Kingdom of Saudi Arabia
Ministry of Education
King Faisal University
College of Computer Sciences & Information Technology

Smart Vest System for Blind

Milestone 5

A project submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science

By

Norah Rashed Almarri (217041639)

Sumyah Mohammed Rageh (217044905)

Elaf Mohammad Al hassan (217141273)

Supervised by

Dr. Rawabi Alsudais

Committee Members

Ms. Sarah Aldossary

November, 2021

ACKNOWLEDGEMENT

First and foremost, we would like to thank God Almighty for giving us the strength, knowledge, ability and opportunity to undertake this project. Without his blessings, this achievement would not have been possible. Special thanks and prayers to our supervisor Dr. Rawabi Alsudais who gave us the golden opportunity to do this wonderful project and helped us throughout this tough semester. Moreover, special thanks and gratitude for the committee members, all KFU management and staff members who made it possible for us to continue our education online during this quarantine period. Finally, we would also like to thank our parents and friends who kept encouraging us all the time. May God bless you all.

UNDERTAKING

This is to declare that the project entitled "**Smart Vest for Blind**" is an original work done by undersigned, in partial fulfillment of the requirements for the degree "Bachelor of Science in Computer Science" at, College of Computer Sciences and Information Technology, King Faisal University.

All the analysis, design and system development have been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or university.

Student Name	Student ID	Signature
1. Norah Rashed Almarri	217041639	
2. Sumyah Mohammed Rageh	217044905	
3. Elaf Mohammad Al hassan	217141273	

ABSTRACT

There are millions of blind people around the world who can’t live satisfying lives because of their disability. Vision impairment is a serious problem because it happens to people of all ages, and both males and females. Many software, and hardware companies contributed in making tools that assist the blind people in their daily life. For graduation project, we chose to develop an assisting tool that can significantly improve blind people’s lives, avoiding bumping accidents while they walking. This tool is a smart vest system that uses ultrasonic sensor to detect the presence of objects and uses image processing to recognize the detected object. The object included in this system are (car, tree, wall). Recognition results will be given as audio output that can be heard by headphones. The most crucial part of the hardware is the raspberry pi, that is going to be powered by a power bank, and the rest of sensors are going to be attached to it. All the electronic hardware will be stitched to a fabric vest that the blind person can wear and Python programming language that will implement the object detection and recognition. The designed system aim to produce affordable and as comfortable as possible system for blind people.

Table of contents

List of Figures vii

1 Introduction 1

1.1 Background of the Project 1

1.2 Motivation for the Project 2

1.3 Problem Statement 2

2 The Scope of The Project 2

3 Project Requirements 3

3.1 Functional Requirements 4

3.2 Non-Functional Requirements 4

3.3 Hardware Requirements 4

3.4 Software Requirement 5

4 Methodology 5

5 Expected Outcomes 6

6 Related Work 7

6.1 Multifunctional Device for Blind People 7

6.2 Smart Glasses for the Visually Impaired People 10

7 Solutions Identification & Justification 11

7.1 Image processing Algorithms 12

7.2 Microcontroller 13

7.3 Programing Language 14

7.4 Holders 16

7.5 Distance Measurements Sensors 17

8 System Appropriate Analysis 18

8.1 Overall System Flowchart 18

8.2 System Use Case Diagram 19

8.3 Activity Diagram 20

9 Proposed Design 21

10 Tools and Techniques Used Proposal Phase 24

11 Tools and Techniques Used During Project Implementation 25

12 Work Plan for the Implementation Phase 28

13 Project Implementation 29

13.1 Implementation Changes on The Project Proposal 29

13.2 Preparation of the Work Environments 30

14 Tools and techniques used during project implementation 35

15 Overall project Outcomes 38
16 Implementation Details 39
17 Analysis of Overall Result 47
17.1 Ultrasonic sensor function(input) 47
17.2 buzzer(output) 47
17.3 ON/OFF 47
17.4 OS Raspberry pi 47
17.5 Connecting the ultrasonic sensor 48
17.6 Connecting buzzer 48
17.7 Connecting vibration motor 49
17.8 Connecting USB Camera 49
17.9 Object Detection 50
17.10 Text-to-Speech 50
17.11 Smart Vest Design 50
18 Comprehensive Remarks on Overall Project Outcome and Achievements 50
19 Project Enhancement and Future Work 52
20 Conclusion 53
References 55
21 Appendix 58
21.1 The source codes. 58

List of Tables

Table 1:comparison among different systems with specific criteria 11
Table 2:Comparstion between Arduino Uno and Rasppberry Pi....................................... 14
Table 3: Java, C++, and Python comparison.. 16
Table 4:Holders Comparison table 16
Table 5:Distance Sensors Comparsion.. 17
Table 6: Work plan for implementation phase next semester 28
Table 7: Implementation changes on the project proposal 30

List of Figures

Figure 1: Expected outcomes 6
Figure 2:Smart Vest for Blind 7
Figure 3: Ultrasonic Sensor....8
Figure 4:Working steps for ultrasonic sensor....8
Figure 5: Flow chart for obstacle detection....9
Figure 6: Overall System Flowchart....19
Figure 7: Use Case diagram for Smart Vest System 20
Figure ٨ : Activity Diagram 21
Figure 9: System parts Connection....22
Figure 10: Outer component of the vest 22
Figure 11: Vest turned ON....23
Figure 12: Object detected 23
Figure 13: Object recognition....24
Figure 14: Vest System (battery is low) 24
Figure ١٥ : Raspberry Pi Imager interface....31
Figure ١٦ : VNC server setup....32
Figure 17: VNC server installing 33
Figure 18: VNC Viewer 34
Figure ١٩ : Illustrate how raspberry pi and viewer connected 34
Figure٢٠ : Raspberry pi screen from desktop 35
Figure 21: First circuit connection 39
Figure 22: Python code for first circuit 40
Figure 23: System with vest connection....41
Figure 24: System with vest connection (in back) 42
Figure 25: The yellow protective sticker....42
Figure 26: Vest after power bank added....43
Figure 27: The push button 44
Figure 28: switch from child's toy....44
Figure 29: The output of USB camera 45
Figure 30: The tablet app used....46
Figure 31: The command in ConnectBot....46
Figure 32: Connection of the ultrasonic sensor pins to the raspberry pi 48
Figure 33: Circuit diagram of the buzzer....49
Figure 34: Circuit diagram of the ultrasonic sensor and the buzzer....49

1 Introduction

According to the World Health Organization, the number of visually impaired people of all ages is estimated to be 285 million, of whom 39 million are blind.[1] Most of these people face daily struggles bumping into walls, people, and get into car accidents! For the sake of helping them, we have decided to develop a smart vest that would help millions of blind people around the world. We aim to make the vest comfortable and as affordable as possible. The vest has three embedded systems coded in python. The first system is an anti-bumping assistant. the second is and third-eye assistant recognizes the presence of humans, cars, and walls. The third is a battery-level alert. This chapter begins by presenting the background of the project and explaining our motivation to make this project. Then, it will discuss the problem statement, provide the methodology, show the scope of the project and the system analysis. Furthermore, this chapter will include the user requirements, define which language will be used, show the project baseline requirement, and report outline.

1.1 Background of the Project

Walking without being aware of the obstacles in front of the person is extremely a tough experience. It is similar to how dangerous it is walk eyes are closed, or how scary it is to walk in a completely dark room. Blind people have a higher chance of getting injured as a result of bumping into objects. This is why many of the blind people have to depend on family members or friends to guide them everywhere. Doing this project, we hope to make blind people a little bit more independent.

Smart vest system aims to detect the presence of the objects around the blind person wearing the smart vest to help them being independent for a certain period of time. In addition, the loud pronouncement of the detected object can reduce the risk of road accidents. Furthermore, the smart vest designed to be easy to use as there is no additional help needed to wear it.

1.2 Motivation for the Project

As computer science students, and after we have three years in college already, we came to know how modern technology can help in solving many of the problems we have today. Given the fact that us, the team members have, or have had blind family members, relatives, and friends, we saw closely how tough their lives were. Seeing a blind person is of course heartbreaking, but what is more heartbreaking is seeing an injured blind person. Sadly, many of the blind people experience serious bone fracture injuries at some point in their lives due to crashing into objects in front of them when nobody was there to guide them. Doing this project, we dedicate all our knowledge, time, and effort to reduce the suffering of our blind loved ones.

1.3 Problem Statement

Although modern technology has found good solutions for many kinds of physical disabilities including artificial limbs, the topic of blind people didn't get enough attention. while amputee, deaf, and mute people have found hope, Blind people are still struggling in the dark. It is important to fix the problem because it impacts the lives of many people of all ages and nationalities. Anyone, and at any day could lose the sense of sight in an accident, that's why we should use technology to solve this. There are sure some inventions out there, but they are extremely expensive and most people can't afford them, that's why we need to come up with an affordable alternative solution.

2 The Scope of The Project

The scope of this project is to develop a system that can help blind people recognize objects such as walls, humans, and cars around them and know the estimated distance to avoid any collision between users and objects. In this system, it will contain three embedded systems which are a battery-level system, an anti-bump system, and a mini-eye system. In the battery-level system, it will use for ON/OFF the embedded system and it will give an alarm when the battery is low via voice.

In the mini-eye system, we will use a camera installed on the front of the vest to take a picture of objects and detect these objects using Faster R-CNN algorithm which is better in localize and recognize the object or any kind of image classification algorithms [2]. Next, we will use the result of the last action to convert the object name from text to speech using the API in Python so that the user can identify the name of the object whose image was taken and that we will do by datasets only for humans, cars, and walls pictures.

On the other hand, we will use the anti-bump system to measure the exact distance between the user and the objects in front of them to avoid an accident by informing the user over the vibration motor that an object is close. In that action, the distance sensor will give an alert when the user closes about one meter to the object. It will work first before the mini-eye system to open the camera and take a picture rather than let the camera work all the time which will consume more power and battery unnecessarily.

