# PROJECT REPORT

ON

# "NayanDrishti": A Revolutionary Navigation/Visual Aid for the Visually Impaired

SUBMITTED TO THE

## UNIVERSITY OF MUMBAI

FOR THE DEGREE OF

## BACHELOR OF ENGINEERING

IN

## COMPUTER ENGINEERING 2020-2021

**Submitted by**

Jordan Dsouza (Roll No.19)

Salil Fernandes (Roll No.22)

Anthony Kattikaren (Roll No.36)

**Supervisor (s):**

Ms. Dipti Jadhav

**DEPARTMENT OF COMPUTER ENGINEERING**

**DON BOSCO INSTITUTE OF TECHNOLOGY**

**(ENGINEERING COLLEGE)**

**PREMIER AUTOMOBILES ROAD, KURLA (W) – 400070**

# "NayanDrishti": A Revolutionary Navigation/Visual Aid for the Visually Impaired

Submitted in partial fulfilment of the requirement for the Degree of

## BACHELOR OF ENGINEERING IN COMPUTER ENGINEERING

by

**Jordan Dsouza Roll No.19**

**Salil Fernandes Roll No.22**

**Anthony Kattikaren Roll No.36**

Supervior (s):

Ms. Dipti Jadhav

**DEPARTMENT OF COMPUTER ENGINEERING**

**DON BOSCO INSTITUTE OF TECHNOLOGY**
**(ENGINEERING COLLEGE), PREMIER AUTOMOBILES ROAD,**
**KURLA (W) – 400070**

**2020-2021**

# CERTIFICATE

This is to certify that the project report entitled "**NayanDrishti: A Revolutionary Navigation/Visual Aid for the Visually Impaired"** is a bonafide work of **Jordan Dsouza Roll No. 19, Salil Fernandes Roll No. 22 and Anthony Kattikaren Roll No. 36** submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the Degree of Bachelor of Engineering in the Computer Engineering.

**Mrs. Dipti Jadhav**
Supervior

**Mrs. Sana Shaikh**
Head of Department

**Dr. P. Nambiar**
Principal

# Project Report Approval for B. E.

This project report entitled **"NayanDrishti": A Revolutionary Navigation/Visual Aid for the Visually Impaired** by **Jordan Dsouza, Salil Fernandes and Anthony Kattikaren** is approved for the degree of Computer Engineering course of Bachelor of Engineering.

Examiners

1.---------------------------------------------

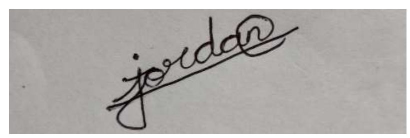
2.---------------------------------------------

Date: 28/5/2021

Place: Mumbai.

# DECLARATION

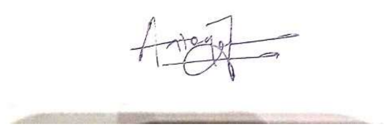
I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.



1. Jordan Dsouza, 19

2. Salil Fernandes, 22

3. Anthony Kattikaren, 36



Date: 28/5/2021

# ABSTRACT

The disabled section in society typically blind people or people suffering from any form of visual impairment have to live in complete uncertainty throughout their lives. The lack of vision due to defects at birth or some accident has forced them to constantly be aware of their surrounding environment and has also made them live with fear. Blind people are often dependent on others to lead a normal life and experience difficulty in performing day to day activities on account of a disability they did not choose to have. Performing any simple task like picking up an object or even walking from one place to another is daunting challenge for blind people. In our project, we are developing a prototype of a product which aids blind people in performing day to day activities. We are working around the domains of computer vision and machine learning to develop this prototype. Since blind people cannot see what objects are in front of them a feature of object recognition would have to be incorporated. While walking or navigating, blind people would not be aware of any obstacles that are in front of them which is why we have decided to implement a feature which would detect these obstructions. Blind people also face uncertainty as to where they are or whether they are walking in the correct direction. To relieve blind people of this burden some sort of GPS navigation feature is used in the prototype. By developing this prototype we aim to demonstrate how machine learning can be used to help the disabled section of society and to inspire computer engineers to build upon this prototype which would lead to development of an all-round navigation assistant for the blind and visually impaired.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ABBREVIATION

| No. | Abbreviations | Meaning |
|---|---|---|
| 1. | **SURF** | Speed Up Robust Features |
| 2. | **PIC** | Peripheral Interface Controllers |
| 3. | **GPS** | Global Positioning System |
| 4. | **RFID** | Radio-Frequency Identification |
| 5. | **WIFI** | Wireless Fidelity |
| 6. | **SLAM** | Simultaneous Localization and Mapping |
| 7. | **IoT** | Internet of Things |
| 8. | **NOIR** | Nominal Ordinal Interval and Ratio. |
| 9. | **CNN** | Convolutional Neural Network |
| 10. | **SIANN** | Space Invariant Artificial Neural Networks |
| 11. | **VCC** | Voltage Common Collector |
| 12. | **TRIG** | Trigger |
| 13. | **RTK** | Real-Time Kinematic |
| 14. | **WAAS** | Wide Area Augmentation System |

| | | |
|---|---|---|
| 15. | **HDOP** | Horizontal Dilution of Precision. |
| 16. | **HAE** | Height Above Ellipsoid |
| 17. | **NMEA** | National Marine Electronics Association |
| 18. | **RAM** | Random Access Memory |
| 19. | **SSD** | Solid state drives |
| 20. | **GPIO** | General-Purpose Input/Output |

# 1. INTRODUCTION

## 1.1 Introduction:

We sometimes wonder how the disabled section our society manage to perform tasks that look impossible pertaining to their disability. How efficiently can they perform their day-to-day tasks and how well can they communicate with normal people. Keeping in mind the difficulties they face, their ability to perform such tasks and how to ease the fear related to it technology has advanced in many fields.

Our target audience are the blind people and the main motive behind the targeted people is to enable them experience freedom in their walks of life and also immediately alert their dear ones if they are facing trouble in anything. Blindness can basically be classified into three types:
**1. Complete Blindness:** A state where the affected person is completely out of sight and is unable to differentiate between objects and humans.
**2. Night Blindness:** A state where the affected person is unable to see once it's about to start getting dark until it is dusk i.e. sunrise or in any place that is poorly lit. People with night blindness often have difficulty driving at night or seeing stars.
**3. Color Blindness:** A state where the person is unable to differentiate between colors. Color blindness is also known as dyschromatopsia.

In the implementation of our project, we wanted to implement a model which would turn out to be beneficial to our user (i.e. the blind person) not only helping them navigate through easily with the help of object prediction and detection but also guide them in aspects such as "How can one reach his/her destination from the source?" which is navigation using the Global Positioning System (GPS) and if time permits even a software that helps our user to use a smartphone in a way that is comfortable to them (A software/app that can understand braille signs or senses the motion of the user to perform tasks such as calling whenever in distress, sending the location of the user to the guardian, etc.). The outcome of our project is based on three of the majorly talked about topics that are the backbone of computer technology and have a plethora of applications in the real world:

**1. Convolutional Neural Networks:** These networks are a pack of layers that filter images using convolution techniques. Convolutional layers convolve the input and pass its result to the next layer and so on until the output layer is reached.
The product will take an input from the surrounding environment of the user and convolve it for a better prediction of the obstructing object which will be done using machine learning.

**2. Machine Learning:** In machine learning, we basically enable the computing machine to learn aspect related to the real world. The machine will eventually be able to detect and predict the outcome of a particular object. The product with the output image can will predict the object that is in hinderance to the user's path.

**3. Global Positioning System:** One of the main aspects of this product, the GPS will eventually help the user reach his/her destination from his/her source. The user will be able to communicate his/her destination to the software and the software will guide the user to the destination. There are many APIs available for the same to integrate it along with the software to show real time location of the user.

**4. IOT (Internet of Things):** Internet of things is a trend in the modern technological world and a boon to society. It helps power up our world with various aspects such as wireless network connectivity, portable processors that can be used anywhere, etc. Our project consists of raspberry pi, GPS module, camera module, etc. that will be effectively used in processing data, capturing images, displaying location of the person, etc. This domain proves as a good link to the above two domains.

### 1.2 Aim And Objectives

The main aim of our project is to develop a prototype that would set a standard to products that are manufactured for the blind and visually impaired. Blind people are often taken advantage of and are left behind in most walks of life. A bitter truth blind people must face is that they cannot drive a vehicle nor can they perform high level jobs like that of an engineer or doctor and hence have to accept relatively lower salaries performing other jobs. Reading too can only be done by means of Braille script which can be difficult even with practice. Thus, we aim to relieve blind people from the hurdles they face on a daily basis. By developing this prototype product we aim to bring to light the endless possibilities machine learning has brought to the table in terms of helping the blind and visually impaired.

To summarise our objectives:

- To set the foundation for a product that aids the visually impaired.
- To inspire engineers to make an effort of providing a means to a normal life to the visually impaired.
- To learn advanced machine learning concepts of deep learning and convolutional neural networks.
- To explore the various functionalities and features of TensorFlow, OpenCV and Raspberry Pi.
- To contribute to the existing knowledge related to object detection, classification and recognition by machine learning techniques.
- To acquire some of the technical machine learning and artificial intelligence skills required in the industry.

### 1.3 Organization of Report

First chapter of the report contains Introduction to this project which explains our motivation for taking up this project and the different domains that it encompasses. Second chapter of the report consists of the Literature Survey we had performed to analyse work that has already been done in our project domain. The third chapter explains the problem statement while the fourth chapter deals with design and architecture of our system. The fifth chapter is divided into three phases, Module-I deals Object Recognition, Module-II deals with Obstacle Sensing and Module-III deals with Location Sensing. The sixth chapter explains the results in detail. The seventh chapter deals with the work distribution while a conclusion is given in the eight and final chapter. The remainder of the report deals with code, future scope and acknowledgement.

# 2. LITERATURE SURVEY

“Electronic walking stick for the blind” published in the year 2018, the use of optical sensors hasbeen highlighted. This is a modern concept of a walkingstick which is completely digital. These sensors essentially convert light reflected from any surface and convert them into electric signal and acts as response to the stimuli and informs the blind person via a speaker onthe handle about the obstacle. The object aimed to give voice assistance to the user and was able to deliver in various cases.[1]
Platform: Microcontroller.

“Portable Blind aid Device” published in the year 2019, it highlights the use of a mobile-based project in which the user can switch his wireless device into blind assistance mode with thehelp of a button. With the help of the camera, GPS and a cloud-based architecture it will be able to give the real-time location of the user and also make himaware of his surroundings. Also plan to include an advanced image recognition algorithm to recognize the faces of stranger and to store them.[2]
Platform: Android or IOS devices or any mobile devices.

“Intelligent glasses for the blind” published in the year 2016, the device is smart glass which uses a camera, an ultrasonic sensor and an electronic touchpad to assist the blind person. Using a mobile device one can activate the glass and the camera will capture and convert the 3D image into a spatial matrix and give appropriate outputs. The touchpad giveslight electric shocks to make the user aware of the obstacles. Future scope was to add a walking cane with a button to give the user audio output of the real-timesurroundings.[3]
Platform: Any computing device like mobile phones, laptops or tablets.

“Object Identification for Visually Impaired.” published in the year 2016, a simple image recognition system that makes use of camera to recognize images, an ultrasonic sensor to detect obstacles is explained. The camera captures the images and if the image can be recognized by the images in the dataset then the device gives and audio output using a speaker that will be attached to the users clothing. Future upgrades include to introduce face recognition and to use a wireless camera.[4]
Platform: Raspberry Pi, PIC controller.

“Real-Time Objects Recognition Approach for Assisting Blind People.” Published in the year 2017, an object recognition project is depicted that with the help of SURF (Speeded Up Robust Features) and light machine learning is able to give accurate information about the objects captured. Using GPS and image recognition it gives 90% accurate results. It makes use of a database and uses machine learning algorithm it is able to identify the objects.[5]
Platform: Windows, Ubuntu, MacOS.

“EarTouch: Facilitating Smartphone Use for Visually Impaired People in Mobile and Public Scenarios.” published in the year 2019, a Blind aid device which uses ear gestures like long swipe, tap, slide, etc. for operating the various function on the smartphone. Making use of one-hand gestures and Ear- Touch feature it enables the user to use the device with ease. And it gives out audio output through the inbuilt speakers. The device was able to read the gestures accurately and activate the respective functions when theuser wanted it.[6]
Platform: Android, IOS, Windows OS or any mobile device.

“An Ultrasonic Navigation System for Blind People” published in the year 2017, introduces with us another device for object detection which not only uses an ultrasonic sensor but also includes an accelerometer, footswitch, microcontroller and vibrationpad. User is made aware of the obstacle and closeness ofit by different degree of vibrations and also through an audio output.[7]
Platform: PIC microcontroller.

“Mobile Blind Navigation System Using RFID”. published in the year 2015, the paper introducesa mobile based communication device which makes useof the WIFI, GPS to determine the user’s location. And also makes use of RFID tags in the surroundings to givemore accurate location to the user and guide them by giving appropriate instructions. The user makes use of asmart walking cane and RFID handled reader.[8]
Platform: Android, iOS and any mobile devices.

“Voice assisted system for the blind” published in the year 2015, the paper introduces us witha simple object/obstacle detector which makes use of anultrasonic senor, microcontroller, mp3 module and SD card. The sensor sends the distance measured and the microcontroller depending the proximity limit it is programmed sends an output to the mp3 module. It makesuse of depth sensing for avoidance of pot holes etc. for the user to walk with smoothness and comfort with normalcy.[9]
Platform: Microcontroller.

“Virtual-Blind-Road Following Based Wearable Navigation Device for Blind People.” published in the year 2019, the paper depicts a navigation device specifically for the indoor environment which makes use of SLAM (simultaneous localization and mapping). The device tries to give the person the best path to an indoor destination by keeping track of the positions of previously encountered obstacles. It makes use of PoI graph to store new points in the route to the destination and using A* algorithm onthe PoI graph it finds the shortest and optimum path to the destination. Obstacle distance is measured by an Ultrasonic range finder and provides audio output.[10]
Platform: Embedded CPU like Raspberry or Ardunio.

# 3. PROBLEM STATEMENT

Our product introduces a three-way combined solution system that will serve as a navigation aid for the blind:

To ease the difficulty and uncertainty faced by the blind or any visually impaired person when they have to walk from one place to another. A simple task like walking is scary for blind people owing to the fact that they just do not know what obstacles, whether dangerous or not, may be in front of them. Through our project we hope to create a product that makes navigation a safer and simpler task for blind people. Object detection and recognition is the core concept which our project revolves around. Our project domain is machine learning and we have decided to make use of the concept of a Convolutional Neural Network built using TensorFlow to process image data and perform the object recognition task after being trained with datasets.

### 3.1. Scope

Our project aims to implement three main features:

1) Object recognition using a camera attached to the pi module and using our machine learning domain and implementing Convolutional Neural Network (CNN) break down the images into 2D frames and train the algorithm to identify various objects in the surrounding.
2) Object Detection using an Ultrasonic sensor which can detect obstacles at a distance of 4m and sends an alert to the user if the proximity of the object is within a certain given range.
3) Giving the real-time location of the user using a GPS module that is programed to communicate with the satellite and on a press of a button give the user his/her location. The live location data would be uploaded to Firebase so that in case the blind person gets lost or robbed in public, the authorities can make use of the last known location of the blind person to track him/her down.

