A Project Report

On

VI-SPECTACLE

Submitted in partial fulfilment of the requirement of

University of Mumbai

For the Degree of

Bachelor of Engineering

in

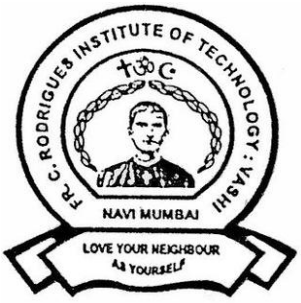
COMPUTER ENGINEERING

Submitted by

Freddy Poly
Dipak Tiwari
Varghese Jacob

Supervised by

Shagufta Rajguru



Department of Computer Engineering

Fr. Conceicao Rodrigues Institute of Technology
Sector 9A, Vashi, Navi Mumbai - 400703

UNIVERSITY OF MUMBAI

2019-2020

APPROVAL SHEET

This is to certify that the project entitled

"VI-SPECTACLE"

Submitted by

Freddy Poly 101616
Dipak Tiwari 101658
Varghese Jacob 101659

Supervisors : _____

Project Coordinator : _____

Examiners : 1. _____

2. _____

Head of Department : _____

Date :

Place :

Declaration

We declare that this written submission for B.E. Declaration entitled "**VI-SPECTACLE**" represent our ideas in our own words and where others' ideas or words have been included. We have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any ideas / data / fact / source in our submission. We understand that any violation of the above will cause for disciplinary action by institute and also evoke penal action from the sources which have thus not been properly cited or from whom paper permission have not been taken when needed.

Project Group Members:

1. Freddy Poly, 101616

_____

2. Dipak Tiwari, 101658

_____

3. Varghese Jacob, 101659

_____

Abstract

It is arduous for the visually impaired people to identify every object in their immediate surroundings. With recent advancements in technologies, various products are being developed to overcome these constraints. In contrast to these products, this paper delivers a solution which is portable and economically viable for the visually impaired. A device is being developed, equipped with an ESP32 cam module, an ultrasonic sensor and a mobile phone. A real time system is created which performs object detection on the live streamed data, generated by the camera of the mobile phone. In accordance with the data from the sensors, the object and its proximity are identified. A graphical user interface acquires this data and cautions the user about the imminent obstacles, thus guiding them through a pertinent path to reach their destination safely.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

According to a survey conducted by the WHO (World Health Organization) in 2012, over 39 million people in the world suffer from complete vision loss [2]. Vision loss causes considerable repercussions on the lives of those who experience it as well as on their family, friends and society. It directly influences the patient's ability to carry out day to day activities like reading, travelling and fraternizing. It is also burdensome for them to move from one location to another as it requires identifying various objects and their positions.

An instrument or a tool is required in order to assist them in carrying out these activities. The white cane is the most frequently used device among the visually impaired people [3]. This device enables its user to scrutinize the neighbourhood to identify obstacles and travel safely. It also assists the bystanders to identify if the user is visually challenged and hence take appropriate care. Even though the cane helps in alerting the user of the surrounding, it does not necessarily let them recognize the type of object. However, other sensory organs can be utilized to identify and pinpoint these objects. Despite the fact that it helps the user to speculate the object type, the accuracy is substandard.

1.2 Motivation

Vision loss is one of the major issues in the world and our community today. Many people facing this issue have risked their lives due to inadequate aid. The main reason behind selecting this topic is to help the less fortunate people perceive the surroundings just like us and deliver a superior sensory experience.

Along with this, it also satisfies our craving to learn and develop products that uses technologies like Deep Learning and IoT, which helps in building a better world.

1.3 Aim and Objective

The project aims to present a solution to problems faced by the visually challenged, furthermore, yielding a higher accuracy. Visually impaired people use a white cane or a long cane along with the help of their other senses to navigate outdoors. Although the cane provides a limited idea about the surroundings, the person still does not get a perfect picture of what exactly is happening in the outside world. The VI spectacle should

enable the user to identify various different obstacles and in real time.

1.4 Report Outline

The idea is to develop a pair of spectacles that are loaded up with state-of-the-art technologies. The initial work comprises of integrating a set of ultrasonic sensors onto the belt of the user. Ultrasonic sensors are placed in three different directions of the user. The data from these sensors are compared and the user is guided through the proper direction.

This initial work is extended by integrating the phone camera into the existing set of hardware. To increase the latency of the system, the total number of sensors are reduced to one. And the belt is replaced by a pair of spectacles. Secondly, the data generated by the sensor is transferred over a network using a Wi-Fi module. A graphical user interface is developed using the Android Studio development environment. The data is being sent to this Android application. Finally, based on this data, object detection is performed, and the user is informed about the obstacle.

Many research papers on similar projects were read to understand more about this project. A comparison was made to understand the advantages and disadvantages of using different deep learning algorithms. It was concluded that YOLO model provides the highest accuracy. Various other hardware projects related to the topic were also researched, it was found that WiFi was the fastest way of communication possible.

Chapter 2

Study Of the System

2.1 About ESP32 Module

The ESP32 cam is a micro controller chip with integrated Wi-Fi and dual mode Bluetooth. It has several GPIOs to connect peripherals, it also features a microSD card slot that can be useful to store files to serve to clients. The pin diagram is as shown in Figure 2.1.Other important parts includes a FTDI programmer, Jumper wires and Breadboard.

Figure 2.1: **Pin Diagram**

It has cutting edge features that enhances the efficiency and efficacy of the project. The development environment is Arduino IDE and the language used is the C programming language. The ESP32 cam does not come with a USB connector, so the code is uploaded by using a USB to TTL serial converter. FTDI232RL is the most commonly used USB to TTL converter. It has a mini-B USB port which is connected to a programming device with a USB input.