3 Project Requirements

In this section, we performed a brief analysis to understand and illustrate the the smart vest system requirements. This analysis is a blueprint to be followed in the development phase.

3.1 Functional Requirements

- The system shall be able to recognize the objects from pictures.
- The system shall be able to tell what the object is via voice.
- The system shall be able to measure the estimate distance between the user and the objects.
- The system shall be able to give alarm when the battery is low.
- The system shall be able to vibrate when there is an object close.

3.2 Non-Functional Requirements

- Useability: The system is easy to use.
- Learnability: new users should learn how to use in from the first time.
- Maintainability: sensors and batteries should be maintainable.

3.3 Hardware Requirements

- Microcontroller (Raspberry Pi)
- USB Camera
- Ultrasonic HCSR04 Sensor
- Earphone
- Jumper Wires
- Breadboard
- Vibration Sensor
- Power Bank
- Power Button (USB-C In-Line Power Switch Cable for Raspberry Pi)
- Vest
- microSD card
- Laptop

3.4 Software Requirement

- Image processing technique
- Text to Speech technique
- Jupyter notebook
- Python Programming Language
- Python packages such as OpenCV, sci-kit image, matplotlib.
- VNC
- Raspberry Pi imager
- NOOB

4 Methodology

The proposed system consists of three embedded systems. The first system is the battery-level system. This system has the following four functionalities: turn the vest on, turn the vest off, watch the battery level while the vest is on, and give an alert when the battery is low. The second system is the anti-bump system. The main hardware piece is the ultrasonic device. It turns on automatically when the first circuit (battery monitoring system) turns on. It has the following functionalities: measure distance between the wearer and obstacles such as walls and give vibration feedback before bumping into obstacles. The last system is the mini-eye system. The main hardware is the camera, and the main process is image processing. It takes images as input, then gives audio output saying if there is human, wall, or car. All the hardware pieces are attached to the fabric vest by stitching, for heavy devices, and by gluing for lighter ones.

5 Expected Outcomes

There are useful expected outcomes from designing this system. Figure **1** below shows the expected outcomes of our system. Figure 2 shows our initial perception of the vest.

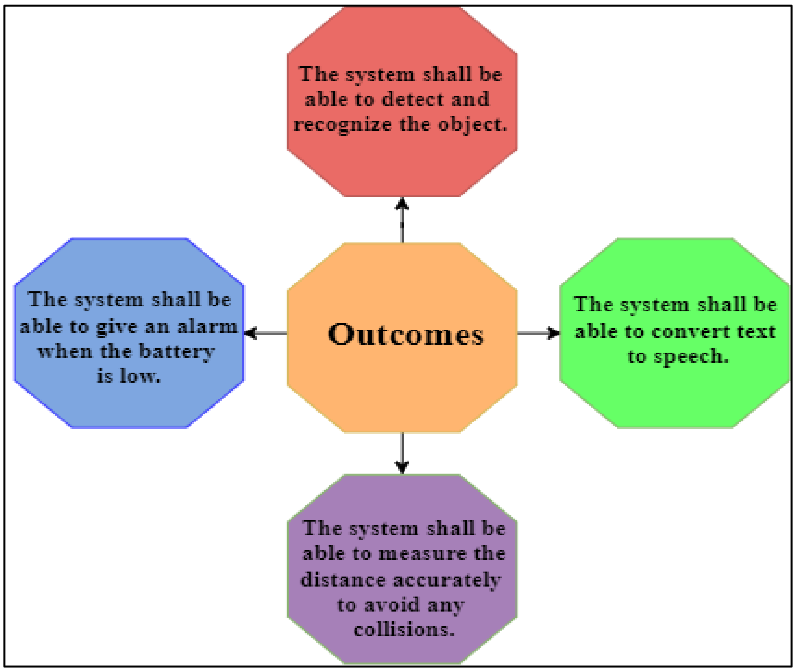

Figure 1: Expected outcomes

Figure 2:Smart Vest for Blind

6 Related Work

This section discusses number of the projects that are related to our project Smart Vest for Blind. The first project is Multifunctional Device for Blind People which implements many functions, and we focus on the most related functions to our project which are obstacle detection and image to speech conversion. The second project is Smart Glasses for the Visually Impaired People.

6.1 Multifunctional Device for Blind People

This project is a smart system to guide the blind which is designed to improve the movement of the blind. The project is about a wearable device with earphones that are connected to raspberry-pi to implement some functions such as face recognition, image to speech conversion, obstacle detection and currency detection to help the blind to move anywhere without needing any help from others. This project used two algorithms which are Hear for image processing and ANN to perform the functions [4].

- **Obstacle Detection Using Ultrasonic HCSR04 Sensor**

The Ultrasonic HCSR04 consists of two parts to detect the object. One part is called Chirp and the second part is called Echo. Chirp emits a short ultrasonic burst and Echo will receive and listen to the ultrasonic as shown in Figure 3. The Ultrasonic sensor is controlled by the microcontroller (trigger pulse) and the sensor emits Ultrasonic which is about 40KHz burst. The burst moves through the air which is about 1130 feet per second. When the burst hits any objects, it will defect back to the sensor. The Ultrasonic sensor produces an output pulse to the microcontroller which will finish when the Echo is detected and the distance to the target match width of this pulse. This output will be converted into an audio signal which will produce as a sound to the blind via earphones [5].

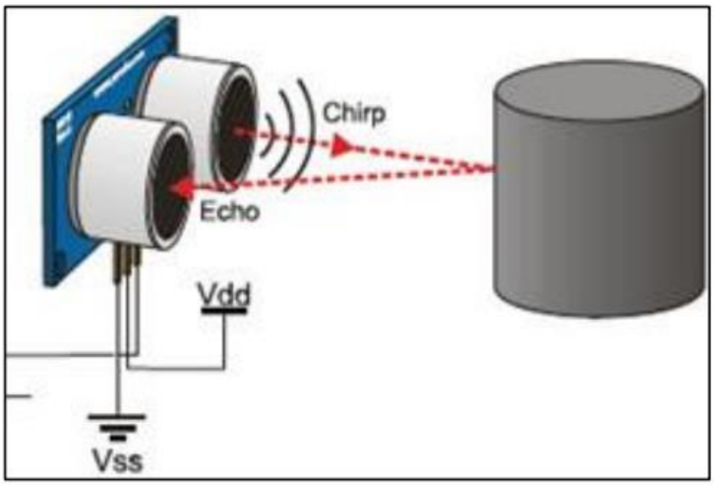

Figure 3: Ultrasonic Sensor

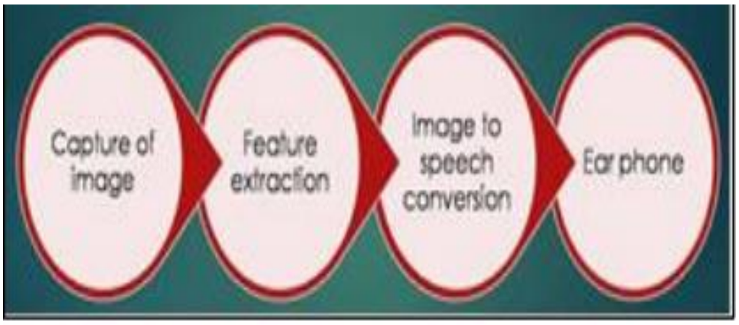

Figure 4:Working steps for ultrasonic sensor

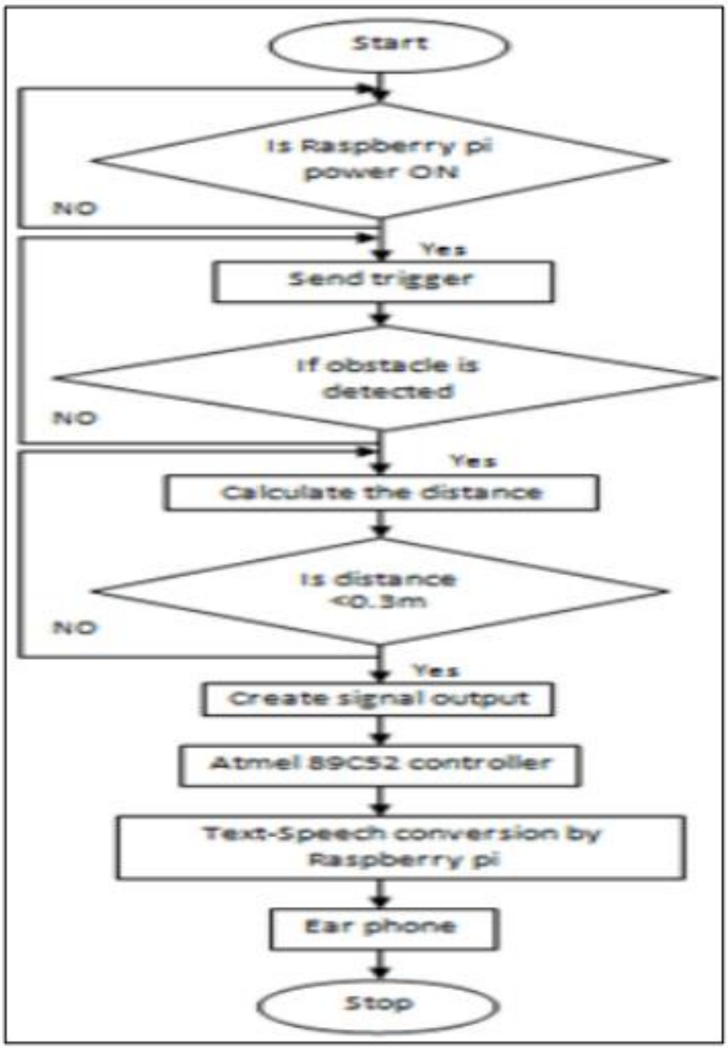

Figure 5: Flow chart for obstacle detection

- **Image to Speech Conversion**

This function helps the blind to know what is written on objects such as paper. The blind will put the camera in front of the text and choose a specific button for the keypad. The camera will capture the text as an image by webcam. Then, the image processing will convert the image into text. Next, the eSpeak algorithm will convert the text into an audio signal which is produced as an output to the blind's earphones. The flow chart of the image to speech conversion is shown in Figure 5 [5].