All the directions and instructions and output from each source and input that is fed will be given to the user via a Bluetooth speaker which will be connected to the Bluetooth in the raspberry module. Since, our final outcome is basically a prototype product there is a vast scope for improvement. With a rapidly increasing trend towards machine learning and AI, additional functionalities can be implemented in the future. Once the prototype model is complete, if all the above-mentioned features are working smoothly, a few additional features can be added pertaining to distinguishing of day and night since blindness. For e.g. if a person has night blindness, he/she can use the glasses during the night to distinguish between various objects that is in hinderance to his/her path. For a person suffering from complete blindness, the day-night distinguishing feature can be implemented as the person then would not have to be dependent on someone else to actually inform him/her whether it is day or night. Since, we are performing this project at the student level, we face restrictions in terms of hardware. If the prototype is taken up by a company, a professional aspect can be incorporated which would make the product give more accurate readings and predictions and would genuinely be of great help to the blind people.

# 4. SYSTEM ANALYSIS

## 4.1. Existing System:

All above proposed papers that we have researched have shown 90% success rate in the field and in all test cases. Although with certain setbacks too. All devices are capable of giving the required results but the reactions time in real-world is less which the devices are not able to keep up with. For real-time outputs more complex and advanced system and cloud-based architecture is required which is difficult on a limited budget. In image recognition few projects weren't able to identify objects in the dark or night. The components used in few projects are fragile and prone to damage. Hence when used by a user could get damaged. And using strong equipment makes the project expensive.

## 4.2. Proposed System:

We have decided upon three clear cut goals we want to accomplish through this project. Firstly, the blind person has to be made aware about the various objects ad entities around him whether it is a common household object like a spoon, a book, a cup or whether if a person is standing in front of him. This is needed to give him/her a sense of the environment around them. Secondly, while walking, the blind person has to be alerted if they are close to walking into an obstacle. Lastly, since blind people can never really know exactly where they are unless they ask someone who is trustworthy, we have decided to implement a feature which would inform the blind person of his physical location. Keeping these goals in mind our proposed system would be capable of the following:

a) Using an object recognition neural network which takes live video feed through a camera and detects and recognises all the objects in the frame so that the blind person knows whats in front of him/her.

b) By means of an ultrasonic sensor, obstacles in front of a blind person's path can be detected in the range of upto 4 m thus allowing the blind person to take preemptive action to avoid collision.

c) In order to inform the blind person of their current location, a GPS module will detect the person's location coordinates and voice their location in terms of an address(street, country, city, pincode and area). This location information would also be directly uploaded to the cloud so it can be used by the authorities to track down the blind person in case of any mishap.

d) All of the above features would be useless without the last feature we are implementing which is a voice feedback feature. A Bluetooth speaker would be able to communicate the objects recognised, the distance between an obstacle and the blind person and the address of the person's current location. Since blind people cannot see, audio message delivery is extremely crucial.

## 4.2. Analysis

Using the Raspberry Pi we are going to be implementing Object Detection and Recognition, Obstacle Distance Sensing and GPS Location Sensing.

### A. Object Detection & Recognition

In order to perform live object detection and recognition, we have to find a way of taking live pictures and transferring it to the program to allow the neural network to perform analysis. Therefore, in order to capture images from a live video feed we have interfaced the NOIR Camera Module to the Raspberry Pi.

The main purpose of our prototype product is object detection and recognition. This is achieved by making use of a Convolutional Neural Network. Deep Learning, a **convolutional neural network** (**CNN**, or **ConvNet**) is a class of deep neural networks, most commonly applied to analysing visual imagery. They are also known as **shift invariant** or **space invariant artificial neural networks** (**SIANN**), based on their shared-weights architecture and translation invariance characteristics. The name “convolutional neural network” indicates that the network employs a mathematical operation called Convolution. Convolutional networks are a specialized type of neural networks that use convolution in place of general matrix multiplication in at least one of their layers.

CNNs are regularised versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "fully-connectedness" of these networks makes them prone to data. Typical ways of regularization include adding some form of magnitude measurement of weights to the loss function. CNNs take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble more complex patterns using smaller and simpler patterns.

Depth-wise convolution and point convolution are the two main operation performed in the network for feature extraction and learning. The combination of these two operations is called depth-separable convolutions.

**1. Depth Wise Convolution:** In ***depth-wise operation***, convolution is applied to a **single channel** at a time unlike standard CNN’s in which it is done for all the M channels. So here the filters/kernels will be of size **K x K x 1**. Given there are M channels in the input data, then M such filters are required. Output will be of size **Q x Q x S**.

**Cost of this operation:**
A single convolution operation requires K x K multiplications.
Since the filter are slided by **Q x Q** times across all the M channels, the total number of multiplications is equal to **S x Q x Q x K x K**.
So for depth wise convolution operation,
Total no of multiplications = S x $K^2$ x $K^2$

**2. Point Wise Convolution:** In ***point-wise operation***, a **1×1 convolution** operation is applied on the M channels. So, the filter size for this operation will be **1 x 1 x Q**. Say we use N such filters, the output size becomes **Q x Q x P**.

**Cost of this operation:**
A single convolution operation requires 1 x S multiplications.
Since the filter is being slided by Q x Q times, the total number of multiplications is equal to S x Q x Q x (no. of filters).
So for point wise convolution operation,
Total no of multiplications = S x $Q^2$ x P

So now we can create a mathematical comparison to prove how our architecture is more efficient and computationally less intensive than standard convolution.

| | |
|---|---|
| Normal Convolution | P x $Q^2$ x $R^2$ x S |
| Depth Wise Separable Convolution | S x $Q^2$ x ($K^2$ + P) |

Where p = no of filters
S = no of channels in the image (eg an RGB image has 3 channels red, green and blue)
Q = image dimensions (height and width)
K = kernel/filter dimensions

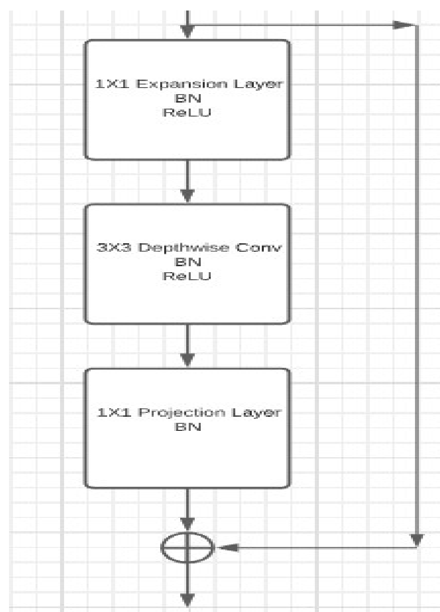



**Fig .1. Single layer of the MobileNet V2 architecture used in our project**

The **input** for the CNN we are using is an image. On the Raspberry Pi, the camera module will take the live video feed and on pressing a button an image will be captured. This image is then fed to the object detection program which has the CNN logic coded into it. We have made use of the Raspberry Pi NOIR Camera Module to take the live image feed.

The **output** for this image analysis by the CNN will be a rectangular box around the object detected by the CNN. The box will have a label which indicates the object prediction made by the CNN. The prediction is nothing but what object the CNN thinks it is based on the training it has received. Once we have successfully integrated a wireless speaker with the Raspberry Pi we will be providing an audio cue by making the speaker sound out the object recognized by the neural network.

### B. Obstacle Distance Sensing

For the purpose of sensing the presence of an obstacle and sensing the distance between the obstacle and person, an ultrasonic sensor is being used.

The input for this would in fact be an object placed in front of the sensor.

The output is a distance reading if the object is in the sensor's accurate range. Once we have interfaced a wireless speaker, an audio warning will be sounded stating that the obstacle is nearby. The program calculates the time it takes for the echo to come back from the obstacle and calculates the distance based on a formula using the speed of sound.

The ultrasonic sensor has 4 pins:

a. VCC which is used to take the power supply.
b. TRIG which will on receiving the input signal from the Raspberry Pi will send out the ultrasonic pulse.
c. ECHO which will gather the echo pulse from the obstacle and send the signal that the obstacle has been detected back to the Raspberry Pi.
d. GND which provides general grounding to the sensor.

For exchanging signals and for the program to receive readings from the sensor the GPIO pins of the Raspberry Pi were connected with the pins of the sensor.

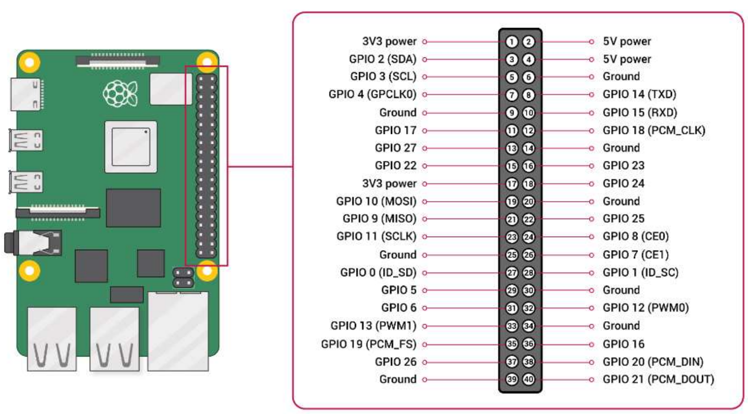



**Fig 2. GPIO Pin layout of the Raspberry Pi**

### C. Location Sensing

In order for prototype to deliver on its promise of informing the blind person of their current physical location, a GPS feature is to be implemented. For the purpose we are using the UBLOX Neo 6M GPS Module.

The **input** would be the satellite signal picked up by the antenna from space. This module simply has an antenna that picks up a signal sent by a GPS satellite in space and circuitry that demultiplexes the signal to transfer the location data to the Raspberry Pi.

The **output** would involve extracting the latitude and longitude coordinates from the input and converting them into a physical location address which would be sounded on the Bluetooth Speaker.
GPS location data comes in various formatted sentences under the NMEA structure.

Eg.***$GPGGA,181908.00,3404.7041778,N,07044.3966270,***

***W,4,13,1.00,495.144,M,29.200,M,0.10,0000*40***

All NMEA messages start with the $ character, and each data field is separated by a comma.

**GP** represent that it is a GPS position (GL would denote GLONASS).

**181908.00** is the time stamp: UTC time in hours, minutes and seconds.

**3404.7041778** is the latitude in the DDMM.MMMMM format. Decimal places are variable.

**N** denotes north latitude.

**07044.3966270** is the longitude in the DDDMM.MMMMM format. Decimal places are variable.

**W** denotes west longitude.

**4** denotes the Quality Indicator:

1 = Uncorrected coordinate

2 = Differentially correct coordinate (e.g., WAAS, DGPS)

4 = RTK Fix coordinate (centimeter precision)

5 = RTK Float (decimeter precision.

**13** denotes number of satellites used in the coordinate.

**1.0** denotes the HDOP (horizontal dilution of precision).

**495.144** denotes altitude of the antenna.

**M** denotes units of altitude (eg. Meters or Feet)

**29.200** denotes the geoidal separation (subtract this from the altitude of the antenna to arrive at the Height Above Ellipsoid (HAE).

**M** denotes the units used by the geoidal separation.

**1.0** denotes the age of the correction (if any).

**0000** denotes the correction station ID (if any).

***40** denotes the checksum.

The $GPGGA is a basic GPS NMEA message. There are alternative and companion NMEA messages that provide similar or additional information.

Here are a couple of popular NMEA messages similar to the $GPGGA message with GPS coordinates in them (these can possibly be used as an alternative to the $GPGGA message): $GPGLL, $GPRMC

In addition to NMEA messages that contain a GPS coordinate, several companion NMEA messages offer additional information besides the GPS coordinate. Following are some of the common ones:

$GPGSA – Detailed GPS DOP and detailed satellite tracking information (eg. individual satellite numbers). $GNGSA for GNSS receivers.

$GPGSV – Detailed GPS satellite information such as azimuth and elevation of each satellite being tracked. $GNGSV for GNSS receivers.

$GPVTG – Speed over ground and tracking offset.

$GPGST – Estimated horizontal and vertical precision. $GNGST for GNSS receivers.

D. **Audio Feedback**

Since blind people are the target audience of our prototype, some form of auditory feedback has to be put in place because blind people cannot see and hence they would require audio messages to be delivered in order make the outputs of the above features useful. For the purpose, we have made use of a wireless Bluetooth speaker.

The input to the speaker would obviously be the objects recognized by the neural network, the distance that is sensed by the ultrasonic sensor in case of obstacles and finally the physical location of the person.

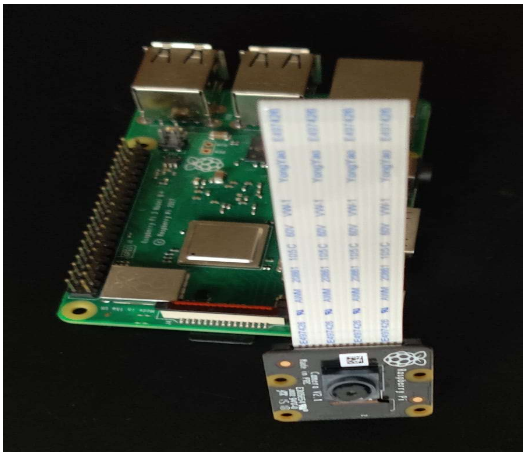
The output of the speaker will be artificial voice generated by Python which would inform the blind person about the above pieces of information.


**Fig 3. Sony XRS Bluetooth speaker which we have used in our project**

## 4.4. Hardware and software details

Hardware:

- Our entire project revolves around the use of the Raspberry Pi computer board. We have used the Raspberry Pi 3 Model 3B+ for our implementation. It has 1GB of RAM with in-built Bluetooth.
- San Disk Micro SD Card with 32 GB for installation of the Raspbian OS and storage of files on the Pi.
- HC-SR04 Ultrasonic Sensor
- Raspberry Pi NOIR Camera Module
- Ublox- NEO 6MV2 GPS Module
- Female to Male, Male to Male, Female to Female Jumper wires
- Breadboard
- 560 ohm and 1000 ohm resistors



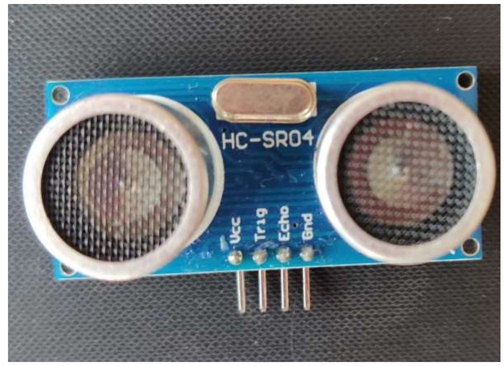



**Fig 4. NOIR Camera interfaced with the Raspberry Pi** **Fig 5. HC-SR04 Ultrasonic Sensor**

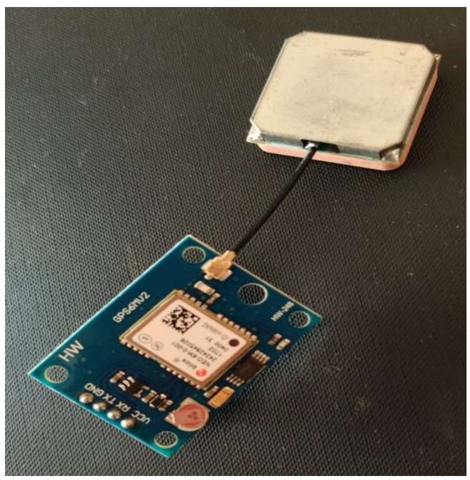


**Fig 6. UBLOX Neo 6M GPS Module**

**Software:**

- Raspbian OS which is the primary operating system of the Raspberry Pi on which programs will be coded.
- TensorFlow which is the machine learning framework by Google whose API is used to code the convolutional neural network for object classification.
- Python which is a high level programming language with massive machine learning capabilities.
- Under Python we will be using several special purpose libraries:
  - RPi.GPIO which is python's package to allow the program to exchange input and output between the program and GPIO pins of the Pi.

