EyeGen

A PROJECT REPORT

Submitted by

Ankit Kumar – 20BBS0206
Mohammad Umair Khan – 20BBS0211
Md Jamshed Alam – 20BBS0214

Course Code: CBS3006

Course Title: Machine Learning

Under the guidance of

Prof. Abdul Gaffar H

Associate Professor, SCOPE,

VIT, Vellore.

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

April 2023

ACKNOWLEDGMENT

We would like to thank Prof. Abdul Gaffar H and Vellore Institute of Technology, Vellore for providing the necessary environment for the successful execution and completion of this project. We appreciate sir's advice and feedback during our execution to help us improve and make our project more holistic and complete.

We feel it’s a pleasure to have indebted him for his valuable support, advice, and encouragement and we thank him for his superb and constant guidance towards this project. We are deeply grateful to all the staff members of Vellore Institute of Technology for supporting us in all aspects. This was a valuable learning opportunity for all of us.

Table of Contents

ABSTRACT

In today's ever-fast-moving world, disabled humans have been suffering more than ever. Many systems exist to help people with such disabilities navigate their surroundings, but they have a bunch of problems and there is a lack of a proper solution that covers all such pain points. Hence, to help users with disabilities such as blindness, a novel object detection methodology has been employed via automated calls to the Google Vision API. Utilizing the results from above, scene detection has been employed to locate the person within their surroundings. (E.g., Bedroom, Bathroom, Garage, etc.)

Both the models shall be integrated to build an efficient and useful system and it has been extensively tested for multiple different scenarios. Natural Language Processing has been extensively utilized to convert object data to easier-to-understand language data, which will help assist people better. The entire above-mentioned process has been enabled via a seamless application flow that is developed in a multi-stack approach with python being the primary choice for handling artificial intelligence- related tasks. The overall system has been integrated into a singular, easy-to- use and intuitive application to help users.

1. INTRODUCTION

1.1.PURPOSE

The authors of the current project are developing an effective object detection system to find items in a certain frame. Additionally, to estimate the user's environment and provide distance information for the discovered items. This was created from the viewpoint of users who could have different infirmities that might make it difficult for them to comprehend and engage with their surroundings.

1.2.SCOPE

Scope of the project includes but is not limited to:

- A novel object detection methodology shall be employed via automated calls to the Google Vision API.

- Utilizing the results from above, scene detection shall be employed to locate the person within their surroundings. (E.g., Bedroom, Bathroom, Garage, etc.)

- Both the models shall be integrated to build an efficient and useful system and will be extensively tested for multiple different scenarios.

- Natural Language Processing shall be extensively utilized to convert object data to easier to understand language data, which will help assist people better.

- The entire above-mentioned process shall be enabled via a seamless application flow that is developed in a multi stack approach with python being the primary choice for handling artificial intelligence related tasks.

- The overall system shall be integrated into a singular, easy to use and intuitive application to help users.

1.3.DEFINITIONS, ACRONYMS, ABBREVIATIONS

This section of the introduction includes details regarding the different terms used in the sections below for reference:

- API – Application Programming Interface
- DBA – DataBase Administrator
- DL – Deep Learning
- FTP – File Transfer Protocol
- GPS – Global Positioning System
- JSON – JavaScript Object Notation
- IoT – Internet of Things
- ML – Machine Learning
- NLP – Natural Language Processing
- SOC – System on a Chip

1.4.OVERVIEW

This paper was created with input from a variety of people, including other developers, testers, and even EyeGen system users. The paper provides a summary of the product that will be provided. To comprehend the conditions necessary to satisfy this product responsibility, one may read through.

The document is organised such that the introduction comes first, then a detailed overview of the project, followed by requirements and the characteristics of the software system. The document is then put together using a set of appendices, which include cover non-functional needs and additional requirements.

To provide the greatest possible experience for impaired users who will ultimately utilise the system, the authors of this project have specified a number of requirements and have generally given equal weight to each and every area of the program.

2. LITERATURE SURVEY

- Dhou et al., circa 2022

  The single credit-card-sized single-board microprocessor, a six-axis accelerometer/gyro, ultrasonic sensors, a GPS sensor, cameras, and a digital motion processor make up the proposed mobile sensors unit. While the user is moving, the device is utilised to gather data on the nearby barriers and the cane user. To recognise the impediments that are identified and warn the user about their nature, an embedded machine learning algorithm is created and saved in the microcomputer memory.

  Additionally, the unit notifies the cane user and their guardian in the event of an emergency, such as a cane fall. Additionally, a mobile app is created for the guardian to use in order to follow the cane user via Google Maps using a mobile device in order to assure safety. A prototype was created and put to the test in order to validate the system.

- Vrindavanam et al., circa 2021

The study presents an alternative technique that can automatically create audio descriptions of images, which can considerably assist the visually handicapped. It is accompanied by a survey of the literature on image description technology. The necessity for the study resulted from the fact that the interaction points for people who are blind are becoming more limited in a world that is becoming more and more digital, and that accessing digital media via an image describer can help people who are blind.

The visuals that the visually challenged cannot see are analysed, appropriate descriptions are produced, and speech output is created. The paper uses the Inception Resnet-V2 model as the feature extractor and decoder (GRU- RNN) along with the Bahdanau attention model to generate the text description of the image, which is then converted to an audio using the Google Text-to-Speech converter as opposed to the standard methods like Computer Vision/CNN. The results were shown to be more accurate, and as a result, they may help the blind access digital media.

- Felix et al., circa 2018

  The goal of this project is to use artificial intelligence, machine learning, image, and text recognition, and assist persons who are blind or visually handicapped. An Android smartphone app that focuses on voice assistant, picture recognition, currency recognition, e- books, chat bots, and other features carries out the concept.

  The software can use voice commands to help you identify nearby items and also use text analysis to help you read text in a hard copy document. It will be a productive approach for blind individuals to use technology to engage with their surroundings and take use of its features.

- El-Taher et al., circa 2021

  The purpose of this article is to provide a thorough analysis of research that is directly related to or concerned with BVIP's assistance outdoor navigation. We divide the navigational space into many stages and duties. In our systematic review of the literature, we analyse articles, methodologies, datasets, and existing limitations according to task using this framework.

We also include a summary of navigation applications for BVIP that are both for-profit and non-profit. Our study adds to the corpus of knowledge by offering a thorough, organised examination of the work in the field, including the state of the art and recommendations for future research. It will assist researchers and other interested parties in the field in developing a knowledgeable perspective on research progress.

3. OVERVIEW OF THE WORK

3.1. OBJECTIVES

The goal is to design a unique object identification mechanism that will be implemented using automated queries to the Google Vision API. Using the aforementioned data, scene detection will be used to locate the individual within their surroundings. (For example, bedroom, bathroom, and garage).