- **eSpeak Algorithm**

This algorithm is open-source software that includes many languages. For the English language, eSpeak provides a good quality speech. It includes different methods from other open-source such as sounds quite different and TTS engines. The process is performed as a command-line program that takes the text from an input or a file. The eSpeak algorithm can be used to produce the output of the speech as a WAV file. Also, it supports SSML (Speech Synthesis Markup Language) and HTML file. For languages, it has many languages which are about 1 Mbyte. Furthermore, it can convert text to sound codes. The code can be written in C++ programming language [5].

6.2 Smart Glasses for the Visually Impaired People

This project helps impaired people to recognize the text correctly by wearing the smart glass. The idea of this project is to take instructions via switches and produce the output as a sound by earphones. The task will start when the user pushes the button. For the text recognition mode, the glass will check the state of the text if it is readable and positioned correctly will accept it. Otherwise, it rejects the image and asks the user to fix the orientation of the image or retake the image again. When the image is accepted, the view will be processed in real-time to send the image to an optical character recognition (OCR) software to take text and convert it into speech by the text-to-speech synthesizer. Then, the sound will be produced as an output [6].

Image processing will be done by using Simulink (MathWorks, MA, Natick). For the reading mode, there are some challenges related to the image such as orientation, quality and text position of the image. So, it must focus first on the orientation of the frame and red borders. The image will be enhanced by using noise filtering, contrast improvement and morphological operations. Tesseract OCR engine is applied to extract the text before converting the text into the audio output [6].

Table 1:comparison among different systems with specific criteria

Criteria	Multifunctional Device for Blind People	Smart Glasses for the Visually Impaired People	Smart Vest for Blind (proposed project)
Support multiple functions	Yes	Yes	Yes
Recognize different objects	Just Faces	No	Yes
Give an alarm to avoid collisions	Yes	No	Yes
Use text-to-speech technique	Yes	Yes	Yes
Use image processing technique	Yes	Yes	Yes
Give an alarm if the battery is low	No	No	Yes

Therefore, our proposed project Smart Vest for Blind is better because it supports the most important functions for the blind people such as detecting different objects, avoid collisions and give an alarm when the battery is low.

7 Solutions Identification & Justification

In this section, we have listed the possible hardware and software solutions for implementing Smart Vest system. The section contains five main subsections. The first section presents image processing algorithms. The second section states Microcontrollers types. The third section lists the programming languages that may be applied for the system. The fourth section discuss the holders that can be used and justifies the reasons of selecting the vest as a system holder. The last section shows different types of distance sensors.

7.1 Image processing Algorithms

There is different algorithm that used for image processing based on their usage and distinct attributes. In our project, we will use Faster Region-Based Convolutional Neural Networks (R-CNN) because it is fast and showed more accurate results comparing with other discussed techniques.

- **Real-time Human Detection with OpenCV**

This algorithm depends on video and it is more accurate and is used in stores such as the gold store because takes real-time input and it works 24 hours to support the security [3]. We didn't take this algorithm as the first option because the camera will take the only picture when it receives the signal from the anti-bump system it will not work for 24 hours. In case the selected algorithm not working, we will use this algorithm as an alternative solution.

- **Mask R-CNN**

This algorithm is used to segment the image and gives three outputs for each object in the image: its class, bounding box coordinates, and object mask. The advantages of using this algorithm as an alternative solution are simple, flexible, and general approach. The only disadvantage is taken a higher time to the training dataset [4].

- **Faster R-CNN**

It is one from the R-CNN family and it was published in 2015. The only difference between Faster R-CNN and it is family is using "Region Proposal Network" rather than selective search which makes the algorithm slower. This algorithm depends on how the previous system performed [16]. This algorithm is the selected one for our project because it is fast and showed more accurate results comparing with other discussed techniques.

7.2 Microcontroller

Microcontrollers are embedded inside devices to control the activities and highlights of an item. It is used in products and devices that are automatically controlled such as remote control of TV [7,8]. We will choose the best type of microcontroller that is appropriate for our project.

- **Arduino Uno**

Arduino Uno is a more simplistic approach and it may attract many people when they want to make a project. It has a simple time interface with the components like motors. This type of microcontroller is simple and used for simple projects that need quickly fetch the data from the sensors and do one activity from this data.[٩]

- **Raspberry Pi 3**

Raspberry Pi 3 is Single Board Computer (SBC) which is mean that the board is a completely useful PC with its devoted processor, memory, and it can run on OS (Operating System). Also, the Raspberry Pi 3 is faster and has many ports such as a Camera port, HDMI port, audio port, and LCD port which help in projects that need many actions [9]. It has advantages and disadvantages we will show it in the table including Arduino Uno advantages and disadvantages.

Table 2:Comparstion between Arduino Uno and Raspberry Pi

	Raspberry Pi 3	Arduino Uno
Pros	**Wi-Fi and Bluetooth capability.** **Stronger and faster processor.** **Great for projects that need to connect online and have multiple tasks to be done at the same time.**	**No need for long set up.** **Great for projects that do not need to accomplish multitasks in one time.**
Cons	**Complex.** **Long set up.**	**No Internet connectivity out of the box.** **Run one code at time.**

As we mention above there are pros and cons for each type of microcontroller shows in Table 1 and depend on that we will choose the best type that is suitable for our project and needs.

We will choose Raspberry Pi 3 as our microcontroller because in our project we need Internet connectivity and multitask processes done at the same time in order to achieve high efficiency.

7.3 Programing Language

It is very important to choose suitable programming language when you are going to start a project. In this section, we presents and compares different types of programming languages. A justification is followed for the selected programming language.

- **Java**

Java is a famous programming language that owned by Oracle and around 3 million devices run Java[10]. This language has many features which are Object-Oriented (OOP), secure, simple, Robust, and high performance. Learning this language is not difficult and it is easy to read and write [11]. In the last of this section, you can find Table 2 for comparison between Java and other programming languages.

- **C++**

C++ is a cross-platform language that can be utilized to make efficiency applications and give the programmer a high level of control over the system. Also, it close or similar to Java and C# languages so it can be easy to switch to C++ [12]. At the end of this section the Table 2 shows the features of C++ comparing to other programming languages.

- **Python**

Python is also a popular programming language and can be used in different platforms such as Windows, Linux, and Raspberry Pi. The language is widely common used for image processing and it has high number of image processes packages[13]. The Tables 2 shows the difference between languages comparing to Python language.

Table 3: Java, C++, and Python comparison

Java	C++	Python
Compiled Programming language.	Compiled Programming language.	Interpreted Programming language.
Has in build multithreading support.	Does not support threads.	Support multithreading.
Has library support for many concepts like UI.	Has limited number of library support.	Has a huge set of libraries that make it fit for AI.
Platform Independent.	Platform dependent.	Platform Independent.

Python have been selected as our programming language because it has features that appropriate for our project such as supporting multithreading and the libraries that will help us when we work with AI to achieve the functions that we decided.

7.4 Holders

There are many holders we can pick up and implement our idea on it, but we decide on the vest for many reasons and Table 3 shows the criteria that we put to measure the suitable holder.

Table 4:Holders Comparison table

Criteria	Eyeglasses	Hat	Vest
Comfortable	No	No	Yes
Handle many devices (like camera, ultrasonic, etc.)	No	No	Yes
Safety	No	It depends on how many devices and which	Yes
Easy to wear	Yes	Yes	Yes

As Table 3 shows, the vest can be worn comfortably with no problem with how many devices will be connected. Also, in case of damage for any of the devices that we will use, it is safe more than eyeglass or hat because it is not close to the eyes or head.

7.5 Distance Measurements Sensors

Distance sensors are used to measure the approximate distance of an object without the need for any physical contact[14]. In this section, we will briefly explain two type and at the end, we will choose one type with justification for our selection.

- **Ultrasonic Sensor**

  It is a common type and is used to detect the distance to objects by emanating high-recurrence ultrasonic waves. It is used in smart cars and distance measurements [14]. In the last section there is a table for present the pros and cons of this sensor.

- **IR distance sensor**

  It measures the approximate distance via sensing emitting IR beam and calculates the reflection of angle. It used in laptop, TV, and computers [14]. In the Table 4, there is a comparison prose and cons of Ultrasonic and IR sensors.

Table 5:Distance Sensors Comparison

	Ultrasonic Sensor	IR distance sensor
Pros	• Consume lower power • Works well in dim places • Not affected by object color	• Able to measure complex surfaces • Secure communication
Cons	• Limited detection range • Slow refresh rate	• Limited measurement range • Affected by environment conditions.

We will choose Ultrasonic sensor type because we only need the distance sensor for specific reason which is measure the approximate distance in different places for long time and we can find the Ultrasonic sensor is the propriate one for our project.

8 System Appropriate Analysis

In this section, we will conduct what the system should do and we will show different system design either flowchart or diagrams in order to understand the system working and how can be used.

8.1 Overall System Flowchart

In Figure 6 we simply draw the overall system flowchart which is showing the basic operations and steps to use the system appropriately. First, when user wear the vest and turn on the power the following processes will work which are ultrasonic detect the near object, camera turn on and detect the objects. In ultra-sonic processes, it will work automatically after the user wear the vest and turn on the power as second circular because the first one is on/off power. If there is any object near to the user around one meter the ultrasonic will send signal to the third circular which is the camera to turn on. Second, after the camera turn on, it will detect the objects that in coco names context and put the accuracy percentage above each object detected. The flowchart present basic operations and we explained in details to be clear and more understandable.