A voltage of 3.3V-5V is provided by its Vcc along with a current supply of 500mA. It contains Rx and Tx pins which helps in transmitting and receiving the data from the ESP32 cam module. The connections are as shown in Figure 2.2. In order to upload the code, ESP32 has to be kept in the flashing mode. After the code is uploaded, it is taken out of flashing mode and an IP address is generated. The data can be received by using any digital device that is connected to the same network using the generated IP address. Various different features provided by the ESP32-CAM are :

- The smallest 802.11b/g/n Wi-Fi BT SoC module
- Low power 32-bit CPU,can also serve the application processor

- Up to 160MHz clock speed, summary computing power up to 600 DMIPS
- Built-in 520 KB SRAM, external 4MPSRAM
- Supports UART/SPI/I2C/PWM/ADC/DAC
- Support OV2640 and OV7670 cameras, built-in flash lamp
- Support image WiFI upload
- Support TF card
- Supports multiple sleep modes
- Embedded Lwip and FreeRTOS

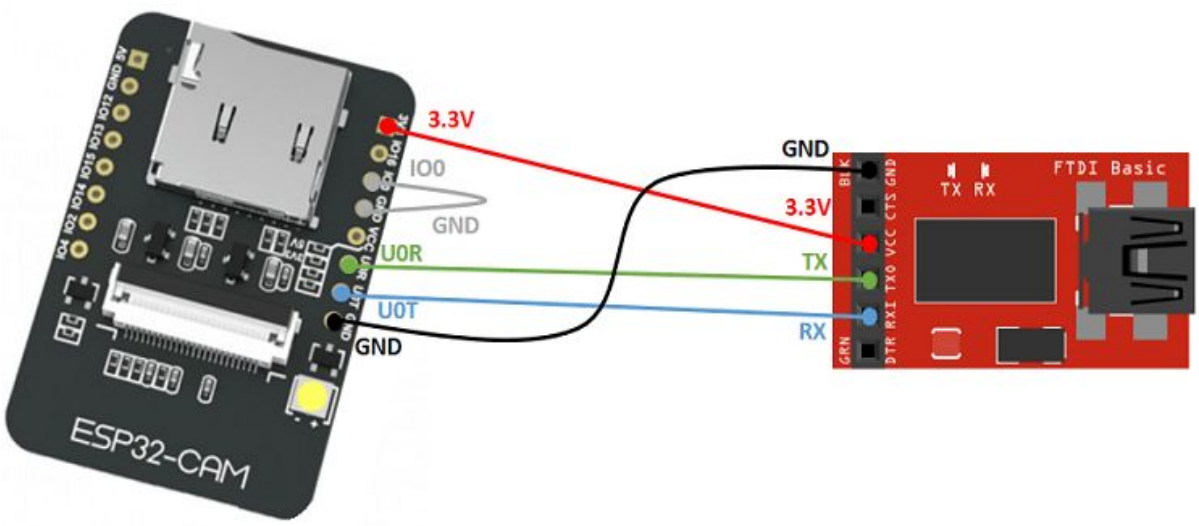


Figure 2.2: **Circuit Diagram. Taken from [1]**

2.2 About the YOLO model

YOLO predicts multiple bounding boxes per grid cell. At training time we only want one bounding box predictor to be responsible for each object. We assign one predictor to be "responsible" for predicting an object based on which prediction has the highest current IOU with the ground truth. This leads to specialization between the bounding box predictors. Each predictor gets better at predicting certain sizes, aspect ratios, or classes of object, improving overall recall.

Compared to other region proposal classification networks (fast RCNN) which perform detection on various region proposals and thus end up performing prediction multiple times for various regions in a image, Yolo architecture is more like FCNN (fully convolutional neural network) and passes the image (nxn) once through the FCNN and output is (mxm) prediction.

YOLO imposes strong spatial constraints on bounding box predictions since each grid cell only predicts two boxes and can only have one class. This spatial constraint limits the number of nearby objects that our model can predict. Our model struggles with small objects that appear in groups, such as flocks of birds. Since our model learns to predict bounding boxes from data, it struggles to generalize to objects in new or unusual aspect ratios or configurations. Our model also uses relatively coarse features for predicting bounding boxes since our architecture has multiple downsampling layers from the input image.

2.3 Related Works

2.3.1 Voice Assistance for Visually Impaired People

A. Karthik et al [4] describes the difficulties faced by the visually challenged in reading messages on posters, billboards, etc in their day to day life. They generally rely on Braille scripts to read these messages. The paper proposes the use of Optical Character Recognition for detecting messages which are not in the Braille format. A Raspberry Pi equipped with a digital camera is used to take input images. The detected messages are then converted from text to audio using a voice synthesizer.

2.3.2 Lightweight smart glass system with audio aid for visually impaired people

Feng L. et al [5] have created a smart glass prototype system using an Intel development board called the Intel Edison. It is a portable device which assists the user in locating public signs like bus stations, subway stations, public toilets, hotels and so on. This system automatically identifies public signs throughout the path that the user is walking. Open CV is being used for efficient image analysis. A voice assistant helps in hinting the user of these signs using a wireless bone conduction headphone.

2.3.3 Smart walking cane for the visually challenged

S. Murali et al [6] proposes a device which is mounted with multiple sensors to guide the user safely. This smart walking cane consists of several sensors,

including a depth sensor, a step sensor and a water sensor. These sensors help in identifying the type of terrain in which the user is in. The cane also consists of GPS and GSM modules that help in tracking the user and alerting the police in case of emergencies. The user is guided by an audio module which directs the user via a secured direction. This helps the user in having a better understanding of their surroundings.

2.3.4 Smart device for visually impaired people

R. Kasthuri et al [7] elucidates the use of Android technology to solve the problems faced by the visually challenged. The paper proposes a voice-based application with a user-friendly UI that enables the user to access any application on the phone, play songs or even call any contact with mere voice commands. These voice commands are converted into text by using the Speech Recognition Engine. The development environment being used is the Selendroid which is an instrumentation-based automation framework.

2.3.5 Smart Assistive Navigation Devices for Visually Impaired People

A. Pardasani et al [8] have designed a set of smart devices that helps the visually impaired people in navigating outdoors. The paper mainly talks about two devices namely, a smart glass and a smart pair of shoes. The smart glass is able to detect objects and convert it into audio messages. Furthermore, it can convert the detected texts into braille scripts, by printing them using the braille printer, to store as permanent records. The second device is a pair of smart shoes equipped with various different sensors and a micro controller. This helps in alerting the user about the object's distance and proximity.

Chapter 3

Proposed System

3.1 Problem Statement

The device is equipped with a mobile phone, ESP32 and an ultrasonic sensor. The data from the sensor is sent to the android application using Wi-Fi. Depending on these values, the camera is switched on and off. The application performs object detection and gives the results accordingly. The result is then converted from text to speech and sent to the user.

3.2 Scope

The object detection algorithm is used to detect a limited set of objects which is found in our surrounding. To improve the efficiency, number of objects are set to 10. The object which is to be detected should at least be at a distance of 1.5 to 2 metre from the user.

If multiple objects are detected, the closest object would be notified first. A delay of one second can be expected due to latency issues. This project is worked upon with an intention of making a contribution towards medical welfare.

3.3 Proposed System

The VI Spectacle consists of multiple hardware modules that are mounted on a pair of glasses. These hardware modules include an ESP32 module, ultrasonic sensors and jumper cables. The mobile application receives the data sent by the hardware modules through the Wi-Fi network. The data consists of the distances between the user and the obstacle. Depending on the values of these distances, computations are performed by this mobile application as shown in Figure 4.5.