Both models will be merged to create an effective and useful system that will be rigorously evaluated in a variety of circumstances. Natural Language Processing will be used widely to translate object data into language data that is simpler to comprehend, hence assisting individuals more effectively. The whole aforementioned procedure will be supported by a multi-stack application flow, with Python serving as the language of choice for artificial intelligence-related operations. To assist users, the whole system must be included into a single, user-friendly, and simple application.

3.2. REQUIREMENTS

3.2.1. USER REQUIREMENTS

1) Assist the visually impaired and other disabled members to understand the world around them in a better manner.
2) The events occurring around the world, which are in 360 view around a person, shall be described to them.
3) Improve the sensory functions of the user, by providing a virtual aid instead of vision.

3.2.2. FUNCTIONAL REQUIREMENTS

1) Object Detection - An automated object detection module is set up using python and the Google Vision API that uses different models such as Yolo, to automatically detect the objects surrounding a person, from which it will automatically infer the indoor location of a user within their house, etc.
2) Natural Language Processing - Multiple models are implemented concurrently to enable the conversion of objects to a more human readable language, helping improve the accessibility of the overall system for end users.
3) Location Detection - The system has the capability to analyse different aspects of its surroundings to find the exact current location of the user without any GPS/GSM, etc.
4) Integrated Application - A fully integrated and seamlessly joined application is built from scratch to ensure the user can avail all the functionalities in a singular, easy to use and intuitive flow.

3.2.3. NON-FUNCTIONAL REQUIREMENTS

1) Easy to use - The system should be easy to use, to ensure that even inexperienced users do not have any issues while understanding and interacting with the system.
2) Clean interface - The interface that is developed for the project shall be intuitive and clean, allowing it to be easy to navigate and the flow should be easy to understand.
3) Accessible to multiple disabilities - The interface must follow the guidelines of international agencies including WWC, to ensure that the entire system is accessible to people with different disabilities.
4) Minimalistic UI/UX - The User Interface / User Experience should be designed with a careful minimalistic flow, to prevent the confusion caused by a cluttered interface. There should be no information fatigue.
5) Smooth and functional experience - The user experience should be smooth without hiccups and the functions should be clear and easy to use, without a lot of intermediate steps that act as a buffer, providing its main functionality in an instant.

6) Scalable development - The development of the project shall incorporate a set of scalable guidelines, such that as the user base of the system grows, it should be able to handle multiple users utilising the system at the same time without significant performance issues occurring.
7) Secure user experience - The user's data and other private information should be secured by the latest web technologies, and should conform to the best data privacy standards, to ensure there are no data leaks.
8) Portability - The system should be designed with a portable usability approach in mind, allowing users to be helped even in environments where they may not have full access to their devices.
9) Flexibility - The system should be flexible enough to provide the users with a customised experience helping different people in the best way they need assistance.

3.2.4. OTHER REQUIREMENTS

Non-functional requirements are those that provide criteria rather than particular behaviours that may be used to evaluate how a system performs. Functional requirements, on the other hand, identify certain behaviours or functions. The three most common non-functional criteria are cost, scalability, and dependability. The words "constraints," "quality characteristics," and "quality of service needs" are also used to describe non-functional requirements.

1) Reliability: Any exceptions that arise while the software is running ought to be captured to stop the system from crashing.

2) Scalability: New modules and capabilities should be able to be added to the system easily, allowing system evolution.

3) Cost: Since software package is available for free, the cost should be minimal.

PERFORMANCE REQUIREMENTS

Performance criteria specify how effectively a software system performs various tasks under circumstances. Examples include the software's throughput, execution time, reaction time, and storage capacity. Supporting end-user tasks is frequently the foundation for the service levels that make up the performance criteria. Performance criteria are essential components in the design and testing of a software product, like the majority of quality qualities.

The system won't be able to recognize a specific object since there will be several detectable objects in the frame at any given time, making it impossible for the model to function in a big area. As a result, it will be difficult to determine the environment in which the user is present. We have increased the product's performance efficiency to between 85 and 90 percent. The app can occasionally be inaccurate since the findings are not always correct.

SAFETY REQUIREMENTS

The application has to be carefully designed as the users will be blindly trusting the system to provide and accurate representation of their surroundings. Even minor mistakes can put the users in serious danger and hence a strict level of safety needs to be considered in the entire process of the application.

SECURITY REQUIREMENTS

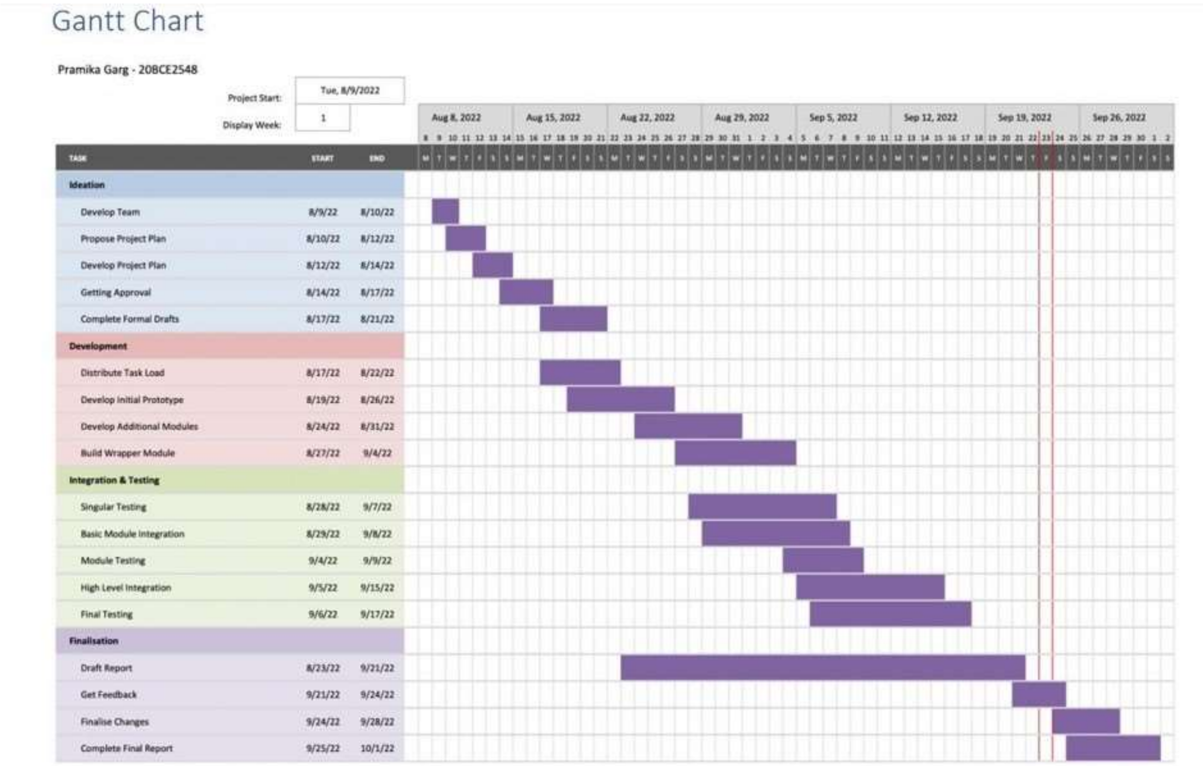
As we are dealing with extremely sensitive private data, even within the dwellings of the user, we have used utmost precaution in handling all the data points. Our application adheres to the best principles for safe software development as set out by the industry. Using the Firebase interface, we used Google Auth to authenticate our application.