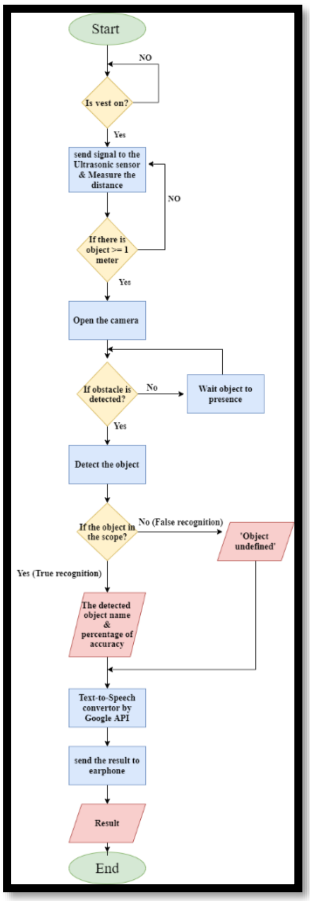

Figure 6: Overall System Flowchart

8.2 System Use Case Diagram

Use Case diagram shows the functional requirements of a system and it helps to trace each function within the system. Furthermore, it represents each function clearly which helps other people who is not a developer to understand the functions of the system easily. Figure 7 represents the Use Case diagram for the Smart Vest System for Blind.

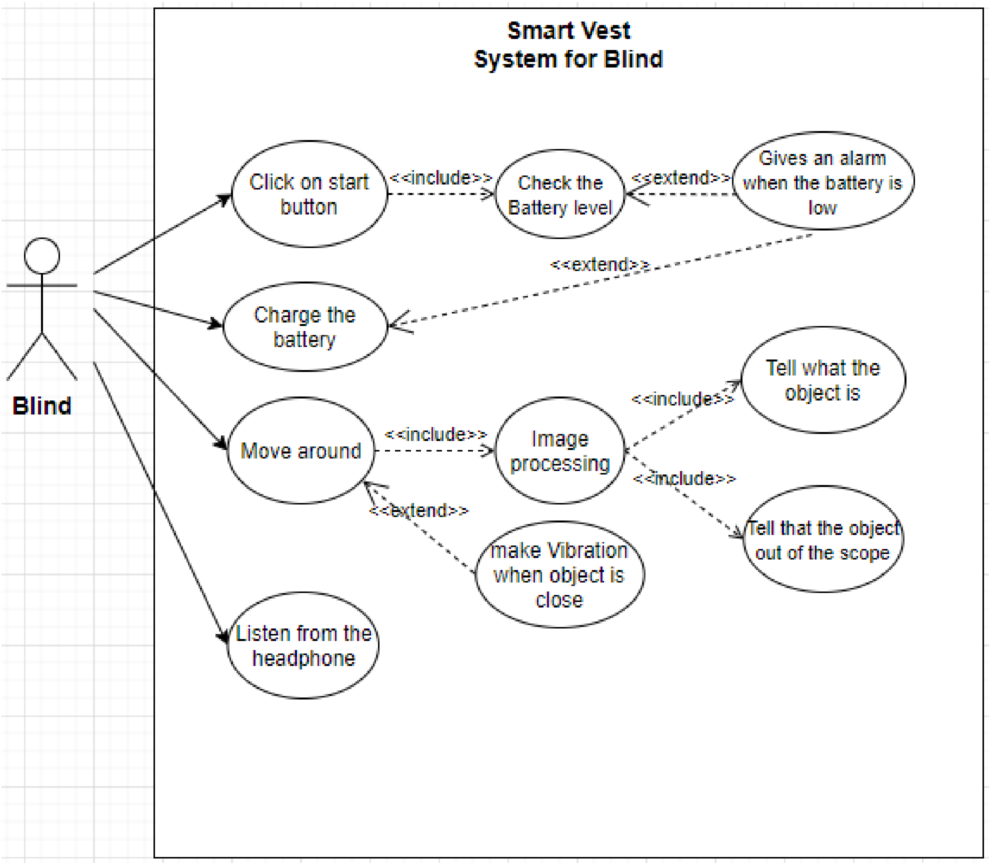

Figure 7: Use Case diagram for Smart Vest System

8.3 Activity Diagram

An activity diagram is a behavioral diagram in the UML diagram. It is used to describe the dynamic aspects of the system. An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Figure 8 shows the activity diagram for the smart vest.

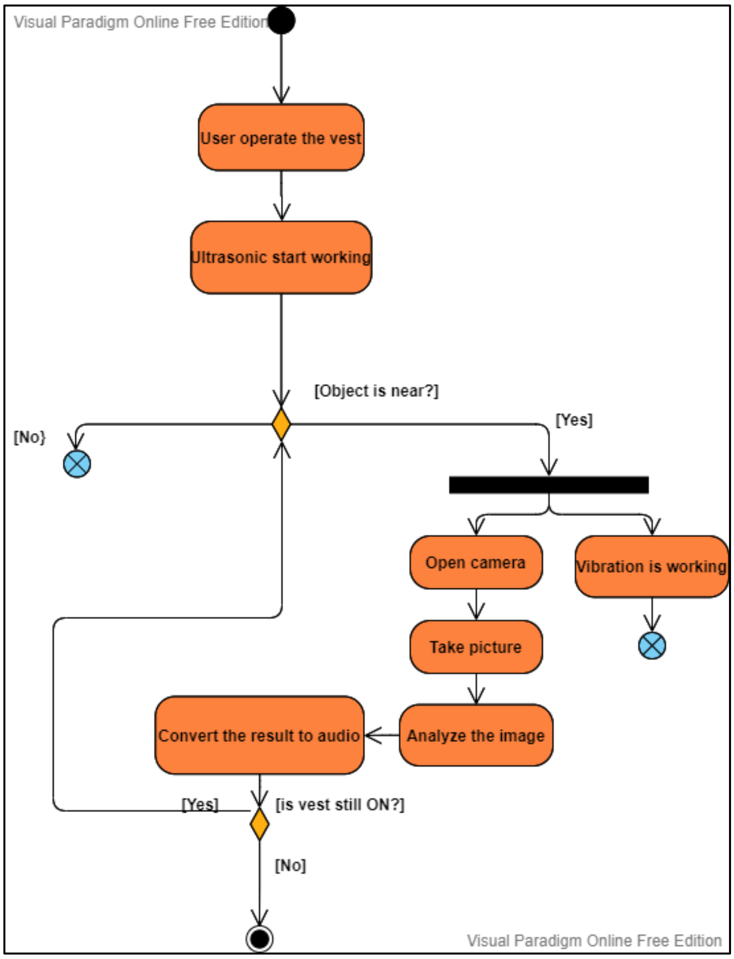

Figure ٨ : Activity Diagram

9 Proposed Design

In this section we will put a simple scenario that shows how can the user use our system and all other cases. First, we will present the connection circular of our project that we plan to do it in Figure 9.