  - time which is needed to measure the time between sending and receiving the pulse in distance sensing.

  -OpenCV, keras for importing certain machine learning functions

-matplotlib for data visualization and mapping

-tensorflow for building the neural network layer by layer

- pyrebase to allow the python program to upload live GPS data to Firebase

## 4.5. Methodology

### A. Object Recognition

We have made use of the MobileNet SSD architecture to program our neural network. This architecture uses two main operations namely depthwise convolutions and pointwise convolutions. Our architecture utilizes the concept of depth-wise separable convolution. For MobileNets the depth-wise convolution applies a single filter to each input channel. The pointwise convolution then applies a 1X1 convolution to produce a linear combination of the outputs the depth-wise convolution. All layers in MobileNet consist of a 3X3 depth-wise separable convolution except for the first layer which has a full convolution. Each layer consists of the depthwise separable convolution followed by a BatchNorm BN and Rectified Linear Unit ReLU nonlinearity with the exception of the final fully connected layer which has no nonlinearity and feeds into a softmax layer for classification. A final average pooling reduces the spatial resolution to 1 before the fully connected layer. Counting depth-wise and pointwise convolutions as separate layers, MobileNet has 28 layers.

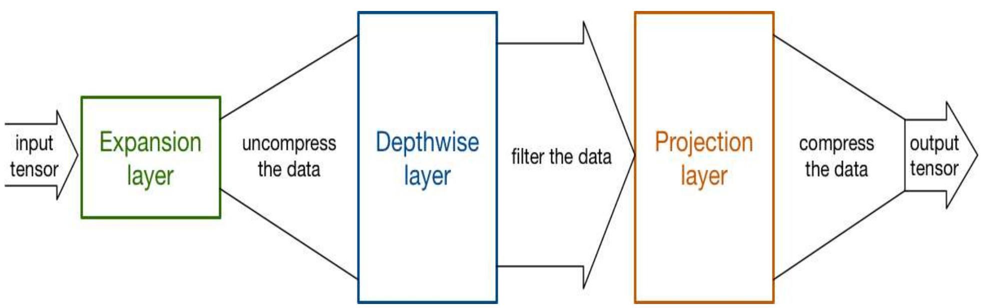



**Fig 7. Data Flow in Convolutions**

The COCO Dataset has been used to train our MobileNet CNN for object detection and recognition. COCO in short is a dataset used for training object detection, segmentation and captioning networks. COCO stands for Common Objects in Context which means that images are taken from everyday objects to prepare the dataset. It has approximately 330,000 images with more that 200,000 of them labelled. For each convolutional layer of the architecture, the ReLU activation function has been used along with Batch Normalization, the rectified linear activation function also called ReLU is the most commonly used activation functions in artificial neural networks. It returns a 0 as output for any negative input. If it receives a positive value x as input it returns x as the output. So in general this function is given as an equation $f(x) = \max(0, x)$. Batch Normalization is a deep learning technique that normalizes the output of each sublayer. It allows for fast processing and deep network training by reducing internal covariate shift.

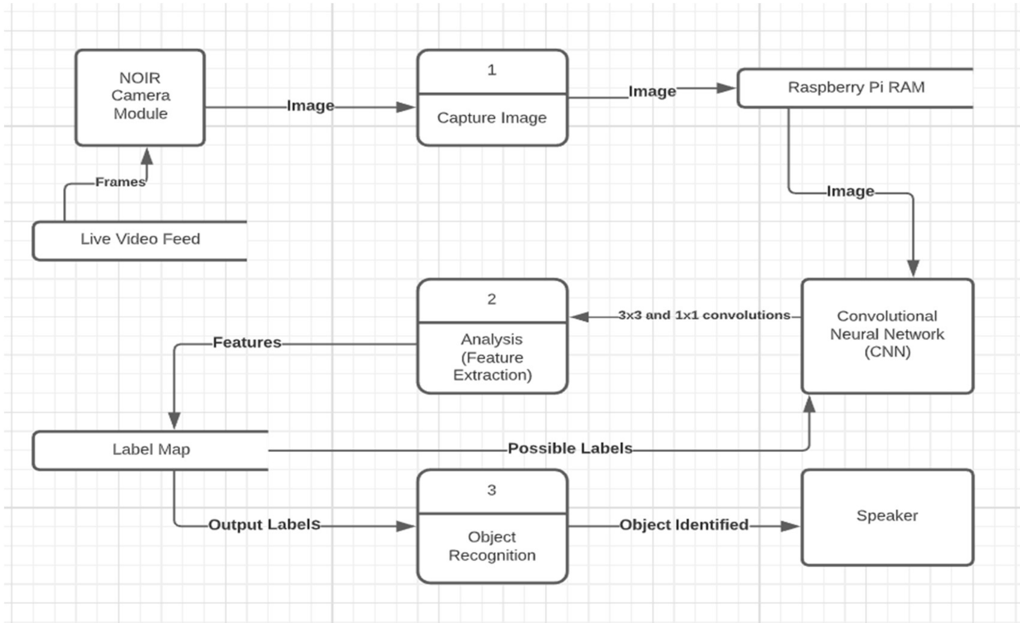



**Fig 8. Data Flow Diagram for Object Recognition**

## B. Obstacle Sensing

We have used the HC-SR04 Ultrasonic Sensor to perform this functionality. The accurate range of this sensor is 2cm to 400cm which means that it can correctly measure distance between a user and obstacle when the obstacle is no less than 2cm and no more than 400cm away from the sensor.

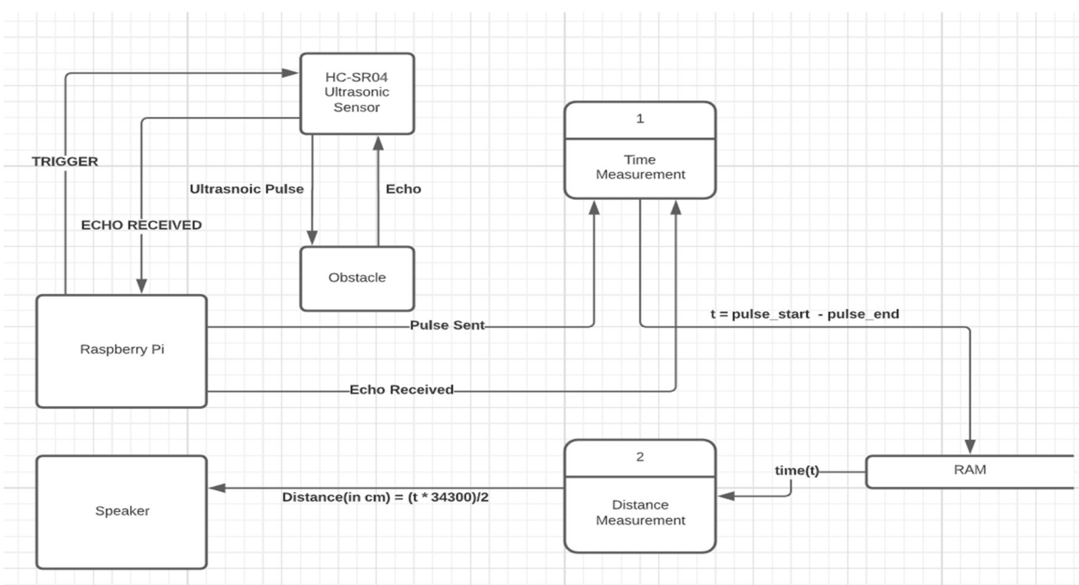



**Fig 9. Data Flow Diagram for Ultrasonic Distance Sensing.**

The VCC pin of the sensor is connected to the GPIO 5V pin 1 on the Pi for power supply. The trigger pin is connected to GPIO pin 23 while echo is connected to GPIO pin 24 on the Raspberry Pi. The ground pin of the sensor will be connected to pin 3 on the Pi. When the code is executed, the sensor will send out an ultrasonic pulse. The obstacle will reflect the pulse back and the sensor picks up this echo. The time interval between sending the echo and receiving the pulse is calculated by the program which is then used to find out the distance of the obstacle. The formula used to calculate distance takes into account the speed of sound in air and is explained in detail in chapter 6.

**C. Location Sensing**

For the purpose of determining the location of the blind person in real time we have made use of a Ublox Neo 6M GPS Module which is easily interfaced with the Raspberry Pi. The VCC pin of the GPS is connected to the 5V pin 1 of the Pi, the transceiver TX pin is connected to the receiver RX GPIO pin 15 of the Raspberry Pi and finally the gorund GND pin is connected to pin 3 on the Pi. After the connections are made the unit is powered up and left in the open to lock a satellite signal. Once the blue LED on the GPS module is blinking it means that location data is being received. The program will extract any of the NMEA formatted sentences and obtain the latitude and longitude coordinates. These coordinates are then converted to a location address and all the location information uploaded to the cloud Firebase.

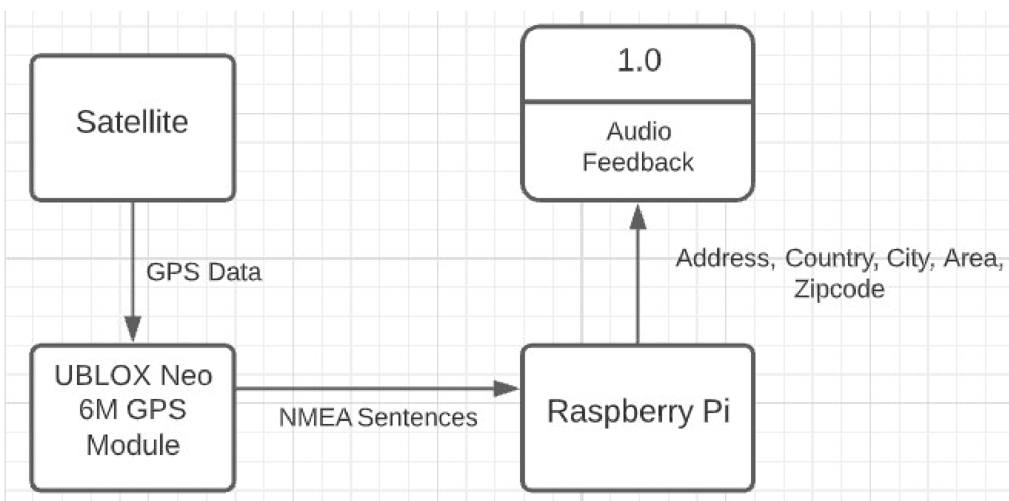



**Fig 10. Data Flow Diagram for GPS Location Sensing**

## 4.6. Design details/ Architecture

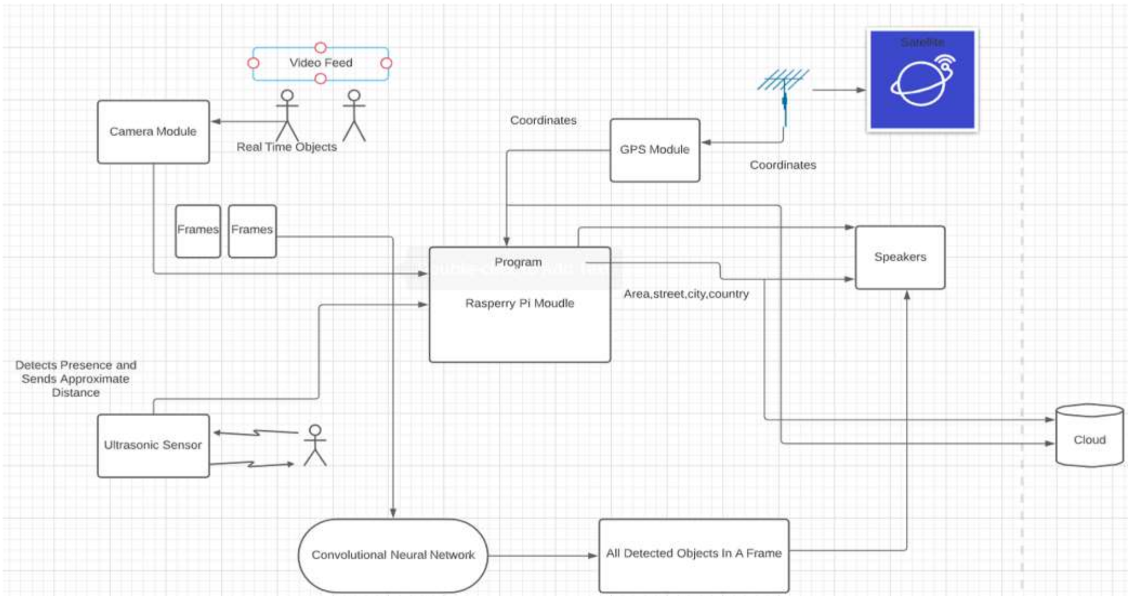



**Fig 11. Architecture of our Proposed System**

The first functionality offered by our prototype product is object detection and recognition. To implement this the camera module will obtain a live video feed and in this feed one image or frame will be sent to the program for analysis. The coded CNN will perform feature extraction and match the object to a specific label. This label is nothing but the prediction made by the neural network as to what the object is.

The second functionality we are implementing is obstacle distance sensing. Here we have interfaced an ultrasonic sensor to send out an ultrasonic pulse in the direction of the obstacle and the echo pulse will be received by the sensor. The exchange of signals will happen at the I/O pins of the Pi and sensor. The time interval between the instant the plse was sent and the instant the echo was received is recorded and the distance between sensor and obstacle is outputted after being calculated by a formula coded in the program.

The final feature we are incorporating in our project is GPS Location sensing. We will be implementing this feature in our final semester i.e. semester 8. For this we have the GPS Module with the antenna ready to interface with the Raspberry Pi. Once the module has been locked on with a satellite GPS data will be retrieved by means of a program and using this data the current location will be fetched in parallel with Google Maps.

For the purpose of giving the visually impaired person audio messages we will be interfacing a Bluetooth Speaker. By means of the speaker, the user will get audio cues about objects identified by the CNN, the distance between the user and an obstacle and also will be given an audio representation of their location in terms of street name, city, country, zipcode, etc.