SOFTWARE QUALITY REQUIREMENTS

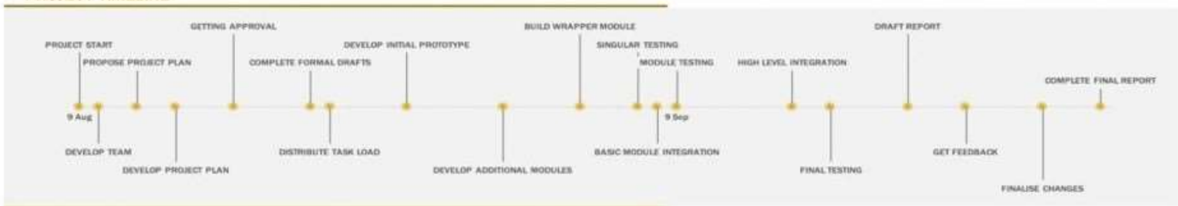
The outcome that will be displayed in the final step will have a significant propensity to be accurate because the dataset we are utilizing is varied. As a result, the product will be dependable and suitable for a wide range of items and settings. In order to reuse the previously gathered knowledge, the program will also use earlier difficulties as a reference. The structure of the app makes it incredibly simple to use; all that is required of the user is a login before they may take a picture of an item, after which the image is immediately analysed by our model to determine the user's presence and environment.

4. SYSTEM DESIGN

4.1. CHARTS



Project Timeline

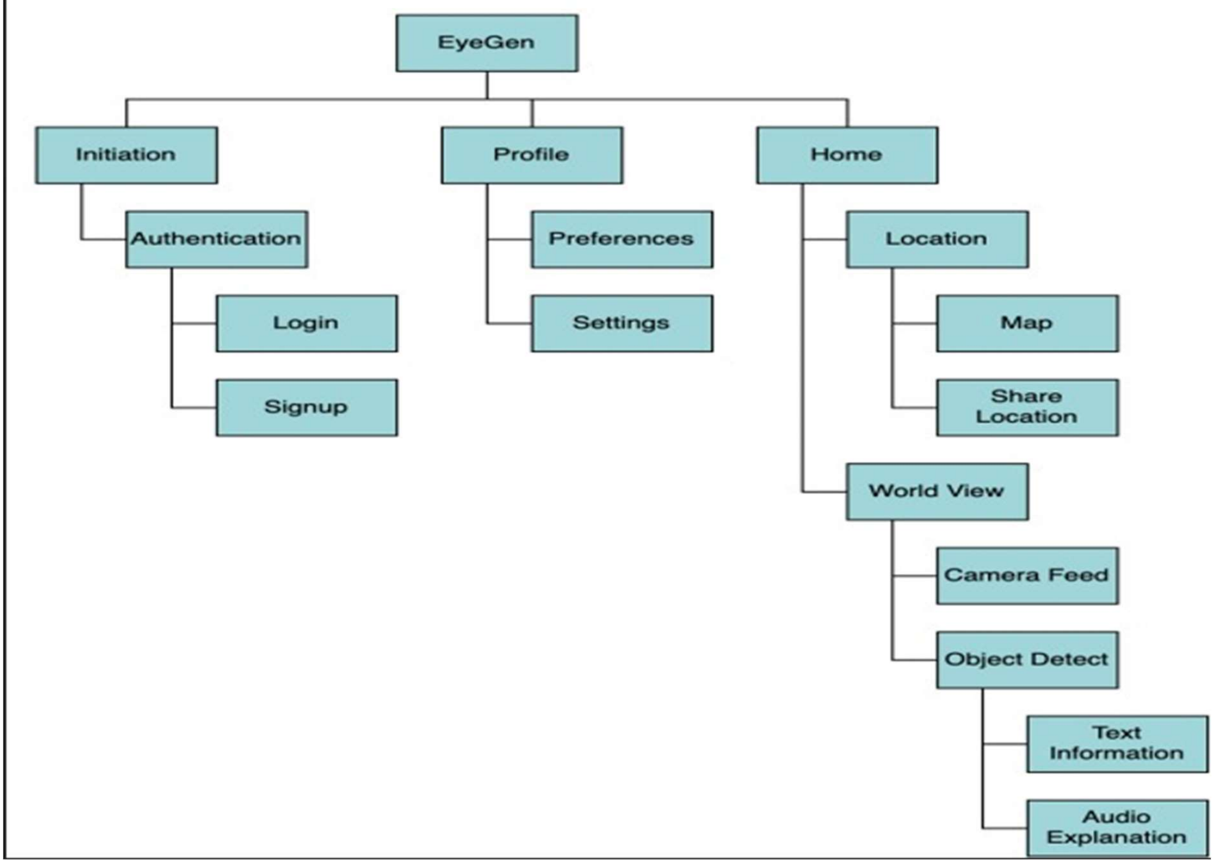


PROJECT DETAILS

DATE	MILESTONE	POSITION
9-Aug	Project Start	15
10-Aug	Develop Team	-10
12-Aug	Propose Project Plan	10
14-Aug	Develop Project Plan	-15
17-Aug	Getting Approval	20
21-Aug	Complete Formal Drafts	10
22-Aug	Distribute Task Load	-10
26-Aug	Develop Initial Prototype	15
31-Aug	Develop Additional Modules	-15
4-Sep	Build Wrapper Module	20
7-Sep	Singular Testing	15
8-Sep	Basic Module Integration	-10
9-Sep	Module Testing	10
15-Sep	High Level Integration	10
17-Sep	Final Testing	-15
21-Sep	Draft Report	10
24-Sep	Get Feedback	20
28-Sep	Finalise Changes	-10
1-Oct	Complete Final Report	-20

4.2.STRUCTURES

Activity Based



4.3.OVERVIEW OF MODULES / COMPONENTS

With the aid of voice instructions, the product's main goal is to assist those who cannot see the outside world. It serves as an alternative for the pricey items on the market that do the same task with a high degree of precision.

The following is a list of the EyeGen system's primary features: -

- Images should be continually recorded to know the current environment, and then quickly processed using the YOLOv3 model.

- Continuously call APIs for knowing user's position via the GPS sensor, which delivers them in JSON format.

- The item must continuously provide an environment change notice by user friendly voice command in addition to other important updates.

