansions and projections are done using convolutional layers with learnable parameters, and so the model is able to learn how to best (de)compress the data at each stage in the network". (Hollemans, 2018:online)

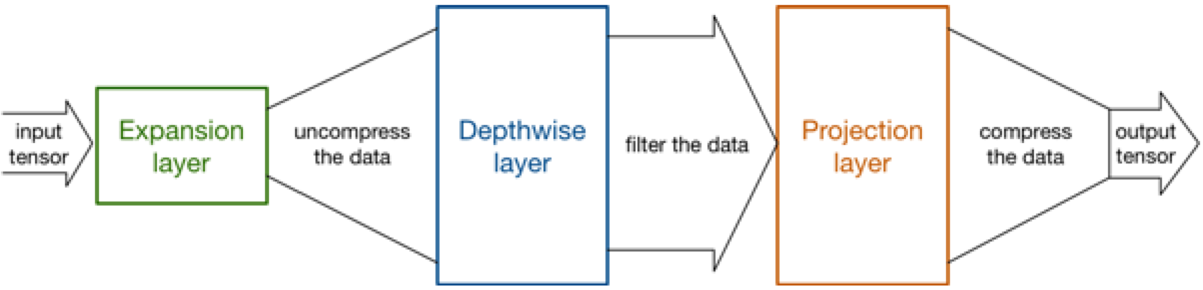


Figure 13 - Compression (Hollemans, 2018:online)

MobileNetV2 uses inverted residula blocks with bottleneck features but with lower a parameter count than compared to MobileNetV1 , other than that both are almost the same.

2.4.3 Relu & Sigmoid (Activation Function)

An activation produces the output of a node in a neural network by applying a function to it in order to determine whether or the node should be activated and at what magnitude.

Activation functions are divided into two types , linear and non-linear activation functions.

Relu is the preferred activation function because it is easy to train and has great performance

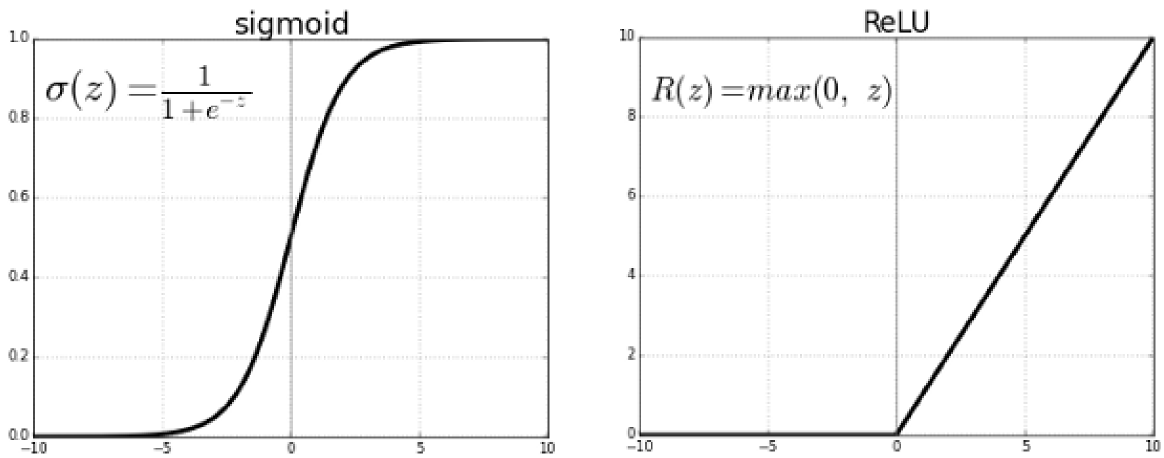


Figure 14- ReLU v/s Logistic Sigmoid (Sharma, 2017:online)

However, the main issue with ReLU is that negative values become 0 immediately which decreases the ability of the model to fit or train the data properly. This has a further effect downstream as it affects the resulting graph by not mapping negative values proportionately and appropriately. This can be corrected by using a slightly negative gradient when the value is below 0.

The sigmoid activation layer is the one used before the image is classified between threat or non-threat. The node output here is between 1 and 0. This is intentional as the image is either classified as a threat or a neutral with the input of 0 being the threshold of 0.5. Values above and below 0.5 will be classified as either threat or non-threat with a probability expressed.

2.5 Confusion Matrix

The confusion matrix is a metric of performance for machine/deep learning and is used for classification. In the proposed project Neutral or Threat were the 2 classes of an image.

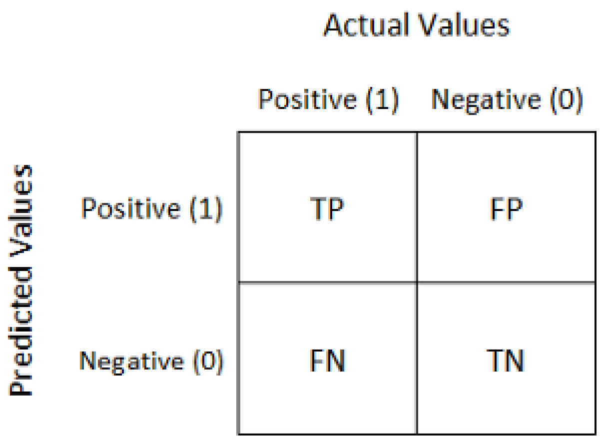


Figure 15- Confusion matrix (Narkhede, 2018:online)

The above shows the actual values and the predicted values.
TP – True Positive
FP- False Positive
FN- False Negative
TN – True Negative

If the actual image was a neutral and the deep learning model predicted it as neutral then it is a True Positive. The inverse of the actual image being misclassified as a Threat means it goes into the False Negative box.

Likewise, if the Image was a threat and successfully predicted to be threat then it is said to be a True negative and for the inverse if the Image was misclassified to be Neutral when that is not that the case then it is said to be a False Positive.

This is done with the images used in the dataset and a total is produced resulting in the sum of each image belonging to the 4 descriptions . Often a proportion is used instead of a number.

In this example of the proposed project for high performance we expect that the captured images actually being a threat and being classified as threat to be above 50% of the total images actually classified as threat. This means the model can determine that a threat is actually present and inform the visually impaired person so as to make a decision in that environment. This would be a high value for True Negative and a low value for False Negative.

Low performance would mean that the threat image being misclassified as a Neutral i.e a False positive with a high percentage would indicate that visually impaired person is being put in danger due to the model's inability to understand that a threat is present.

For the proposed project a low False Positive (FP) and high True Negative (TN) would mean the model is protecting the visually impaired person. It does not matter so much as if the image detected be neutral and be misclassified as a threat (False Negative) as it would simply mean the visually impaired person is going to be overly cautious, but at least his/her life is not as risk as it's in the False Positive user case scenario.

2.5.1 Precision

How precise the model is determined by :

Precision = TP/TP +FP

The higher this value is the more Precise the model is. This means that from all the Neutral Images classified what percentage were actually Neutral.

2.5.2 Recall

Recall is determined by:

Recall = TP/TP + FN

The higher this value is the better the model is at determining from all the images classified as Neutral how many did the Model predict to be correctly Neutral .

2.5.3 Accuracy

Accuracy is from all the classes positive and negative ; how many were predicted correctly.

2.5.4 F1 score

Is difficult to understand the relationship between the Recall and Precision scores between 2 models so F1 uses a harmonic mean to provide a value that one can use for comparisons.

It extenuates extreme scenarios more so that a good mathematical comparison can be made.

F-measure = 2*Recall*Precision/Recall +Precision

2.5.5 AUC-ROC

The Receiver operating characteristic is a metric of evaluation in as such that the True Positive rate is plotted against the False Positive rate at various threshold values.

"The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.

The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.

When AUC = 1, then the classifier is able to perfectly distinguish between all the Positive and the Negative class points correctly. If, however, the AUC had been 0, then the classifier would be predicting all Negatives as Positives, and all Positives as Negatives.

When 0.5<AUC<1, there is a high chance that the classifier will be able to distinguish the positive class values from the negative class values. This is so because the classifier is able to detect more numbers of True positives and True negatives than False negatives and False positives.

When AUC=0.5, then the classifier is not able to distinguish between Positive and Negative class points. Meaning either the classifier is predicting random class or constant class for all the data points.

So, the higher the AUC value for a classifier, the better its ability to distinguish between positive and negative classes." (Bhandari, 2016:Online)

2.6 Tensorflow

Machine learning is very complex and implementing models was more difficult in the past than it was today, Google's Tensorflow has eased the process of acquiring data, training models, serving predictions and refining future results.

Tensorflow is an open-source library for numerical computation and large-scale machine learning. Tensorflow is a consortium of machine learning and deep learning models and algorithms and combines to make them useful through the use of common programmatic metaphors. Programming languages of Python or JavaScript are primarily used to provide convenient front-end API for building applications with the much faster C++ used to execute those applications providing the highest performance. (Yegulalp, 2018:online)

Tensorflow lite is part of Tensorflow and one that allows its use on mobile devices and reduced computational power and memory but with the same accuracy.
A Tensorflow lite model was used in the project and was trained on approximately 1000 images using the keras library.

The model used was trained first on google colab and can be accessed at : **https://colab.research.google.com/drive/1oE0yKXBjJhKSjD5WPJxESuVSBkucWxUD?usp=sharing**

The optimization algorithm selected for training was SGD and validation was used so the machine could validate as it trained on the dataset and ensure that the gradient descent was of the correct vector.

2.7 Google colab

Google colab is a free Jupyter notebook that runs on the cloud. Colab supports many popular machine learning libraries such as Tensorflow and Keras can be easily loaded

Code can be written and executed in Python and the free cloud services provides a free GPU.

2.8 Raspberry Pi

The Raspberry Pi is a cheap, credit card size computer that runs a Linux operating system. Its capability allows it to run like a regular desktop computer such as browsing the internet, playing high-definition video and word processing.

First released in 29th of February 2012, the Raspberry Pi has the ability with sensor attachments to interact with its environment and its simplicity makes it the preferred device for IOT projects and was used for the proposed project, the reason being that its availability and affordability would help it become more accessible to those who need the IOT device.

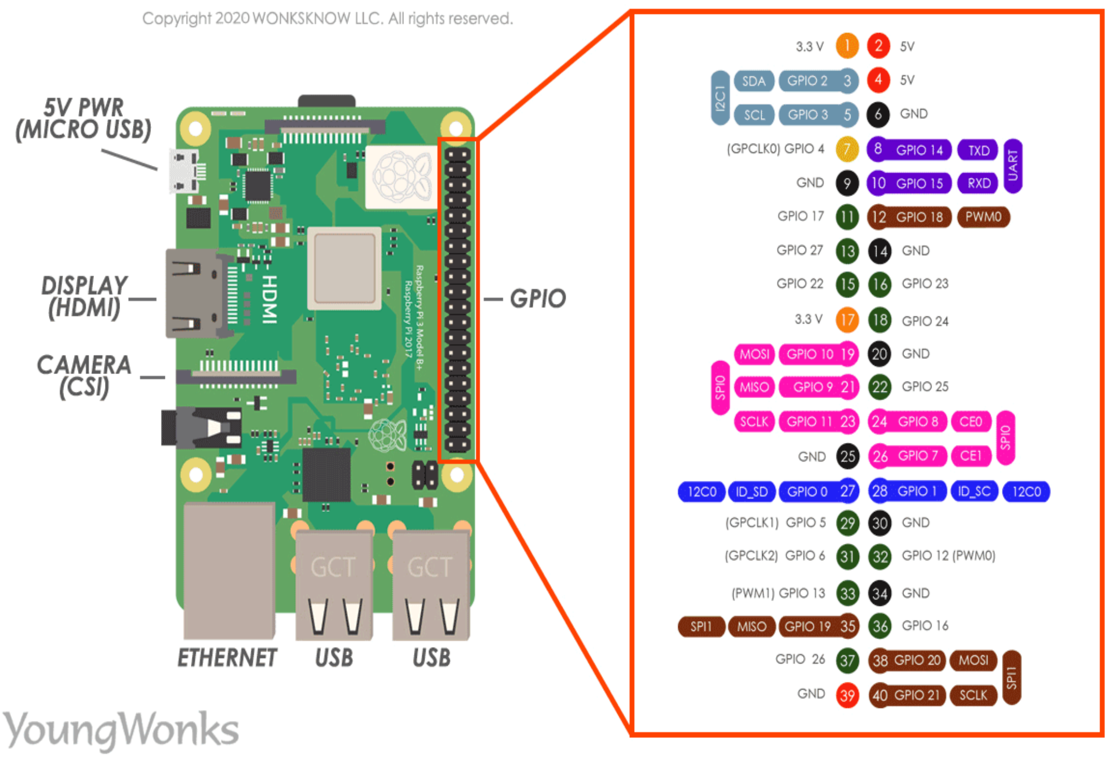


Figure 16- Raspberry layout ((Aggarwal, no date : online)

2.9 Ultrasonic Sensor

The Ultrasonic Sensor used in this project is the HC-SR04.

The HC-SR04 has:

- operational voltage of 5V dc,
- operating current of 15mA,
- measure angle of 15 degrees
- ranging distance is between 2cm and 4cm.

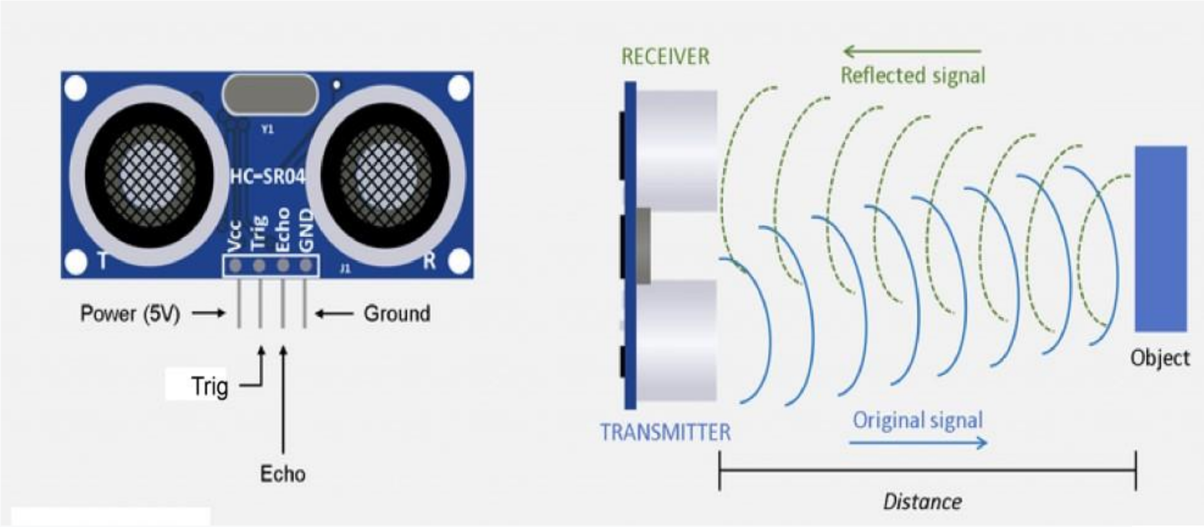


Figure 17 - About Ultrasonic Sensor HC-SR04 (osyoo, 2018:online)

The Raspberry Pi Supplies the power.