The Front End of the system is developed using the integrated development environment called Android Studio. The application asks the user to turn the device on. Once the device is turned on, the mobile application connects itself to the device using Wi-Fi. The device starts streaming the distances immediately after the connection is established. Depending on these values, the application decides if object detection is to be performed. As soon as an obstacle comes close to the user, the application opens the camera and performs object detection. An interactive UI helps in elucidating the type of object. The whole process is repeated again once it reaches the last step.

Chapter 4

Design Of the System

4.1 Requirement Engineering

4.1.1 Requirement Elicitation

The user needs to open the application before switching on the device. The application enables the user to start the Wi-Fi hotspot automatically. After pressing the button on the ESP, the module tries to establish a connection with the Wi-Fi network using the SSID and password provided in the source code of device. Permissions for the camera and user credentials should be provided by the user. The initial set up enables the user to save time in the future.

4.1.2 Software lifecycle model

Agile process model is being used to develop this project. Agile process model refers to a software development approach based on iterative development. The project was broken down into small tasks which were improved after each iteration. The project scope and requirements were laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration were clearly defined in advance.

Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is completely developed. The process flow is as shown in the Figure 4.1.

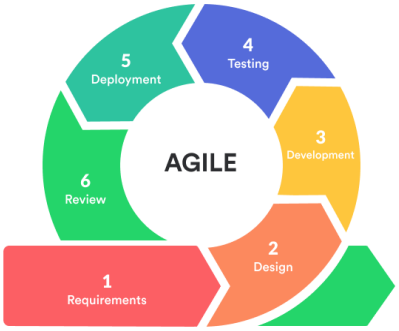


Figure 4.1: **Agile Process Model**

4.1.3 Requirement Analysis

4.1.3.1 UML/ Sequence diagrams

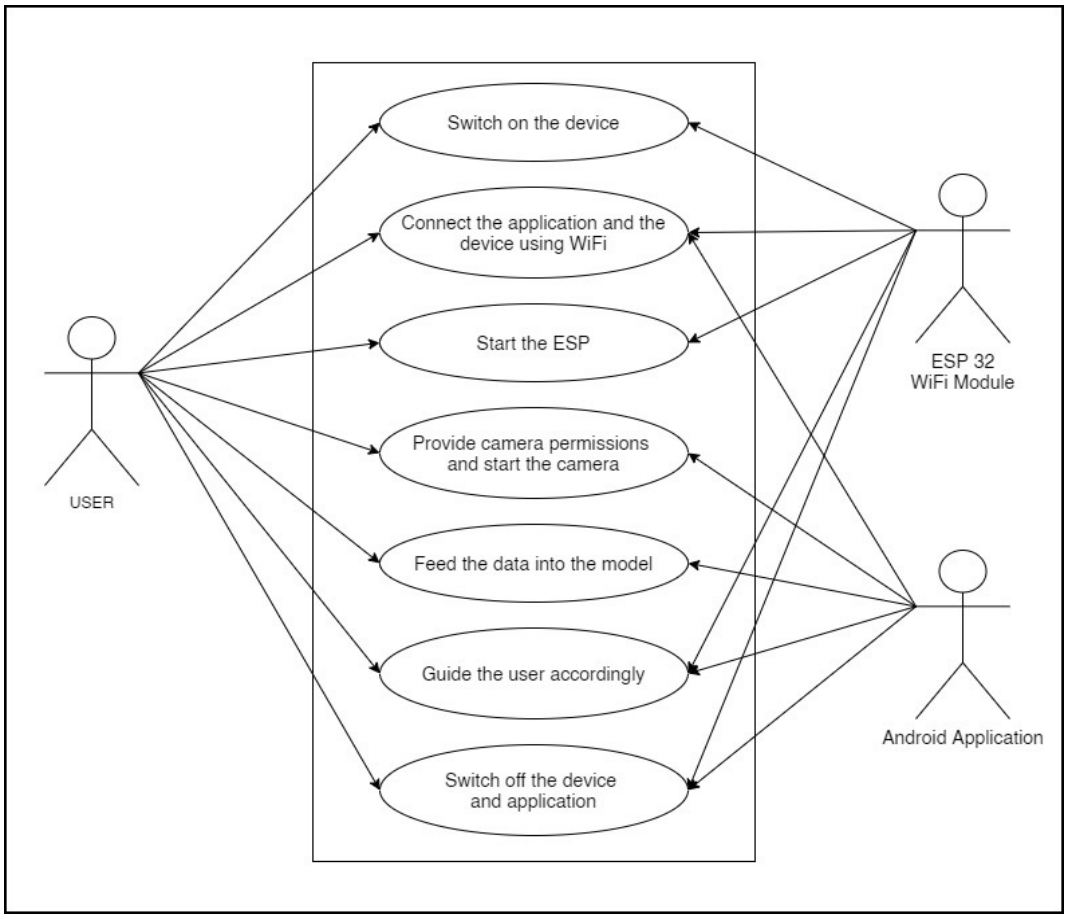


Figure 4.2: **Use Case UML Diagram**

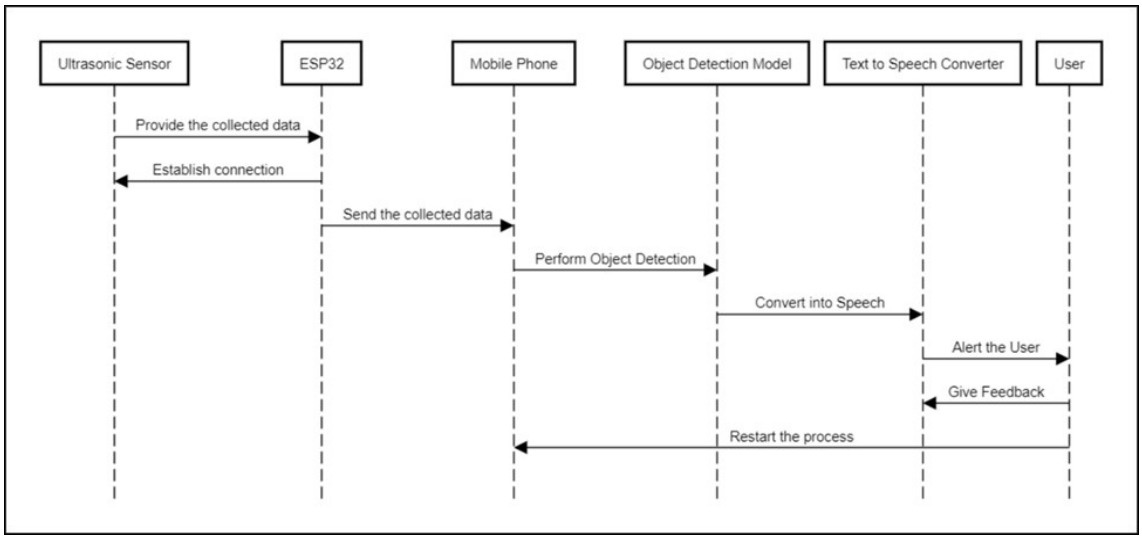


Figure 4.3: **Sequence Diagram**

4.1.3.2 Cost Analysis

The cost of our project completely depends on the hardware components used. This project delivers a solution which is portable and economically viable for the visually impaired. The cost distribution is as shown in the Table 4.1.