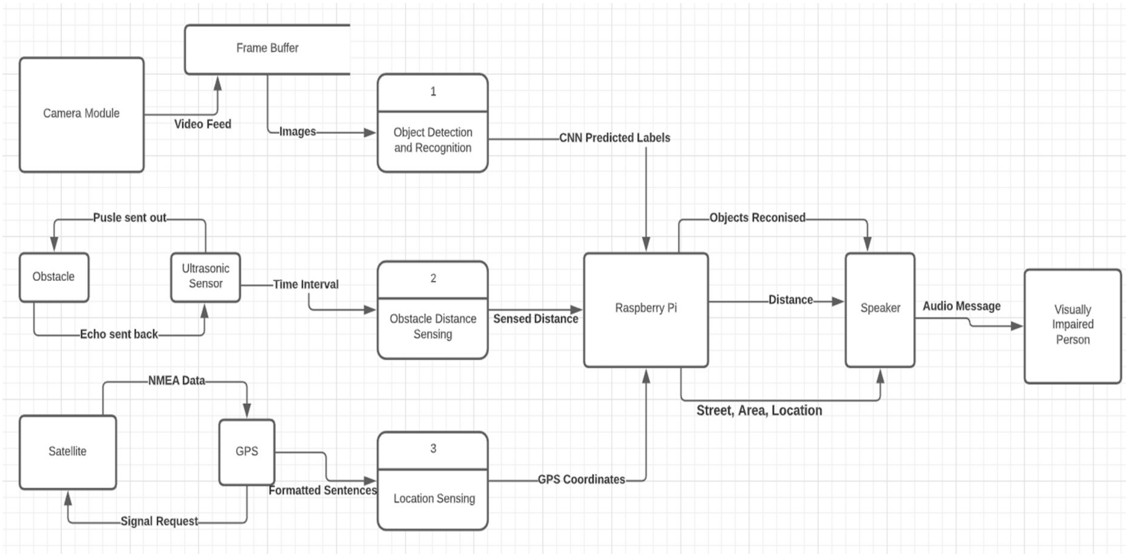



**Fig 12. Data Flow Diagram for the Architecture**

# 5. IMPLEMENTATION

## 5.1. Module-I Object Recognition:

We have used the convolutional neural network (CNN) model to implement object detection and recognition. Convolutional Neural networks are a family of deep learning algorithms that take parts of images as inputs, apply certain weights and biases and differentiate one part of the image from another. For object detection due to the limited RAM and processing power available on the Raspberry Pi, we have made use of a neural network that is computationally less intensive but still accurate namely the MobileNet architecture.

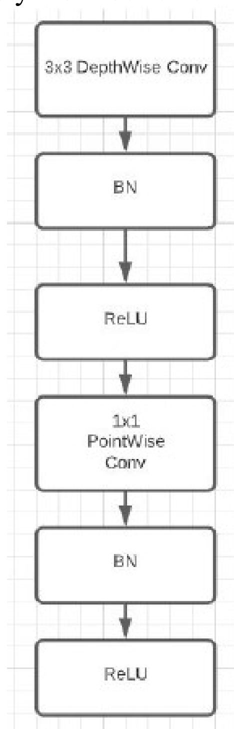



**Fig 13. A single layer in our MobileNet architecture**

For the purpose of training the neural network and generating the label map, we have used the COCO image dataset. COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

- Object segmentation
- Recognition in context
- Superpixel stuff segmentation
- 330K images (>200K labeled)
- 1.5 million object instances
- 80 object categories.

The NOIR Camera Module of the Raspberry Pi once interfaced provides the Python program the live video feed. We have implemented the MobileNet Neural Network using the interfaces provided by Tensorflow. The video frames are given as input to the network which then performs a feature extraction process and compares it to the features that it learnt during the training phase. Once the probability of the labels are calculated , a rectangular box is drawn around the object with the label of that object and its probability score. The Bluetooth speaker then sounds a message to indicate what object has been recognized. This entire process happens in a matter of seconds.

## 5.2. Module-II Obstacle Detection

We made use of an ultrasonic sensor to detect the presence of obstacles and inform the user about how far or close they are. Extra caution had to be taken when implementing the circuit between the sensor and GPIO pins. The ultrasonic sensor operates at 5V and this voltage was supplied by connecting GPIO pin 1 to the VCC pin of the sensor. Since all the GPIO pins (except for pin 1) operate at 3.3V and the signals sent by the sensor at 5V, care has to be taken to lower the voltage. To do this we made use of a voltage divider circuit implemented on a breadboard. We used 560 ohm and 1000 ohm resistors to lower the 5V echo signal to a 3.3V signal.

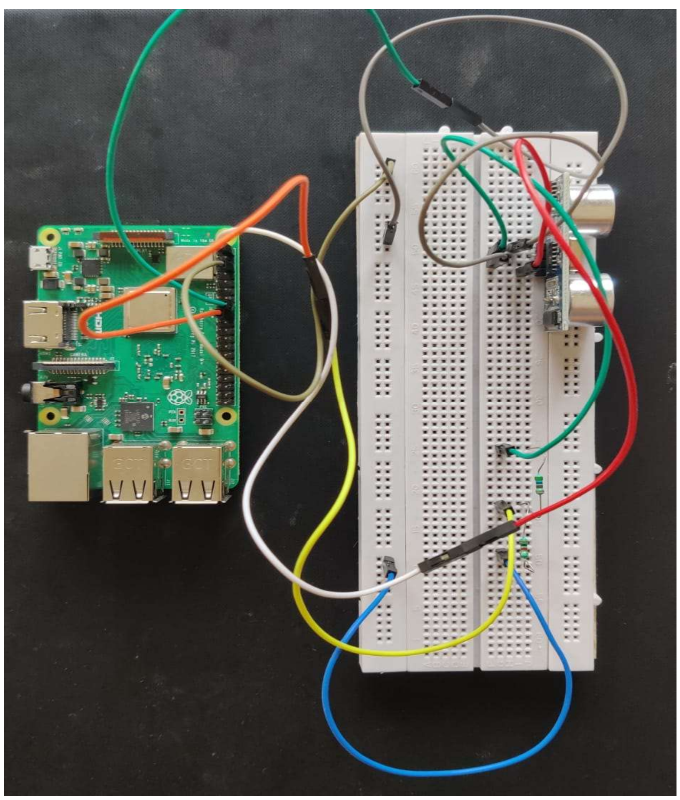


**Fig 14. Setup for Ultrasonic Distance Measurement**

When the program is executed, the GPIO Pin 24 will send a signal to the TRIG pin of the sensor triggering it to send an ultrasonic pulse out. The program will note this as the start of the time interval. The echo from the obstacle will return to the sensor causing a signal to be sent to the ECHO pin of the sensor. The ECHO signal is lowered from 5V to 3.3V by the voltage divider and then sent to the GPIO Pin 23 of the Raspberry Pi. The program notes this as the end of the pulse interval. The time taken for the pulse to hit the object and for its echo to return is calculated as,

t = pulse_end – pulse_start

Finally the distance between obstacle and sensor is calculated as,

D = (T * 34300)/2,

where speed of sound is 343 m per sec or 34300 cm per sec.

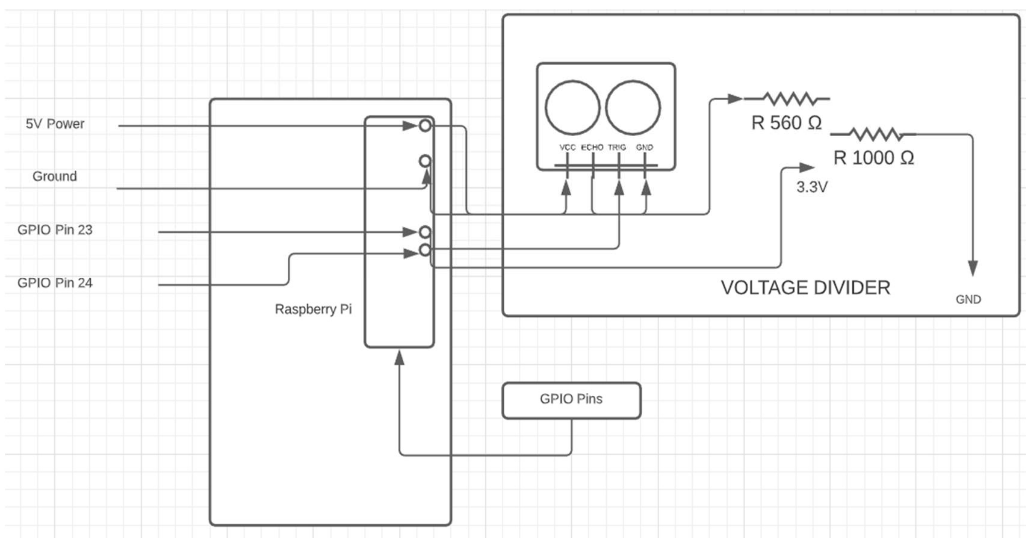



**Fig 15. Diagrammatic Representation of the setup**

## 5.3. Module-III Location Sensing

In order to allow the blind person to know where exactly he/she is when navigating outdoors, a GPS sensor has been used. The Ublox Neo 6M GPS Module allows for easy interfacing with the Raspberry Pi. The antenna once attached to the sensor allows for GPS data to be received. The antenna has to be left exposed to the sky in order for a satellite to lock onto it and start transmitting GPS data.

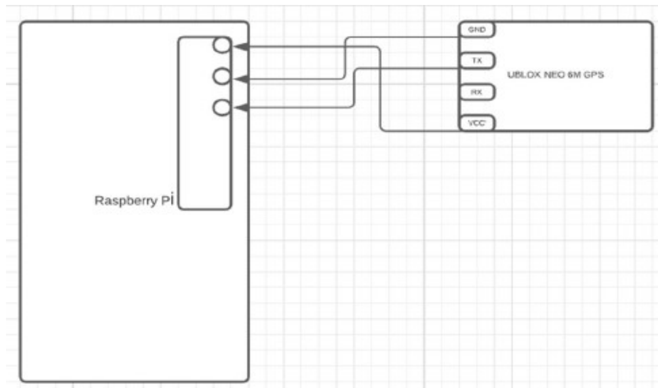



**Fig 16. Diagrammatic representation of GPS Location Sensing setup**

There are 3 pins that are using to take the received data from the module:
a. GND: this pin is connected to pin 3 of the Pi for providing electrical grounding.
b. VCC: this pin is connected to pin 1 of the Pi which provides a 5V power supply.
c. TX: the transceiver pin of the module is connected to the RX pin of the Pi which is pin 5. RX is the receiver pin of the Pi which will transfer received GPS data

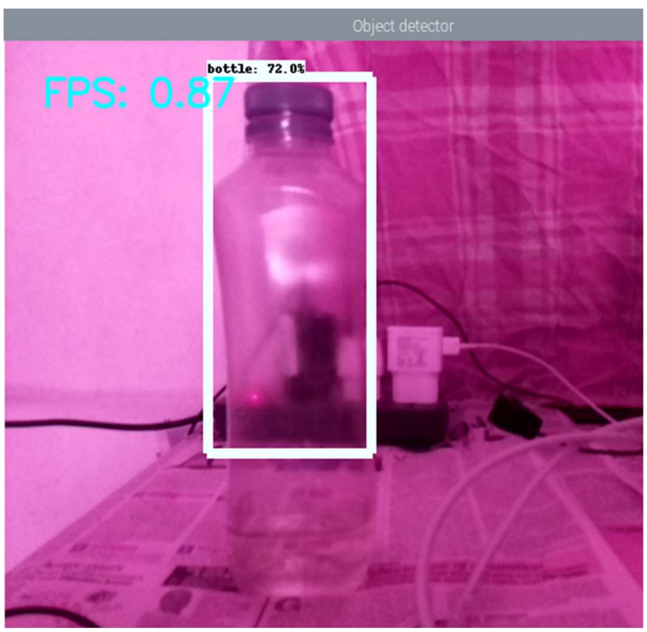
In python the GeoPy, PyNMEA2 and GeoPandas libraries have been used. The GPS Module receives location data in the form of NMEA sentences. In the Python code, the GPRMC(Recommended minimum specific GPS/Transit data) sentence has been extracted and used to obtain latitude and longitude coordinates of the module's current location.
An example of a GPRMC sentence:
eg1. $GPRMC,081836,A,3751.65,S,14507.36,E,000.0,360.0,130998,011.3,E*62
eg2. $GPRMC,225446,A,4916.45,N,12311.12,W,000.5,054.7,191194,020.3,E*68

| | |
|---|---|
| 225446 | Time of fix 22:54:46 UTC |
| A | Navigation receiver warning A = OK, V = warning |
| 4916.45,N | Latitude 49 deg. 16.45 min North |
| 12311.12,W | Longitude 123 deg. 11.12 min West |
| 000.5 | Speed over ground, Knots |
| 054.7 | Course Made Good, True |
| 191194 | Date of fix 19 November 1994 |
| 020.3,E | Magnetic variation 20.3 deg East |
| *68 | mandatory checksum |

Reverse GeoCoding i.e. coordinates to address has been performed by the program to obtain the address. As usual this location address is sounded to the blind person by means of the Bluetooth speaker.
Since we wanted to upload the location data to the cloud we have made use of the pyrebase library as well. The library allows us to create nodes in Firebase and upload latitude, longitude and address to a new node each time the blind person uses the GPS.

# 6. RESULTS

## A. Object Recognition:


**Figs 17, 18. 19. Object Recognition Output for a few objects**

The output we have taken for object recognition was done using some common household objects. The neural network was able to accurately recognise the objects we put in front of the camera withing a matter of just 5 or less seconds which is extremely good considering that the Raspberry Pi model we are using has just 1 GB of RAM. We testout out the neural network on a few other items and a similarly accurate speed output was obtained. The objects that were recognised were sounded on the Bluetooth speaker using a particular format. For example, if the cup was detected the message sounded was “Cup Detected!”.