For implementing the functionality mentioned above, the following modules are defined and used within the product system:

1. Object Identification

a. Description and Priority - This module is the primary functionality of the application and hence is the maximum priority in ensuring proper operation.

b. Stimulus/Response Sequences - The user will start the camera-based detection and the raw feed will automatically be sent to the YOLOv3 model which detects all the objects within each frame and automatically logs them into JSON based text files.

c. Functional Requirements - An automated object detection module is set up using python and the Google Vision API that uses different models such as YOLOv3, to automatically detect the objects surrounding a person, from which it will automatically infer the indoor location of a user within their house, etc.

2. Speech to Text

a. Description and Priority

This module is also the primary functionality of the application and hence is the maximum priority in ensuring proper operation. b. Stimulus/Response Sequences

The system will read the JSON files from the GPS sensor and the YOLOv3 model and will then start the natural language processing model and the objects will be converted to text form and then into a voice output which will be broadcasted by the speaker. c. Functional Requirements

Multiple models are implemented concurrently to enable the conversion of objects to a more human readable language, helping improve the accessibility of the overall system for end users.

3. GPS Location Data

a. Description and Priority

This module is part of the optional setting of the EyeGen system and hence lies the lowest on the priority system as it is non-critical. b. Stimulus/Response Sequences

The system will read the JSON files from the GPS sensor via FTP connection and store them to the text local store.

c. Functional Requirements

The system has the capability to analyse different aspects of its surroundings to find the exact current location of the user without any GPS/GSM, etc.

4. Integrated System Application

a. Description and Priority

This module is also the primary user interface of the system, but is not the highest priority as the speaker is providing the main functionality and the application just acts as a authentication and start / stop endpoint. b. Stimulus/Response Sequences

The system will authorize the user after a successful login (or allow for registration and then login), and then allow the user to modify their settings and start / stop the advanced detection system and audio output of the user’s environment details. c. Functional Requirements

A fully integrated and seamlessly joined application is built from scratch to ensure the user can avail all the functionalities in a singular, easy to use and intuitive flow. It should support authentication, start / stop and settings modification.

5. IMPLEMENTATION

5.1. COMPONENT TEMPLATE DESCRIPTION

- GPS Module Integration – This module works on taking the input from the GPS sensor with a 5 second time period and sends it to the Google Coral.

  In this particular module, the GPS sensor is connected to power and enabled. It gets the accurate GPS coordinates at a high frequency and makes it available to the GPS Service. The service takes the coordinates at a 5 second interval and exposes it via a FTP endpoint serving JSON files. The files are received by the Google Coral Edge Computing Device and then are processed into the local store as well as sent to the database for long term storage. This module works on detecting the user's exact location and the direction of their movement via a connected compass instrument.

- Camera Sensor Module – The sensor captures the raw feed live and sends it to the backend of the EyeGen system where models process it.

  In this particular module, the Camera sensor is switched on, and the entire raw feed is processed and sent via a pipeline to the Google Coral Edge Computing Device. It is also used to calculate the distance between the objects detected by the YOLOv3 model and the user, to alert the user about their surroundings.

- Google Coral Edge Computing – This device is used to power all the edge computing needs forming the backend and a centralized node for all actions within EyeGen.

  The Google Coral Edge Computing Device acts like the central node and primary service endpoint, where all the other parts of the system get linked together. It takes the input of the raw image feed from the Camera Sensor Module, detects all the objects and then works on computing the distance to those objects in collaboration with the Camera Sensor Module. It also takes the GPS coordinates and works extracting the latest positional and directional information from the GPS Module. Then, it takes all these inputs and runs through a NLP model to convert the text to human hearable and understandable audio which is then passed along to the Speaker Module.

- Speaker Module – This module is utilized in sending out the final output to the user in the form of speech.

  In this particular module, the speakers are activated by the Google Coral Edge Computing Device, which starts a service which looks for incoming audio files from the Google Coral Edge Computing Device and pushes it out via the speaker system for the use of the user.

5.2.DESIGN DECISIONS & TRADE OFFS

In the process of preparing for this project, the team has utilized their best capabilities in enabling good system design and software engineering practices throughout the entire project. To enable this to become a reality, design considerations were made at the primal stage of the planning and development phase of the EyeGen system and we have implemented reusability in the following ways throughout EyeGen:

- API – In the process of preparing the designs of the API which will be hostel within the GPS Module as well as the primary API's within the Google Coral Edge Computing Device in the main backend frame of EyeGen, common API patterns have been used for multiple different sensors and there was a focus on enabling the same API base code to be utilized across different actors.
- Database – The process of designing the databases and other local stores within the EyeGen system was streamlined in such a fashion to ensure that common identification and proper normalization was used for the tables to prevent duplication and mismanagement of data. The DBA work was quite simplified by following a list of fixed patterns while developing the database.

- FOSS Tooling – Free and Open Source tooling was utilized extensively within the codebase, with multiple libraries and modules being taken from pre-existing well developed modular packages. This enabled the reduction of a lot of boilerplate as well as preventing the reinvention of the wheel within the development process.

The entire design process as well as the implementation was done with a focus on reusability to reduce the amount of unnecessary and less efficient versions of code being written, as well as to seamlessly incorporate safe and well audited open source libraries to ensure that all data is processed in a secure manner without any potential for leakage.

An added benefit was the ability to develop the EyeGen modules at a very rapid pace due to the availability of pre-existing modules for several aspects of the EyeGen software.

In the entire design process for EyeGen, there were a lot of assumptions made, characteristics defined and ideation done. The following paragraphs explain the final conclusion of the decisions taken and the conclusion shall explain the reasonings behind the choices.

The device is designed to run on a fundamental Linux-based operating system, making setup quick and easy. The hardware platform will be Google Coral, the operating system will be Ubuntu (Linux), there will be a camera for capturing images, and a GPS module for providing location data.

Since the data is stored in text format, a complex database system that may shorten the product's battery life is not required. Temperatures between -20 and
+50 Celsius are suitable for carrying and using the gadget physically.

The following criteria were taken into account for design and execution decisions when creating the EyeGen product:

- Textual local data storage serve as the foundation for the database used.
- YOLOv3 is the algorithm employed for object recognition and picture processing.
- Python is the primary language used for the backend, coupled with a number of libraries to implement various functionalities inside the application.
- To interface with the frontend, which also handles JSON-formatted FTP data migration from the GPS sensor, an API has been developed.
- Communication is conducted via the Voice Control protocol.
- To integrate a huge number of data points in real time for the finished voice output, a high- speed CPU is necessary.

- To enhance performance, a parallelized multi-threaded technique is used.
- End-to-end encryption is utilised to totally protect the privacy of all sensitive data.
- The user is only accountable for keeping the equipment safe from harm and sometimes clearing the camera's field of vision during operation.

Following testing, dependencies were chosen based on the following assumptions, which were established when creating the product:

- Different items are specified using their dimensions in order to set them apart from one another.
- The compass is used to determine the direction that the user is pointing in, and the GPS sensor is utilised to obtain accurate position coordinates.
- The project is based on the YOLOv3 detection algorithm, which has an accuracy of up to 90% when identifying objects within photos.
- Forward-moving objects are used in the training of the YOLOv3 model.