Component	**Unit Price**
ESP32	Rs 500
Ultrasonic Sensor	Rs 120
Lithium Ion Battery	Rs 200
MT3608 Step Up Boost Converter	Rs 90
Miscellaneous	Rs 50
Total	**Rs 960**

Table 4.1: Cost Distribution

4.1.3.3 Hardware Requirements

- ESP32 Module
- FTDI232RL USB to TTL Serial Converter
- Ultrasonic Sensor
- Lithium Ion Battery
- MT3608 Step Up Boost Converter
- Micro USB Connector
- Printed Circuit Boards (PCBs)
- Slide Switch
- Jumper Cables
- Mobile Phone

4.1.3.4 Software Requirements

- Arduino IDE
- Android Studio
- Anaconda Navigator
- Tensorflow

4.2 System architecture

4.2.1 Activity diagram

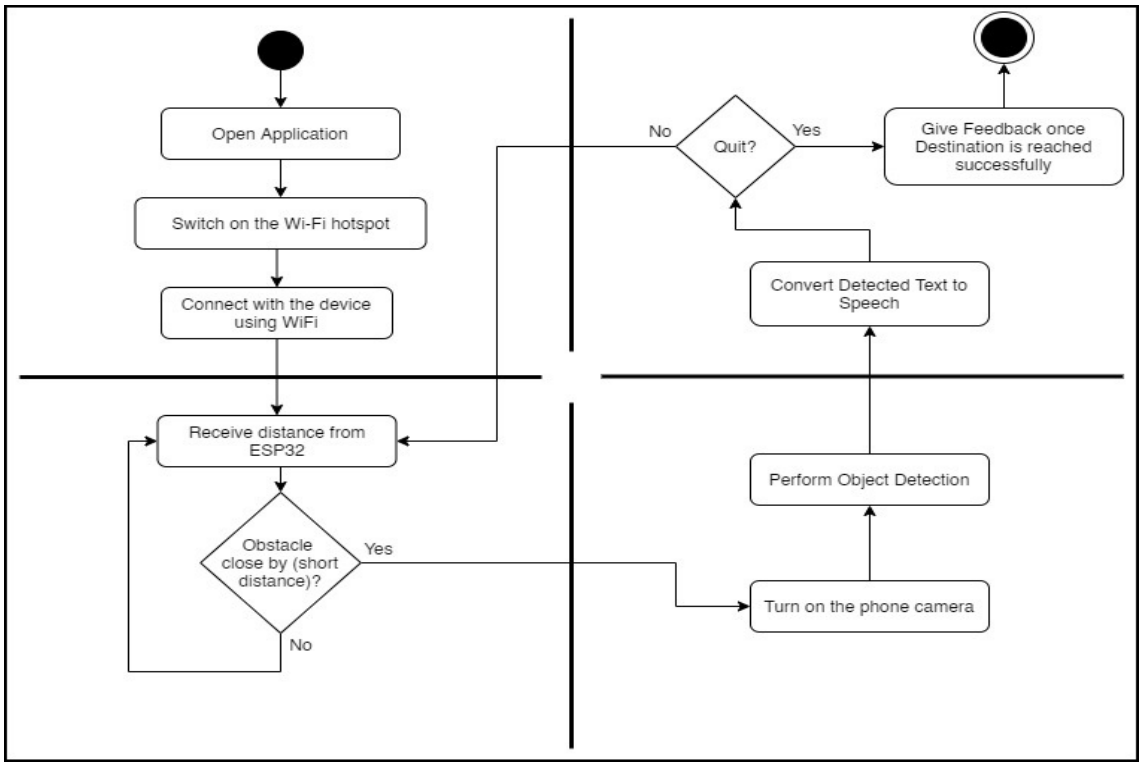


Figure 4.4: **Activity Diagram**

4.2.2 Block Diagram

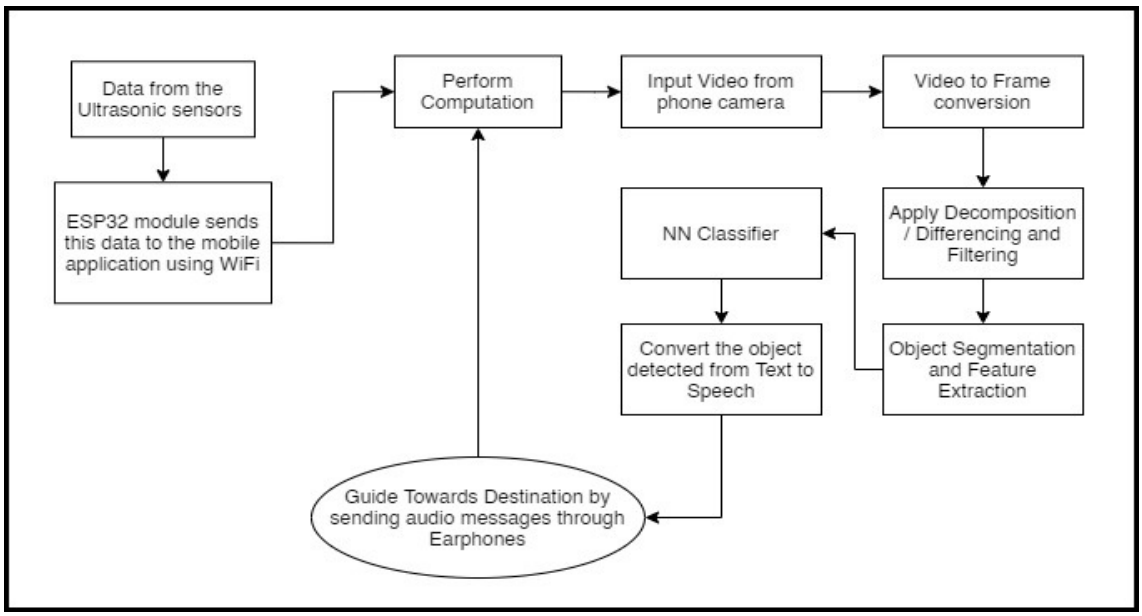


Figure 4.5: **Block Diagram**

Chapter 5

Result and Discussion

5.1 Screenshots of the System

5.1.1 Screenshots of the Device

Figure 5.1: **Device Picture (Front View)**

Figure 5.2: **Device Picture (Side View)**

Figure 5.3: **Device Circuit View**

5.1.2 Screenshots of the Application

Figure 5.4: **Opening Screen**

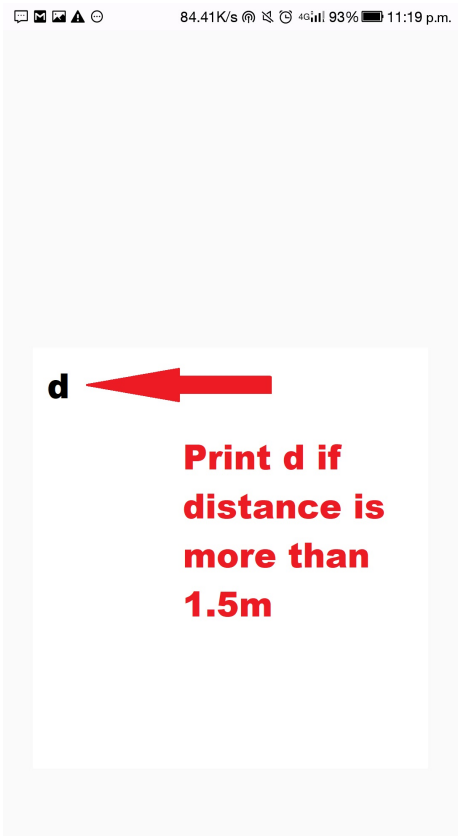


Figure 5.5: **Main Screen I (Data received by ESP)**

Figure 5.6: **Main Screen II (Data received by ESP)**

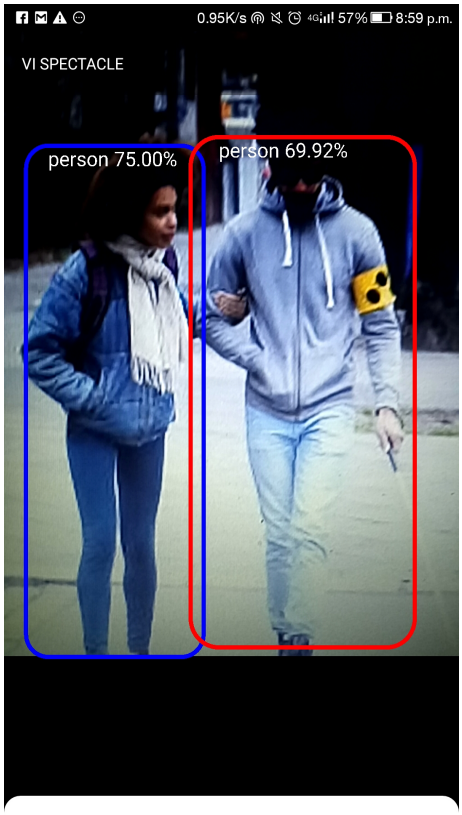


Figure 5.7: **Object Detection (People)**