## B. Obstacle Sensing:

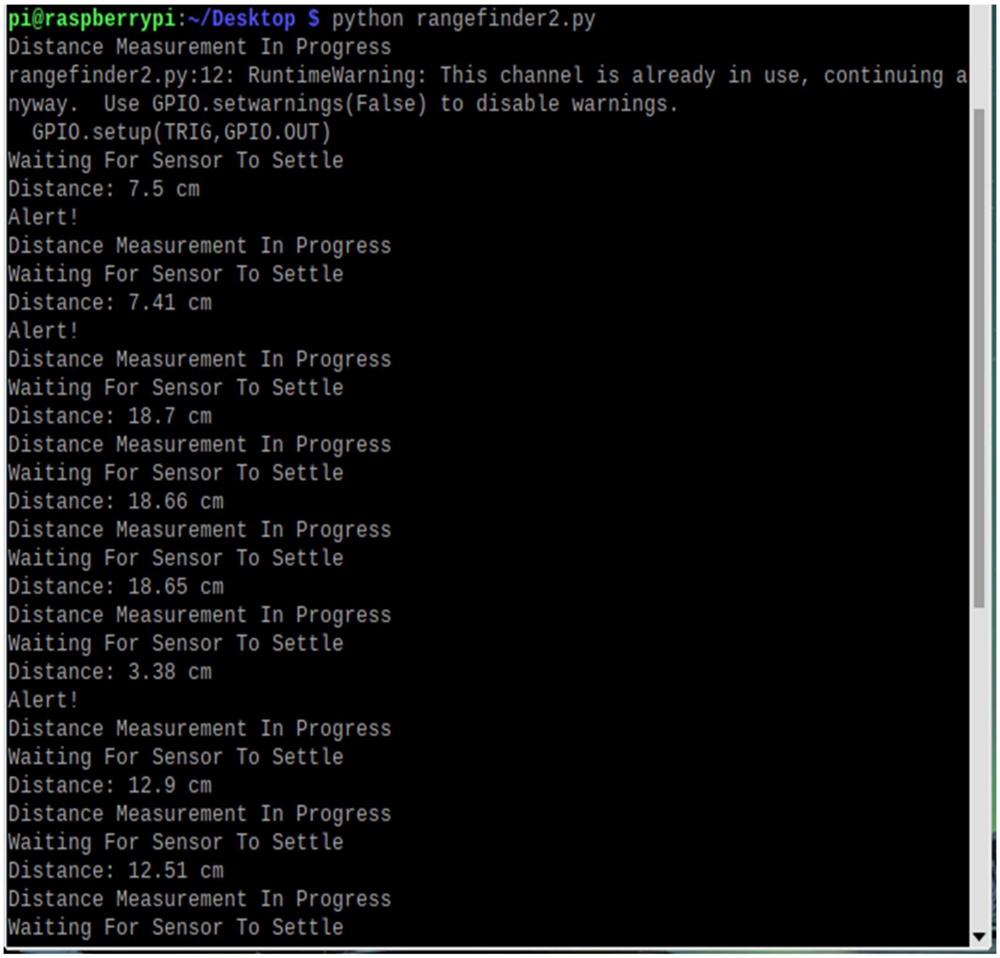



**Fig 20. Obstacle Sensing Output**

In the obstacle sensing feature using the ultrasonic sensor, whenever the obstacle is at a distance of 15cm or less then an Alert message is sounded on the Bluetooth speaker to warn the blind person that they are walking into something. We have chosen to keep 15cm as the range at which the  warning is given however the range can always be changed to whatever is needed in the program. The speaker gave messages in a fixed format. For example if the obstacle was 10 cm away the message sounded was “Warning obstacle is 10 centimetres away!”.

### C. Location Sensing

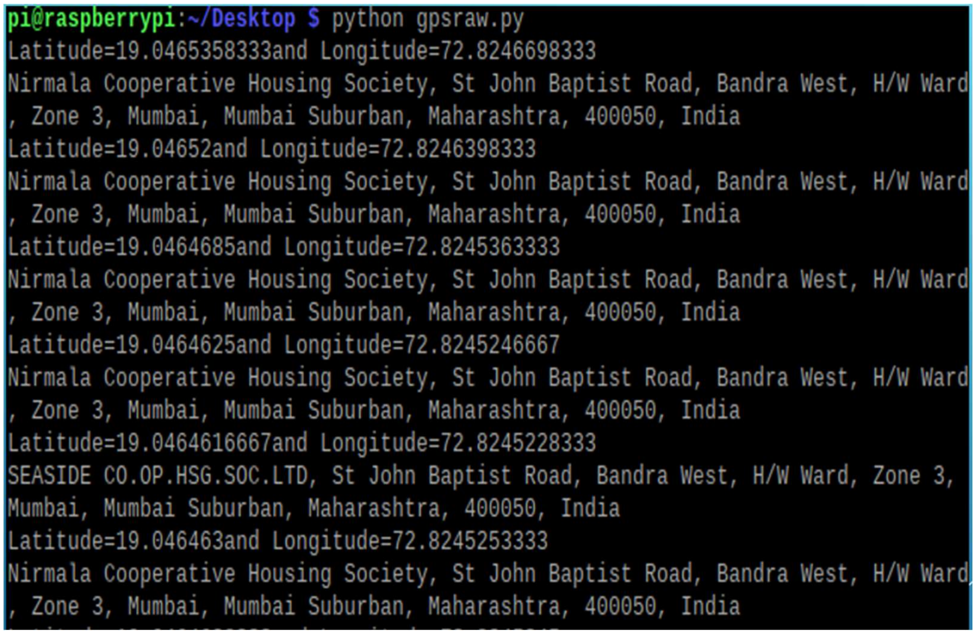



**Fig 21. GPS Location Sensing Output**

The GPS Module was collecting a strong satellite signals outdoors. The $GPGGA NMEA sentence was used to extract the latitude and longitude coordinates as seen in the screenshot. The coordinates were then reverse geocoded to address. The message sounded by the speaker for the address given in the screenshot is “You are currently at Nirmala Coorperative Housing Society, St John Baptist Rd, Bandra West, Mumbai Suburban, Maharashtra, 400050 India.”

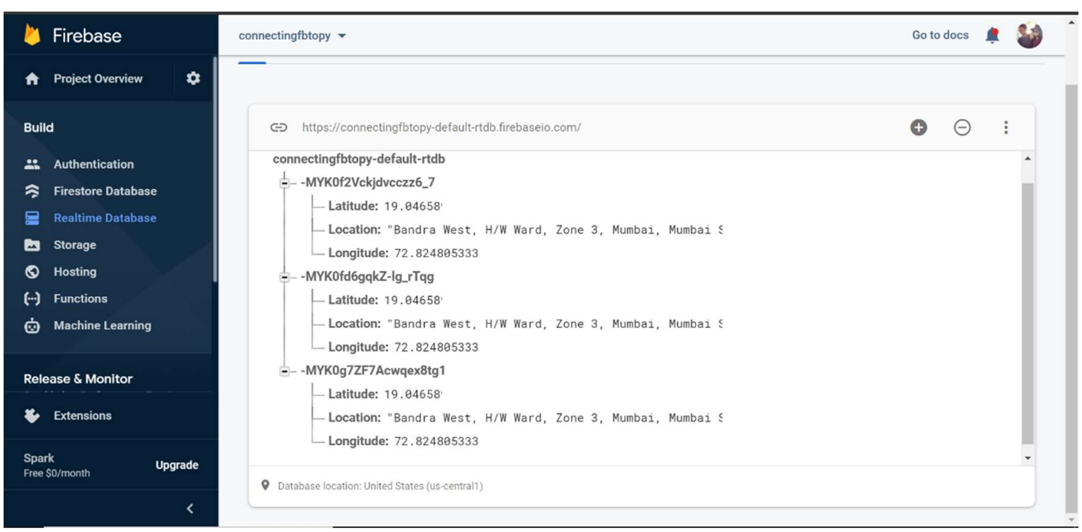



**Fig 22. GPS Location Sensing Output in Firebase Database.**

# 7. PROJECT TIMELINE AND TASK DISTRIBUTION

**Project Timeline:**
The timeline of work and events of our project can be split across semesters 7 and 8 in the form of tables and charts.

**Semester 7:**
We had managed to complete implementation of the Object Recognition and Obstacle Detection features of our prototype.

| **Month, Year** | **Tasks/Responsibilities** |
|---|---|
| September 2020 | Collecting all the remaining hardware that is required for initiating and completing the project. We will finish off all basic technical activities like installing the Raspbian OS on the Raspberry Pi and interfacing all the components ensuring that they are functioning correctly. |
| October 2020 | We will begin the main technical tasks of coding out the convolution neural network incorporating Google's Tensorflow framework. Appropriate datasets will be used for training the network. Coding will also be done for sensing and perceiving the distance of the object by the ultrasonic sensor. |
| November 2020 | Since we want GPS location to be a feature of our project whereby audio output of street, city, etc. is given so the person knows where he/she is , coding for utilizing the GPS Module will be completed. A cloud storage will be used here. |
| December 2020 | Any bugs in the code will be rectified or if any feature is not working as expected it will be taken care of. We will test the device in different scenarios to see how fast and accurate the results are being generated. |
| January 2021 | Any additional features like a reading feature or face recognition feature which recognizes faces of people known to the blind person may be developed depending on how the status of the work of previous months. |

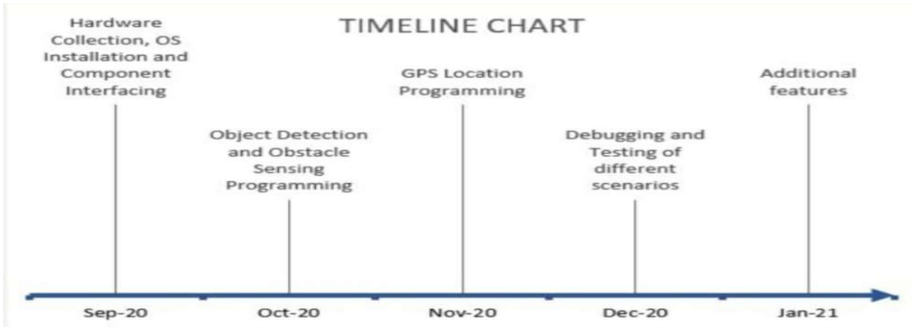



**Fig 23. Timeline chart for semester 7**

**Semester 8:**

| **Month, Year** | **Tasks/Responsibilities** |
|---|---|
| January 2021 | Find out background information on the GPS and Bluetooth Speaker and interface the components with the Raspberry Pi. |
| February 2021 | Complete programming for Location sensing and voice feedback. |
| March 2021 | Integrating Voice feedback with the previous features implemented and brush up on any remaining coding. |
| April 2021 | Mainly final documentation work. |

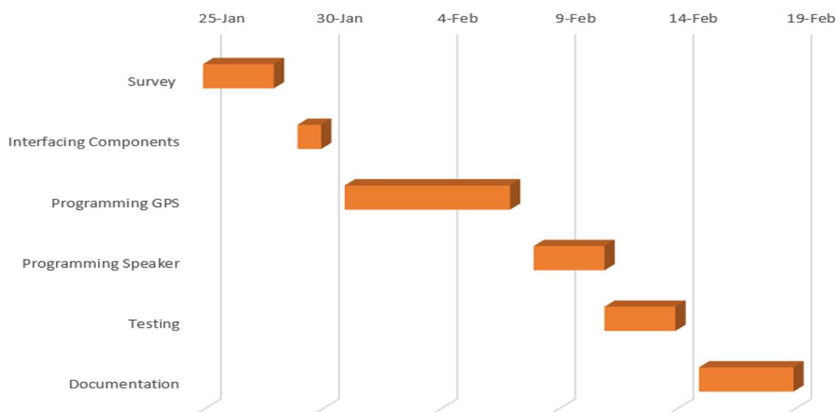



**Fig 24. Timeline chart for semester 8.**

## Task Distribution:

| Role Title | Salil | Jordan | Anthony |
|---|---|---|---|
| Hardware Collection | I,C | A | R |
| Assembly | R,A | I | I |
| Programming | R,A | I,C | I,C |
| Testing | R | I,C | A |
| Debugging | R | I,C | A |
| Documentation | I,C | A | R |

R – Responsible
A – Accountable
C – Consulted
I - Informed

**Fig 25. Responsibilty Matrix**

We decided on how the overall work needed to complete the project could be broken up into six main activities of collecting necessary hardware, interfacing components, programming/ coding, testing under different test cases, debugging any errors and finally documentation of the work. The responsibility matrix we came up with ensured an equitable distribution of tasks such that each group member contributed in every aspect of the project equally.

# 8. CONCLUSION

As stated clearly in the introduction, we are committed to the comfort of the user. He/she should be able lead a near normal life with the help of our proposed project. We were also able to decide and design the model as well as complete a part of it i.e. object detection. The code that we implemented was able to successfully distinguish between a non-living object and a human being with considerable amount of accuracy.

The objectives that were planned out of this semester have been successfully fulfilled, from idea formulation to research to design to code implementation. The work was divided successfully among the group members be it research of IEEE papers for idea formulation, to designing the architecture of the system and eventually coding object detection part.

We had our guide who reviewed our work weekly and gave us necessary instructions wherever possible.

Last but not least, we had our project panel who approved of the work done from idea formulation to object detection and lauded us for the work done as well as constructively criticized us in certain aspects that needed amendment. We have planned for the next semester and hopefully we achieve our targets well in advance so that if possible, we can integrate more functionalities that will appeal
the user.

# REFERENCES

[1] Ren, Shaoqing, et al. “Faster R-CNN: Towards real-time object detection with region proposal networks.” Advances in neural information processing systems, 2015.

[2] Chi-Sheng, Hsieh. “Electronic walking stick for the blind.” U.S. Patent No. 5,097,856, 24 Mar. 1992.

[3] Evanitsky, Eugene. “Portable blind aid device.” U.S. Patent No. 8,606,316, 10 Dec. 2013.

[4] Cervantes, Humberto Orozco. “Intelligent glasses for the visually impaired.” U.S. Patent No. 9,488,833. 8 Nov. 2016.

[5] Jothimani, A., Shirly Edward, and G. K. Divyashree. “Object Identification for Visually Impaired.” Indian Journal of Science and Technology 9.S1, 2016.

[6] Zraqou, Jamal S., Wissam M. Alkhadour, and Mohammad Z. Siam. “Real-Time Objects Recognition Approach for Assisting Blind People.”, 2017.

[7] Ananth Noorithaya, M. Kishore Kumar, A. Sridevi. “Voice assisted system for the blind”, 2015. https://doi.org/10.1109/CIMCA.2014.7057785

[8] Ruolin Wang, Chun Yu, Xing-Dong Yang, Weijie He, Yuanchun Shi. 2019. EarTouch: Facilitating Smartphone Use for Visually Impaired People in Mobile and Public Scenarios. In CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland Uk. ACM, New York, NY, USA, 13 pages. https: //doi.org/10.1145/3290605.3300254

[9] Conference on Human Factors in Computing Systems (CHI ’11). ACM, New York, NY, USA, 3247–3256. https://doi.org/10.1145/1978942.1979424 Mounir Bousbia-Salah, Abdelghani Redjati, Mohamed Fezari, Maamar Bettayeb. “An Ultrasonic Navigation System For Blind People”.2007. https://www.researchgate.net/publication/251851635

[10] Rachid Sammouda, Ahmad AlRjoub. “Mobile Blind Navigation System Using RFID”.2015. https://ieeeexplore.ieee.org/document/7353325

[11] Zraqou, Jamal S., Wissam M. Alkhadour, and Mohammad Z. Siam. “Real-Time Objects Recognition Approach for Assisting Blind People.”, 2017.

[12] Simonyan, Karen, and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition.” arXiv preprint arXiv:1409.1556 (2014).

[13] Abadi, Martn, et al. “Tensorflow: Large-scale machine learning on heterogeneous distributed systems.” arXiv preprint arXiv:1603.04467 (2016).

[14] Ren, Shaoqing, et al. "Object detection networks on convolutional feature maps." IEEE transactions on pattern analysis and machine intelligence 39.7 (2017): 1476-1481.

[15] Anika Nawer, Farhana Hossain, Md. Galib Anwar. "Ultrasonic Navigation System for the visually impaired & blind pedestrians" .2015.
https://www.researchgate.net/publication/283153904

[16] Shiri Azenkot, Sanjana Prasain, Alan Borning, Emily Fortuna, Richard E. Ladner, and Jacob O. Wobbrock. 2011. Enhancing Independence and Safety for Blind People.