The decisions that are mentioned above were taken for a wide variety of reasons after a lot of careful consideration and observation and evaluation of different possible designs and implementation methodologies. The (micro)service-based architecture instead of a monolith was chosen due to the ability to easily add functionality, reduce load and heat on the central Google Coral system while delegating responsibilities efficiently to different components via reusable API architecture. The choice of the model was based on implementing a state of the art and highly performant model which was developed for usage in low power and high frequency runs use cases, which matched perfectly with the designed scenario of the EyeGen system.

5.3.PSEUDOCODE

GPS Sensor

```
from machine IMPORT Pin, UART, I2Cfrom ssd1306 IMPORT
SSD1306_I2C
```

```
#Import utime library to implement delayIMPORT utime, time

#Oled I2C connection

i2c=I2C(0,sda=Pin(0), scl=Pin(1), freq=400000)SET oled TO
SSD1306_I2C(128, 64, i2c)

#GPS Module UART Connection

SET gps_module TO UART(1, baudrate=9600, tx=Pin(4),
rx=Pin(5))

#OUTPUT gps module connection detailsOUTPUT(gps_module) #Used
to Store NMEA SentencesSET buff TO bytearray(255) SET TIMEOUT
TO False

#store the status of satellite is fixed or notSET FIX_STATUS
TO False



#Store GPS CoordinatesSET latitude TO ""
SET longitude TO "" SET satellites TO ""SET gpsTime TO ""
#function to get gps Coordinates
DEFINE FUNCTION getPositionData(gps_module):
 global FIX_STATUS, TIMEOUT, latitude, longitude,
satellites, gpsTime

#run WHILE loop to get gps data

#or terminate WHILE loop after 5 seconds timeout

SET timeout TO time.time() + 8 # 8 seconds from nowWHILE
True: gps_module.readline()
```

```
SET buff TO str(gps_module.readline())#parse $GPGGA term

#b'$GPGGA,094840.000,2941.8543,N,07232.5745,E,1,09,0.9,102.1
,M,0.0
,M,
,*6C\r\n' #OUTPUT(buff) SET
parts TO buff.split(',')

#if no gps displayed remove "and len(parts) EQUALS 15"
frombelow IF condition
IF (parts[0] EQUALS "b'$GPGGA" and len(parts) EQUALS 15):
if(parts[1] and parts[2] and parts[3] and parts[4] and
parts[5] and parts[6] and parts[7]):

OUTPUT(buff)

#OUTPUT("Message ID : " + parts[0])#OUTPUT("UTC time : " +
parts[1]) #OUTPUT("Latitude : " + parts[2]) #OUTPUT("N/S : "
+ parts[3]) #OUTPUT("Longitude : " + parts[4])#OUTPUT("E/W :
" + parts[5]) #OUTPUT("Position Fix: " + parts[6])#OUTPUT("n
sat: " + parts[7])

SET latitude TO convertToDigree(parts[2])# parts[3] contain
'N' or 'S'
IF (parts[3] EQUALS 'S'):

SET latitude TO -latitude

SET longitude TO convertToDigree(parts[4])# parts[5] contain
'E' or 'W'
IF (parts[5] EQUALS 'W'):
```

```
SET longitude TO -longitudeSET satellites TO parts[7] SET
gpsTime TO parts[1][0:2] + ":" + parts[1][2:4] +":" +
parts[1][4:6]

SET FIX_STATUS TO True
 break

IF (time.time() > timeout):SET TIMEOUT TO True break
 utime.sleep_ms(500)

#function to convert raw Latitude and Longitude#to actual
Latitude and Longitude
DEFINE FUNCTION convertToDigree(RawDegrees):

SET RawAsFloat TO float(RawDegrees)

SET firstdigits TO int(RawAsFloat/100) #degrees

SET nexttwodigits TO RawAsFloat - float(firstdigits*100)
#minutes

SET Converted TO float(firstdigits + nexttwodigits/60.0)

SET Converted TO '{0:.6f}'.format(Converted) # to 6
decimalplaces

RETURN str(Converted) WHILE True:
 getPositionData(gps_module)

#if gps data is found then OUTPUT it on lcdif(FIX_STATUS
EQUALS True):
OUTPUT("fix. ")
```

```
 oled.fill(0) oled.text("Lat: "+latitude,
0, 0) oled.text("Lng: "+longitude, 0, 10)
oled.text("No of Sat: "+satellites, 0, 20)
oled.text("Time:      "+gpsTime,      0,       30)oled.show()
OUTPUT(latitude) OUTPUT(longitude) OUTPUT(satellites)

OUTPUT(gpsTime)

SET FIX_STATUS TO False
 if(TIMEOUT EQUALS
True):

OUTPUT("Request Timeout: No GPS data is found.")

#  #updated on 5-May-2022 oled.fill(0)
 oled.text("No GPS data is found", 0,
0)oled.show()
#  SET TIMEOUT TO False
```