The Trigger pulse is a 10-microsecond pulse is sent from the Pi.

Then a simple audio burst is created and sent by the transmitter, and then the Echo Pin becomes High.

The Object then reflects the transmitted audio signal, which is then picked up by the receiver , then the Echo Pin becomes Low.

The Time it takes for the Echo Pin to go from High to Low is the Time it takes to travel to and from the object, so Time is known.

Using;

Distance = Velocity x Time

The signal has travelled twice the distance from the sensor to the object so the calculation must half the total distance to get the distance from the object.

2xdistance_from_object = Velocity x Time

The velocity of Sound at sea level is 343 m/s.

distance_from_object = 343/2 x Time.

The distance from the object is calculated and then used in the project to determine how close an object is by varying the frequency of the buzzer.

When an object is detected the camera activates and captures an image.

2.9 Buzzer

The Buzzer used is a Piezo Passive Buzzer Sounder.
When the distance returned by the sensor approaches 0 , its frequency increases.
By using specific intervals of distance and assigning a specific frequency the visually impaired person can know how far away an object is by frequency alone.

2.10 Camera

The NextPro Raspberry Pi Camera Module 5M 1080p was used.

2.11 IOT

When devices and objects with built in sensors are connected to an internet of things platform, the data is integrated, and analytics are applied in order to gain key insights and ensure that necessary applications address specific needs.

In the Project specific scenario, the sensors/interfaces used were Camera, Speaker and Ultrasonic sensor. All this information was centrally processed using the Raspberry Pi which is normally connected to the internet of things. The ultrasonic sensor provided object detection and returned a distance. The camera returned pictures once an object had been detected. The picture then was image classified with its label being returned and via the speaker as its output. All of these procedures used the Raspberry Pi module as the main interface which compiled all this information and through using the tensor flow lite deep learning CNN model (MobileNetV2), the image was classified correctly the object detected.

In the ideal world a machine/deep learning model is constantly updated with streaming data to learn and become more accurate, this forms the machine/deep learning pipeline for IOT projects.

2.11 Related Works

2.11.1 An_Android-based_Portable_Smart_Cane_for_Visually_Impaired_People

The evaluation of the project was very basic. It only analyzed whether or not functionality worked for the project. However, the evaluation of the project could be improved by assessing the overall effectiveness of the smart cane and how it achieves its specific purpose in helping the visually impaired person. This could have been achieved by using real use case scenarios of the common and important issues the visually impaired will experience such as roads, edges, holes and moving objects and assessing its performance respectively. A real use case scenario such as determining whether people are neutral or threatening to a visually impaired person has been used in the proposed project as the data suggests that most visually impaired live in 3rd world countries and such countries are in constant states of conflict and difficult circumstances making it more important to discern whether or not someone is there to help or there to take advantage of the visually impaired person.

A benefit to this particular smart cane is how it utilizes 5 different frequencies of sound to determine how far and how close the object detected is. The simplicity and effectiveness are something the proposed project could make use of. The below extract explains this in further detail. “ Obstacle detection in the range of 5 feet with varying buzzer frequency after every 12 inches to give batter understanding of distance to obstacle ”.This enables the visually impaired person to learn just how close the obstacle is and as they approach, or the obstacle approaches them In a simplistic way.

When an obstacle is detected by the sensor , a buzzer of varying frequency is activated. The buzzer sound frequency changes at intervals of 5 feet , 4 feet, 3 feet, 2 feet and 1 foot. At either 1- or 2-feet obstacle detection the visually impaired is able to touch the object. The process and mechanism are not quite stated clearly by this paper, but its concept and its simplicity has been utilised in the proposed project. This allows the visually impaired person to intuitively "know" how far an object it is based on frequency rather than numerically providing values for them to figure out thus enhancing their user experience.

This paper is similar to the proposed project because there will be sensors deployed to detect the obstacle using distance. The sensors to carry out this operation are ultrasound.

The proposed project will not make use of user-android application but could very well make use of both GPS and google maps in the future , as well as Voice and Call functions to the microprocessor in order to allow user communication to the device. The proposed project will make use of Raspberry pi as the main processing unit but there is scope for the Arduino to be included later if it is proven to make a viable contribution.

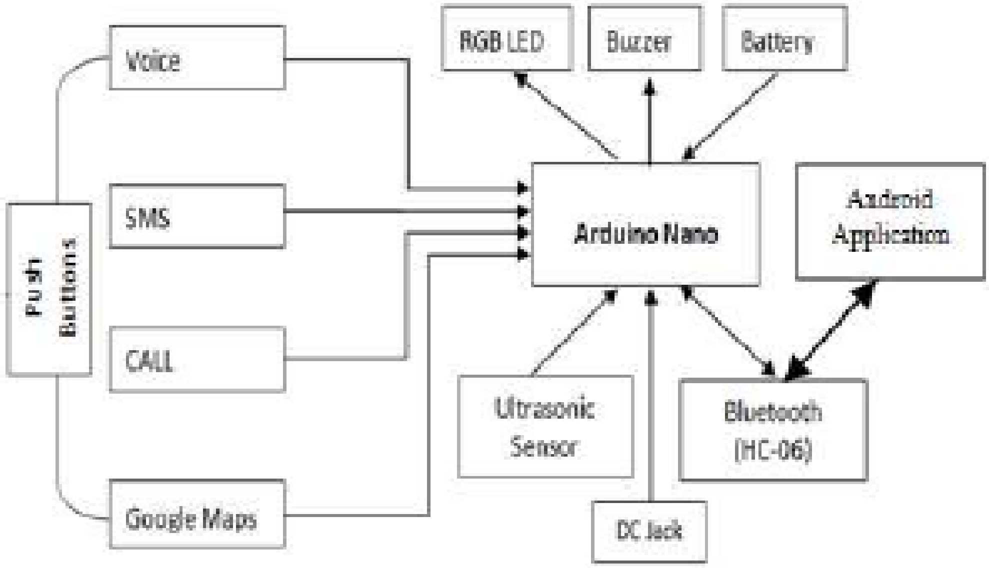


Fig. 1. Android Application Model

Figure 18- Diagram of Project model (M. Adil Khan et al , 2021)

This article admits that in order to advance the smart stick it could use A.I to detect and correctly identify the obstacle ahead. In this respect the proposed project will make use of Machine learning and Deep learning in a way that helps the blind identify the obstacle ahead and make use of text to speech to communicate those results to the visually impaired person.

Overall, for this paper there were too many features which made it quite complex, the android app was useful however, it did not make the blind person more autonomous and still required a person who could see to maximize the use of the app. This could be improved by making everything voice activated in the future. (Al M. K., 2021)

2.11.2 Computer Vision-based Assistance System for the Visually Impaired Using Mobile Edge Artificial Intelligence

This project is multifaceted and makes use of Image classification, object detection, semantic image segmentation and 3D point cloud processing to extract complex perceptual information about the environment.

Semantic image segmentation models can detect road, curb and road marking pixels. As such the proposed project could make use of this particular feature and is something that could be explored.

Many such techniques such as curb edge detection devices are already being used in autonomous vehicles, so the technology is well detailed.

This project is similar to the proposed project, in as such that it is uses Deep learning to preform object classification but uses more advanced computer vision hardware and software for object detection resulting in a higher cost and more processing power, things which detract from the goal of making technology for the visually impaired cost effective as most do not come from high income backgrounds.
This project also makes use of 3D point cloud computing processing to detect elevation changes.

This project also recognized that many visual aids to not have advanced perception such as edge detection hence the need for Deep learning/Computer Vision. This project is similar to the proposed one in such that a voice interactive interface is to enable useability for the visually impaired person.

A major advantage of this paper is its approach through carrying out various interviews with the visually impaired (Lighthouse for the blind) , the challenges and obstacles have been clearly identified to make the project more effective and to have their concerns addressed correctly from the beginning.

From these interviews it was clarified that the main things the blind will have issues with are:

Road markings, pedestrian crossings, edges , curbs, sidewalk, trees, holes, barriers to construction , cars , cyclists , cycle lanes , lamp posts, posts boxes, post office, cars on the road and this report highlighted these things effectively.

The proposed project will aim to make use of the research and interviews carried out in this paper in order to have a clearer evaluation for specific objects and use case scenarios. It is of interest to note that to identify road markings and pedestrian crossings the proposed project will need to notify the visually impaired at where the road marking is located and how far in addition to detecting an obstacle. The proposed project aims to be cost effective however in order to increase distribution amongst lower economies in order to be accessible for more people. It is important to note that the items described above are for more developed countries and such as data for the 3rd world countries is missing.

The things highlighted above are mainly the things that are hazardous or beneficial to the visually impaired person which fed into the concept of people and putting them in to two categories, Threat or Neutral for the proposed project.

The system designed of this project is as shown below:

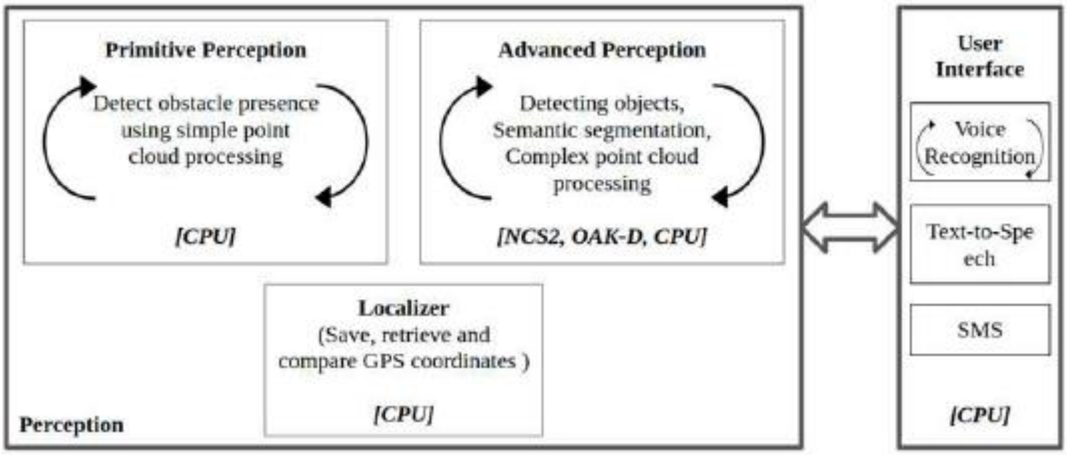


Figure 19 -System design block diagram ((Al J. K., 2021))

The proposed project will only make use of ultrasonic sensors and then deploy deep learning to identify the object with text to speech communicating this to the person. Distance will be communicated via frequency buzzer which is far more simplistic and easier to use.

The equipment deployed in this project is advanced but expensive with NCS2 and OAK – D costing around £325.

However, because the sensors can sense top, bottom ,left and right it supersedes the walking stick because it can sense hanging objects as obstacles. The proposed object could remediate such a problem by using extra sensors and cameras at differing heights and could be something to consider.

For this project, Point cloud processing is carried out using Open 3D and models are optimized on Open Vino techniques.

The datasets deployed in the models were;

- Google open image (GOI) dataset
- Laboratory for Intelligent and Safe Automobiles (LISA)
- German Traffic Sign Recognition Benchmark (GTSRB)
- Traffic Cone Dataset
- Cityscape Dataset
- Pascal VOC dataset from Luxonis' DepthAI library
- ADEK20

The GOI dataset is large, images with class labels relevant to this project, i.e., traffic lights, traffic signs and street names, were chosen and their labels converted to the PASCAL VOC format for training purposes.

For simplicity the proposed project , a new dataset of images from Google search will be collected , inspected and utilised.

The Tensorflow lite models are pre trained on the city scape for this project. It would be interesting to use pre trained deep learning models for object classification and deploy it on the proposed walking stick but for originality of research a new deep learning model will be developed with a custom-built dataset and labels.
Tensorflow lite has the best results for indoor application and is something that will be considered for the proposed project.

For this paper:

OpenVINO and TensorFlow model zoos which include;

- DepthAI's SSD PASCAL object detection model,
- OpenVINO's ADAS models and TensorFlow Lite's model pretrained on the Cityscapes and ADEK20 datasets.

SSD-mobile net was chosen for its speed and single stage detection capability. Pixels were trained on size 600x400 pixels to 300x300 pixel for 300,000 steps and initial learning rate of 8x10^-4 and decay of 0.95 with batch size of 24 . Training :70 , validation : 20 and test:10 ratios were used with the best results being obtain on 300x300 pixel.

For Detecting Objects Apart from DepthAI's SSD-MobileNet object detection model pretrained on the PASCAL VOC dataset, custom models were trained to detect traffic-related classes such as traffic signs, traffic lights, traffic cones, fire hydrants, yellow pavements, crosswalk buttons, public trash cans etc.

For Crosswalk detection alternate approaches such as machine learning on depth images and semantic image segmentation were employed.
This was achieved by preforming edge detection followed by the Hough transform to detect lanes and something the proposed project could consider.

This project utilizes GPS and made use of a VK-162 G-Mouse USB-enabled GPS for geolocalization. Twilio was used to send SMS coordinates of the GPS.

For detecting elevation changes an ensemble of 2 MiniVGGNet classifiers for color images and depth images that had been trained for up-curb, down-curb and flat-surface classes.

A separate dataset of 9000 RGB color and depth images was collected in the vicinity of curbs for this purpose. (The dataset can be found at https://app.speechify.com/item/2266620c-6abd-4398-a9cd-5c874f990e8c#:~:text=Xplore.%20Restrictions%20apply.-,segmentation%20models%20such%20as,%3B,-Settings
)

For the proposed project GPS could be incorporated in the future but the concept of the custom dataset was utilized.

For text to speech, the online packages used were GoogleTTS , Microsoft Speech Engine and TTS-Watson. For offline Festival was higher performing than Pyttsx3.Python's play sound package with google audio was used. The proposed project could make use of all the above but aims to be autonomous without the need of internet to make use of amazon echo as a voice/speech interface module. An alternative could be considered due to possible mobility issues of the amazon echo.

The way of evaluation used needs a clear explanation to those who may not have understanding of the technology they deployed. However, the desired breakdown of using F1 scores for object classification was satisfactory. It would be an improvement if they explained better how object detection performance is measured with a visually impaired persons specific use case scenarios. Responses such as ease of visually impaired person's navigation ease, useability and safety could be indicators of the effectiveness of their project.

F1 classification report will be used for the proposed project as well as recall and precision.

The use of Viso suite on this project with Open CV enabled the project to avoid using code to optimize computer Vision.