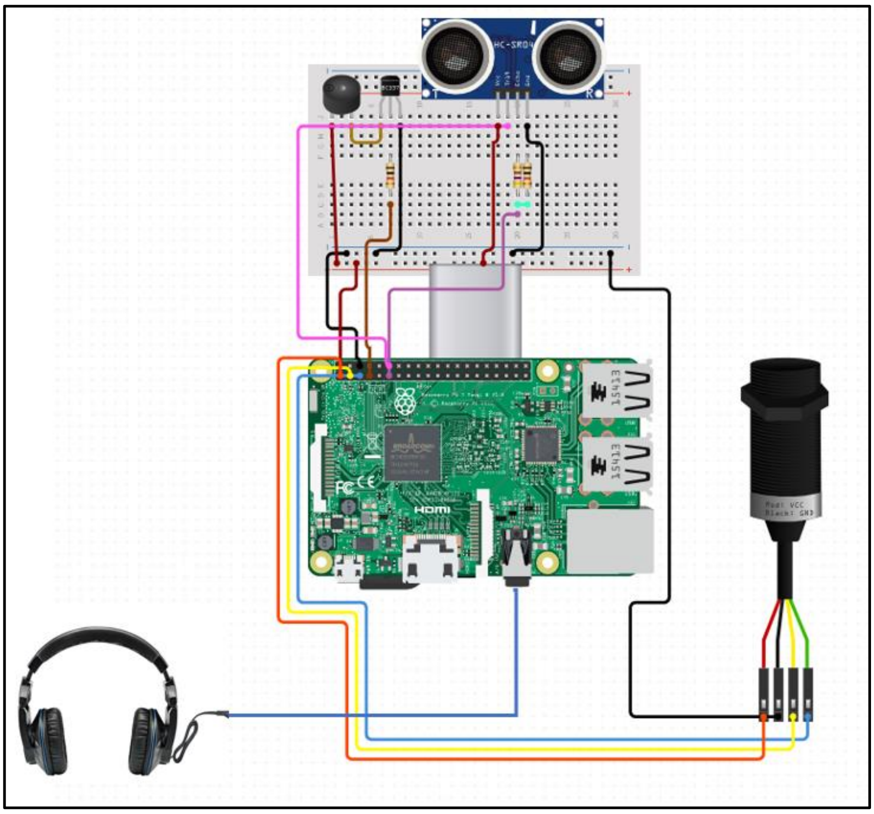

Figure 9: System parts Connection

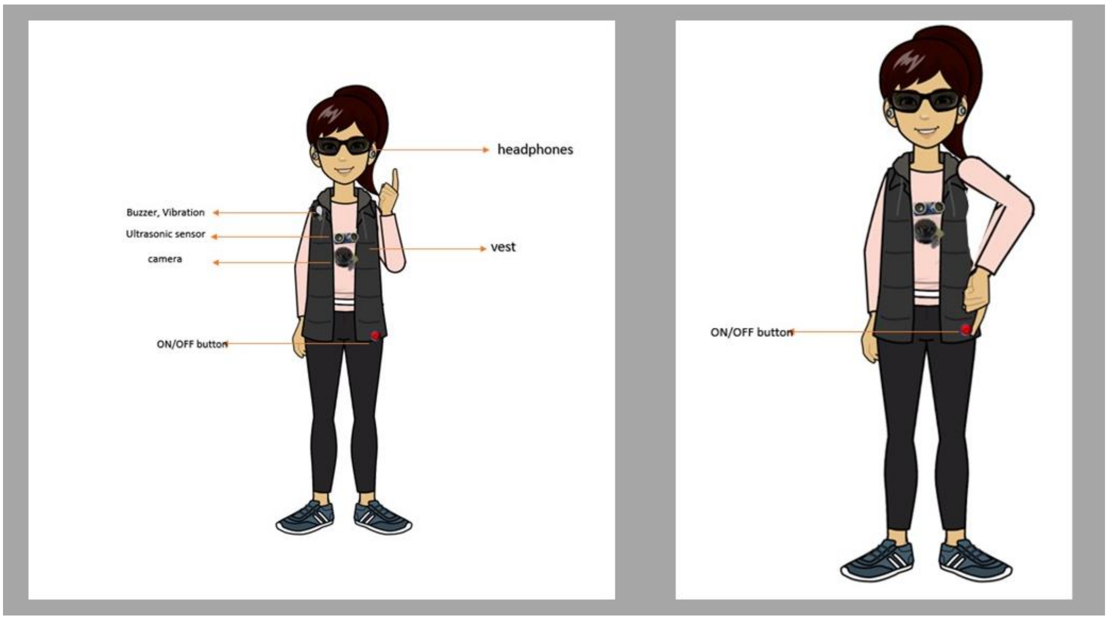

Figure 10: Outer component of the vest

The Figure 10. shows the outer components of the vest(it shows the visible parts from outside, there are other functional parts stitched in the vest from the inside). The vest should be zipped closed and the camera and ultrasonic are placed in the chest area on the surface of the vest. In the image it is not zipped closed because it was the only avatar we found. After the blind person wears the vest, it is turned on by a button as shown in the figure.

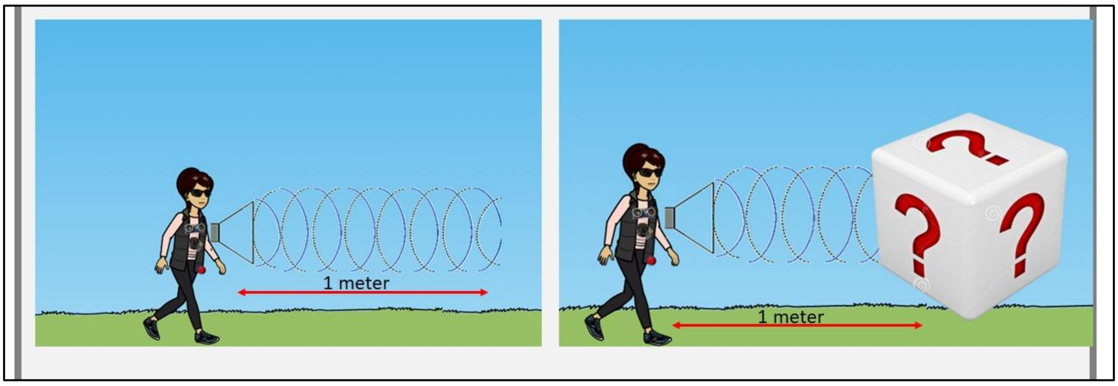

Figure 11: Vest turned ON

Once the vest is turned on, the ultrasonic sensor is turned on and starts scanning. The person is walking in Figure 11. An object faces her that is one meter close. The presence of object is detected by the ultrasonic sensor.

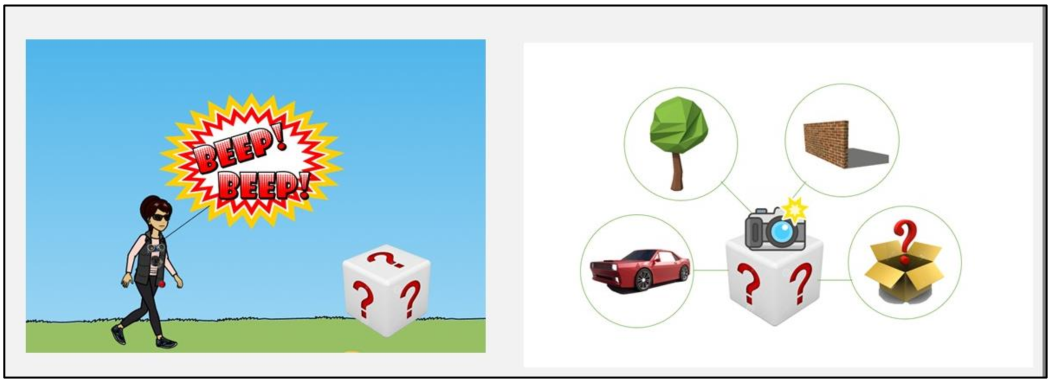

Figure 12: Object detected

The buzzer turns on. The person becomes aware that there is an object in front of her. The camera turns on and captures a picture of the object as showing in Figure 12. Image processing is done to detect the type of object. It could be one of the defined objects (tree, car, wall) or something else unrecognized by the vest.

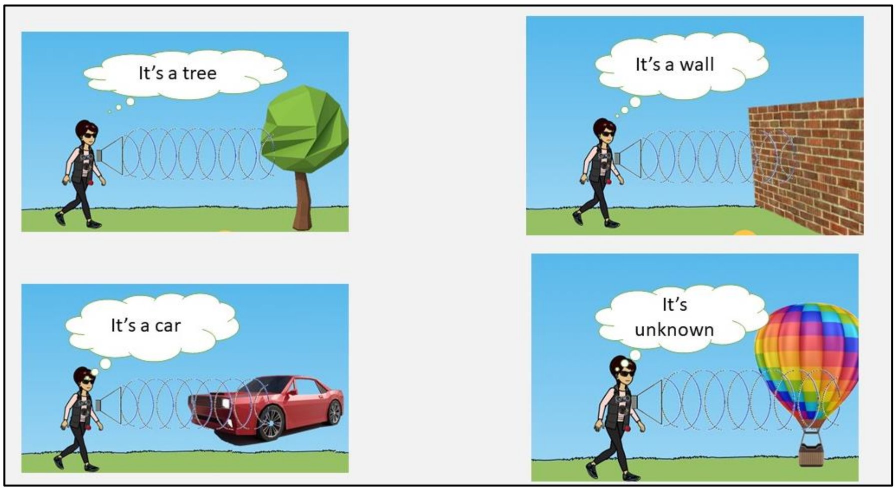

Figure 13: Object recognition

If the object falls to the predefined criteria, the wearer will get audio feedback in headphones saying the name of the object in English sentences. For example, "it is a car". Otherwise, the wearer will hear the sentence "it's unknown" as showing in Figure 13.

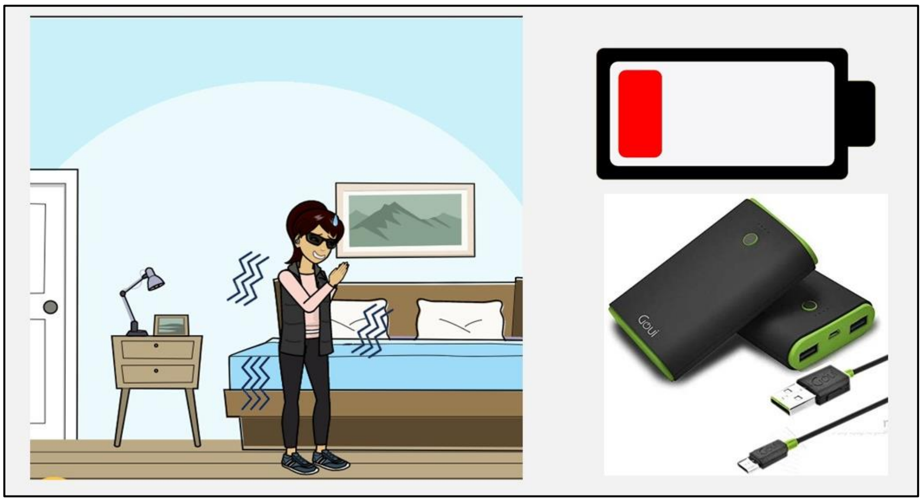
Figure 14: Vest System (battery is low)

In Figure 14, when the battery level is low, 15% left, the wearer gets notification. The person then must recharge the power bank.

10 Tools and Techniques Used Proposal Phase

- **Untitled Diagram Draw IO**

it is a web site that helps to draw different kinds of diagrams such as Flowchart, Entity Relation, UML, and Use Case diagram. We used this web site to design the Use Case diagram and the flowchart for our project.

- **Circuito.io**

  It is a web site that helps to design the hardware circuits. We used this tool to visualize the connection of the devices in our project.

- **Plan 365**

  It is a program that helps to make plan by dividing the time for each task. We used it to make the Gantt chart for our project which helps us to organize the working weeks and assign specific tasks for each week.

- **Canva**

  It is a web site that helps to design anything. We used this web site to design the Smart Vest.

11 Tools and Techniques Used During Project Implementation

This section is discussing all techniques that will be used during the project implementation.