Object Detection

```
IMPORT numpy as np

IMPORT matplotlib.pyplot as mtlbIMPORT cv2 as cv from PIL
IMPORT Image as imIMPORT threading IMPORT time

OUTPUT(cv.version )

#reading the network

SET yolo TO cv.dnn.readNet(r"D:\Software
Engennering\yolov3.weights",r"C:\Users\Asus\Desktop\darknet-
master\cfg\yolov3.cfg")
```

```
SET classes TO []
 with
open(r"C:\Users\Asus\Desktop\darknet-
master\data\coco.names",'r') as f :

SET classes TO f.read().splitlines()

DEFINE FUNCTION processing(lis=[]) :OUTPUT("Hello")

DEFINE FUNCTION image_capture() :cap=cv.VideoCapture(0) SET
dic TO {}

SET start TO time.time()

WHILE True :
SET ret , img TO cap.read()#cv.imshow('Input',img) c=
cv.waitKey(1)


IF c==27:  break  height,width,_=img.shape #converting
to normal frame#grey scale
SET blob TO cv.dnn.blobFromImage(img , 1/255 , (320,320)
,(0,0,0) , swapRB=True, crop TO False )

#mtlb.imshow(blob) #OUTPUT
frame0

SET i TO blob[0].reshape(320,320,3)mtlb.imshow(i)
yolo.setInput(blob)

SET output_layes_name TO yolo.getUnconnectedOutLayersNames()
SET layeroutput TO yolo.forward(output_layes_name)

SET boxs TO []
```

```
SET confidences TO []SET class_ids TO []

FOR output IN layeroutput : FOR detection IN output : SET
score TO detection[5:]

SET class_id TO np.argmax(score) SET confidence TO
score[class_id]

IF confidence > 0.7 :
SET center_x TO int(detection[0]*width) SET center_y TO
int(detection[0]*height)SET w TO int(detection[0]*width) SET
h TO int(detection[0]*height)SET x TO int(center_x - w/2)
SET y TO int(center_y - h/2)

boxs.append([x,y,w,h])
confidences.append(float(confidence))
class_ids.append(class_id) len(boxs)

SET indexes TO cv.dnn.NMSBoxes(boxs,confidences,0.5,0.4) SET
font TO cv.FONT_HERSHEY_COMPLEX
SET colors TO np.random.uniform(0,255,size=(len(boxs),3)) SET
dici TO {}

FOR i IN indexes.flatten():SET x,y,w,h TO boxs[i] SET label
TO str(classes[class_ids[i]]) SET confi TO
str(round(confidences[i],2))SET color TO colors[i] IF
(label IN dic) and (label IN dici) :

SET dic[label] TO max(dic[label] , dici[label]) SET
dici[label] TO dici[label] + 1else : SET dici[label] TO
1
cv.rectangle(img,(x,y),(x+w,y+h),color,1)cv.putText(img
, label +"
```

```
"+confi,(x,y+20),font,2,(255,255,255),1)
cv.imshow('video',img) FOR a IN dici : IF a IN dic : SET
dic[a] TO max(dic[a],dici[a])else : SET dic[a] TO dici[a]

OUTPUT(dic)

mtlb.imshow(img)
SET end TO time.time()IF (end-start>5) :
processing(dic)time.sleep(5)

DEFINE FUNCTION main_task():global x # setting global
variable x as 0SET x TO 0 # creating a lock
SET lock TO threading.Lock() # creating threads
SET t1 TO threading.Thread(target=image_capture, args=())SET
#t2 TO threading.Thread(target=processing, args=())
# start threadst1.start() #t2.start()
# wait UNTIL threads finish their jobt1.join() #t2.join()
IF  name EQUALS " main ": FOR i IN range(10):main_task()
```

NLP Model

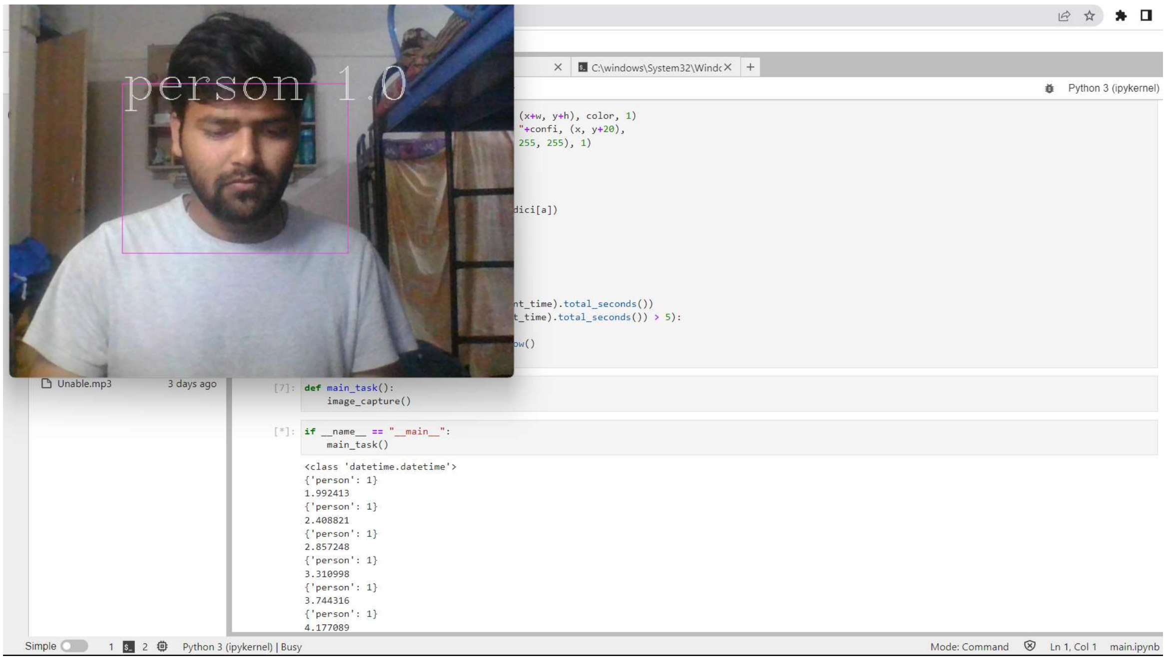
```
from gtts IMPORT gTTSIMPORT os
# The text that you want to convert to audioSET mytext TO 'You
are at Anna Auditorium' # Language IN which you want to
convert
SET language TO 'en'

SET myobj TO gTTS(text=mytext, lang=language,
slow=False)# welcome myobj.save("Anna_Audi.mp3") #
Playing the converted file os.system("Anna_Audi.mp3")
```