# APPENDIX

| Alphabet | Formal Definition |
|---|---|
| A | **1. Arduino:** Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices.<br><br>**2. Artificial Intelligence (AI):** Artificial intelligence, is intelligence demonstrated by machines, unlike the natural intelligence displayed by humans and animals. |
| B | **1. Blind:** unable to see because of injury, disease, or a congenital condition.<br><br>**2. Batch-Normalization:** Batch normalization is a method used to make artificial neural networks faster and more stable through normalization of the input layer by re-centering and re-scaling.<br><br>**3. Bluetooth:** Bluetooth is a wireless technology standard used for exchanging data between fixed and mobile devices over short distances using UHF radio waves in the industrial, scientific and medical radio bands, from 2.402 GHz to 2.480 GHz, and building personal area networks (PANs). It was originally conceived as a wireless alternative to RS-232 data cables. |
| C | **Convolutional Neural Network (CNN):** In deep learning, a convolutional neural network is a class of deep neural networks, most commonly applied to analyzing visual imagery. |

| | |
|---|---|
| **D** | **Deep Learning:** Deep learning is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised. |
| **E** | **Expansion Layer**: The expansion layer acts as a decompressor (like unzip) that first restores the data to its full form, then the depth-wise 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. |
| **F** | **Frames:** Frames are an artificial intelligence data structure used to divide knowledge into substructures by representing "stereotyped situations". ... Frames are the primary data structure used in artificial intelligence frame language; they are stored as ontologies of sets. |
| **G** | **1. Global Positioning System(GPS):** The Global Positioning System, originally Navstar GPS, is a satellite-based radionavigation system owned by the United States government and operated by the United States Space Force.<br><br>**2. Gantt Chart:** A Gantt chart is a type of bar chart that illustrates a project schedule, named after its inventor, Henry Gantt. |
| **K** | **Keras:** Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. |
| **M** | **MobileNet:** MobileNet is a streamlined architecture that uses depth-wise separable convolutions to construct lightweight deep convolutional neural networks and provides an efficient model for mobile and embedded vision applications. |

| | |
|---|---|
| **O** | **1. Object Segmentation:** A process of image analysis. Its purpose is to extract the interesting region from image (corresponding to the interesting objects in scene).<br><br>**2. Object Detection:** Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. |
| **P** | **1. Portable:** Able to be easily carried or moved, especially because being of a lighter and smaller version than usual.<br><br>**2. Plethora: A** large or excessive amount of something. |
| **R** | **1. RTK:** Real-time kinematic (RTK) positioning is a satellite navigation technique used to enhance the precision of position data derived from satellite-based positioning systems (global navigation satellite systems, GNSS) such as GPS, GLONASS, Galileo, NavIC and BeiDou.<br><br>**2. RAM:** RAM is basically the short-term memory of your computer. This is the place where your computer keeps track of the programs and data you are using at the moment.<br><br>**3. Raspberry pi:** It is a small microcontroller that can act as a processor too. It is open source and has many applications.<br><br>**4. RFID**: Radio-frequency identification (RFID) uses electromagnetic fields to automatically identify and track tags attached to objects. An RFID tag consists of a tiny radio transponder; a radio receiver and transmitter.<br><br>**5. RELU:** Rectified Linear Activation Function is the most common choice of activation function in the world of deep learning. **Relu** provides state of the art results and is computationally very efficient at the same time. |

| | |
|---|---|
| **S** | **1. Scope:** the extent of the area or subject matter that something deals with or to which it is relevant.<br><br>**2. SLAM** **(Simultaneous Localization and Mapping):** In computational geometry and robotics, simultaneous localization and mapping (SLAM) is the computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it. |
| **U** | **Ultrasonic sensor:** The HC-SR04 ultrasonic sensor uses SONAR to determine the distance of an object just like the bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package from 2 cm to 400 cm or 1" to 13 feet. |
| **V** | **Visualization:** The representation of an object, situation, or set of information as a chart or other image |

# Coding Conventions

**Source code:**

**A. Object Recognition Using Tensorflow**

```
# Import packages
import os
import cv2
import numpy as np
from picamera.array import PiRGBArray
from picamera import PiCamera
import tensorflow as tf
import argparse
import sys
import subprocess

import subprocess

#function to for the Bluetooth speaker to speak using eSpeak
def execute_unix(inputcommand):
p = subprocess.Popen(inputcommand, stdout=subprocess.PIPE, shell=True)
(output, err) = p.communicate()
return output

# Set up camera constants
IM_WIDTH = 640
IM_HEIGHT = 480
#IM_WIDTH = 640    Use smaller resolution for
#IM_HEIGHT = 480   slightly faster framerate

# Select camera type (if user enters --usbcam when calling this script,
# a USB webcam will be used)
camera_type = 'picamera'
parser = argparse.ArgumentParser()
parser.add_argument('--usbcam', help='Use a USB webcam instead of picamera',
action='store_true')
args = parser.parse_args()
```

```
if args.usbcam:
camera_type = 'usb'

# This is needed since the working directory is the object_detection folder.
sys.path.append('..')

# Import utilites
from utils import label_map_util
from utils import visualization_utils as vis_util

# Name of the directory containing the object detection module we're using
MODEL_NAME = 'ssdlite_mobilenet_v2_coco_2018_05_09'

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

# Path to frozen detection graph .pb file, which contains the model that is used
# for object detection.
PATH_TO_CKPT =
os.path.join(CWD_PATH,MODEL_NAME,'frozen_inference_graph.pb')

# Path to label map file
PATH_TO_LABELS = os.path.join(CWD_PATH,'data','mscoco_label_map.pbtxt')

# Number of classes the object detector can identify
NUM_CLASSES = 90

## Load the label map.
label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
categories = label_map_util.convert_label_map_to_categories(label_map,
max_num_classes=NUM_CLASSES, use_display_name=True)
category_index = label_map_util.create_category_index(categories)

# Load the Tensorflow model into memory.
detection_graph = tf.Graph()
with detection_graph.as_default():
od_graph_def = tf.compat.v1.GraphDef()
with tf.io.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
serialized_graph = fid.read()
```

```
od_graph_def.ParseFromString(serialized_graph)
tf.import_graph_def(od_graph_def, name='')

sess = tf.compat.v1.Session(graph=detection_graph)


# Define input and output tensors (i.e. data) for the object detection classifier

# Input tensor is the image
image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')

# Output tensors are the detection boxes, scores, and classes
# Each box represents a part of the image where a particular object was detected
detection_boxes = detection_graph.get_tensor_by_name('detection_boxes:0')

# Each score represents level of confidence for each of the objects.
# The score is shown on the result image, together with the class label.
detection_scores = detection_graph.get_tensor_by_name('detection_scores:0')
detection_classes = detection_graph.get_tensor_by_name('detection_classes:0')

# Number of objects detected
num_detections = detection_graph.get_tensor_by_name('num_detections:0')

# Initialize frame rate calculation
frame_rate_calc = 1
freq = cv2.getTickFrequency()
font = cv2.FONT_HERSHEY_SIMPLEX

# Initialize camera and perform object detection.

### Picamera ###
if camera_type == 'picamera':
# Initialize Picamera and grab reference to the raw capture
camera = PiCamera()
camera.resolution = (IM_WIDTH,IM_HEIGHT)
camera.framerate = 10
rawCapture = PiRGBArray(camera, size=(IM_WIDTH,IM_HEIGHT))
rawCapture.truncate(0)
```

```
for frame1 in camera.capture_continuous(rawCapture, format="bgr",use_video_port=True):

t1 = cv2.getTickCount()

# Acquire frame and expand frame dimensions to have shape: [1, None, None, 3]
# i.e. a single-column array, where each item in the column has the pixel RGB value
frame = np.copy(frame1.array)
frame.setflags(write=1)
frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
frame_expanded = np.expand_dims(frame_rgb, axis=0)

# Perform the actual detection by running the model with the image as input
(boxes, scores, classes, num) = sess.run(
[detection_boxes, detection_scores, detection_classes, num_detections],
feed_dict={image_tensor: frame_expanded})

# Draw the results of the detection (aka 'visulaize the results')
vis_util.visualize_boxes_and_labels_on_image_array(
frame,
np.squeeze(boxes),
np.squeeze(classes).astype(np.int32),
np.squeeze(scores),
category_index,
use_normalized_coordinates=True,
line_thickness=8,
min_score_thresh=0.40)
obs=[category_index.get(value) for index,value in enumerate(classes[0]) if scores[0,index]>0.5]
if(len(obs)!=0):
det_ob = obs[0]['name']
print(det_ob)
string = det_ob+" Detected"
c = 'espeak -ven+m4 -k5 -s140 --punct="<characters>" "%s" 2>>/dev/null' % string
execute_unix(c)
cv2.putText(frame,"FPS: {0:.2f}".format(frame_rate_calc),(30,50),font,1,(255,255,0),2,cv2.LINE_AA)

# All the results have been drawn on the frame, so it's time to display it.
```

```
cv2.imshow('Object detector', frame)

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

# Press 'q' to quit
if cv2.waitKey(1) == ord('q'):
break

rawCapture.truncate(0)

camera.close()

### USB webcam ###
elif camera_type == 'usb':
# Initialize USB webcam feed
camera = cv2.VideoCapture(0)
ret = camera.set(3,IM_WIDTH)
ret = camera.set(4,IM_HEIGHT)

while(True):

t1 = cv2.getTickCount()

# Acquire frame and expand frame dimensions to have shape: [1, None, None, 3]
# i.e. a single-column array, where each item in the column has the pixel RGB value
ret, frame = camera.read()
frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
frame_expanded = np.expand_dims(frame_rgb, axis=0)

# Perform the actual detection by running the model with the image as input
(boxes, scores, classes, num) = sess.run(
[detection_boxes, detection_scores, detection_classes, num_detections],
feed_dict={image_tensor: frame_expanded})

# Draw the results of the detection (aka 'visulaize the results')
vis_util.visualize_boxes_and_labels_on_image_array(
frame,
```

```
np.squeeze(boxes),
np.squeeze(classes).astype(np.int32),
np.squeeze(scores),
category_index,
use_normalized_coordinates=True,
line_thickness=8,
min_score_thresh=0.85)

cv2.putText(frame,"FPS:
{0:.2f}".format(frame_rate_calc),(30,50),font,1,(255,255,0),2,cv2.LINE_AA)

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

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

# Press 'q' to quit
if cv2.waitKey(1) == ord('q'):
break
if p is not None:
string=p+" detected"
print(string)
camera.release()

cv2.destroyAllWindows()
```

**B. Obstacle Detection Using Ultrasonic Sensing**

```
import RPi.GPIO as GPIO
import time
import subprocess

def execute_unix(inputcommand):
p = subprocess.Popen(inputcommand, stdout=subprocess.PIPE, shell=True)
(output, err) = p.communicate()
return output
```

```
GPIO.setmode(GPIO.BCM)

TRIG = 23
ECHO = 24
while 1:

GPIO.setmode(GPIO.BCM)
print("Distance Measurement In Progress")
GPIO.setup(TRIG,GPIO.OUT)
GPIO.setup(ECHO,GPIO.IN)

GPIO.output(TRIG, False)
print("Waiting For Sensor To Settle")
time.sleep(2)

GPIO.output(TRIG, True)
time.sleep(0.00001)
GPIO.output(TRIG, False)

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 * 17000

distance = round(distance, 2)

print("Distance:",distance,"cm")

# 1 foot = 12 inches = 30.48cm
if distance <= 31:
dist = str(distance)
string = "Careful obstacle is "+dist+" centimeters away"
c = 'espeak -ven+m4 -k5 -s140 --punct="<characters>" "%s" 2>>/dev/null' % string
execute_unix(c)
```

```
    print("Alert!")
    time.sleep(2)
    GPIO.cleanup()
```

**C. Location Sensing using GPS and FireBase**

```
import serial
import time
import string
import pynmea2
import pandas as pd
#import geopandas as gpd
import geopy
import subprocess
import pyrebase

firebaseConfig = {
"apiKey": "AIzaSyBRYIpKhCOMrf9wSJhKGoupsaRq-AxYq0o",
"authDomain": "connectingfbtopy.firebaseapp.com",
"projectId": "connectingfbtopy",
"databaseURL": "https://connectingfbtopy-default-rtdb.firebaseio.com/",
"storageBucket": "connectingfbtopy.appspot.com",
"messagingSenderId": "921973152097",
"appId": "1:921973152097:web:56e04327d2f65c039f8d20",
"measurementId": "G-DPMCW1GJSB"
};
firebase = pyrebase.initialize_app(firebaseConfig)
db = firebase.database()


def execute_unix(inputcommand):
p = subprocess.Popen(inputcommand, stdout=subprocess.PIPE, shell=True)
(output, err) = p.communicate()
return output

from geopy.geocoders import Nominatim
#from geopy.extra.rate_limiter import Ratelimiter

while True:
```

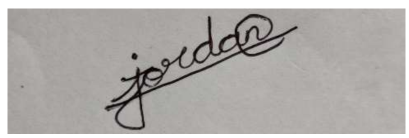
```
port="/dev/ttyS0"
ser=serial.Serial(port, baudrate=9600, timeout=0.5)
dataout = pynmea2.NMEAStreamReader()
newdata=ser.readline()

if newdata[0:6] == "$GPRMC":
newmsg=pynmea2.parse(newdata)
lat=newmsg.latitude
lng=newmsg.longitude
gps = "Latitude:" + str(lat) + " Longitude:" + str(lng)
print(gps)
locator = Nominatim(user_agent="myGeocoder")
latitude= str(lat)
longitude=str(lng)
coordinates = ""+latitude+", "+longitude
location = locator.reverse(coordinates)
addr=location.address
data = {"Latitude": latitude, "Longitude": longitude, "Location":addr}
db.push(data)
print(addr)
string = "You are currently at "+addr
c = 'espeak -ven+m4 -k5 -s140 --punct="?" "%s" 2>>/dev/null' % string
execute_unix(c)
```

# ACKNOWLEDGEMENT

Success of project like this is which involves high technical expertise, patience beyond limits to sit and keep watching black and white terminal screen popping messages after messages, and impeccable support of guides, is possible with every team member working together. So big congratulations to my team-mates.
We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We would like to show our greatest appreciation to **Mrs. Dipti Jhadav** for her tremendous support and help. Without their encouragement and support this project would have been dangling in its midway. She made sure that we were on time always. We would also to like to mention our gratitude to all the panel members and project mentors for their valuable inputs during the mock presentations. Thank you to our HOD Ms. Sana Shaikh for her constant support and motivation. Thank you all for helping us achieve this.