- **Raspberry Pi 4 Model B**

  Raspberry Pi is a Microcontroller device that works as a small computer which contains CPUs, memory, and I/O peripherals. It will be used to connect all sensors and program them [15].

- **Buzzer Sensor**

  It is like a magnetic speaker. It needs different frequency to make the sound. Alarm clocks, machines, timers, and confirmation of user feedback such as a mouse click or keystroke all use buzzers [22].

- **Vibration Motor**

  It is a sensor that can be controlled and decided when it will generate the vibration. It used to send a notification to the user via vibration.

- **USB Camera**

  It is a small camera that will be used to capture the objects. It will be connected to the microcontroller to implement the image processing technique.

- **Ultrasonic HCSR04 Sensor**

  It is a sensor that measures the distance into the in front objects.

- **Light Emitting Diode (LED)**

  A light-emitting diode is a semiconductor light source that emits light when there is a current move through it. In this project, we used two different colors of led which are red and green.

- **Power bank (Anker)**

  It is a portable battery that will be used as a power supplier for the microcontroller which can be recharged when it is low.

- **Earphone**

  Earphone helps to move the sound from the microcontroller into the blind's ears. Any type of earphone can be used in our project.

- **Programming Language (Python)**

  Python is one of the modern programming languages which is object-oriented and high-level programming language that will be used to develop our system.

- **Jumper Wires**

These wires will be used to implement the connection for each device to the microcontroller. We used three types of Jumper wires which are male-to-male, female-to-female and female-to-male.

- **Breadboard**

  It is a thin plastic board used to hold electronic components. We will use the breadboard to make the electrical circuits for each sensor.

- **Faster R-CNN Algorithm**

  It is one of the most useful algorithms for image processing technique which will be used to detect the objects.

- **Python packages**

  When we develop the system using Python language, we need some packages in order to help us operate the system and doing the process of image correctly. We will present some basic packages which are:

  - OpenCV: this library is easy to use and read which can be used effectively in image processing [17].
  - Scikit- image: it is an effective and fast library used for image processing [17].
  - Mahotas: it has features to compute the 3D and 2D images and it can extract information from pictures using advanced image processing [17].

- **Virtual Network Computing (VNC)**

  It is a graphical desktop-sharing system that allows remote access and control between two devices via Buffer protocol. In this project, we downloaded the VNC server on the Raspberry and the VNC viewer on our computers in order to remotely access the Raspberry from our computers.

- **microSD card**

  It is a non-volatile memory card that we use to install the noob of the raspberry pi [18].

- **Raspberry Pi imager**

  A list of the latest versions of Raspbian, supported by the Raspberry Pi, is downloaded automatically from the Raspberry Pi Imager so that people don't worry if an outdated version of the operating system will happen by accident [20].

- **NOOBS**

  It is stands for New Out of Box which works as a simple operating system for the Raspberry Pi [21].

- **Laptop**

  A personal computer, we use it to download the Raspberry Pi OS and code the ultrasonic sensor.

12 Work Plan for the Implementation Phase

This section states the tasks that are planned to be implemented in order to accomplish the proposed graduation project. Tasks distribution through the 2nd semester is shown in Table 6. It also shows the duration and the expected date of each task.

Table 6: Work plan for implementation phase next semester

	The work plan of the project implementation		
	Task	**Start Week**	**End Week**
Mile stone	Start working on Milestone 4	**1**	**7**
	Find the details of the proposal implementation	**1**	**2**

	Collect the requirements	**1**	**3**
	Commands of tools and techniques being used during project implementation	**2**	**3**
	Learn Python language	**1**	**5**
	Analyze how to establish the 1st system circuit	**3**	**5**
	Make the 1st circuit with testing	**5**	**7**
	Edit milestone draft based on supervisor comments	**6**	**7**
	Submit the milestone 4 report	**7**	**7**
	Milestone 4 presentation	**8**	**8**
	Edit milestone 4 based on Committee Members comments	**8**	**9**
Milestone #5	Start working of Milestone 5	**10**	**15**
	Install raspberry pi platform and set up.	**10**	**11**
	Start programming for both circuits.	**10**	**11**
	Build anti-bump and mini eye circuits.	**10**	**14**
	Testing the system after each change in the system.	**11**	**14**
	Identify the overall project outcome/achievements	**11**	**12**
	Analysis of overall result	**12**	**15**
	Validate and test the whole project.	**13**	**14**
	Edit milestone 5 drafts based on supervisor comments	**14**	**15**
	Submit the milestone 5 report	**15**	**15**
	Milestone 5 presentation	**16**	**16**

13 Project Implementation

In this section, we have presented the project proposal changes with their justifications and details of the proposed system implementation for the first circuit which is the ultrasonic sensor circuit. This implementation includes preparing the work environment, drawing a sketch of the circuit, wiring the hardware components, and finally writing and running the python code.

13.1 Implementation Changes on The Project Proposal

In the previous stage of this project, the complete requirements' list was determined. However, during the implementation phase four changes have been made to the hardware and software requirements. The table 7 represents the changes during the implementation phase with their justifications.

Table 7: Implementation changes on the project proposal

No.	Changes	Justification
1	Raspberry Pi camera sensor to USB camera.	Raspberry Pi camera sensor has a short wire and we needed a camera that has a long wire to be more flexible. Furthermore, USB camera is more efficient for image processing procedures because it has higher resolution.
2	LED added to the circuit.	It has been added to indicate that the current is flowing throw the circuit.
3	YOLO algorithm used to detect the objects.	It has been used due to the limited time for training the database after semester become shorter. Also, this method is pre-trained.
4	Use coco names dataset.	It has a defined weight and don't need to train.

13.2 Preparation of the Work Environments

The project implementation is divided into two parts as shown in the work plan section 12. The preparation of the work plan environment of the first part consists of hardware and software. The hardware we needed was raspberry pi 4, laptop, router, 32 GB microSD card, microSD card reader, power bank for the raspberry pi 4 as we mention in section 3.3. The software apps we needed were raspberry pi imager, VNC viewer to run in the laptop, and VNC server to run in the raspberry pi as we mention in section 3.4.

➢ **Installing raspberry pi imager and write to the microSD card**

Raspberry pi imager is a desktop application that is used to install Raspberry Pi OS to a microSD card via a microSD card reader. Figure 15 shows the interface of the Raspberry Pi Imager.

Figure ١٥ : Raspberry Pi Imager interface

The first step is installing the raspberry pi imager from the official website. [1] Then, insert the microSD card into a microSD card reader and connect the reader to a laptop. After that, run the raspberry pi imager. For operating system we are selecting Raspbian, while for the microSD, the name of the microSD card should be visible in a drop-down menu to be selected. Lastly, after the button “write” is pressed, the microSD is a dedicated for running the operating system of raspberry pi and storing the code that will be written later. The Wi-Fi network ID and password are stored in the microSD card, in a file in the physical address “/etc/wpa_supplicant/wpa_supplicant.conf”. When the raspberry pi is powered, it connects automatically to the identified network in the configuration file that was mentioned above.

➢ **Installing REAL VNC viewer in the computer.**

VNC is a graphical desktop communication system that helps you to remotely monitor the desktop interface of one machine (running VNC Server) from another computer or mobile device (running VNC Viewer). VNC Viewer sends keyboard, mouse, and touch events to VNC Server, and VNC Server responds with screen notifications. The graphical interface of Raspberry Pi will be displayed on PC monitor or mobile device. It enables you to manipulate it as though you were operating on the Raspberry Pi itself. A free version of the app is available on the official website https://www.realvnc.com/en/connect/download/viewer/. After the extension is downloaded, the installation setup is guided by a wizard as shown in figure 16 and figure 17.