The future work suggests that traditional point cloud methods to detect elevation changes using Generation 2 Depth AI module could be used. The use of a laptop is limiting to mobility and Google Pixel 2 and Nvidia Jetson (or TX2) are used to process the data.

Functionality, useability and cost needs to be considered as this was a very expensive project that made use of laptops. The proposed project will aim to reduce cost and improve accessibility through simpler and more effective ways such as ultrasonic sensor, buzzers, GPS ,Camera and raspberry pi to avoid using power consuming processing equipment. (Al J. K., 2021)

2.11.3 Raspberry PI Based Smart Walking Stick

This project is simple yet effective. It uses an application called IP webcam.
This project makes use of a GPS module SIM28ML and transmits its data first to the Arduino , via the spare contact it transmits this data to GSM module.

MobileNet was utilised and the transfer learning concepts were also utilised to ensure optimum performance which is a feature that the proposed project will make use of. The effectiveness of MobileNet as CNN image classifier will also be used in the proposed project.

The SD card utilised contains the Keras and Tensor flow libraries. The proposed project uses raspberry pi SD card functionality and will therefore make use of Keras and Tensorflow.

Overall, this paper lacks the attention to detail . The intention was good but a more succinct step by step approach is required with what they used and how they implemented and met the projects evaluation objectives. The proposed project will address this weakness by explaining in detail what every phase and interface of the project does and will explain its objective and how the objective lies within the overall bigger picture. In detail will be explained the cameras operational capacity , how the buzzer communicates distances to the user and in detail of the specific use case and its relevancy to the visually impaired person.

This paper spent too much providing a general overview , it could improve by providing stronger detail in its processes and how it achieved its overall strategic processes . The analysis and conclusion were very weak and needs to be backed up with data and progress milestones. The proposed project will eliminate this with strong data , clear methodology ,milestones and future considerations to expand and improve the research and its objectives. The data the proposed project will use is object detection , object classification and will make use of F1 classification reports. Use case scenarios and measurable evaluation such as navigation will be tested as well as useability . Specific scores will be used and where possible statistical analysis to be used such as Confusion Matrices. Milestones for the Proposed project has already been included in the ToR but a further break down with specificity will be used. (al M. G., 2021)

2.11.4 Iot Based Smart Stick Used For Visually Impaired People by Using Image Processing

This project provides a good overview and provides background motivation and knowledge of object detection and the overall objective.

A benefit of this project is upon detecting the obstacle, it provides an instruction such as move left , move right to the user in order to avoid the obstacle all together. The proposed project will make considerations to include a narrative for visually impaired person to decide what action to take if at all necessary.

This project goes into detail of the current infrastructure for assistance of visually impaired such as guide dogs, canes and human guides. It breaks down the advantages and drawbacks of each option. One benefit of this the project was trialled with visually impaired users and the results were noted to determine the efficacy of the trial.

An important thing to note is the use of a belt with sensors as walking sticks can only pick up objects below the waist usually and not above the waist around the heart and head.
The object recognition technique detects indoor structures such as stairways, doorways, etc. and things such as tables, computer monitors, etc. by a Gaussian Mixture Model-based pattern gratitude method.

This project differs in that it breaks down signage's text in the environment and converts that into audio for the blind person , for example stop sign with "STOP" being converted to text then audio.

The proposed project could contribute to this field by using image label to audio to provide the blind person with exactly what the object type is and also how far it is. It would enable the Visually impaired to have greater attention to detail from the environment and make sense of it rather than only avoid obstacles.

Object character recognition captures the image, and it will be converted to text. Text to speech will then be done using NLP.

The main advantage of our work is as follows:

- It can be detect more distant obstacles and easy to recognize the destination.
- Having feature to left and right turn alarm signal.
- Simple to use and low cost.
- It can detect the digs and water present in the ground. Device can be converted easily.
- Fewer accidents will be accrued from blind people.

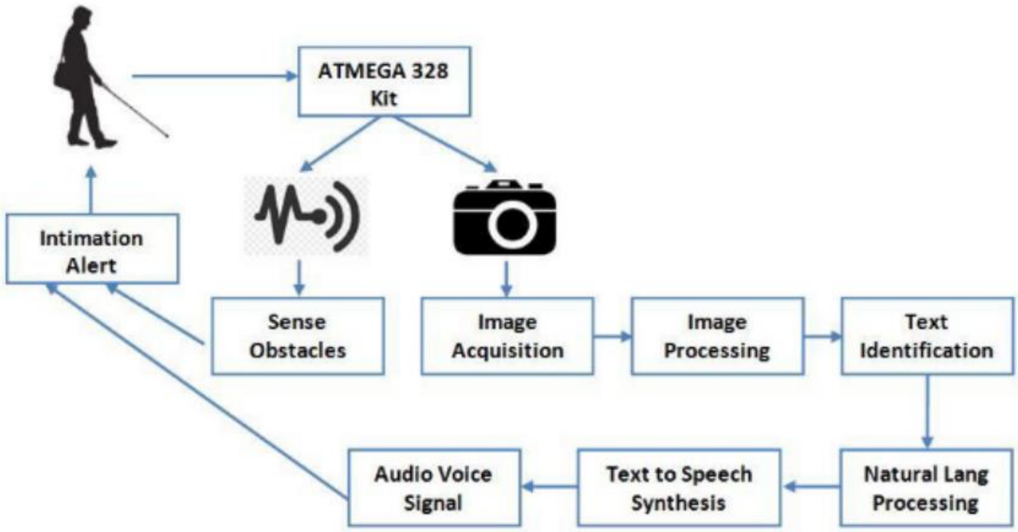


Figure 20- System architecture (Mr. M. Kannan et al., 2020)

Once obstacle is detected , camera activates, and image is sent to the cloud via Cloud Connect

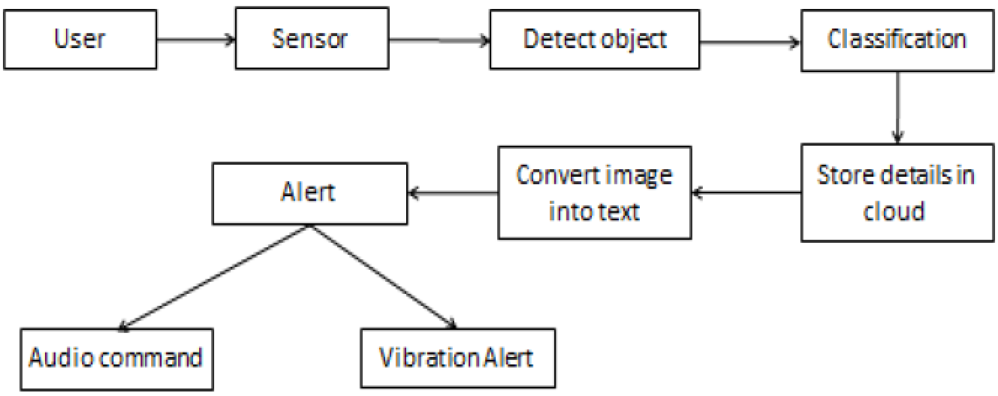


Figure 21- System processes diagram (Mr. M. Kannan et al. , 2020)

For this paper , the image must contain text and then this text is identified and voiced to the visually impaired person. Another benefit of this paper is its use of clear diagrams to explain its system architecture and processes and one the proposed project will make use of.

The proposed project could make an improvement in as such that identifies the object type i.e car , person , lamppost and perhaps it could determine threat level i.e “Person on chair” – threat level 25% . Or “moving car” threat level 60% . “Highway” – threat level 80%. Perhaps it could be utilised in scenarios where there is risk i.e “Weapon identified” threat level 90% or constantly be alert and scanning the environment for potential threats. It could identify road types and potential speed limits so as to determine level of risk for any pedestrian related activities. It could act as another way to determine the overall safety of their environment in Real time warning and detection and determine probabilities of risk involved. (al M. M., 2020)

2.12 The Proposed Project conclusion on the related works

An important thing to note for the related works was the unconventional use of a belt with sensors as a walking stick can only pick up objects below the waist usually and not above the waist around the heart and head.

The proposed project could expand the research in this area in as such as determining when a person is detected in the visually impaired persons area of concern it could provide obstacle detection and determine how safe the individual that triggers the obstacle detection is. In this way it helps the visually impaired person be constantly alert and scan the environment for potential threats as often people determine how safe an environment actually is.

The simplicity and effectiveness of obstacle detection corresponding to a passive buzzer frequency in order to provide an intuitive and simplistic way of detecting distance from the obstacle are something the proposed project could make use of.

This enables the visually impaired person to learn just how close the person is and if they are approaching or moving away from them in a simplistic way.

The related works are similar to the proposed project because there will be sensors deployed to detect the obstacle using distance. The sensors to carry out this operation are sonar.

The proposed project will not make use of user-android application but could very well make use of both GPS and google maps , as well as Voice and Call functions to the microprocessor in order to allow user communication to the device. The proposed project will make use of Raspberry pi as the main processing unit but there is scope for the Arduino to be included later if it is proven to make a viable contribution.

From the interviews involving the visually impaired, it was clarified that the main things the blind will have issues with are: Road markings, pedestrian crossings, edges , curbs, sidewalk, trees, holes, barriers to construction , cars , cyclists , cycle lanes , lamp posts, posts boxes, post office, cars on the road and this report highlighted these things effectively. Interesting to note however is that these obstacles belong to those in first world countries and the majority of visually impaired are low income or live in 3rd tier countries.

The proposed project will aim to make use of the research and interview carried out in this paper in order to have a clearer evaluation for specific objects and use case scenarios as such will determine whether the obstacle/person is beneficial or threatening to the visually impaired person.

Finally proposed project will make use of MobileNet CNN for image classification , Tensorflow lite models and a custom-built dataset with F1 classification reports in order to assess performance for its prototype.

Chapter 3 – System Design of Project

3.1 Introduction

This chapter will go through the software and hardware of the project and how it is implemented. Diagrams of the system presented to showcase how components communicate with one another. Data Collection will be explored and explained as well as the Deep learning Model.

Software:
- Tensorflow with Tensorflow lite and Keras
- Raspberry pi OS
- Deep learning CNN model : MobileNetV2
- Thonny python IDE
- Espeak TTS engine
- Google Colab
- Download All Images (Google Chrome Extension)

Datasets:-
- Google Images (selected from Google search, split into two folders)

Hardware:-
- Raspberry Pi 4
- SD Card
- 3.5mm audio cable
- HC-SR04 (Ultrasonic Sensor)
- Piezo Passive Buzzer Sounder
- 1k ohm and 2kohm resistor to divide voltage
- Adeept GPIO Expansion Kit for Raspberry Pi T-Type GPIO Breakout Expansion Board
- 40 Pin Rainbow Ribbon Cable
- 830 Points Solderless Breadboard
- NextPro Raspberry Pi Camera Module 5M 1080p
- Anker Soundcore mini (Speaker)

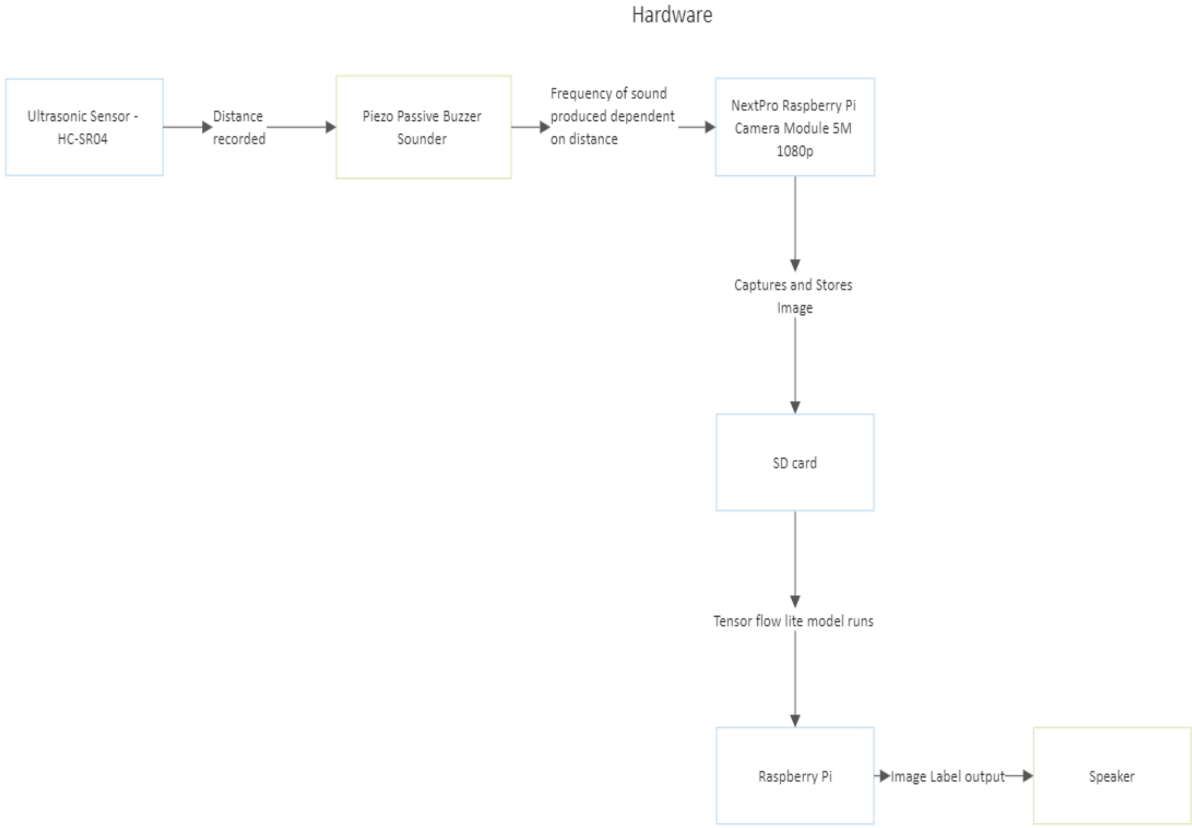


Figure 22- Hardware process of Proposed Project

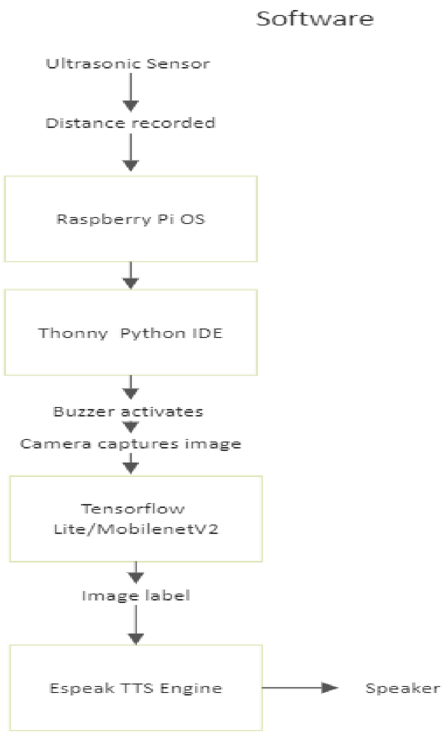


Figure 23 – Software process of Proposed Project.