1.Jordan Dsouza, 19

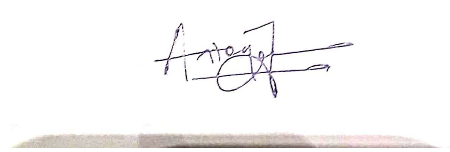


--------------------------------------------------

2. Salil Fernandes, 22

--------------------------------------------------

3. Anthony Kattikaren, 36



--------------------------------------------------

Date: 28/5/2021

# PUBLICATION

| Sr. No. | Specifications | Numerical total |
|---|---|---|
| 1. | **Number of pages** | **8** |
| 2. | **Number of figures** | **11** |
| 3. | **Number of tables** | **1** |
| 4. | **Number of equations** | **1** |
| 5. | **Number of references** | **10** |
| 6. | **Number of words** | **4341** |

| | |
|---|---|
| **Paper Title** | **NAYAN-DRISHTI: A Revolutionary Navigation/Visual Aid for the Visually Impaired.** |
| **Name of the Author:** | Salil Fernandes, Jordan D'souza, Anthony Kattikaren, Mrs. Dipti Jadhav. |
| **Publication Conference:** | ICT4SD 2021, Goa. |
| **Status:** | Paper accepted pending the filling of permission to publish and other nomination forms. |
| **Reviews by the panel members of the conference:** | 1. The work is encouraging.<br>2. Abstract well written.<br>3. The originality and scientific quality of this paper is acceptable.<br>4. Statistical analysis in this paper is suitable and infographs are satisfactory.<br>5. Recommended for inclusion. |

# NAYAN-DRISHTI: A Revolutionary Navigation/Visual Aid for the Visually Impaired.

Salil Fernandes [1], Jordan D'souza [1], Anthony Kattikaren [1], Mrs. Dipti Jadhav [1]

Don Bosco Institute of Technology, Mumbai-70.

salilfernandes72@gmail.com, jordandsouza99@hotmail.com, anthonyj3199@gmail.com, dipti.dbit@dbclmumbai.org

**Abstract:** The project/proposed product hinges on three domains of computational technology i.e., Machine learning, Convolutional Neural Networks and Internet of things. The aim of the project is to invent a product that is helpful to the disabled section of society as ideally as possible try to as well as to acquaint ourselves with the much talked about and ever growing domains of computer technology. The main functions that our proposed product will offer are detection of the obstructing object and alerting via a speaker (along with classification and distance of the object from the user) and a navigation system (which obtains live data of the current location of the user with the help of the UBLOX GPS module). The proposed product is designed in such a way so as to provide an all in one multitasking and hassle-free solution to our user and to ease the burden that come along with the disability of blindness. The proposed product is touted as a boon to our users since it not only will help them in identifying the obstructions ahead them but will also help them to navigate from their current location to their destination with freedom and no fear.

**Keywords:** Convolutional Neural Networks (CNN), Rectified Linear Unit (ReLU), Batch Normalization (BN).

## 1. Introduction:

We sometimes wonder how the disabled section our society manage to perform tasks that look impossible pertaining to their disability. We have to ponder as to how efficiently can they perform their day-to-day tasks and how well can they communicate with normal people. Keeping in mind the difficulties they face, their ability to perform such tasks and how to ease the fear related to it, technology has advanced in many fields.
Blindness can basically be classified into three types:
Complete Blindness, Night Blindness and Colour Blindness.

The proposed product will function accordingly as explained below using the following computing domains:
1. Convolutional Neural Networks: The product will take an input from the surrounding environment of the user and convolve it for a better prediction of the obstructing object which will be doneusing machine learning.
2. Machine Learning: The product with the output image can will predict the object that is in hinderance to the user's path with the help of pre-trained datasets.
3. Global Positioning System: The user will be able to communicate his/her destination to the software and the software will guide the user to the destination. The UBLOX GPS Module plays an important role in communicating the satellite in space to obtain the real time location of the user.
4. IOT (Internet of Things): Our project consists of raspberry pi, GPS module, camera module, etc. that will be effectively used in processing data, capturing images, displaying location of the person, etc.

## 2. Literature Survey:

### A. Survey of Existing Systems:

"Electronic walking stick for the blind" published in the year 2018, the use of optical sensors hasbeen highlighted. This is a modern concept of a walkingstick which is completely digital. These sensors essentially convert light reflected from any surface and convert them into electric signal and acts as response to the stimuli and informs the blind person via a speaker onthe handle about the obstacle. The object aimed to give voice assistance to the user and was able to deliver in various cases.[1]
"Portable Blind aid Device" published in the year 2019, it highlights the use of a mobile-based project in which the user can switch his wireless device into blind assistance mode with thehelp of a button. With the help of the camera, GPS and a cloud-based architecture it will be able to give the real-time location of the user and also make himaware of his surroundings. Also plan to include an advanced image recognition algorithm to recognize the faces of stranger and to store them.[2]
"Intelligent glasses for the blind" published in the year 2016, the device is smart glass which uses a camera, an ultrasonic sensor

and an electronic touchpad to assist the blind person. Using a mobile device one can activate the glass and the camera will capture and convert the 3D image into a spatial matrix and give appropriate outputs. The touchpad giveslight electric shocks to make the user aware of the obstacles. Future scope was to add a walking cane with a button to give the user audio output of the real-time surroundings.[3]

"Object Identification for Visually Impaired." published in the year 2016, a simple image recognition system that makes use of camera to recognize images, an ultrasonic sensor to detect obstacles is explained. The camera captures the images and if the image can be recognized by the images in the dataset then the device gives and audio output using a speaker that will be attached to the users clothing. Future upgrades include to introduce face recognition and to use a wireless camera.[4]

"Real-Time Objects Recognition Approach for Assisting Blind People." Published in the year 2017, an object recognition project is depicted that with the help of SURF (Speeded Up Robust Features) and light machine learning is able to give accurate information about the objects captured. Using GPS and image recognition it gives 90% accurate results. It makes use of a database and uses machine learning algorithm it is able to identify the objects.[5]

"EarTouch: Facilitating Smartphone Use for Visually Impaired People in Mobile and Public Scenarios." published in the year 2019, a Blind aid device which uses ear gestures like long swipe, tap, slide, etc. for operating the various function on the smartphone. Making use of one-hand gestures and Ear- Touch feature it enables the user to use the device with ease. And it gives out audio output through the inbuilt speakers. The device was able to read the gestures accurately and activate the respective functions when theuser wanted it.[6]

"An Ultrasonic Navigation System for Blind People" published in the year 2017, introduces with us another device for object detection which not only uses an ultrasonic sensor but also includes an accelerometer, footswitch, microcontroller and vibration pad. User is made aware of the obstacle and closeness ofit by different degree of vibrations and also through an audio output.[7]

"Mobile Blind Navigation System UsingRFID". published in the year 2015, the paper introducesa mobile based communication device which makes useof the WIFI, GPS to determine the user's location. And also makes use of RFID tags in the surroundings to givemore accurate location to the user and guide them by giving appropriate instructions. The user makes use of asmart walking cane and RFID handled reader.[8]

"Voice assisted system for the blind" published in the year 2015, the paper introduces us witha simple object/obstacle detector which makes use of anultrasonic senor, microcontroller, mp3 module and SD card. The sensor sends the distance measured and the microcontroller depending the proximity limit it is programmed sends an output to the mp3 module. It makesuse of depth sensing for avoidance of pot holes etc. for the user to walk with smoothness and comfort with normalcy.[9]

"Virtual-Blind-Road Following Based Wearable Navigation Device for Blind People." published in the year 2019, the paper depicts a navigation device specifically for the indoor environment which makes use of SLAM (simultaneous localization and mapping). The device tries to give the person the best path to an indoor destination by keeping track of the positions of previously encountered obstacles. It makes use of PoI graph to store new points in the route to the destination and using A* algorithm onthe PoI graph it finds the shortest and optimum path to the destination. Obstacle distance is measured by an Ultrasonic range finder and provides audio output.[10]

**B. Limitations of the systems on research gap:**

All above proposed papers that we have researched haveshown 90% success rate in the field and in all test cases. Although with certain setbacks too. All devices are capable of giving the required results but the reactions time in real-world is less which the devices are not able to keep up with. For real-time outputs more complex andadvanced system and cloud-based architecture is required which is difficult on a limited budget. In image recognition few projects weren't able to identify objects in the dark or night. The components used in few projects are fragile and prone to damage. Hence when used by a user could get damaged. And using strong equipment makes the project expensive.

**C. Problem Statement and Objective:**

Our product introduces a three-way combined solution system that will serve as a navigation aid for the blind:

1. To ease the difficulty and uncertainty faced by the blind or any visually impaired person when they have to walk from one place to another. A simple task like walking is scary for blind people owing to the fact that they just do not know what obstacles, whether dangerous or not,may be in front of them.
2. Through our project we hope to create a product that makes navigation a safer and simpler task for blind people. Object detection and recognition is the core concept which our projectrevolves around.
3. Our project domain is machine learning and wehave decided to make use of the concept of a Convolutional Neural Network built using TensorFlow to process image data and performthe object recognition task after being trained with datasets.

**D. Scope:**
Our project aims to implement three mainfeatures:

1) Object recognition using a camera attached to the pi module and using our machine learning domain and implementing Convolutional Neural Network (CNN) break down the images into 2D frames andtrain the algorithm to identify various objects in the surrounding.
2) Object Detection using an Ultrasonic sensorwhich can detect obstacles at a distance of 4m and sends an alert to the user if the proximity of the object is within a certain given range.
3) Giving the real-time location of the user using a GPS module that is programmed tocommunicate with the satellite and on a press of a button give the user his/her location.

## 3. Models and Setups:

For object detection we have used a convolutional neuralnetwork to perform object recognition on a given image. With respect to sensing the distance of a user from a given obstacle, we have utilized the concept of ultrasonicdistance sensing.

**A. Object Detection and Recognition:**

The primary goal of our project is to perform image analysis to lead to object recognition which would enable a blind person to get a sense of his/her environment. By understanding what objects surround them, blind people can make better decisions about where they are and in what direction they should move. We have used the convolutional neural network (CNN) model to implement object detection and recognition. Convolutional Neural networks are a family of deep learning algorithms that take parts of images as inputs, apply certain weights and biases and differentiate one part of the image from another. Most images now a daysare RGB images meaning that there are three channels tothe image. CNNs make it easier to understanding the image by capturing the spatial and temporal dependencies by using filters are different levels or depths. It offers faster analysis by reducing the number of parameters and reusing weights.

Under CNNs there are several architectures that come into the picture like the Faster R-CNN model, Mask R- CNN Inception model and the SSD ResNet models to mention but a few. In our project we are using a Raspberry Pi 3 Model B+ to program and execute our programs. Due to limited memory of 1GB RAM and lower processing power available we have to make use of a neural network which is computationally less intensive but without making a compromise in accuracy.The architecture which suited this purpose is the SSD- MobileNet v2 architecture.

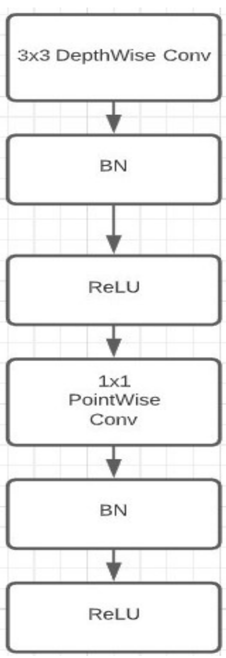



**Fig 1.** Single convolution layer in MobileNet.

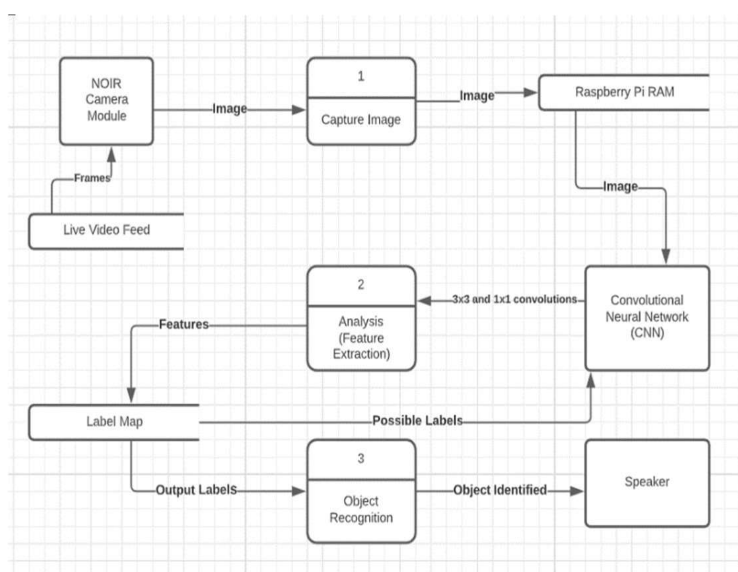



**Fig 2.** Data Flow diagram for Object Recognition.

The architecture uses the concept of depth-wise convolutions. Depth-wise Convolution is a type ofconvolution wherein a single convolutional filter isapplied to each input channel. In the regular
2D convolution performed over multiple input channels,the filter is as deep as the input and lets us freely mix channels to generate each element in the output. In contrast, depth-wise convolutions keep each channel separated from each other. In general, the steps used to perform a depth-wise convolution are:

1. Break up the input and filter into channels.
2. Convolve each input with the appropriate filter and combine the convolved outputs.

A depth-wise convolution consists of a depth separable convolution and a pointwise convolution. Spatial separable convolution works mainly with the spatial dimensions of an image and kernel which are the widthand height. (The third dimension is called depth which isthe number of channels of each image and is not taken into account by spatial separable convolutions). A spatial separable convolution breaks a kernel into twosmaller kernels. The most common case would be to divide a 3x3 kernel into a 3x1 and 1x3 kernel. In place ofperforming a single convolution by 9 multiplications, weperform two convolutions with 3 multiplications each to get the same result. With fewer multiplications, computational complexity goes down, and the network isable to run more efficiently. Unlike spatial separable convolutions, depth-wise separable convolutions use kernels that are not divided into two smaller kernels. The depth-wise separable convolution is so named because in addition to spatial dimensions, it deals with the depth dimension — the number of channels — as well. An input image can have3 channels: RGB. After a few convolutions, an image may have multiple channels. An image with 64 channelswould have 64 different versions of the same image. Analogous to spatial separable convolution, a depth-wiseseparable convolution divides the kernel into two kernelsthat do two convolutions: a depth-wise convolution and apointwise convolution.

**B. Dataset Used:**

The COCO Dataset has been used to train our MobileNetCNN for object detection and recognition. COCO in short is a dataset used for training object detection, segmentation and captioning networks. COCO stands forCommon Objects in Context which means that images are taken from everyday objects to prepare the dataset. It has approximately 330,000 images with more that 200,000 of them labelled.

**C. Activation Function Used:**

For each convolutional layer of the architecture, the ReLU activation function has been used along with BatchNormalization, the rectified linear activation function also called ReLU is the most commonly used activationfunctions in artificial neural networks. It returns a 0 as output for any negative input. If it receives a positive value x as input it returns x as the output. So in generalthis function is given as an equation $f(x) = \max(0, x)$.
Batch Normalization is a deep learning technique that normalizes the output of each sublayer. It allows for fastprocessing and deep network training by reducing internal covariate shift.

**D. Obstacle Distance Sensing:**

For the purpose of sensing the presence of an obstacleand sensing the distance between the obstacle and person, an ultrasonic sensor is being used. The input for this would in fact be an object placed in front of the sensor. The output is a distance reading if theobject is in the sensor's accurate range. By means of a wireless Bluetooth speaker, an audio warning will be sounded stating that the obstacle is nearby.