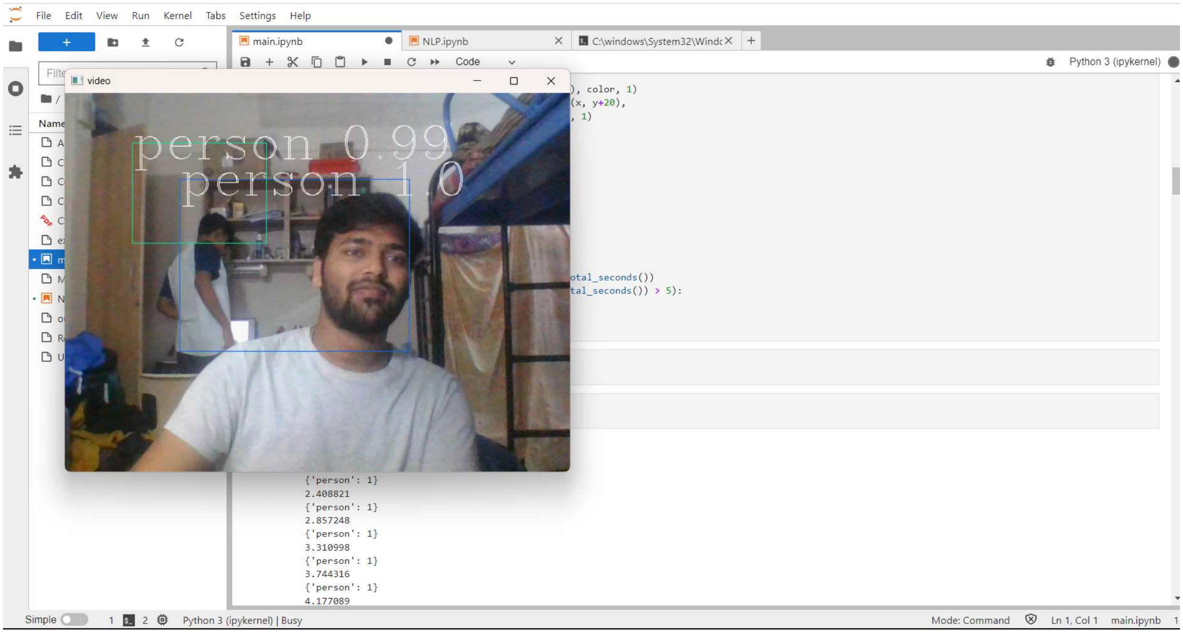
5.4. TEST CASES

Test Case Id	Test Case Objective	Prerequisite	Input Data	Expected Output	Actual Output	Status
1	To detect the location of the Person in an open Space	Device Must be Swiched On	Placed Outside Sjt	Must tell -- At SJT	You are at SJT	Pass
2	To detect the environment in a Buiding	Device Must be Swiched On	SJT -- 317 Lab	Must tell -- At SJT LAB	You are in Lab	Pass
3	To detect a classroom inside SJT	Device Must be Swiched On	SJT -- 303 Class	Must tell -- In SJT Class	You are in Class	Pass
4	To detect road	Device Must be Swiched On	Road in from of SJT	Must tell -- On Road	Be carefull , on Road	Pass
5	To detect Boys R block	Device Must be Swiched On	Front area of R block	Must tell -- At R Block	You are at R Block	Pass
6	To detect Girls G block	Device Must be Swiched On	Front area of G block	Must tell -- At G Block	You are at G Block	Pass
7	To detect a lab at TT	Device Must be Swiched On	TT G17	Must tell -- At TT lab	You are at Lab	Pass
8	To detect Crowd	Device Must be Swiched On	Area near lift	Must tell -- In Crowd	You in a crowd	Pass
9	To detect Anna Auditorium	Device Must be Swiched On	Front area of Anna Auditorium	Must tell -- At Anna Auditorium	You are at Anna Auditorium	Pass
10	To detect a Auditorium	Device Must be Swiched On	Inside Channa Reddy Auditorium	Must tell -- In Auditorium	You are in an Auditorium	Pass
11	To detect the crowd at Enzo	Device Must be Swiched On	Enzo Shop	Must tell -- Incrowd	You in a crowd	Pass