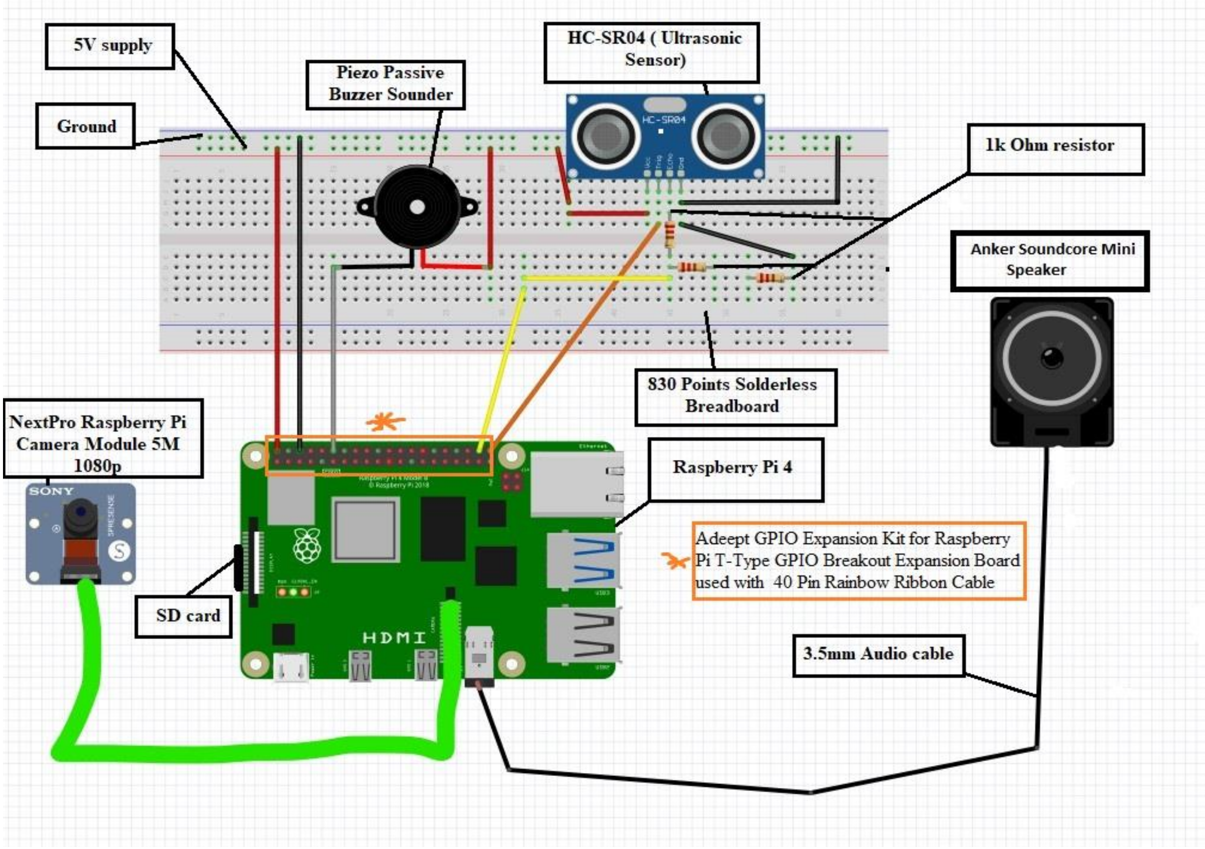


Figure 24- Schematic of Hardware of proposed project

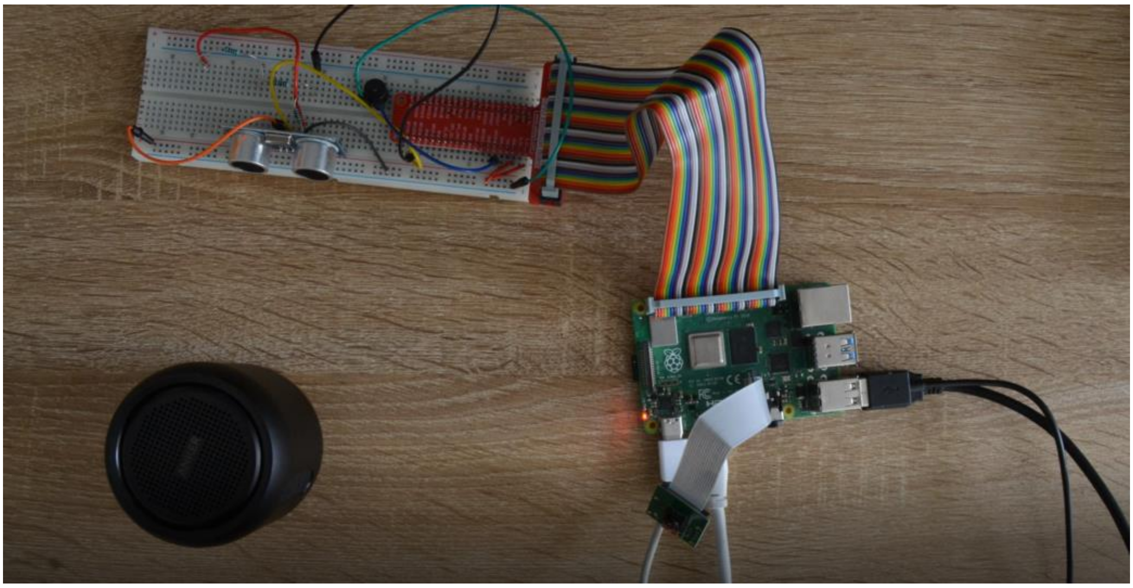


Figure 25- Capture of the proposed project hardware

3.2 Data collection

The Data was collected using a google search. There were total neutral images of 510 and total of threat images of 547 . Data downloaded using chrome extension application “Download All Images” with image type specified to be JPEG. Using Image Data Generator , images were processed into a target size of (224,224) and rescaled into the 256 channels for Red , Gren and Blue. The Data was split into Training: 80% (845 images) , Validation: 10% (105 images) and Test: 10% (107 images) .

Neutral images consisted of human beings of different race and gender with no weapons and effort exuded in order to include humans holding drills and other household tools. Threat images consisted of an assortment of human beings carrying explosive devices , guns from small to large and knives. All images were then trained using Google Collaboratory cloud environment.

3.3 Deep learning and Tensorflow lite model

Created using a custom dataset and using Google Collaboratory.
MobileNetV2 was chosen for its speed and single stage detection capability. Batch size of 32 was chosen throughput. Images were split into Training : 80% : validation :10% and test: 10%. Learning speed of 0.001 used with optimiser SGD and epoch size of 14.

Google Colab Code accessible here:
https://colab.research.google.com/drive/1oE0yKXBjJhKSjD5WPJxESuVSBkucWxUD?usp=sharing

CPU used on Colab :

Intel(R) Xeon(R) CPU @ 2.20GH

Essentials	
Product Collection	Legacy Intel® Xeon® Processors
Code Name	Products formerly Prestonia
Vertical Segment	Server
Status	Discontinued
Lithography ?	130 nm
Performance Specifications	
Total Cores ?	1
Processor Base Frequency ?	2.20 GHz
Cache ?	512 KB L2 Cache
Bus Speed ?	400 MHz
TDP ?	61 W
VID Voltage Range ?	1.5V
Supplemental Information	
Embedded Options Available ?	Yes
Package Specifications	
Sockets Supported ?	PPGA603
T_{CASE} ?	B0=72°C; C1+D1=75°C
Processing Die Size	131 mm²
# of Processing Die Transistors	55 million
Advanced Technologies	
Intel® Turbo Boost Technology ‡ ?	No
Intel® Hyper-Threading Technology ‡ ?	Yes
Intel® Virtualization Technology (VT-x) ‡ ?	No
Instruction Set ?	32-bit

Figure 26- Intel(R)Xeon(R) CPU performance

Figure 27- Custom built dataset for Deep learning

```
# Flow from directory method is possibe in this scenario because the class folders are aleady named.
train_generator = train_datagen.flow_from_directory(
        '/content/output/train/', # This is the source directory for training images
        target_size=(224, 224),
        batch_size=32,
        # Since we use binary_crossentropy loss, we need binary labels
        class_mode='binary')

validation_datagen = ImageDataGenerator(rescale=1/255)

# Flow training images in batches of 128 using train_datagen generator
validation_generator = validation_datagen.flow_from_directory(
        '/content/output/val/',  # This is the source directory for training images
        target_size=(224, 224),
        batch_size=32,
        # Since we use binary_crossentropy loss, we need binary labels
        class_mode='binary')


test_datagen = ImageDataGenerator(rescale=1/255)
test_generator  = test_datagen.flow_from_directory(
    '/content/output/test',
    target_size=(224,224),
    batch_size = 32,
    class_mode = 'binary'
)
```

```
Found 845 images belonging to 2 classes.
Found 105 images belonging to 2 classes.
Found 107 images belonging to 2 classes.
```

Figure 28- Training , Validation and Test images into corresponding imagedatagenerator with rescale

```
[ ] mobilev2_base = MobileNetV2(include_top = False,
                             weights="imagenet",
                             input_shape=(224,224,3)
                            )


    mobilev2_base.trainable = False #Base is not trainable in this case. Only want to train the last two layers only.

    model = keras.Sequential([
        mobilev2_base,
        layers.Flatten(),
        layers.Dropout(0.25),
        layers.Dense(256, activation = 'relu'),
        layers.Dense(1, activation = 'sigmoid')
    ])
```

Model Summary

```
[ ] model.summary()

    Model: "sequential_2"
    _________________________________________________________________
     Layer (type)                Output Shape              Param #
    =================================================================
     mobilenetv2_1.00_224 (Funct  (None, 7, 7, 1280)       2257984
     ional)

     flatten_2 (Flatten)         (None, 62720)             0

     dropout_2 (Dropout)         (None, 62720)             0

     dense_4 (Dense)             (None, 256)               16056576

     dense_5 (Dense)             (None, 1)                 257

    =================================================================
    Total params: 18,314,817
    Trainable params: 16,056,833
    Non-trainable params: 2,257,984
```

Figure 29- Model architecture

Model Alogrithm , Batch_size and Epoch congiguration parameters

```
#Model Algorithm selection

#model.compile(optimizer = Adam(learning_rate=0.001, decay = 1e-8, beta_1 = 0.9, beta_2 = 0.999), loss='bin
#model.compile(optimizer = RMSprop(learning_rate = 0.001), loss = 'binary_crossentropy', metrics = ['binary
#model.compile(optimizer = Adagrad(learning_rate=0.0001), loss='binary_crossentropy', metrics=['binary_accu
model.compile(optimizer = SGD(learning_rate=0.001), loss='binary_crossentropy', metrics=['binary_accuracy']
```

Model fitting

```
[ ] history = model.fit(
        train_generator,
        #steps_per_epoch=8,
        epochs=14,
        verbose=1,
        validation_data=validation_generator,
        #validation_steps=8,
        #callbacks=[callbacks]
```

Figure 30- Model testing parameters - note optimiser SGD , learning rate of 0.001 , epochs of 14 and batch size of 32 throughout.

```
# Convert the model.
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()

converter = tf.lite.TFLiteConverter.from_keras_model(model)
converter.optimizations = [tf.lite.Optimize.DEFAULT]

tflite_model_quant = converter.convert()

# Save the model.
with open('model.tflite', 'wb') as f:
  f.write(tflite_model)

with open('model_quant.tflite', 'wb') as f: # Neccesary to run in the raspberry pi environment
  f.write(tflite_model_quant)

WARNING:absl:Function `_wrapped_model` contains input name(s) mobilenetv2_1.00_224_input with unsupported characters which will be renamed to mobilenetv2_1_00_224_input in the SavedModel.
WARNING:absl:Function `_wrapped_model` contains input name(s) mobilenetv2_1.00_224_input with unsupported characters which will be renamed to mobilenetv2_1_00_224_input in the SavedModel.

[ ] TF_MODEL_FILE_PATH = 'model.tflite' # The default path to the saved TensorFlow Lite model

interpreter = tf.lite.Interpreter(model_path=TF_MODEL_FILE_PATH)

[ ] interpreter.get_signature_list()

{'serving_default': {'inputs': ['mobilenetv2_1.00_224_input'],
  'outputs': ['dense_5']}}

[ ] classify_lite = interpreter.get_signature_runner('serving_default')
classify_lite

<tensorflow.lite.python.interpreter.SignatureRunner at 0x7f0bc87f01d0>
```

Figure 31- Model trained , Converted into Tensorflow lite model which is quantised to work in the raspberry pi environment.