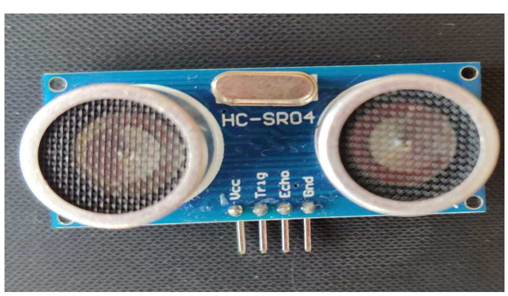



**Fig 3.** Ultrasonic Sensor.

The ultrasonic sensor has 4 pins:

a. VCC which is used to take the power supply.
b. TRIG which will on receiving the input signalfrom the Raspberry Pi will send out the ultrasonic pulse.

c. ECHO which will gather the echo pulse from theobstacle and send the signal that the obstacle hasbeen detected back to the Raspberry Pi.
d. GND which provides general grounding to the sensor.

For exchanging signals and for the program to receive readings from the sensor the GPIO pins of the RaspberryPi were connected with the pins of the sensor. We have used the HC-SR04 Ultrasonic Sensor to perform this functionality. The accurate range of this sensor is 2cm to 400cm which means that it can correctly measure distance between a user and obstacle when the obstacleis no less than 2cm and no more than 00cm away fromthe sensor. Extra caution had to be taken when implementing the circuit between the sensor and GPIO pins. The ultrasonic sensor operates at 5V and this voltage was supplied by connecting GPIO pin 1 to the VCC pin of the sensor. Since all the GPIO pins (except for pin 1) operate at 3.3V and the signals sent by the sensor at 5V, care has tobe taken to lower the voltage. To do this we made use ofa voltage divider circuit implemented on a breadboard. We used 560 ohm and 1000 ohm resistors to lower the 5V echo signal to a 3.3V signal.

**E. Location Sensing:**

In order to allow the blind person to know where exactly he/she is when navigating outdoors, a GPS sensor has been used. The Ublox Neo 6M GPS Module allows for easy interfacing with the Raspberry Pi.

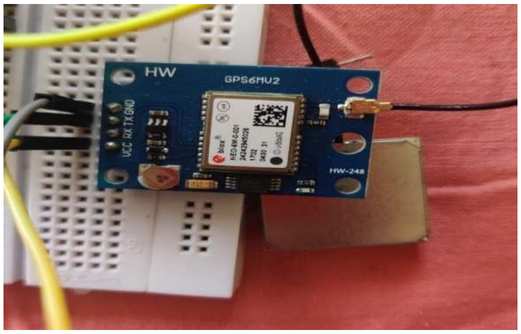



**Fig 4.** GPS Module.

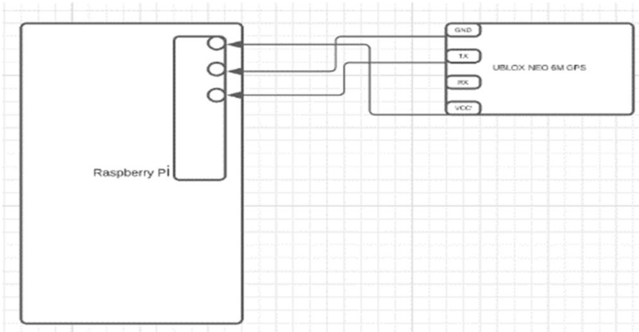



**Fig 5.** Diagrammatic location sensing setup.

The antenna once attached to the sensor allows for GPS data to be received. The antenna has to be left exposed to the sky in order for a satellite to lock onto it and start transmitting GPS data. There are 3 pins that are using to take the received data from the module:

a. GND: this pin is connected to pin 3 of the Pi for providing electrical grounding.
b. VCC: this pin is connected to pin 1 of the Pi which provides a 5V power supply.
c. TX: the transceiver pin of the module is connected to the RX pin of the Pi which is pin 5. RX is the receiver pin of the Pi which will transfer received GPS data to the serial port to be read by the Python program.

In python the GeoPy, PyNMEA2 and GeoPandas libraries have been used. The GPS Module receives location data in the form of NMEA sentences. In the Python code, the GPGLL (Geographic position, latitude / longitude) sentence has been extracted and used to obtain latitude and longitude coordinates of the module's current location. Reverse GeoCoding i.e. coordinates to address has been performed by the program to obtain the address. As usual this location address is sounded to the blind person by means of the Bluetooth speaker.

**F. Wireless Sound:**

The wireless Bluetooth chip that comes with the Raspberry Pi has been used to provide audio messages. A Bluetooth speaker is connected to the Pi with the help of this chip. For the purpose of making use of Bluetooth several packages had to be installed first namely: Bluez, Alsa, Bluetooth Manager and PulseAUDIO.
Since our device aims to help blind people navigate audio messages have to be delivered. This is the primary reason for using a wireless speaker. It is possible for blind people to get visual cues about their environment, hence audio hints have to be delivered to inform them.
In the object detection and recognition feature, once an object is successfully detected, the speaker delivers a message for eg. "Cup Detected".
In the obstacle sensing feature, whenever the ultrasonic sensor senses the obstacles distance to be less than 10 cm a message is

sounded, eg. “Careful Obstacle is 10 cm away”
In the location sensing feature, the speaker sounds the address of the current location obtained by reverse geocoding the coordinates.

## 4. Mathematical Model:

### A. Object Detection and Recognition:

In each convolutional layer, the first layer is a pointwiseconvolution followed by ReLU. The second layer is a depth-wise convolution and the third is another pointwise convolution but without any non-linearity.

| Input | Operator | Output |
|---|---|---|
| H x W x K | 1x1 conv, ReLU | H x W x TK |
| H x W x TK | 3x3 conv S=s,ReLU | $\frac{H}{s} \times \frac{W}{s} \times TK$ |
| $\frac{H}{s} \times \frac{W}{s} \times TK$ | 1x1 conv | $\frac{H}{s} \times \frac{W}{s} \times K'$ |

T is called the expansion factor, s is called the stride, Hand W are the height and width of the image respectively. For our MobileNet, the depth-wise convolution applies asingle filter to each input channel. The pointwise convolution then applies a 1X1 convolution to produce alinear combination of the outputs the depth-wise convolution. All layers in MobileNet consist of a 3X3 depth-wise separable convolution except for the first layer which has a full convolution. A final average pooling reduces the spatial resolution to 1 before the fully connected layer. Counting depth-wise and pointwise convolutions as separate layers, MobileNet has 28 layers.

### B. Ultrasonic Distance Sensing:

We have assumed the speed of sound to be approximately 340 metres per second or in other words 34,000 cm per second. To calculate distance between user and obstacle, only the distance one way has to be calculated since the time interval being measured is from the instant the ultrasonic pulse is sent to the instant theecho is received.

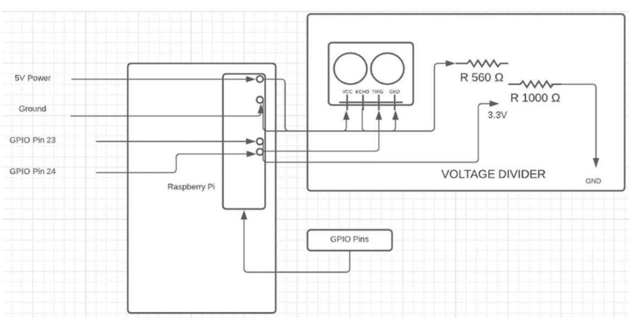



**Fig 6.** Diagrammatic ultrasonic sensing setup.

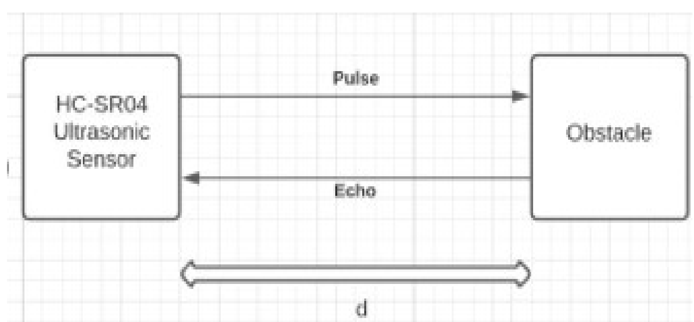



**Fig 7.** Distance Calculation.

Let t be this time interval. Therefore, the formula for distance is,

$$d = (t * 34000) / 2 \quad (1)$$

## 5. Results:

### A. Object Detection and Recognition:

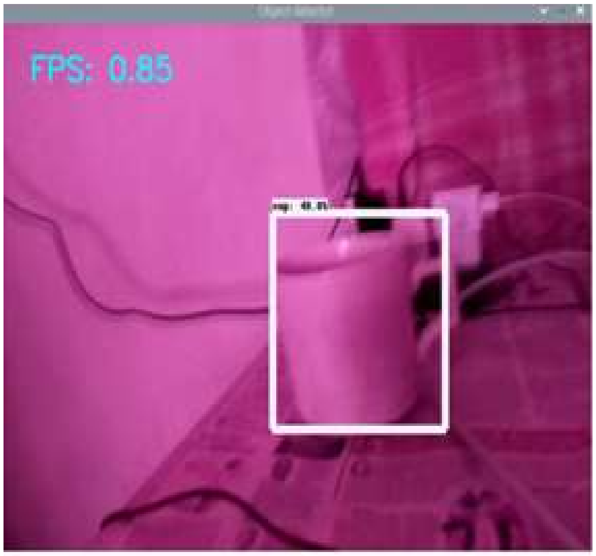
The frames per second (FPS) can be improved by manually changing the resolution of the detector. For objects that the model was trainedfor, the detections and recognitions were instantand highly accurate which is a very good result considering that the program was executed on the Raspberry Pi with only 1GB of memory.

### B. Obstacle Distance Sensing:

In the program we preset the range for which analert message is to be given. Whenever the distance sensed was less than 10 cm an alert is generated. The range can be manually changed in the program.

**C. Live Location Sensing:**

This program enables the GPS module to acquire the signal from the orbital satellite and latch onto it. After the latching of the signal, it detects the location of the user and informs the user about the current location.





**Figs 8, 9.** Object Recognition output.

## 6. Future Scope:

To get more accurate obstacle/distance sensing results, a sonar sensor can replace the ultrasonic sensor as it has a larger sensing range. Whenever the Raspberry Pi foundation manages to integrate more RAM onto the computer board, the neural network can be trained a lot faster and give speedy and accurate predictions. If our prototype is manufactured and improved on by a professional company, the resulting product would change the lives of visually impaired people forever. The functionalities of the proposed system can also be increased from the proposed three to as much as we can in order to benefit the user even better. One such functionality can be a guided software to use the smartphone according to the user's comfort.

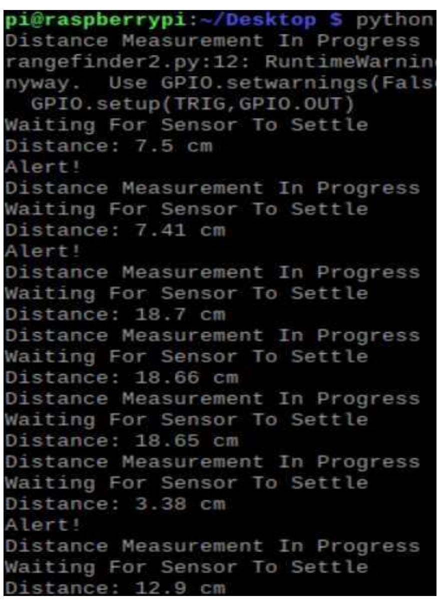



**Fig 10.** Ultrasonic Distance output depicting an alert for distances less than or equal to 10 cm.


**Fig 11.** Live Location Detection.

## 7. Conclusions:

Through the results depicted in the above section, it is clear that our object detection was overall a success in terms of efficiency and accuracy. The system could differentiate between basic objects easily with an efficiency rate ranging between 75-98% depending on the rate of frames per second. On testing the object distance measurement, we found out that the system was again efficient in displaying/sounding an alert as the object was nearing the user. An alert message will be sounded when the user is around 3-5m from the obstructing object. But here for test purposes we chose 10 cm as our minimal limit to sound an alert message. Thus,object detection and object recognition were successfully implemented.

## 8. References and Citations:


[1] Chi-Sheng, Hsieh. "Electronic walking stick for theblind." U.S. Patent No. 5,097,856, 24 Mar. 2010.

[2] Evanitsky, Eugene. "Portable blind aid device." U.S.Patent No. 8,606,316, 10 Dec. 2013.

[3] Cervantes, Humberto Orozco. "Intelligent glasses forthe visually impaired." U.S. Patent No. 9,488,833. 8 Nov. 2016.

[4] Jothimani, A., Shirly Edward, and G. K. Divyashree."Object Identification for Visually Impaired." Indian Journal of Science and Technology 9.S1, 2016.

[5] Zraqou, Jamal S., Wissam M. Alkhadour, and Mohammad Z. Siam. "Real-Time Objects RecognitionApproach for Assisting Blind People.", 2017.

[6] Ruolin Wang, Chun Yu, Xing-Dong Yang, Weijie He, Yuanchun Shi. 2019. EarTouch: Facilitating Smartphone Use for Visually Impaired People in Mobileand Public Scenarios. In CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019),May 4–9, 2019, Glasgow, Scotland Uk. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3290605.3300254

[7] Conference on Human Factors in Computing Systems (CHI '11). ACM, New York, NY, USA, 3247– 3256. https://doi.org/10.1145/1978942.1979424 Mounir Bousbia-Salah, Abdelghani Redjati, Mohamed Fezari, Maamar Bettayeb. "An Ultrasonic Navigation System For Blind People".2007. https://www.researchgate.net/publication/251851635

[8] Rachid Sammouda, Ahmad AlRjoub. "MobileBlind Navigation System Using RFID".2015. https://ieeeexplore.ieee.org/document/7353325

[9] Ananth Noorithaya, M. Kishore Kumar, A. Sridevi."Voice assisted system for the blind", 2015. https://doi.org/10.1109/CIMCA.2014.7057785

[10] Jinqiang Bai, Shiguo Lian, Zhaoxiang Liu, Kai Wang, Dijun Liu "Virtual-Blind-Road Following Based Wearable Navigation Device for Blind People", 2018. https://ieeexplore.ieee.org/document/8307352

[11] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[12] Abadi, Martn, et al. "Tensorflow: Large-scale machine learning on heterogeneous distributed systems."arXiv preprint arXiv:1603.04467 (2016).

[13] Ren, Shaoqing, et al. "Object detection networks onconvolutional feature maps." IEEE transactions on pattern analysis and machine intelligence 39.7 (2017): 1476-1481.

[14] Shiri Azenkot, Sanjana Prasain, Alan Borning, Emily Fortuna, Richard E. Ladner, and Jacob O. Wobbrock. 2011. Enhancing Independence and Safetyfor Blind and Deaf-blind Public Transit Riders. In Proceedings of the SIGCHI.

[15] Ren, Shaoqing, et al. "Faster R-CNN: Towards real-time object detection with region proposal networks." Advances in neural information processing systems, 2015.