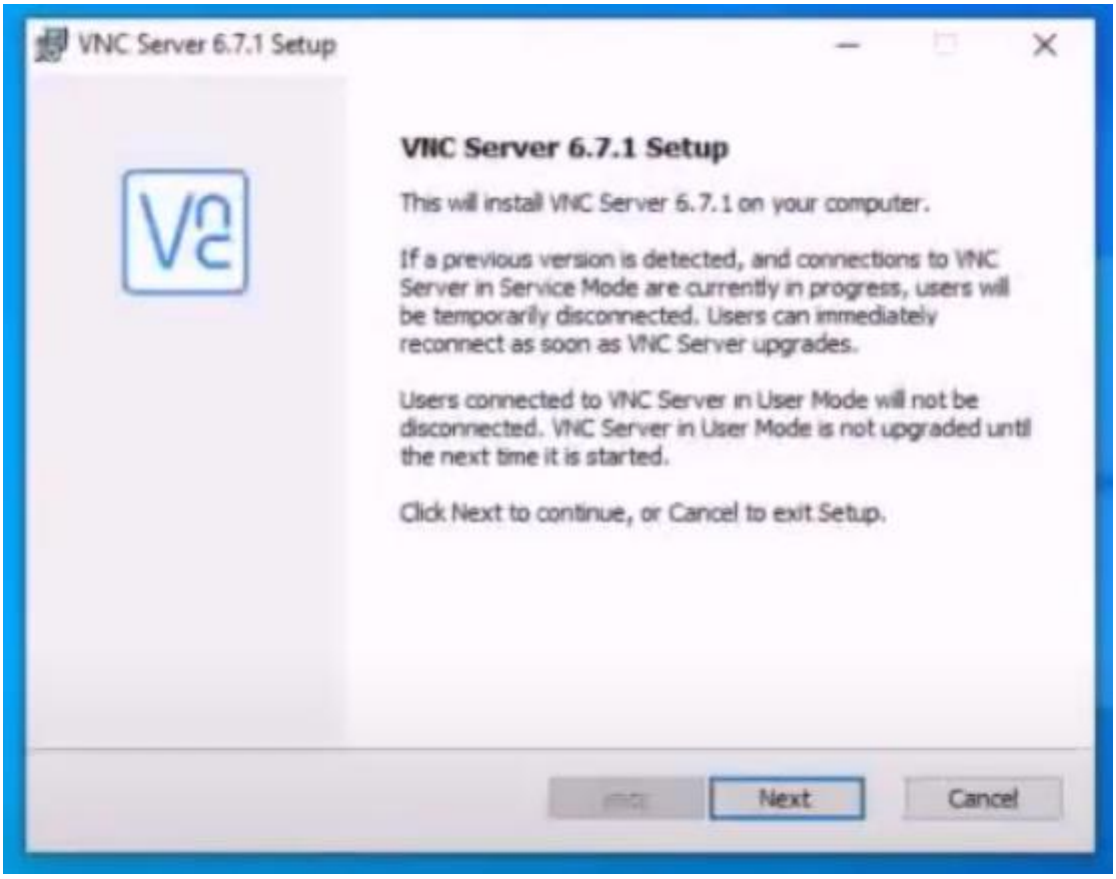

Figure ١٦ : VNC server setup

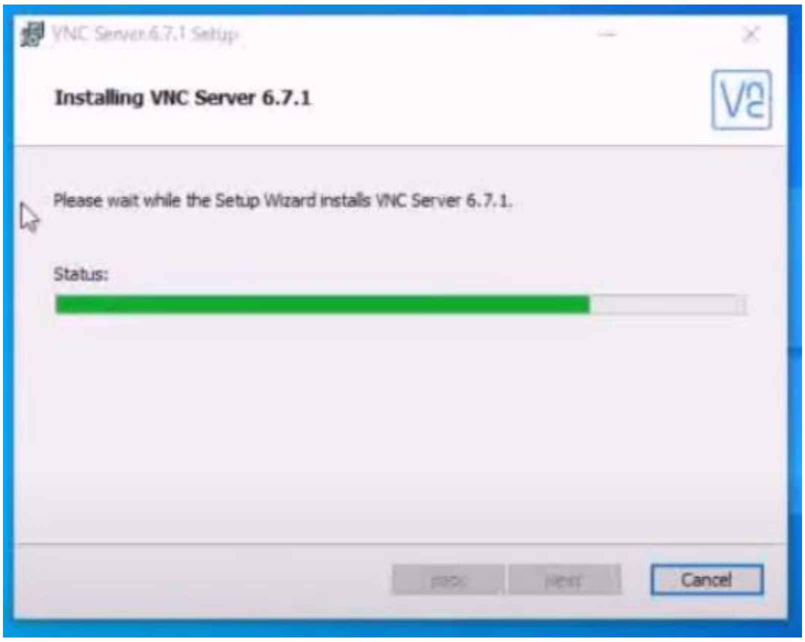

Figure 17: VNC server installing

- **Installing REAL VNC server in the raspberry pi**

In the command line:

```
sudo apt update
sudo apt install realvnc-vnc-server realvnc-vnc-viewer
```

Then:

```
sudo raspi-config
```

The VNC window will appear, plug in the raspberry pi into power, it was connected to the router. Then we opened the router phone app to see the IP of the raspberry pi. In the VNC viewer window, we put this IP.

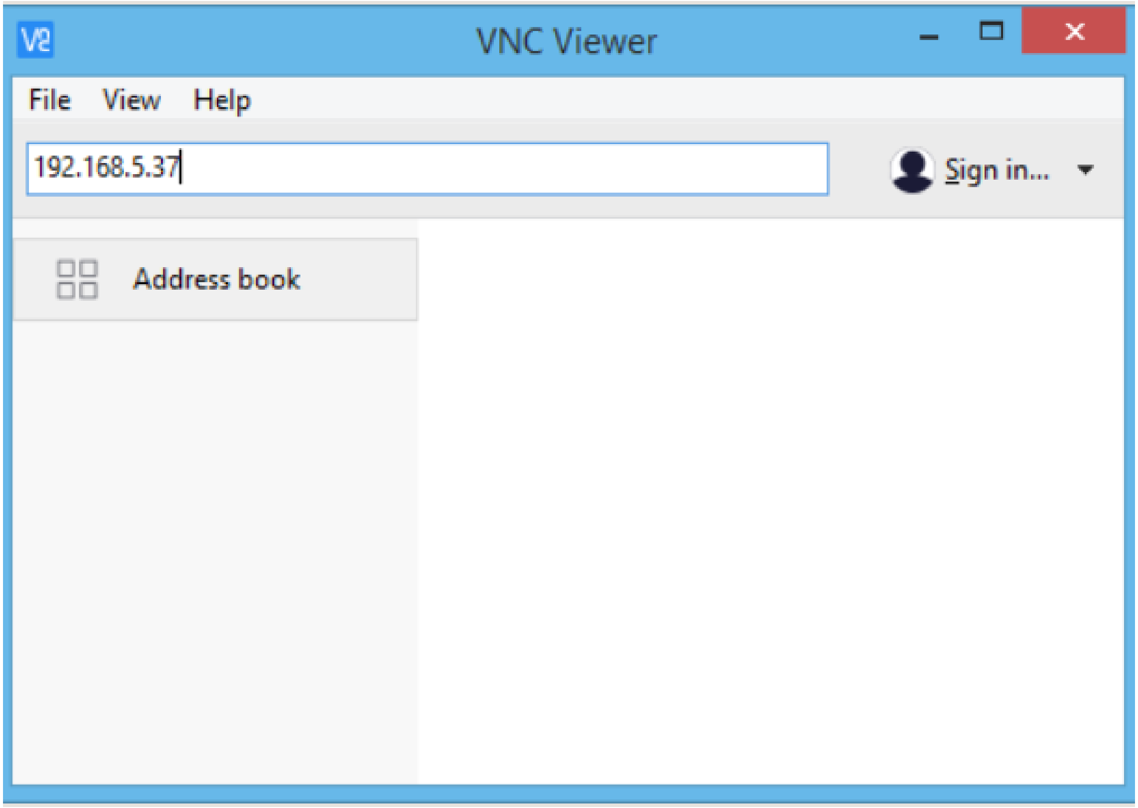

Figure 18: VNC Viewer

Now the computer and the raspberry pi are connected to the same network and can communicate through the VNC. VNC creates a virtual desktop.

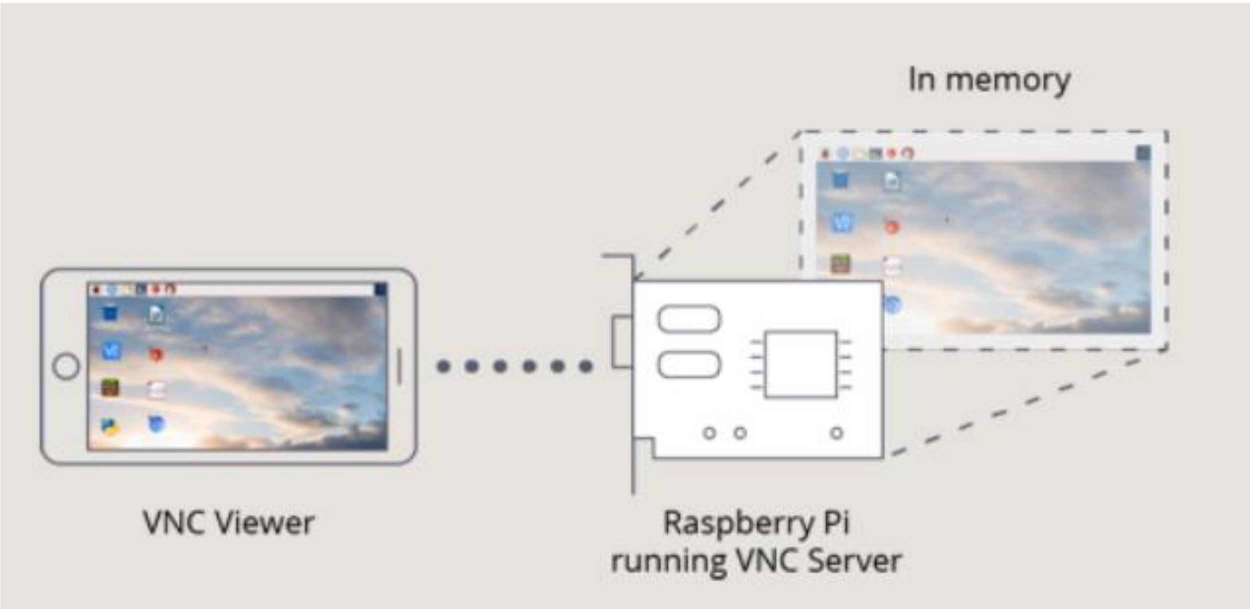

Figure ١٩ : Illustrate how raspberry pi and viewer connected.