3.4 Implementation – Software and Hardware

Link for the main() python file here :

https://drive.google.com/file/d/14Nr5S3lCq8fGmgiU-NoqSJ6T8a3whzMf/view?usp=sharing

```
import time
import picamera
from picamera import PiCamera
import numpy as np
import cv2
from tflite_runtime.interpreter import Interpreter
from PIL import Image
import numpy as np
from picamera import PiCamera
from time import sleep
from num2words import num2words
from subprocess import call

def load_labels(path): # Read the labels from the text file as a Python list.
  with open(path, 'r') as f:
    return [line.strip() for i, line in enumerate(f.readlines())]

def set_input_tensor(interpreter, image):
  tensor_index = interpreter.get_input_details()[0]['index']
  input_tensor = interpreter.tensor(tensor_index)()[0]
  input_tensor[:, :] = image

def classify_image(interpreter, image, top_k=1):
  set_input_tensor(interpreter, image)

  interpreter.invoke()
  output_details = interpreter.get_output_details()[0]
  output = np.squeeze(interpreter.get_tensor(output_details['index']))

  scale, zero_point = output_details['quantization']
  output = scale * (output - zero_point)

  ordered = np.argpartition(-output, 1)
  return [(i, output[i]) for i in ordered[:top_k]][0]

import RPi.GPIO as GPIO
TRIG = 21# I Trigger at GPIO 21
ECHO = 20 #I set the reutrn signal to 20
GPIO.setmode(GPIO.BCM)
GPIO.setup(TRIG,GPIO.OUT) # output the audio
GPIO.setup(ECHO,GPIO.IN) # recieve the audio


while True:
```

Figure 32 - Overview of libraries used in Python

1 – On the Raspberry Pi OS , in Thonny IDE the main python code is executed in a while loop. The ultrasonic sensor detects obstacles and returns how far away they are.

```
while True:

#1
### This operation will determine how far an object is by its frequency
### using ultrasound sensors

    TRIG = 21# I Trigger at GPIO 21
    ECHO = 20 #I set the reutrn signal to 20
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(TRIG,GPIO.OUT) # output the audio
    GPIO.setup(ECHO,GPIO.IN) # recieve the audio

    print("Distance measurement in progress")
    GPIO.output(TRIG,False) # Trigger pin to stabilise the sensor
    print("Sensor is settling")
    time.sleep(0.2)
    GPIO.output(TRIG,True)# Trigger pin high to send burst of audio
    time.sleep(0.00001)

    GPIO.output(TRIG,False)

    echo_state = 0
    while echo_state == 0:
        echo_state = GPIO.input(ECHO)
        pulse_start = time.time()

    #while GPIO.input(ECHO)==0:
        #pulse_start = time.time()

    while GPIO.input(ECHO)==1:
        pulse_end = time.time()


    pulse_duration = pulse_end - pulse_start
    distance = pulse_duration*17150 # speed of sound in air at sea level
    distance=round(distance,2) # round the distance

    print("distance:",distance,"cm") # I have my distance from the sensor to the object
```

Figure 33- Ultrasonic sensor - code in Python

2 – A distance is recorded, and the python main code has limits placed between 0.3m and 1.2m in which ever increasing intervals of 0.3m result in a lower frequency noise that corresponds to that distance interval which is generated by the passive buzzer so as to inform the user. The further an obstacle the lower the frequency generated. The user can learn the frequencies in order to quickly determine how far away someone is intuitively and whether or not they are moving away from them or approaching them. Less than 0.3m a frequency of 2kHz is produced , between 0.3m and 0.6m a frequency of 1.8kHz , between 0.6m and 0.9m a frequency of 1.5kHz and finally between 0.9m and 1.2m a frequency of 1kHz is produced.

```
if 0<distance<30:
    buzzer = 18
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(buzzer,GPIO.OUT)

    #a PWM signal has been defined and frequency will vary as the objects distance
    #to the person varies
    buzzer = GPIO.PWM(buzzer, 2000) # Set frequency to 2 Khz
    buzzer.start(20) # Set dutycycle to 20
    time.sleep(0.5)

    GPIO.cleanup() # reset pins

elif 30<distance<60:
    buzzer = 18
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(buzzer,GPIO.OUT)
    buzzer = GPIO.PWM(buzzer, 1800) # Higher frequency means objects is closer
    buzzer.start(20)
    time.sleep(0.8)

    GPIO.cleanup()
    ###

elif 60<distance<90:
    buzzer = 18
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(buzzer,GPIO.OUT)

    buzzer = GPIO.PWM(buzzer, 1500)
    buzzer.start(20)
    time.sleep(0.6)

    GPIO.cleanup()

elif 90<distance<120:
    buzzer = 18
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(buzzer,GPIO.OUT)

    buzzer = GPIO.PWM(buzzer, 1000)
    buzzer.start(20)
    time.sleep(0.8)

    GPIO.cleanup()
```

Figure 34 - Passive piezo buzzer and the distance corresponding to frequency produced.

3 – The camera activates and captures the image and stores it in the SD card memory allocation on the Raspberry Pi OS and for further processing using the Tensorflow lite model.

```
#3
### Captured image and label retrived and Tensor flow lite DL deployed
    print("Camera is activating")
    camera = PiCamera()
    camera.start_preview()
    sleep(2)
    camera.capture('/home/pi/TFLite_MobileNet/image.jpg')
    camera.stop_preview()
    camera.close()




    data_folder = "/home/pi/TFLite_MobileNet/"

    model_path = data_folder + "model.tflite"
    label_path = data_folder + "labels.txt"

    interpreter = Interpreter(model_path)
    print("Model Loaded Successfully.")

    interpreter.allocate_tensors()
    _, height, width, _ = interpreter.get_input_details()[0]['shape']
    print("Image Shape (", width, ",", height, ")")

## Load an image to be classified.
    image = Image.open(data_folder + "image.jpg").convert('RGB').resize((width, height))

##Classify the image.
    time1 = time.time()
    label_id, prob = classify_image(interpreter, image)
    time2 = time.time()
    classification_time = np.round(time2-time1, 3)
    print("Classificaiton Time =", classification_time, "seconds.")

## Read class labels.
    labels = load_labels(label_path)

## Return the classification label of the image.
    classification_label = labels[label_id]
    accuracy = np.round(prob*100, 2)
    print("Image Label is :", classification_label, ", with Accuracy :",accuracy , "%.")

    accuracy_str = str(accuracy)
## Prepare the label and probability to be concatenated into a string ready for speech
    text_speech = classification_label+" with "+accuracy_str+" percent accuracy"
    print(text_speech)
```

Figure 35- Camera code in Python

4- The Tensorflow lite model which uses Tensorflow interpreter allocates a tensor to the image and begins image classification using the CNN MobileNetV2 architecture that was already trained before in the Google Colab cloud environment on a dataset of 1000 images. The label is output along with its degree of certainty.

```
def load_labels(path): # Read the labels from the text file as a Python list.
  with open(path, 'r') as f:
    return [line.strip() for i, line in enumerate(f.readlines())]

def set_input_tensor(interpreter, image):
  tensor_index = interpreter.get_input_details()[0]['index']
  input_tensor = interpreter.tensor(tensor_index)()[0]
  input_tensor[:, :] = image

def classify_image(interpreter, image, top_k=1):
  set_input_tensor(interpreter, image)

  interpreter.invoke()
  output_details = interpreter.get_output_details()[0]
  output = np.squeeze(interpreter.get_tensor(output_details['index']))

  scale, zero_point = output_details['quantization']
  output = scale * (output - zero_point)

  ordered = np.argpartition(-output, 1)
  return [(i, output[i]) for i in ordered[:top_k]][0]
```

Figure 36- DL model functions used for image classification.

5 – Using the Espeak TTS engine , the label is transformed into speech and then the speaker outputs this to user to identify whether the obstacle is a threat or neutral and its accuracy (degree of certainty).

```
#4
### Final part is to return the device detected via audio device interface
### throug text to speech

    cmd_beg= 'espeak '
    cmd_end= ' | aplay /home/pi/Desktop/Text.wav  2>/dev/null' # To play back the stored .wav file and to dump the std errors to /dev/null
    cmd_out= '--stdout > /home/pi/Desktop/Text.wav ' # To store the voice file

#Replacing ' ' with '_' to identify words in the text entered
    text_speech = text_speech.replace(' ', '_')

#Calls the Espeak TTS Engine to read aloud a Text
    call([cmd_beg+cmd_out+text_speech+cmd_end], shell=True)

    time.sleep(0.3)

### The accuracy enables the visually impaired person to make a discernment
### with regards to the object detetced.
```

Figure 37-Text to speech using Espeak TTS in Python

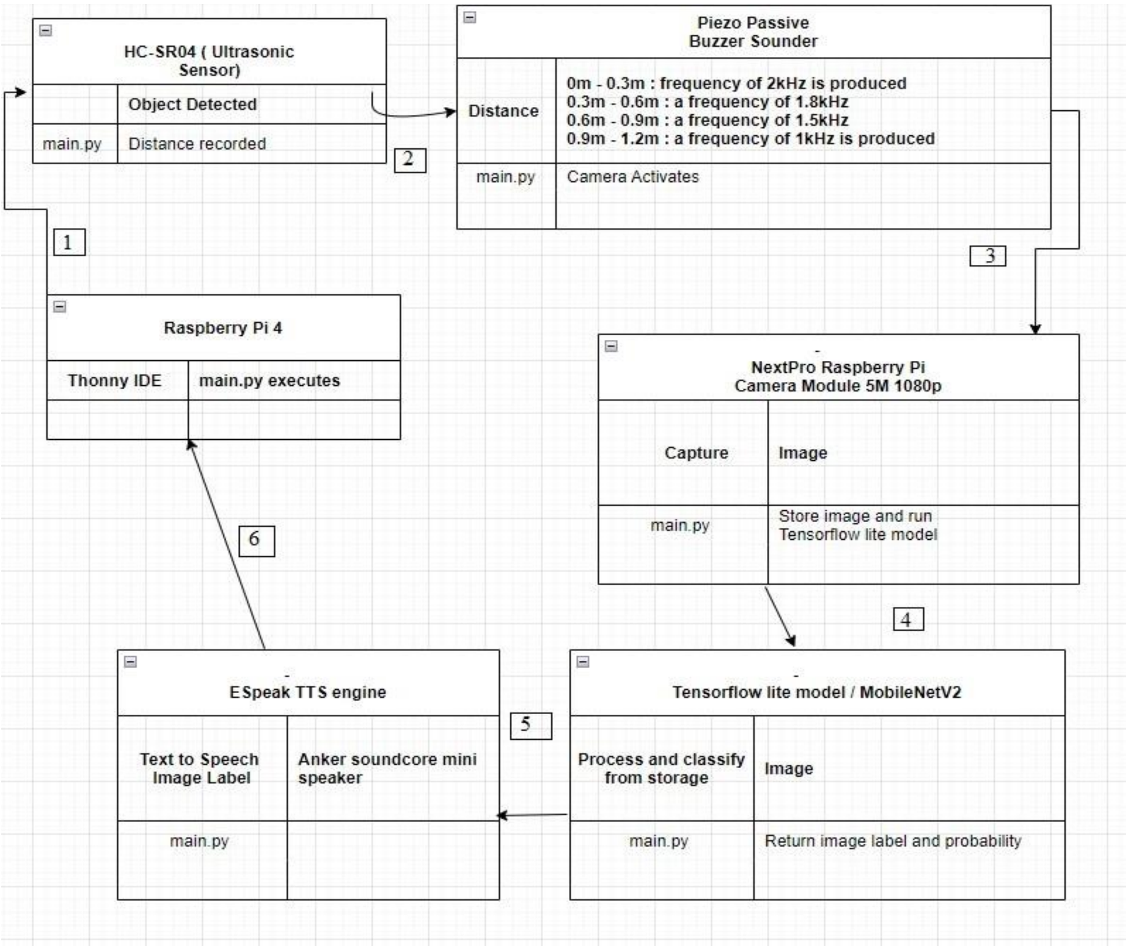


Figure 38- Transactional Model

Chapter 4 – Testing and Evaluation

4.1 Introduction

The assistive device was tested at different distances at which the passive buzzer frequency changed and notified the user how far the obstacle is. This operation was fully functional , however there were a few issues in regard to the Espeak TTS engine which didn't activate and caused audio issues, and which needs to be addressed , probably with the replacement of better software. Below will be the evaluation of how well the assistive device protects and warns the user in different use case scenarios as well as the accuracy of the deep learning model itself. Because the project developed was only a prototype , the practical aspects such as useability and durability were unable to be tested and evaluated at this stage but will be conducted in future iterations if a more developed prototype is made.

4.2 Deep learning model

The projects CNN model was evaluated on its accuracy , confusion matrix and its area under curve.

The CNN architecture used in the project is shown below:

Summary of model :

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 mobilenetv2_1.00_224 (Funct  (None, 7, 7, 1280)       2257984
 ional)