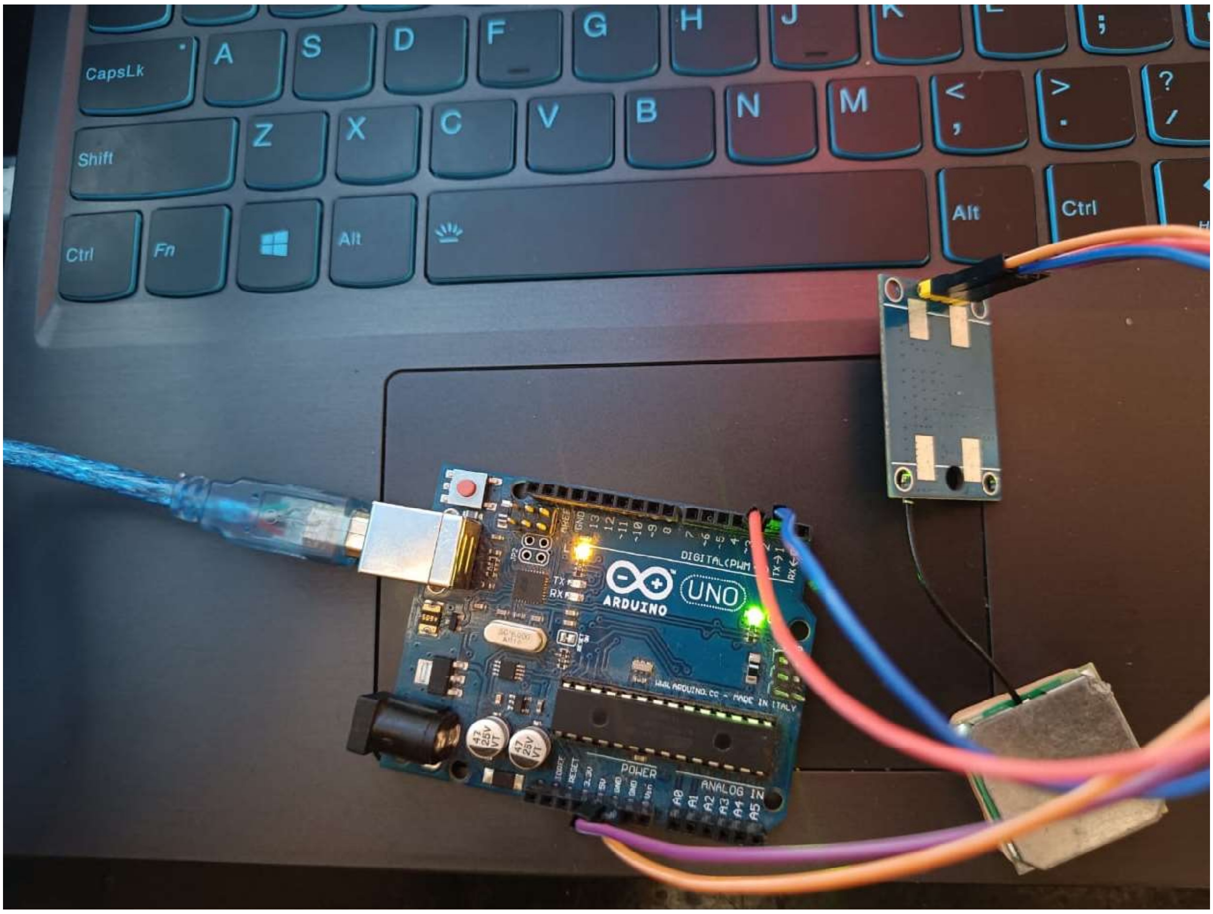
6. OUTPUT & PERFORMANCE ANALYSIS

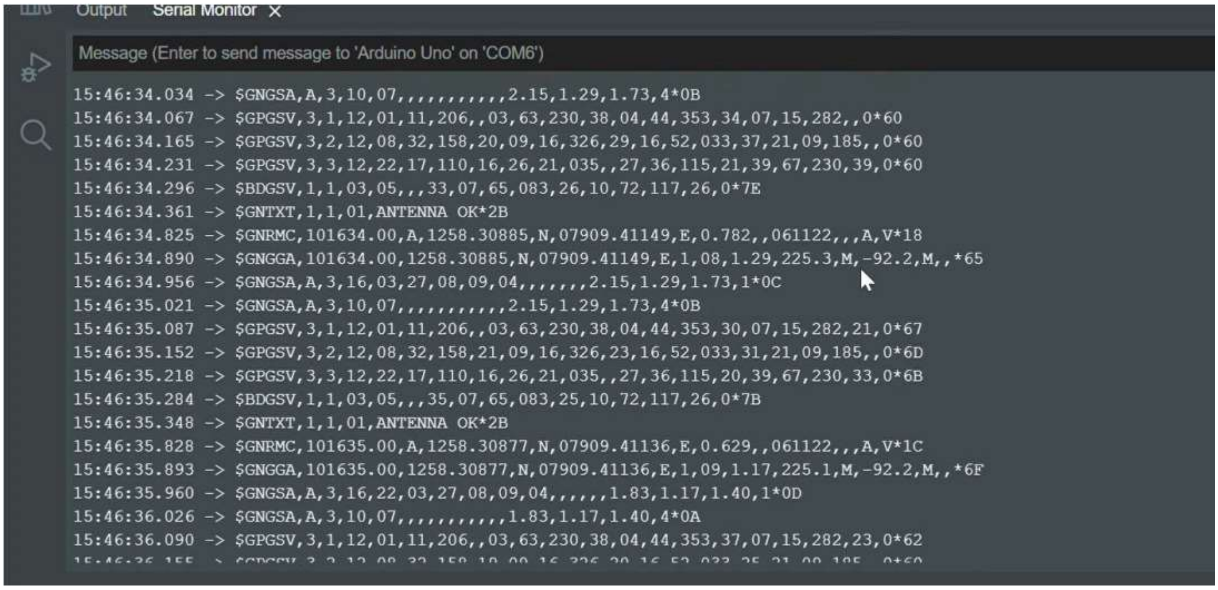
6.1.EXECUTION SCREENSHOTS




File Edit View Run Kernel Tabs Settings Help
main.ipynb
NLP.ipynb
video
person 0.99
person 1.0
Python 3 (ipykernel)
{'person': 1}
2.408821
{'person': 1}
2.857248
{'person': 1}
3.310998
{'person': 1}
3.744316
{'person': 1}
4.177089
Simple
Python 3 (ipykernel) | Busy
Mode: Command
Ln 1, Col 1
main.ipynb

GPS MODULE:


ARDUINO
UNO



6.2.PERFORMANCE METRICS

The system designed by us is utilizing multiple Single Shot Detectors internally through the YOLOv3 model, and hence it can detect many objects together. It does not focus on a singular object in the entire environment. It does become harder for the model to properly output in environments where there are many objects, due to object collisions and viewpoint limitations. The objects are spoken out loudly with priority given to those that are more unusual/risky and hence can be constituted as being more important.

Overall, it can be clearly evaluated that the model is having an accuracy metric of approximately 88-90th percentile. Due to the aforementioned problems, the environmental detection accuracy does suffer but still reaches as high as 75-80th percentile. This can vary widely in precision based on the environment of choice. Therefore, the system presents a relatively accurate measure of the surroundings around it and can work to help those with disabilities.

6.3. PERFORMANCE COMPARISON

Most previous works in this field are quite limited in the working schematics and are not even close to as functional as our solution. Most of them also fail at multiple object detection and accurately prioritizing the alerts sent to the user, making them actively dangerous to the user's health and well-being when in an emergency. Our system also uses the greatest number of sensors, from GPS, GSM, Compass, IR, and several more like accelerometers to define the user's environment and movement is much easier to understand for the user themselves.

7. CONCLUSION & FUTURE WORKS

People with disabilities are suffering more than ever in today's environment of constant change. There are several technologies available to aid those with these limitations in navigating their environment, but they all have several issues and lack a comprehensive answer. Therefore, using automated queries to the Google Vision API, a unique object identification technology has been used to assist users with infirmities such as blindness. Scene detection has been used to find the individual within their surroundings using the findings from the aforementioned test. (For instance, a bedroom, bathroom, or garage.)

The apparatus works well and is effective in guiding the blind. This allows it to identify the danger of the item in front of the user. Because of the optimized technique utilized in the background, the processing speed is great. We want to implement this code using Google Coral, an edge computing device with an externally connected camera. We aim to add additional workspaces and make it a useful product. The gadget should recognize the class numbers stated outside of the classrooms as well. The gadget should be linked to a mapping system that will direct the user to the requested place. In addition, the gadget should have a health monitoring system that detects the user's pulse rate and guides him appropriately. The gadget must be made multilingual so that it may be used by everyone.

The two models have been combined to create a system that is effective and practical, and it has undergone considerable testing under several different conditions. In order to better assist people, natural language processing has been widely used to translate object data into understandable English data. A smooth application flow that is designed using a multi-stack strategy and uses Python as the main language of choice for handling activities involving artificial intelligence has allowed the complete procedure outlined above. To assist consumers, the whole system has been combined into a single, user-friendly application.

Overall, the entire system is much more robust and safer than existing solutions, while also being easier to use and better in terms of all performance metrics. Although more can be done to improve the operational performance in certain crowded and fast-moving environmental situations, this can be the subject of future research to help make the system even better for application to the lives of the disabled, helping them live a life closer to normality.

8. REFERENCES

[1] Dhou, S., Alnabulsi, A., Al-Ali, A. R., Arshi, M., Darwish, F., Almaazmi, S., & Alameeri, R. (2022). An IoT machine learning-based mobile sensors unit for visually impaired people. Sensors, 22(14), 5202.

[2] Vrindavanam, J., Srinath, R., Fathima, A., Arpitha, S., Rao, C. S., & Kavya, T. (2021, August). Machine Learning based approach to Image Description for the Visually Impaired. In 2021 Asian Conference on Innovation in Technology (ASIANCON) (pp. 1-6). IEEE.

[3] Felix, S. M., Kumar, S., & Veeramuthu, A. (2018, May). A smart personal AI assistant for visually impaired people. In 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 1245-1250). IEEE.

[4] El-Taher, F. E. Z., Taha, A., Courtney, J., & Mckeever, S. (2021). A systematic review of urban navigation systems for visually impaired people. Sensors, 21(9), 3103.

[5] Aravindan, C., Arthi, R., Kishankumar, R., Gokul, V., & Giridaran, S. (2021). A Smart Assistive System for Visually Impaired to Inform Acquaintance Using Image Processing (ML) Supported by IoT. In Hybrid Artificial Intelligence and IoT in Healthcare (pp. 149-164). Springer, Singapore.

[6] Hengle, A., Kulkarni, A., Bavadekar, N., Kulkarni, N., & Udyawar, R. (2020, August). Smart Cap: A Deep Learning and IoT Based Assistant for the Visually Impaired. In 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT) (pp. 1109-1116). IEEE.

[7] Malhotra, S., Balasubramanian, K., Sultana, H. P., & Kumar, T. P. An IoT-ML based Proactive Walking aid for the Visually Challenged.

[8] Varalakshmi, I., & Kumarakrishnan, S. (2019, March). Navigation system for the visually challenged using Internet of Things. In 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN) (pp. 1-4). IEEE.

[9] Ghosh, A., Al Mahmud, S. A., Uday, T. I. R., & Farid, D. M. (2020, June). Assistive technology for visually impaired using tensor flow object detection in raspberry Pi and coral USB accelerator. In 2020 IEEE Region 10 Symposium (TENSYMP) (pp. 186189). IEEE.