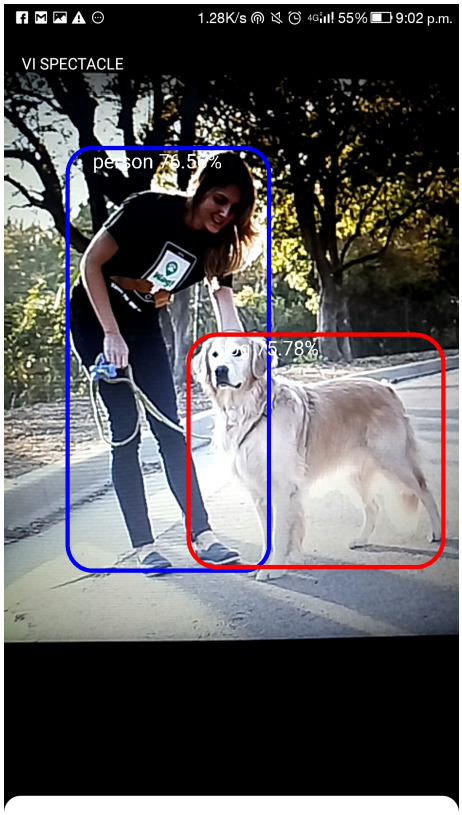


Figure 5.8: **Object Detection (Person, Dog)**

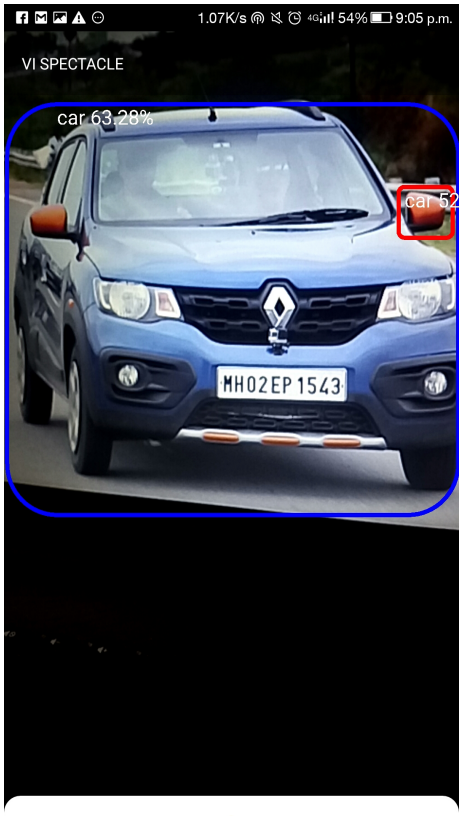


Figure 5.9: **Object Detection (Car)**

Figure 5.10: **Object Detection (Chair)**

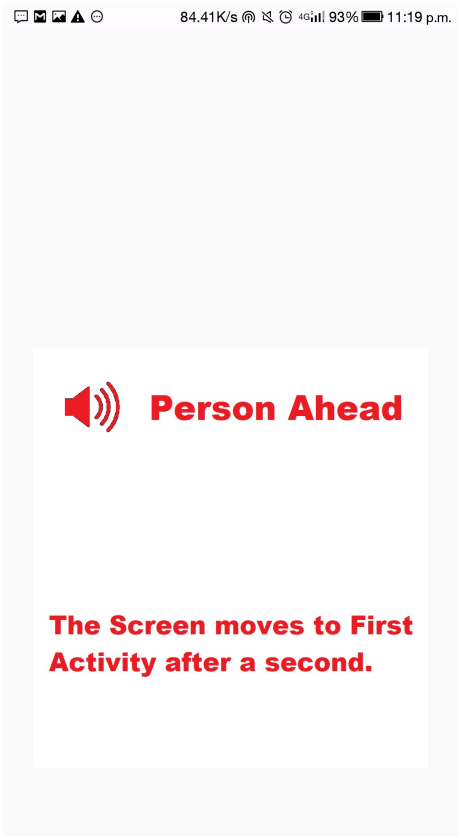


Figure 5.11: **Voice Command (Person)**

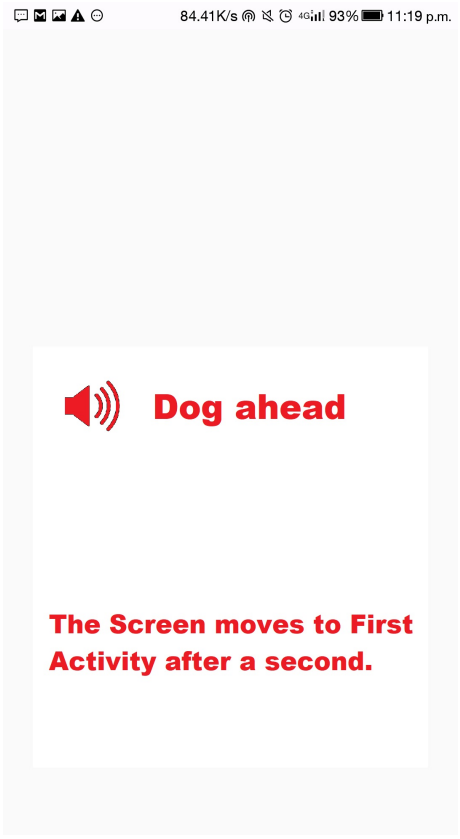


Figure 5.12: **Voice Command (Dog)**

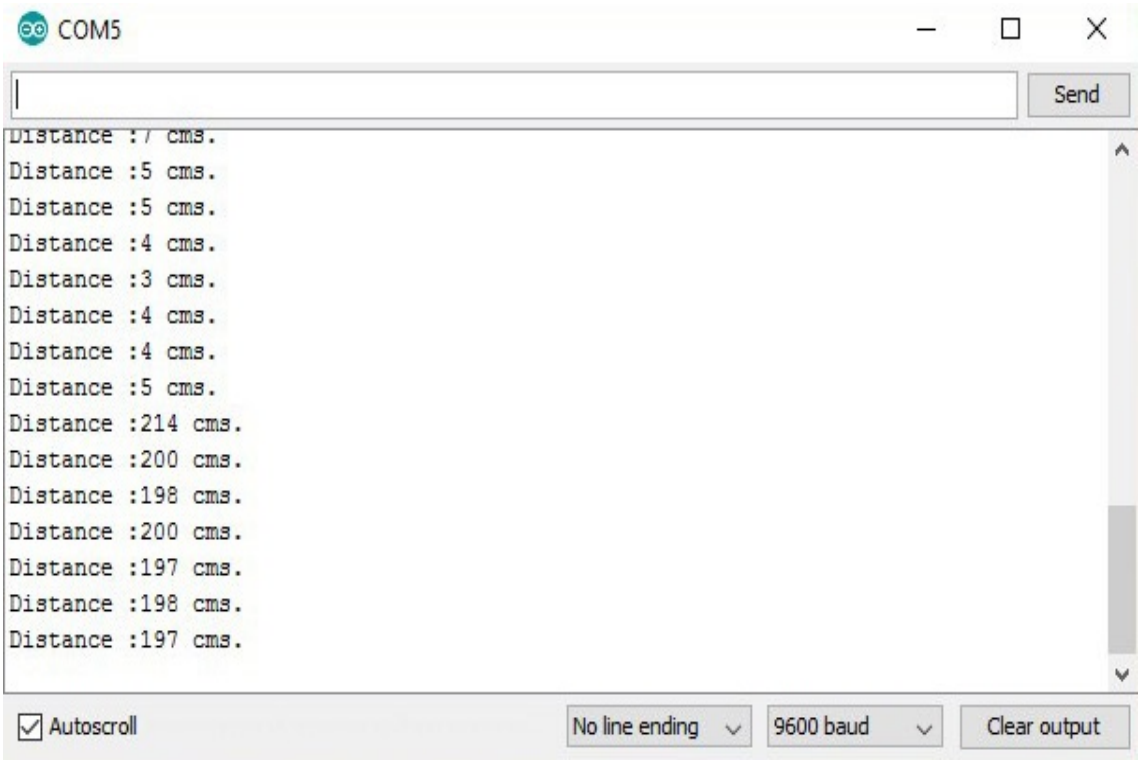


Figure 5.13: **Serial Monitor of Arduino IDE**

5.2 Important Code Snippets

5.2.1 Arduino IDE - Code in the ESP

```
#include <WiFi.h>
const int trigPin1 = 13;
const int echoPin1 = 12;
long duration1;
int distance1;

const char* wifi_name = "Le";
const char* wifi_pass = "12345678";
WiFiServer server(80);

void setup()
{
    pinMode(trigPin1, OUTPUT);
    pinMode(echoPin1, INPUT);
    Serial.begin(115200);

    Serial.print("Connecting to ");
    Serial.print(wifi_name);
    WiFi.begin(wifi_name, wifi_pass);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("Connection Successful");
    Serial.print("IP address: ");
    Serial.println(WiFi.localIP());
    server.begin();
}

void loop()
{
    WiFiClient client = server.available();

    digitalWrite(trigPin1, LOW);
```

```
    delayMicroseconds(2);
    digitalWrite(trigPin1, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin1, LOW);

    duration1 = pulseIn(echoPin1, HIGH);
    distance1= duration1*0.034/2;
    if (client)
    {
        if(distance1<150)
        {
            client.print("<body><h1>l</h1>");
        }
        else
        {
            client.print("<body><h1>d</h1>");
        }
    }
    delay(2000);
}
```

5.2.2 First Activity (Receive data from ESP)

```
String url = "http://192.168.43.12/";
wv1.loadUrl(url);
wv1.setFindListener(new WebView.FindListener() {
    @Override
    public void onFindResultReceived(int
    activeMatchOrdinal, int numberOfMatches,
    boolean isDoneCounting) {
            if(numberOfMatches==1){
                Intent i = new Intent(
                getBaseContext(),
                DetectorActivity.class);
                startActivity(i);
                finish();
            }
    }
});
Runnable myRunnable = new Runnable() {
```

```
    @Override
    public void run() {
        while (1==1) {
            try {
                Thread.sleep(3000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            textView.post(new Runnable() {
                @Override
                public void run() {
                    wv1.reload();
                    wv1.findAllAsync("l");
                }
            });
        }
    };
};
Thread myThread = new Thread(myRunnable);
myThread.start();
```