 flatten_2 (Flatten)         (None, 62720)             0

 dropout_2 (Dropout)         (None, 62720)             0

 dense_4 (Dense)             (None, 256)               16056576

 dense_5 (Dense)             (None, 1)                 257

=================================================================
Total params: 18,314,817
Trainable params: 16,056,833
Non-trainable params: 2,257,984
```

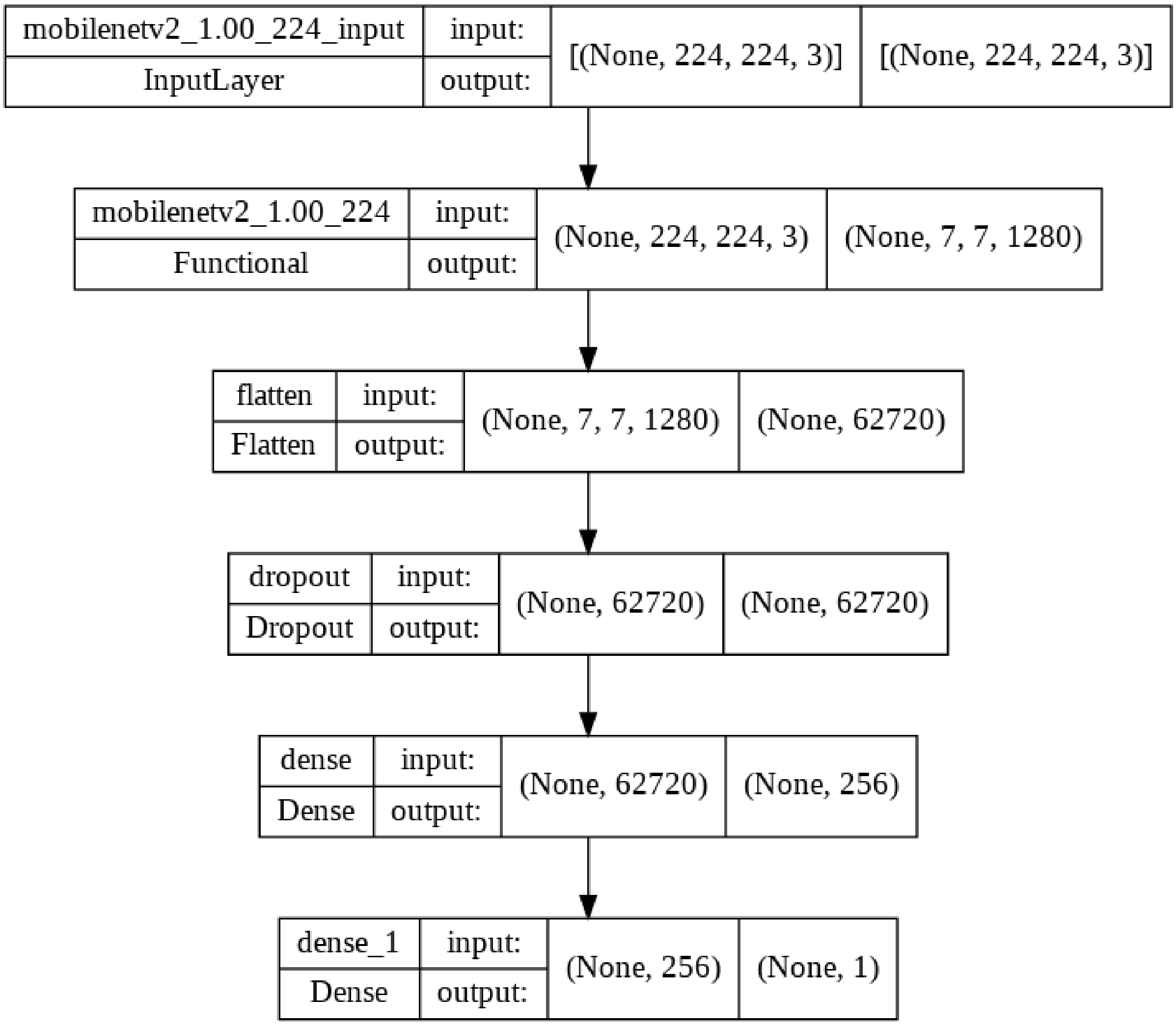


Figure 39- CNN architecture used in project

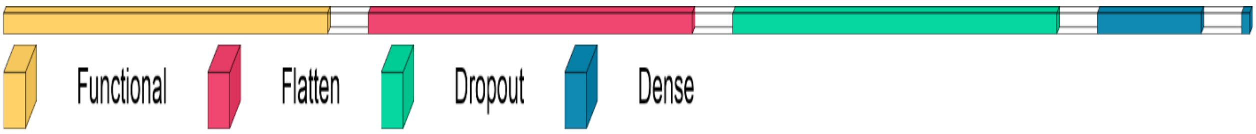


Figure 40- Visualization of the CNN used in project

Table 2- Classification report for Deep learning model

Classification	**Precision**	**Recall**	**F1 - score**	**Accuracy**
Neutral(0)	0.49	0.49	0.49	0.51
Threat (1)	0.54	0.54	0.54	0.51

The model had a higher f1 score for threat (1) than for neutral (0) which is important to mention because it is crucial that threats are detected because their consequence is of higher magnitude.

A precision score of 0.54 for threat meant that a higher proportion of the actual threat images were successfully predicted as being a Threat with precision . Neutral's precision score of 0.49 is lower than half but this has far less consequences on safety because it meant less Threat images were being mispredicted as being Neutral which is a positive. Recall scores replicated the precision score across threat(1) and neutral (0). Threat has the higher score which means there is less likelihood of a threat image being mis predicted than a neutral image is.

Precision , recall and F1 scores are higher for threat(1) which is good because this is where performance matters the most.

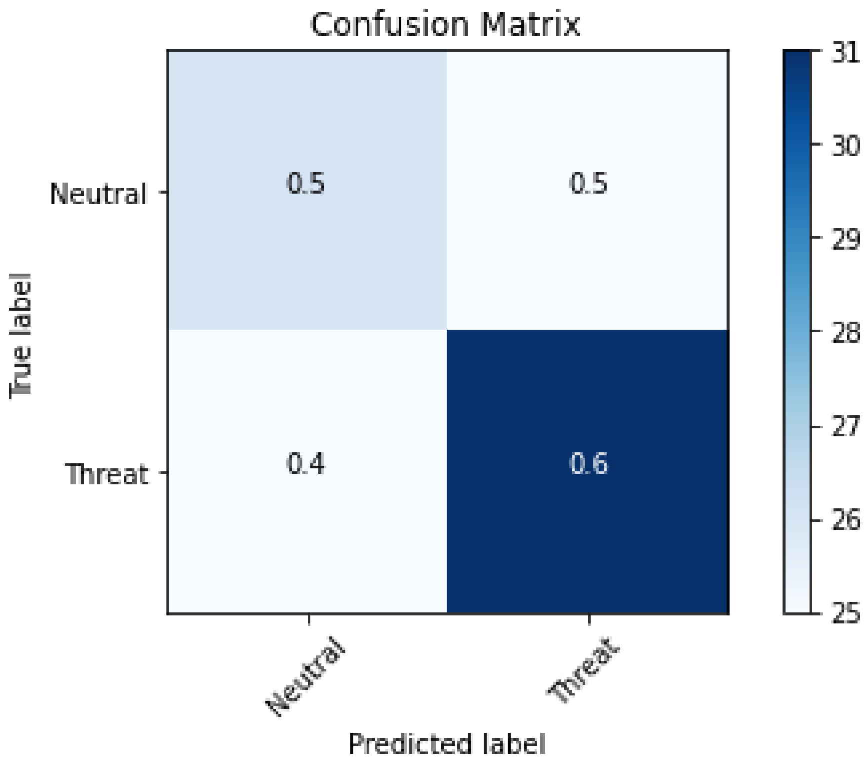


Figure 41- Confusion matrix for project

The confusion matrix reflects what the classification report has shown in that the threat(1) has a proportion of 60% of being successfully predicted as threat(1) and only 40% as being misclassified for neutral(0). The neutral has a 50% proportion across its prediction which is okay because the consequences of mis predicting a neutral as being a threat would mean hypervigilance which is better than compromising on safety. This shows that the model is high performing for this particular context.

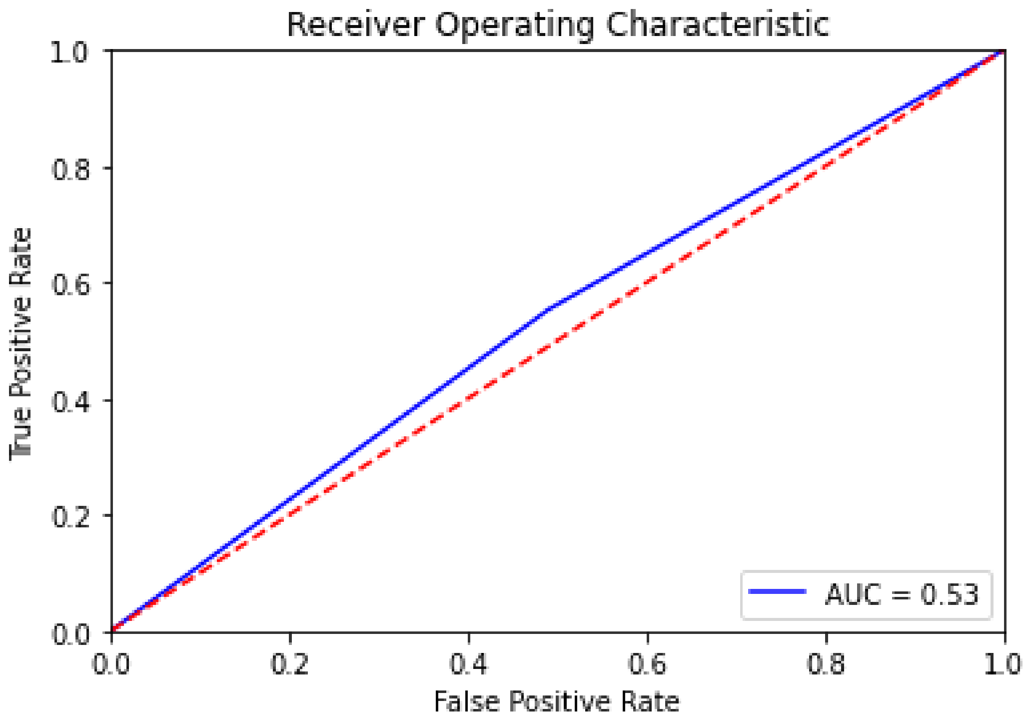


Figure 42- AUC for the project

The AUC is 0.53 which is higher than 0.5 meaning that there is a high chance that the classifier will be able to distinguish the positive class values(Neutral) from the negative class values(Positive) . This models discernment overall is good and it is not predicting class values at random even though it preforms poorly for the neutral class, it excels for the threat class.

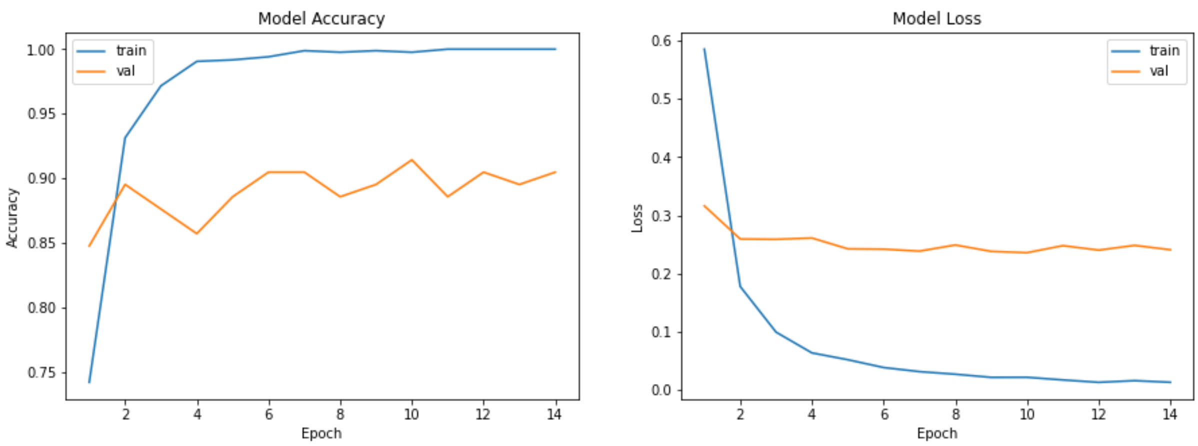


Figure 43- Accuracy for the project

The overall accuracy which is the ability to predict correctly from all positive and negative classes in general was around 85-90% for the validation set , the loss remained around 20%-30%

4.3 Use Case Scenarios

There were 2 use case scenarios to be carried out , which were the following ;

- Scenario 1: Ability for the assistive device to identify that a tool holding individual i.e Drill is not a threat.
- Scenario 2: Effectiveness and accuracy for the assistive device at differing distances for scenarios of Drill , Knife, Gun and Neutral individual.

  It is important to note that any references to gun in the project refers to a toy gun replica in accordance with the UK's gun laws.

Figure 44- Assistive device accuracy at distance 0m - 0.3m

Figure 45- - Assistive device accuracy at distance 0.3m - 0.6m

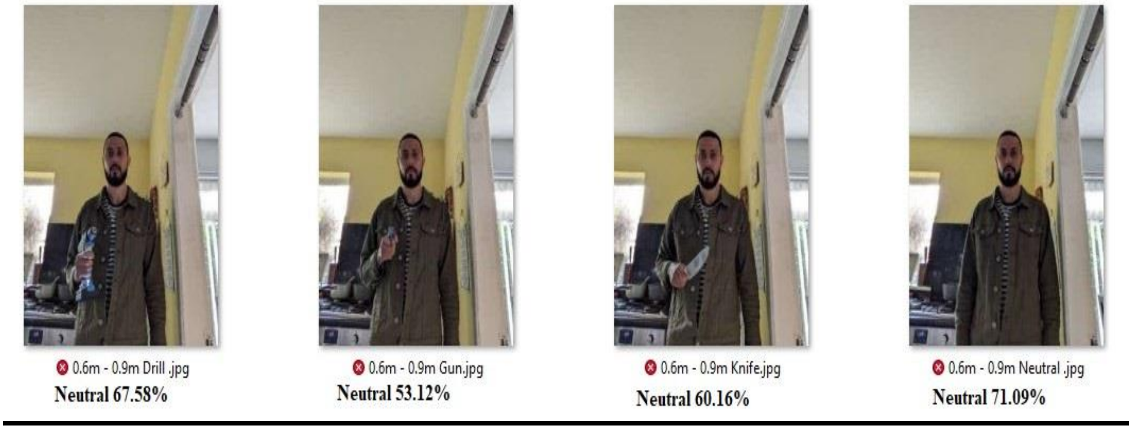


Figure 46- Assistive device accuracy at distance 0.6m - 0.9m

Figure 47- Assistive device accuracy at distance 0.9m - 1.2m

Scenario 1: Ability for the model to tell the difference between a Drill holding individual and the Knife and the Gun wielding individual

The conclusion for this is at close range of 0m – 0.3m regarding the Drill, the assistive device identified the person as being a Threat with 78.52% accuracy which would mean that at this range the visually impaired person is in danger. For the other distances the visually impaired person was not a threat and thus his life not at risk. This is a positive because it shows intelligence that a person who uses normal handheld tools is not a risk. The argument could be made that the CNN model is intelligent and thus identified the person with the drill at close range as being a risk as a Drill can be used to cause damage but only at close range and thus is a threat , but this is only speculation at the moment and needs other handheld tools , tools that don't have the ability to cause damage at the range 0m – 0.3m to prove whether this is true or not true.

Otherwise for distances greater than 0.3m the model has no problem in identifying a person with a Drill is not a threat and does not mistake this as a weapon.

Scenario 2: Effectiveness and accuracy at differing distances for scenarios of Drill , Knife, Gun and Neutral individual.

At distance 0.9m – 1.2m:
The assistive device was perfect in that it correctly predicted that the Gun and the Knife were the Threat and thus protected the Visually assistive device , although it only gave a score of 51.95% accuracy for the Gun wielding individual. The Neutral person was neutral with a score of 51.17% which is close to 50% but its repercussions are not as high as the main aim of the device is to be protect and be overly cautious is not a concern. Perhaps the lighting enhanced the photo quality.

At distance 0.3m – 0.6m and 0.6m – 0.9m:
The assistive device declared all images as being Neutral which was a major concern , especially as the Gun and the Knife certainty increased in score as the distance became closer. The deep learning model needs a larger dataset to learn from and greater emphasis on identifying weapon wielding individuals with extra attention on making sure images at this distance interval are used. The camera should also be upgraded to improve image quality.

At distance 0m – 0.3m:
The assistive device correctly predicted what was Neutral and what was a Threat with the exception of the Drill which has already been discussed in scenario 1. This is a great positive, but it needs to improve its efficacy at the distance 0.3m – 0.6m and 0.6m – 0.9m.

Overall, the assistive device is effective at 0-0.3m and 0.9-1.2m which are the extremes. However, it performed poorly at distance interval 0.3m – 0.9m and needs a larger dataset of which to train from , especially at this range in order to be effective and useful for the visually impaired person.

4.2 Evaluation conclusion and comparison to projects in the related works

Overall, in comparison this proposed project used cheaper hardware such as Pi and Ultrasonic sensors which is what it aimed to achieve as most of the visually impaired come from lower income backgrounds and this makes it more accessible.

The project is similar to some of those in the related works in that a passive frequency buzzer was used to intuitively provide distance at different frequencies to the visually impaired at different distance intervals and the use of camera for image classification was implemented with a text to speech engine implemented to communicate those results.

It differs however from the projects in related works in that it takes consideration to its environment regarding people and not inanimate objects which has been extensively explored. This formed the contribution of this research as a unique point of view whereby it took consideration as to environment as being hazardous or non-hazardous but only due to the people that inhabit it. This approach being that 3rd world countries are dangerous because of the people who live there and not only because of obstacles such as roads or streetlamps which are equally important but the reality being some parts of the world don't have streetlamps or a road to identify and things like safety often being determined by who is around them. See Related works section in Chapter 2 and in particular proposed project conclusion on the related works.

The proposed project met all its aims and objectives with the exception of making a portable compact walking stick that housed the electronics but nonetheless can be considered a success. Its ability to be accurate held up well when the deep learning model was assessed but it remained inconclusive when assessed during the Use Case Scenarios as it performed inconsistently. Thus, more works needs to be carried out in order in making a stronger assistive device. Finally , in the relation to related works the proposed project remains open on a decision to be made as to whether or not a walking stick or something that could be implemented fully body such as protective vest or utility belt that was used in the related works should be utilised.

Chapter 5 – Recommended further work

5.1 Introduction

This chapter is to give a sense of direction to any unexplored areas of the research that would advance this project and contribute to the research area.

5.2 Next steps

The work can be taken further by building the next prototype and making it compact and battery powered with multiple ultrasonic sensors with due consideration for obstacles at feet level , waist level, eye level , front and behind the visually impaired person. Cameras of higher quality should also be used.

A larger dataset of varying distances (especially between 0.3m and 0.9m) can be given to train and transfer learning used for an already trained image classifier to be used on whether the person is of threat or of neutral significance during the training stage. The consideration of implementing a system that is an object detector with constant video streaming that only provides feedback when a danger to the person is detected would be the next step as it would constantly be alert and thus more akin to human nature especially when safety is concerned vs the current wait to the ultrasonic sensor is triggered operation and then capture and classify an image.

GPS , maps and emergency SMS texts to the relevant authorities and kin, should be explored as methods to escape danger keeping in mind that the objective and main intention for this research and body of work is to keep the visually impaired mobile with safety as the highest priority.

Future work considerations need to be considered to undertaking image segmentation research for small portable devices for the visually impaired and how it can provide warnings for them in order to enable them to decide whether or not the outside surface that they are on is safe or not, whether there is a hole in the ground, or they are on a busy road with many cars. This can be taken further as such for discerning areas that are known for extreme hazards such as land mines or deep excavations . Processes that feed into identifying whether or not road is present are fundamental but a discernment for pedestrian crossings and stationary vehicles to enable crossing would also be an excellent contribution to this body of work.

Importantly, the cost of the project must be taken into account because as of today the people who are visually impaired live primarily in 3rd world countries , countries where there are a whole set of different risks than those associated living in the 1st world countries. Thus, it is recommended that one thinks not only about the daily hazards of a high-income country and their interviews but really breakdown the daily living of those in 3rd world countries and try to understand their habitual daily rituals and potential opportunities to make an improvement and contribution.

This project has tried to meet the risks with solutions such as the discernment of a threat or a neutral person. If certain technology such as Intel or Coral stick can be implemented to accelerate the models learning rate, then this should be considered in the future with due consideration made on price.

Finally , future bodies of work could explore the ideas of facial expressions in order to determine a person’s mood and associate that with how they dress , if they carry a weapon and general overall demeanour. More situations can be trained for, and more research is needed to understand these situations and to meet the problem with an excellent solution.

Chapter 6 – Conclusion

6.1 Introduction

The end of this report and with the achievements , findings and obstacles discussed along with reflection of the project.

6.2 Reflections and contribution

Overall, a great deal of time had been spent on the research phase of the project which involved reading and trying to gather the project research requirements , aims and objectives in order to form something original.

A lot of time was spent on training and learning how the Raspberry Pi and its Linux OS works and then the subsequent construction an assistive device prototype along with the relevant code in order to make something functional. This drained a lot of time from making greater explorations in this research area but was a crucial part of the project.

More time could have been used in creating a custom-built dataset of different Use Case scenarios at different distances but again the time used on learning the fundamentals of the IOT on Raspberry Pi was used instead.

On reflection the project was unique and contributed something of value to the research area. The related works really helped form and decide the direction of the proposed project. By no means is this the end , but merely the beginning in that its aim is to inspire research in this area to think differently about the hazards for the visually impaired and satisfactorily address their needs adequately.

The achievements of the project were successful in regard to the original aims and objectives and the obstacles along the way were related to making the hardware communicate with another using the relevant software structure as well as implementing a custom-built deep learning model. All of this took a lot of time, and many struggles were experienced in exporting a deep learning model ready for use on the Raspberry Pi without any prior experience with the Pi's operating system requirements and functionality. With more time , the use of a portable power source as well as attachments for camera , sensors and Pi would have been implemented.

The findings of this research were that a visually assistive device can protect someone at distances of 0 -0.3m and 0.9m – 1.2m adequately but failed in those distances in between meaning its effectiveness was inconclusive. A more diverse and larger dataset must be considered in the next version of this project to ensure the visually impaired person is protected as well as identifying threats in real time using an object detector that can operate at even larger distances than those quoted. A safe system of escape thus must also be devised with use of GPS and SMS to let the relevant people and authorities know if the visually impaired person needs to be extracted from a hostile situation.

The project can be improved by having multiple sensors and cameras and being constantly alert to dangers rather than using the slower process of capturing an image then classifying it , this would then make it an obstacle detection process in real time and not only rely on the ultrasonic sensors being triggered. Due consideration must be given so as not to overwhelm the visually impaired person with too much information.

References

Aggarwal, A. (no date : online, n/a n/a). *Raspberry Pi Layout*. Retrieved from s3.amazonaws: https://s3.amazonaws.com/yw.public/GPIO-diagram-Raspberry-Pi-4.png - [Accessed on 02.09.2022]

Al, J. K. (2021). Computer Vision-based Assistance System for the Visually Impaired Using Mobile. *IEEE*, 10.

al, M. G. (2021). Raspberry PI Based Smart Walking Stick. *Research Gate*, 10.

Al, M. K. (2021). An Android-based Portable Smart Cane for Visually. *IEEE*, 6.

al, M. M. (2020). Iot Based Smart Stick Used for Visually Impaired People By Using Image Processing. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 7.

Bhandari, A. (2016:Online, June 2020). *AUC-ROC Curve in Machine Learning Clearly Explained*. Retrieved from analyticsvidhya: https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/#:~:text=The%20Area%20Under%20the%20Curve,the%20positive%20and%20negative%20classes. - [Accessed on 20.09.2022]

Data Driven Investor. (2021:Online, January 18). *What is the Tensor in deep learning?* Retrieved from Data Driven Investor: https://miro.medium.com/max/651/0*r3diVuJ6bnCeh9ri [Accessed on 04.09.2022]

Free Code Camp. (2017:online, October 23). *Want to know how Deep Learning works? Here's a quick guide for everyone*. Retrieved from freecodecamp: https://www.freecodecamp.org/news/want-to-know-how-deep-learning-works-heres-a-quick-guide-for-everyone-1aedeca88076/ [Accessed on 04.09.2022]

Freecodecamp - gradient descent. (2017:online, Ocotober 23). *Want to know how Deep Learning works?* Retrieved from freecodecamp: https://cdn-media-1.freecodecamp.org/images/d58SH7gTWXH9ZnoEd91iL-X3OUQcPCJj4J-j - [Accessed on 13.09.2022]

freecodecamp image. (2017:Online, October 23). *Want to know how Deep Learning works?* Retrieved from freecodecamp: https://cdn-media-1.freecodecamp.org/images/GDzpkmRqzE6KPOBfTfuShEziKgHWJa9djQ2O- [Accessed on 04.09.2022]

Hollemans, M. (2018:online, April 22). *Compression*. Retrieved from machinethink: https://machinethink.net/images/mobilenet-v2/Compression.png - [Accessed on 20.09.2022]

Hollemans, M. (2018:online, April 22). *MobileNet Version 2 - Depthwise Seperable Convolution png*. Retrieved from macchine think: https://machinethink.net/images/mobilenet-v2/DepthwiseSeparableConvolution.png - [Accessed on 12.09.2022]

Hollemans, M. (2018:online, April 22). *MobileNet Version2*. Retrieved from machinethink: https://machinethink.net/blog/mobilenet-v2/ - [Accessed on 12.09.2022]

IBM Cloud Education. (2020:online, July 15). *What is Machine Learning*. Retrieved from IBM: https://www.ibm.com/uk-en/cloud/learn/machine-learning - [Accessed on 26.06.2022]

Narkhede, S. (2018:online, May 9). *Confusion Matrix*. Retrieved from miromedium: https://miro.medium.com/max/324/1*Z54JgbS4DUwWSknhDCvNTQ.png - [Accessed on 12.09.22]

OpenCV. (2022:online, no date no date). *About*. Retrieved from OpenCV: https://opencv.org/about/ - [Accessed on 13.09.2022]

osyoo. (2018:online, September 18). *Micro bit Lesson — Using the Ultrasonic Module*. Retrieved from osoyoo: https://osoyoo.com/wp-content/uploads/2018/09/hc-sr04-1024x410.png = [Accessed on 13.09.2022]

Prabhu. (2018:Online, March 4). *Array of RGB Matrix*. Retrieved from Medium: https://miro.medium.com/max/462/1*CBY94wikMUCZMB4-Xxs-pw.png - [Accessed on 13/09/2022]

Prabhu. (2018:Online, March 4). *Image matrix multiplies kernel or filter matrix*. Retrieved from medium: https://miro.medium.com/max/922/1*kYSsNpy0b3fIonQya66VSQ.png - [Accessed on 13.09.2022]

Prabhu. (2018:Online, March 4). *Neural network with many convolutional layers*. Retrieved from medium: https://miro.medium.com/max/1400/1*XbuW8WuRrAY5pC4t-9DZAQ.jpeg - [Accessed on 13.09.2022]

Prabhu. (2018:Online, March 4). *Some common filters*. Retrieved from medium: https://miro.medium.com/max/698/1*uJpkfkm2Lr72mJtRaqoKZg.png - [Accessed on 13.09.2022]

Routley, N. (2021, October 4). *Mapped: Where are the World's Ongoing Conflicts Today?* Retrieved from Visual Capitalist: https://www.visualcapitalist.com/mapped-where-are-the-worlds-ongoing-conflicts-today/ - [Accessed on 31.08.2021]

Sarkar, A. (2021:online, June 25). *MobileNet parameter and accuracy comparison against GoogleNet and VGG 16*. Retrieved from towardsdatascience: https://miro.medium.com/max/640/1*oIJtvbsHhXuKPKKl-f3L6g.png -[Accessed on 21.09.2022]

Sharma, S. (2017:online, September 6). *ReLU v/s Logistic Sigmoid*. Retrieved from towardsdatascience: https://miro.medium.com/max/640/1*XxxiA0jJvPrHEJHD4z893g.png - [Accessed on 21.09.2022]

Uniqtech. (2019:online, February 18). *Illustration of a pixel and its RGB Value*. Retrieved from medium.com: https://miro.medium.com/max/628/1*KzSu3ytpLj8Dl0v2qVC1zw.png - [Accessed on 29.08.2022]

Uniqtech. (2019:online, Feb 18). *Illustration of a tensor of pixels with RGB values*. Retrieved from medium: https://miro.medium.com/max/624/1*nGOdf2Otg95W1LF5BflEhg.png - [Acessesed on 20.09.2022]

Venkatesh, B. (2021:Online, January 18). *What is Tensor in Deep Learning*. Retrieved from Data Driven Investor: https://medium.datadriveninvestor.com/what-is-the-tensor-in-deep-learning-77c2af7224a1 [Accessed on 04.09.2022]

WHO. (2021:online, October 14). *Blindness and vision impairment*. Retrieved from who: https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment -[Accessed on 13.09.2022]

Yegulalp, S. (2018:online, June 3). *What is TensorFlow? The machine learning library explained*. Retrieved from infoworld: https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html - [Accessed on 13.09.2022]

Appendix A

Design Materials

Software:

- Tensorflow with Tensorflow lite and Keras
- Raspberry pi OS
- Deep learning CNN model : MobileNetV2
- Thonny python IDE
- Espeak TTS engine
- Google Colab
- Download All Images (Google Chrome Extension)

Datasets:-

- Google Images (selected from Google search, split into two folders)

Hardware:-

- Raspberry Pi 4
- SD Card
- 3.5mm audio cable
- HC-SR04 (Ultrasonic Sensor)
- Piezo Passive Buzzer Sounder
- 1k ohm and 2kohm resistor to divide voltage
- Adeept GPIO Expansion Kit for Raspberry Pi T-Type GPIO Breakout Expansion Board
- 40 Pin Rainbow Ribbon Cable
- 830 Points Solderless Breadboard
- NextPro Raspberry Pi Camera Module 5M 1080p
- Anker Soundcore mini (Speaker)

Code for Hardware on Pi OS (Python)