5.2.3 Object Detection Activity

```
protected void processImage() {
  ++timestamp;
  final long currTimestamp = timestamp;
  trackingOverlay.postInvalidate();
    setRequestedOrientation(ActivityInfo.
    SCREEN_ORIENTATION_LANDSCAPE);
  if (computingDetection) {
    readyForNextImage();
    return;
  }
  computingDetection = true;
  LOGGER.i("Preparing image " + currTimestamp +
  " for detection in bg thread.");
  rgbFrameBitmap.setPixels(getRgbBytes(),
  0, previewWidth, 0, 0, previewWidth,
  previewHeight);
  readyForNextImage();
```

```
final Canvas canvas = new Canvas
(croppedBitmap);
canvas.drawBitmap(rgbFrameBitmap,
frameToCropTransform, null);
// For examining the actual TF input.
if (SAVE_PREVIEW_BITMAP) {
  ImageUtils.saveBitmap(croppedBitmap);
}

runInBackground(
    new Runnable() {
      @Override
      public void run() {
        LOGGER.i("Running detection on
        image " + currTimestamp);
        final long startTime = SystemClock
        .uptimeMillis();
        final List<Classifier.Recognition>
        results = detector.recognizeImage
        (croppedBitmap);
        lastProcessingTimeMs = SystemClock
        .uptimeMillis() - startTime;

        cropCopyBitmap =
        Bitmap.createBitmap(croppedBitmap);
        final Canvas canvas = new Canvas
        (cropCopyBitmap);
        final Paint paint = new Paint();
        paint.setColor(Color.RED);
        paint.setStyle(Style.STROKE);
        paint.setStrokeWidth(2.0f);

        float minimumConfidence =
        MINIMUM_CONFIDENCE_TF_OD_API;
        switch (MODE) {
          case TF_OD_API:
            minimumConfidence =
            MINIMUM_CONFIDENCE_TF_OD_API;
            break;
        }
```

```
final List<Classifier.Recognition>
mappedRecognitions =
    new LinkedList<Classifier.
    Recognition>();

for (final Classifier.Recognition
result : results) {
  final RectF location = result.
  getLocation();
  if (location != null && result.
  getConfidence() >= minimumConfidence)
  {
    canvas.drawRect(location, paint);
    cropToFrameTransform.mapRect
    (location);
    result.setLocation(location);
    mappedRecognitions.add(result);
    Log.i("Identified Objects: ",
    Arrays.toString(new
    Classifier.Recognition[]{result}));
    String val = Arrays.toString(new
    Classifier.Recognition[]{result});
    String[] arr = val.split(" ");
    String objectname = arr[1];
    String percentage = arr[2];
    percentage = percentage.substring
    (0, percentage.indexOf("."));
    percentage = percentage.
    replaceAll("[\\D]", "");
    Integer percent = Integer.
    parseInt(percentage);
    if (percent>65) {
        Intent i = new Intent
        (getBaseContext(), third.class);
        if (objectname.equals("person")
        &&(percent>70)){
            i.putExtra("name",name);
            startActivity(i);
            finish();
```

```
                    }
                    else if(objectname.equals("chair"))
                    {
                        i.putExtra("name", objectname);
                        startActivity(i);
                        finish();
                    }
                }
            }
        }

        tracker.trackResults(mappedRecognitions
        , currTimestamp);
        trackingOverlay.postInvalidate();

        computingDetection = false;

        runOnUiThread(
            new Runnable() {
              @Override
              public void run() {
                showFrameInfo(previewWidth +
                "x" + previewHeight);
                showCropInfo(cropCopyBitmap.
                getWidth() + "x" +
                cropCopyBitmap.getHeight());
                showInference(
                lastProcessingTimeMs + "ms");
              }
            });
      }
    });
}
```

5.3 Testing

5.3.1 Unit Testing

Unit Testing is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected.

Test Case	**Expected Output**	**Actual Output**	**Test Cases Executed**	**Test Cases Passed**	**Remark**
Display Sensor value on Serial Monitor	Display Distance	Displays Distance	10	9	Pass
Send values from ESP to Android Application	Receive values	Receives values	10	8	Pass
Thread continuously runs in the background and reloads the webpage	Should not crash	Doesn't crash, works smoothly	10	7	Pass
Perform Object Detection	Identify Objects	Identifies Objects	10	8	Pass
Convert Text to Speech	Convert TTS	Converts TTS	10	10	Pass

Table 5.1: Unit Testing

5.3.2 Integration Testing

Integration testing tests integration or interfaces between components, interactions to different parts of the system such as an operating system, file system and hardware or interfaces between systems. Integration testing is a key aspect of software testing.

Test Case	**Expected Output**	**Actual Output**	**Test Cases Executed**	**Test Cases Passed**	**Remark**
Is ESP receiving distance from the Ultrasonic Sensor	Receive Distances	Receives Distances	10	10	Pass
Send Distances from ESP to Android Application	Receive Distances	Receives Distances	10	9	Pass
Go to next page if received distance is less than distance threshold	Go to next page	Moves to next page	10	9	Pass
Go to next page if object is detected	Go to next page	Moves to next page	10	9	Pass
Convert the detected object to speech	Convert TTS	Converts TTS	10	10	Pass
Move back to first page after completion	Go to first page	Moves to first page	10	10	Pass

Table 5.2: Integration Testing

5.3.3 Blackbox Testing

Blackbox testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance.

Test Case	**Expected Output**	**Actual Output**	**Test Cases Executed**	**Test Cases Passed**	**Remark**
Switch on Wi-Fi Hotspot automatically	Hotspot should start	Hotspot switches on	10	10	Pass
User connects the phone and device	Connection should take place	Connection successful	10	8	Pass
Perform Object Detection	Identify Objects	Identifies Objects	10	8	Pass
Convert Text to Speech	Convert TTS	Converts TTS	10	10	Pass

Table 5.3: Blackbox Testing

Chapter 6

Conclusion & Future Scope

6.1 Conclusion

This project presents a prototype of the VI Spectacle for the visually challenged. The methodologies used to design this product have been explained in detail. The proposed system involves two major parts which includes the VI Spectacle and the mobile application. VI Spectacle mainly consists of a ESP32 module that sends the distances to the application using WiFi. All the computations take place inside the mobile application. The mobile application performs a variety of tasks like image capturing, computations, object detection and guiding the user. The system is able to detect and recognize objects in real time. This device helps the visually challenged to walk around cities without any difficulties.