```
import time
import picamera
from picamera import PiCamera
import numpy as np
from tflite_runtime.interpreter import Interpreter
from PIL import Image
import numpy as np
from picamera import PiCamera
from time import sleep
from num2words import num2words
from subprocess import call

def load_labels(path): # Read the labels from the text file as a Python list.
  with open(path, 'r') as f:
    return [line.strip() for i, line in enumerate(f.readlines())]

def set_input_tensor(interpreter, image):
  tensor_index = interpreter.get_input_details()[0]['index']
  input_tensor = interpreter.tensor(tensor_index)()[0]
  input_tensor[:, :] = image

def classify_image(interpreter, image, top_k=1):
  set_input_tensor(interpreter, image)

  interpreter.invoke()
  output_details = interpreter.get_output_details()[0]
  output = np.squeeze(interpreter.get_tensor(output_details['index']))

  scale, zero_point = output_details['quantization']
  output = scale * (output - zero_point)

  ordered = np.argpartition(-output, 1)
  return [(i, output[i]) for i in ordered[:top_k]][0]
```

```
import RPi.GPIO as GPIO
TRIG = 21# I Trigger at GPIO 21
ECHO = 20 #I set the reutrn signal to 20
GPIO.setmode(GPIO.BCM)
GPIO.setup(TRIG,GPIO.OUT) # output the audio
GPIO.setup(ECHO,GPIO.IN) # recieve the audio


while True:

#1
### This operation will determine how far an object is by its frequency
### using ultrasound sensors

  TRIG = 21# I Trigger at GPIO 21
  ECHO = 20 #I set the reutrn signal to 20
  GPIO.setmode(GPIO.BCM)
  GPIO.setup(TRIG,GPIO.OUT) # output the audio
  GPIO.setup(ECHO,GPIO.IN) # recieve the audio

  print("Distance measurement in progress")
  GPIO.output(TRIG,False) # Trigger pin to stabilise the sensor
  print("Sensor is settling")
  time.sleep(0.2)
  GPIO.output(TRIG,True)# Trigger pin high to send burst of audio
  time.sleep(0.00001)

  GPIO.output(TRIG,False)

  echo_state = 0
  while echo_state == 0:
    echo_state = GPIO.input(ECHO)
    pulse_start = time.time()

  #while GPIO.input(ECHO)==0:
    #pulse_start = time.time()
```

```
while GPIO.input(ECHO)==1:
    pulse_end = time.time()


pulse_duration = pulse_end - pulse_start
distance = pulse_duration*17150 # speed of sound in air at sea level
distance=round(distance,2) # round the distance

print("distance:",distance,"cm") # I have my distance from the sensor to the object


if 0<distance<30:
    buzzer = 18
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(buzzer,GPIO.OUT)

    #a PWM signal has been defined and frequency will vary as the objects distance
    #to the person varies
    buzzer = GPIO.PWM(buzzer, 2000) # Set frequency to 2 Khz
    buzzer.start(20) # Set dutycycle to 20
    time.sleep(0.5)