6.2 Future Scope

The device can further be upgraded to differentiate between moving and stationary objects. As well as incorporating the device into a single unit, thus eliminating the use of the mobile application entirely.

References

[1] "Esp32-cam video streaming web server (works with home assistant) — random nerd tutorials."

[2] D. Pascolini and S. P. Mariotti, "Global estimates of visual impairment: 2010," *British Journal of Ophthalmology*, vol. 96, no. 5, pp. 614–618, 2011.

[3] I. Khan, S. Khusro, and I. Ullah, "Technology-assisted white cane: evaluation and future directions," *PeerJ*, vol. 6, p. e6058, 2018.

[4] A. KARTHIK, V. K. RAJA, and S. PRABAKARAN, "Voice assistance for visually impaired people," *2018 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, 2018.

[5] F. Lan, G. Zhai, and W. Lin, "Lightweight smart glass system with audio aid for visually impaired people," *TENCON 2015 - 2015 IEEE Region 10 Conference*, 2015.

[6] S. Murali, R. Shrivatsan, V. Sreenivas, S. Vijjappu, S. J. Gladwin, and R. Rajavel, "Smart walking cane for the visually challenged," *2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, 2016.

[7] R. Kasthuri, B. Nivetha, S. Shabana, M. Veluchamy, and S. Sivakumar, "Smart device for visually impaired people," *2017 Third International Conference on Science Technology Engineering Management (ICONSTEM)*, 2017.

[8] A. Pardasani, P. N. Indi, S. Banerjee, A. Kamal, and V. Garg, "Smart assistive navigation devices for visually impaired people," *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, 2019.

Acknowledgement

Success of a project like this involving high technical expertise, patience and massive support of guides, is possible when team members work together. We take this opportunity to express our gratitude to those who have been instrumental in the successful completion of this project. We would like to show our appreciation to **Mrs. Shagufta Rajguru** for their tremendous support and help, without them this project would have reached nowhere. We would also like to thank our project coordinator **Mrs. Rakhi Kalantri** for providing us with regular inputs about documentation and project timeline. A big thanks to our HOD **Dr. Lata Ragha** for all the encouragement given to our team. We would also like to thank our principal, **Dr. S. M. Khot**, and our college, **Fr. C. Rodrigues Institute of Technology, Vashi**, for giving us the opportunity and the environment to learn and grow.

Project Group Members:

1. Freddy Poly, 101616

2. Dipak Tiwari, 101658

3. Varghese Jacob, 101659

Appendix A : Timeline Chart

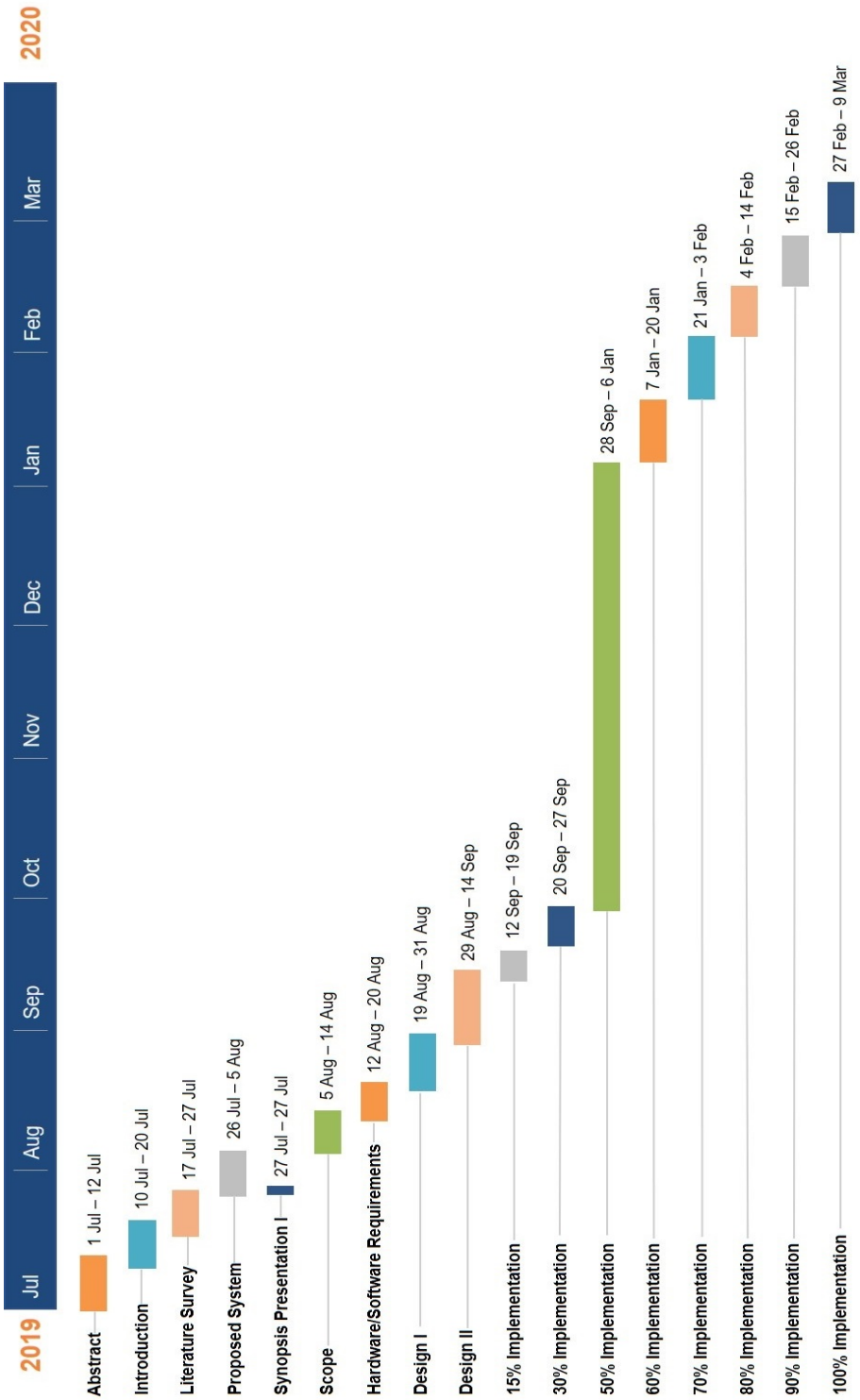


Figure 6.1: **Timeline Chart**

Appendix B : Publication Details

"VI Spectacle – A Visual Aid for the Visually Impaired", *International Research Journal of Engineering and Technology (IRJET)*, Volume 07, Issue 02, February 2020, S.No.:241

https://www.irjet.net/archives/V7/i2/IRJET-V7I2241.pdf