    GPIO.cleanup() # reset pins

elif 30<distance<60:
    buzzer = 18
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(buzzer,GPIO.OUT)
    buzzer = GPIO.PWM(buzzer, 1800) # Higher frequency means objects is closer
    buzzer.start(20)
    time.sleep(0.8)

    GPIO.cleanup()
    ###

elif 60<distance<90:
    buzzer = 18
```

```
        GPIO.setmode(GPIO.BCM)
        GPIO.setup(buzzer,GPIO.OUT)

        buzzer = GPIO.PWM(buzzer, 1500)
        buzzer.start(20)
        time.sleep(0.6)

        GPIO.cleanup()

    elif 90<distance<120:
        buzzer = 18
        GPIO.setmode(GPIO.BCM)
        GPIO.setup(buzzer,GPIO.OUT)

        buzzer = GPIO.PWM(buzzer, 1000)
        buzzer.start(20)
        time.sleep(0.8)

        GPIO.cleanup()

    #else:
        #buzzer = 18
        #GPIO.setmode(GPIO.BCM)
        #GPIO.setup(buzzer,GPIO.OUT)

        #buzzer = GPIO.PWM(buzzer, 2500) ## standard frequency for objects detected
above 1.2m
        #buzzer.start(10)
        #time.sleep(1)

        #GPIO.cleanup()

#2
### Next part of operation is to take an image of the obstalce



#3
```

```
### Captured image and label retrived and Tensor flow lite DL deployed
    print("Camera is activating")
    camera = PiCamera()
    camera.start_preview()
    sleep(2)
    camera.capture('/home/pi/TFLite_MobileNet/image.jpg')
    camera.stop_preview()
    camera.close()






    data_folder = "/home/pi/TFLite_MobileNet/"

    model_path = data_folder + "model.tflite"
    label_path = data_folder + "labels.txt"

    interpreter = Interpreter(model_path)
    print("Model Loaded Successfully.")

    interpreter.allocate_tensors()
    _, height, width, _ = interpreter.get_input_details()[0]['shape']
    print("Image Shape (", width, ",", height, ")")

## Load an image to be classified.
    image = Image.open(data_folder + "image.jpg").convert('RGB').resize((width,
height))

##Classify the image.
    time1 = time.time()
    label_id, prob = classify_image(interpreter, image)
    time2 = time.time()
    classification_time = np.round(time2-time1, 3)
    print("Classificaiton Time =", classification_time, "seconds.")

## Read class labels.
```

```
    labels = load_labels(label_path)

## Return the classification label of the image.
    classification_label = labels[label_id]
    accuracy = np.round(prob*100, 2)
    print("Image Label is :", classification_label, ", with Accuracy :",accuracy , "%.")

    accuracy_str = str(accuracy)
## Prepare the label and probability to be concatenated into a string ready for speech
    text_speech = classification_label+" with "+accuracy_str+" percent accuracy"
    print(text_speech)

#4
### Final part is to return the device detected via audio device interface
### throug text to speech

    cmd_beg= 'espeak '
    cmd_end= ' | aplay /home/pi/Desktop/Text.wav  2>/dev/null' # To play back the stored .wav file and to dump the std errors to /dev/null
    cmd_out= '--stdout > /home/pi/Desktop/Text.wav ' # To store the voice file

#Replacing ' ' with '_' to identify words in the text entered
    text_speech = text_speech.replace(' ', '_')

#Calls the Espeak TTS Engine to read aloud a Text
    call([cmd_beg+cmd_out+text_speech+cmd_end], shell=True)

    time.sleep(0.3)

### The accuracy enables the visually impaired person to make a discernment
### with regards to the object detetced.
```

Code for Deep learning model (Python)

The model used was trained first on google colab and can be accessed at : **https://colab.research.google.com/drive/1oE0yKXBjJhKSjD5WPJxESuVSBkucWxUD?usp=sharing**

Appendix B

Terms of reference

Department of Computing and Mathematics Computing and Digital Technology Postgraduate Programmes Terms of Reference Coversheet	
Student name:	Paul Samuel Christopherson
University I.D.:	21436364
Academic supervisor:	Dr Amna Eleyan
External collaborator (optional):	
Project title:	Smart Stick Prototype for Visually Impaired People using Raspberry Pi or Arduino
Degree title:	MSc Data Science
Project unit code:	AE_MSc.05
Credit rating:	
Start date:	13/06/2022
ToR date:	01/07/2022
Intended submission date:	23/09/2022
Signature and date student:	29.06.2022 S. Christopherson
Signature and date external collaborator (if involved):	

This sheet should be attached to the front of the completed ToR and uploaded with it to Moodle.

Project Background

Globally, at least 2.2 billion people have near or distance vision impairment. Most people with vision impairment are over the age of 50. Those who live in low- and middle-income regions are 4x more likely to have vision impairment than those in high income regions. (WHO ,2021) [1]

In order to help the navigate their environment, visually impaired individuals may;

- Rely on other people
- Use a white stick
- Use guide dogs to assist them

The environment can include many obstacles such as roads, walls, bridges, pedestrians, other moving objects such as cars and changes in height such as side walks and edges all which could potentially pose a danger to the visually impaired person and may even extend to those around them.

The main aim of this research is to explore an assistive device that is inexpensive, mobile and life enhancing to its user. This will help the user to move and explore their environment without limits and dramatically improve the quality of life.

The assistive device in this project will make use of an existing walking stick and then enhance it using smart technology such as sensors and voice commands to help the visually impaired identify and avoid potential obstacles as they move. This will be done using Machine/Deep learning.

Machine learning is a branch of artificial intelligence and computer science which imitates how humans learn using data and algorithms gradually improving its accuracy through feedback. Through its use of statistical methods, algorithms are trained to make classifications or predictions. It is in this way that obstacles can be picked up by sensors and then classified into what the object may be using a voice module.

Deep learning is a subfield of Machine learning. The way in which Deep Learning and Machine Learning differ is in how each algorithm learns. Deep learning automates much of the feature extraction piece of the process, eliminating some of the manual human intervention required and enabling the use of larger data sets. This is useful as It can ingest unstructured data in its raw form (e.g., text, images), and it can automatically determine the set of features which distinguish different categories of data from one another. In this case what the obstacle is that the Smart Stick sensors detect.

(IBM Cloud Education , 2020:online) [2]

Aims & Objectives

This project is to develop a prototype stick device that helps visually impaired people to identify obstacles in front of it using Machine/Deep Learning. The smart stick will use ultrasonic sensors, a camera for image detection ,GPS and a Raspberry Pi or Arduino microcontroller. This microcontroller receives and processes the sensor signals, and it connects it to a voice module to generate an alert voice to aid the blind person to avoid the obstacle. Machine learning and Deep learning will be used to identify the obstacle and GPS could be used to validate the object type. The objects information will then be conveyed to the user via the voice module.

The objectives for the project are:

- Write up terms of reference.
- Conduct research into relevant papers and similar projects.
- Build the Prototype stick device using suitable material for the walking stick. An Arduino or Raspberry Pi will be used as the micro controller combined with sensors and camera to detect and capture object/obstacle(s). The information captured via sensors will be stored in a secure database(in an SD card or otherwise cloud) and the voice module will relay obstacle type to the blind person.
- Access and store the relevant data stored from the prototype readings to be fed into the Machine learning model.
- Preprocess the data by applying the relevant algorithms and filters. Discard non relevant information that has been captured.
- Sort the data into the relevant object folder type and correct size if it is an image.
- Train and build the Machine/Deep learning model to classify objects for the user. To be done using Python and Scikit learn in a Jupiter Notebook environment. Cloud environment to be determined.
- Implement the now classified objects in a feedback loop using a voice module to the User.
- Test and evaluate the actual objects to the predicted object classification during the experimentation.
- Write up the report

Learning Outcomes

The nature of this project is interdisciplinary , hence a combination of learning outcomes from different degree streams are applicable to this project.

The project will allow the development of the following knowledge and skills:

• Choose, customise, and integrate core techniques to build sophisticated solutions for real-world Data Science problems involving Machine/Deep learning.

• Process and analyse, effectively and efficiently, data of varying scales and from heterogeneous sources, formats, and systems, using a range of suitable languages, tools, and environments.

• Build data science products with good software practices such as in code reuse, separation of concerns, modularity, testing, and documentation.

• The working practice of the software industry, and the typical tools and techniques employed in the software development process to deliver the outcome of models which can classify objects.

• The role of computer science in the development of the wider technological community and the relevant ethical and societal issues with regards to the visually impaired using a smart stick.

• Critical thinking and ability to communicate ideas and solutions both in writing and orally to specialist and non-specialist audiences.

Related works

- Chiranjevulu, D., Sanjula, D., Kumar, K. P., Murali, U. B., & Santosh, S. (2020). Intelligent walking stick for blind people using Arduino. Int. J. Eng. Res. Appl, 10, 42-45.

- Anandan, M., Manikandan, M., & Karthick, T. (2020). Advanced indoor and outdoor navigation system for blind people using raspberry-pi. Journal of Internet Technology, 21(1), 183-195.

- Gbenga, D. E., Shani, A. I., & Adekunle, A. L. (2017). Smart walking stick for visually impaired people using ultrasonic sensors and Arduino. International journal of engineering and technology, 9(5), 3435-3447.

- Jawale, R. V., Kadam, M. V., Gaikawad, R. S., & Kondaka, L. S. (2017, September). Ultrasonic navigation based blind aid for the visually impaired. In 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI) (pp. 923-928). IEEE.

- Abd Khafar, N. H. B., Soomro, D. M., Abd Rahman, R., Abdullah, M. N., Soho, I. A., & Rahimoon, A. A. (2019, December). Intelligent navigation system using ultrasonic sensor for visually impaired person (VIP). In 2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS) (pp. 1-4). IEEE.

Literature Review

A thorough literature review to be carried out is not limited to, but will include the below;

- SURF Algorithm
- Bivariate gaussian mixture model
- Machine/Deep learning in Cloud environments
- Text to Speech
- IoT projects with Deep learning
- Arduino/Raspberry Pi IoT projects with sensors
- Neural Networks and Convolutional Neural Network
- Support Vector Machines

Evaluation plan

The aims & objectives will be evaluated with regards to their completion and effectiveness.

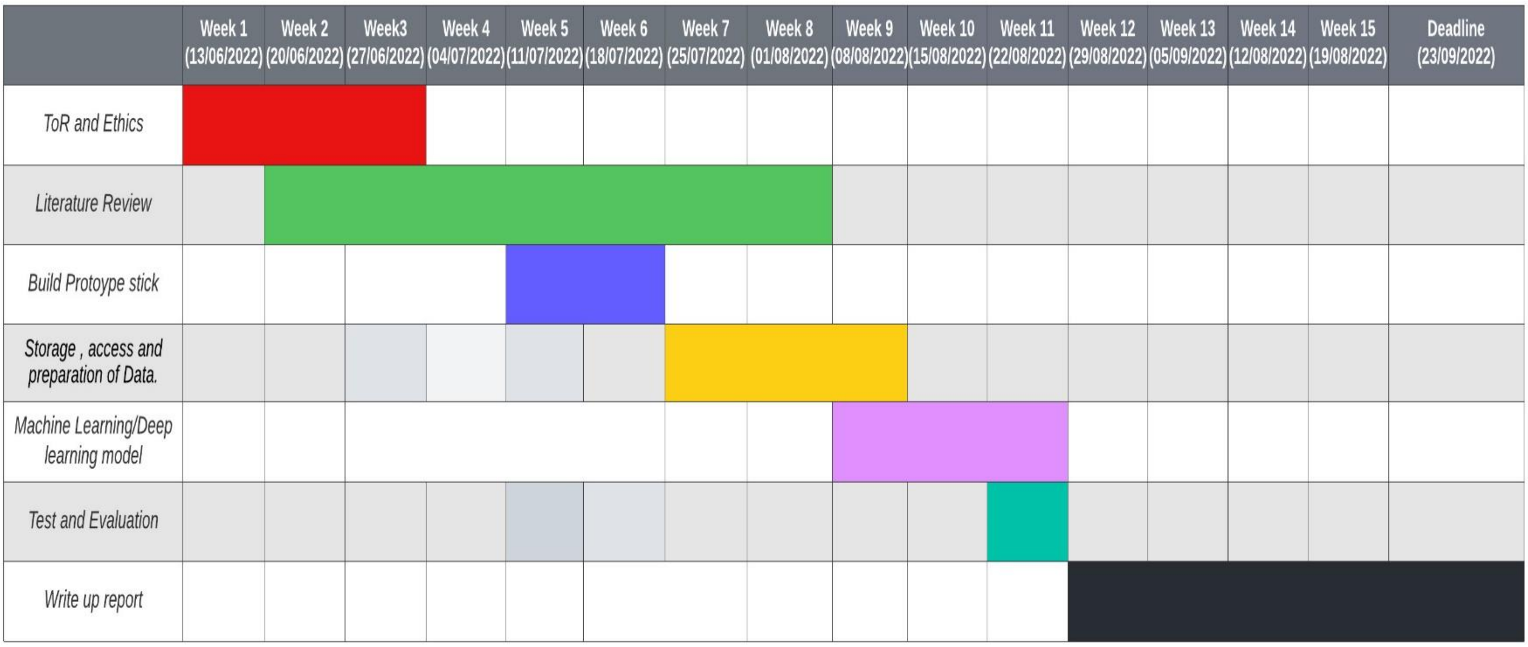
These are:

- Terms of Reference – A detailed plan about the project to be carried out and its required objectives and activities
- Build and completion of the Prototype Smart Stick and its associated Database that will feed into a Machine/Deep learning model as required.
- Data Analysis and Evaluation - Results from training and testing from all the machine/deep learning algorithms and the upgrades required to make the Prototype Smart Stick more effective. Evaluate the product by comparing it with the existing ones in the literature.
- Final Report - Completed report detailing the previous objectives and results obtained from these steps to evaluate the effectiveness of the Smart Stick, the accuracy of each of the machine/deep learning algorithms classification

Activity Schedule

TASK	START DATE	END DATE	DURATION (DAYS)
TOR AND ETHICS	13/06/2022	29/06/2022	13
LITERATURE REVIEW	22/06/2022	05/08/2022	33
BUILD PROTOTYPE STICK	11/07/2022	22/07/2022	5
STORAGE , ACCESS, AND PREPARATION OF DATA.	25/07/2022	05/08/2022	10
MACHINE LEARNING/DEEP LEARNING MODEL	08/08/2022	26/08/2022	15
TEST&EVALUATION	22/08/2022	26/08/2022	5
WRITE UP REPORT	29/08/2022	23/09/2022	20

Task and Subtasks



References

[1] WHO, 2021, *Blindness and vision impairment, [Online] [Accessed on 29.06.2021]* https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment

[2] IBM Cloud Education, 2020, *Machine Learning, [Online] [Accessed on 26.06.2021]* https://www.ibm.com/uk-en/cloud/learn/machine-learning