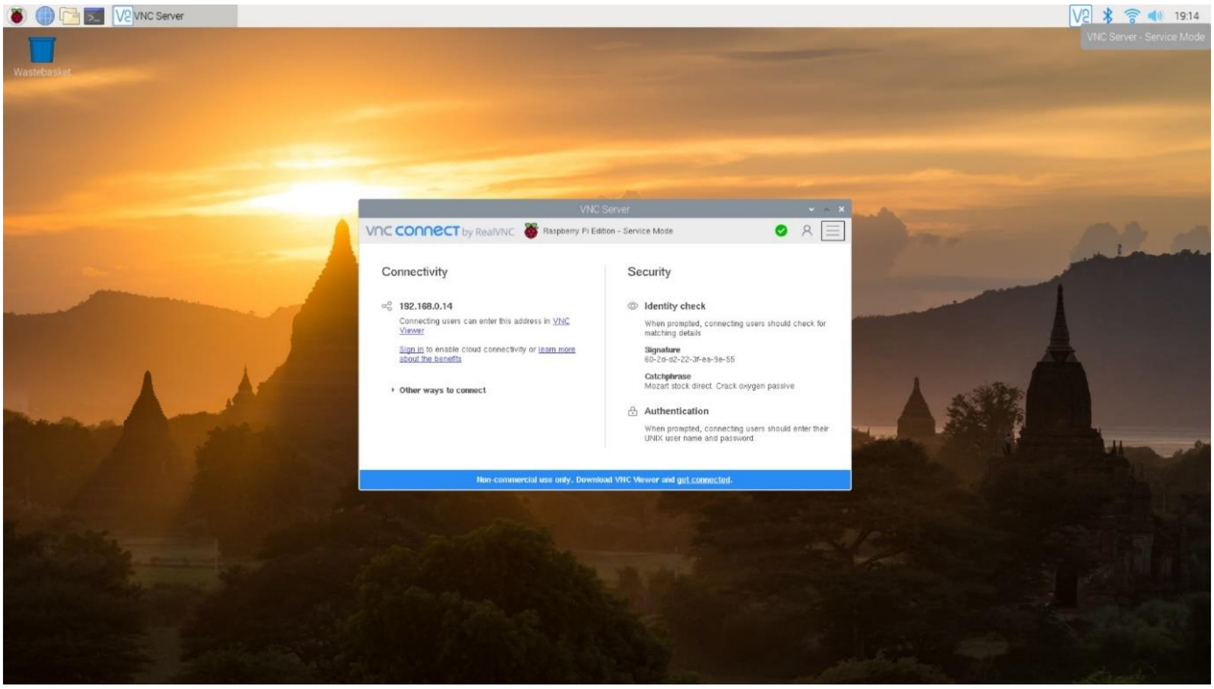

Figure٢٠ : Raspberry pi screen from desktop

14 Tools and techniques used during project implementation.

In the preparation of the project implementation phase, a list of hardware and software components were determined. This section demonstrates these requirements.

- **Raspberry Pi 4 Model B**

Raspberry Pi is a Microcontroller device that works as a small computer which contains CPUs, memory, and I/O peripherals. It will be used to connect all sensors and program them [15].

- **Buzzer Sensor**

It is like a magnetic speaker. It needs different frequency to make the sound. Alarm clocks, machines, timers, and confirmation of user feedback such as a mouse click or keystroke all use buzzers [22].

- **Vibration Motor**

It is a sensor that can be controlled and decided when it will generate the vibration. It used to send a notification to the user via vibration.

- **USB Camera**

It is a small camera that will be used to capture the objects. It will be connected to the microcontroller to implement the image processing technique.

- **Ultrasonic HCSR04 Sensor**

It is a sensor that measures the distance into the in front objects.

- **Light Emitting Diode (LED)**

A light-emitting diode is a semiconductor light source that emits light when there is a current move through it. In this project, we used two different colors of led which are red and green.

- **Power bank (Anker)**

It is a portable battery that will be used as a power supplier for the microcontroller which can be recharged when it is low.

- **Earphone**

Earphone helps to deliver the audio output from the microcontroller into the blind's ears. Any type of personal earphones can be used in our project.

- **Programming Language (Python)**

Python is one of the leading programming languages in the field of AI as it has great libraries and frameworks for AI. It is object-oriented and high-level programming language.

- **Jumper Wires**

These wires will be used to implement the connection between microcontrollers, sensors, and the raspberry pi. We used three types of Jumper wires which are male-to-male, female-to-female and female-to-male.

- **Breadboard**

  It is a plastic board with holes used to connect electronic components via pins emerging from them or pins of the wires. We will use the breadboard to make the electrical circuits for each sensor.

- **Faster R-CNN Algorithm**

  It is one of the most useful algorithms for object detection, and image processing technique which will be used to detect the objects.

- **Python packages**

  When we develop the system using Python language, we need some packages in order to help us operate the system and doing the process of image correctly. We will present some basic packages which are:

  - OpenCV: this library is easy to use and read which can be used effectively in image processing [17].
  - Scikit- image: it is an effective and fast library used for image processing [17].
  - Mahotas: it has features to compute the 3D and 2D images and it can extract information from pictures using advanced image processing [17].

- **Virtual Network Computing (VNC)**

  It is a graphical desktop-sharing system that allows remote access and control between two devices via Buffer protocol. In this project, we downloaded the VNC server on the Raspberry and the VNC viewer on our computers in order to remotely access the Raspberry from our computers.

- **microSD card**

It is a non-volatile memory card that we use to install the noob of the raspberry pi [18].

- **Raspberry Pi imager**

  A list of the latest versions of Raspbian, supported by the Raspberry Pi, is downloaded automatically from the Raspberry Pi Imager so that people don't worry if an outdated version of the operating system will happen by accident [20].

- **NOOBS**

  It is stands for New Out of Box which works as a simple operating system for the Raspberry Pi [21].

- **Laptop**

  A personal computer, we use it to download the Raspberry Pi imager, run virtual desktop of the Raspberry Pi OS, and code the ultrasonic sensor.

15 Overall project Outcomes

The final product of this project is a smart vest for blind people. This equipment was designed to be as an assisting tool that can significantly improve blind people's lives, avoiding bumping accidents while they are walking. This tool uses ultrasonic sensor to detect the presence of objects and uses image processing to recognize the detected object. The objects included in this system are Common Object in Context (COCO).

The first part of developing the smart vest involves working on a physical vest and physical hardware components. This includes the blind user's interaction with the system through audio and vibration manners. Therefore, a circuits system for the project have been built.

For the backend, Python was used to create the logic and the core of our application's page and operations. In addition, the commands of the ultrasonic sensor, USB camera, Text-to-Speech, and the buzzer in the Raspberry Pi are coded using Python language.

16 Implementation Details

In figure 21, it shows the final connection for the first circuit. Also, it shows the final design of our project after connected to the vest.

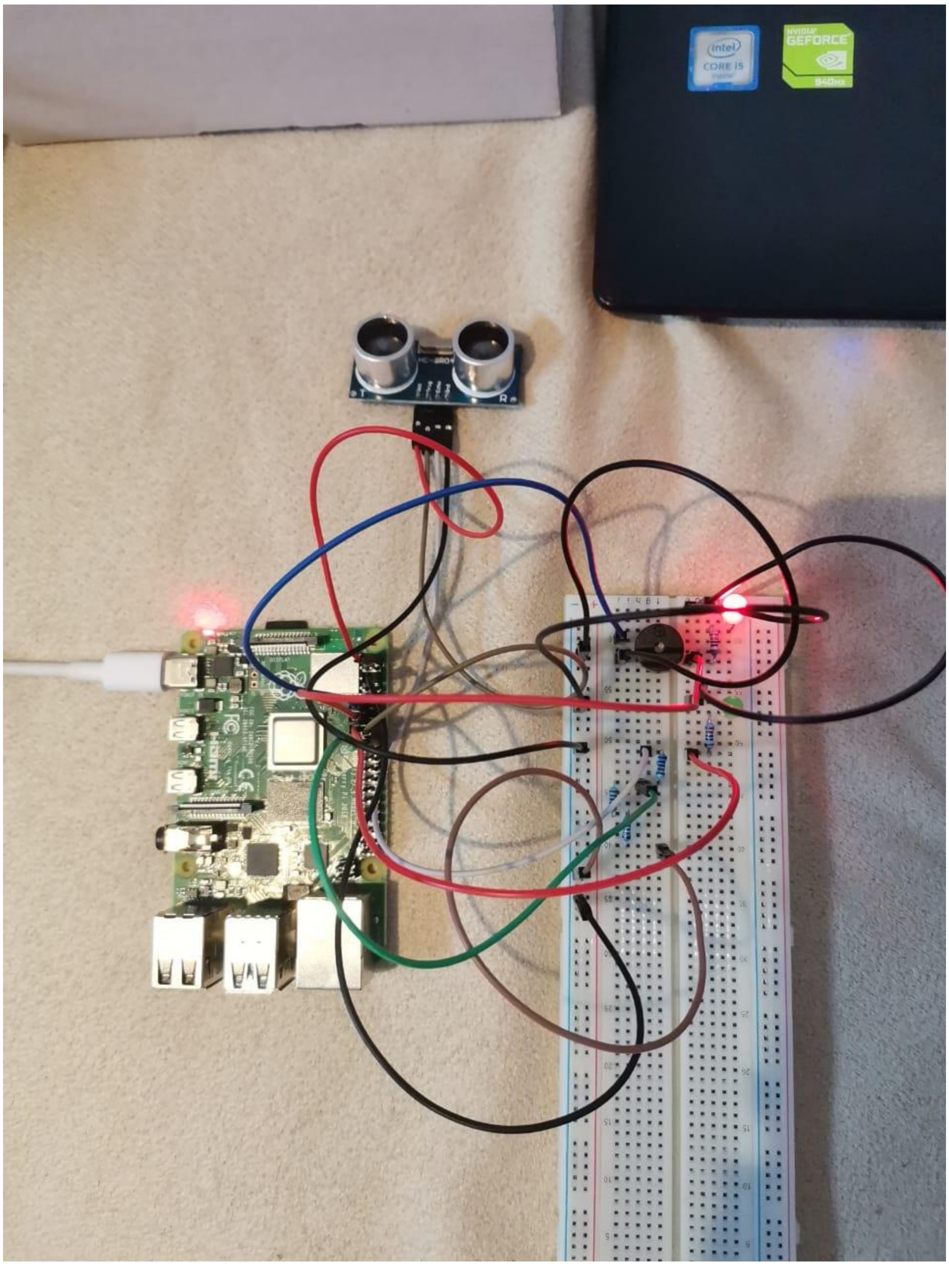

Figure 21: First circuit connection

ultrasonic_with_buzzer_and_led.py

```python
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO_TRIGGER = 18
GPIO_ECHO = 24
GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
GPIO.setup(GPIO_ECHO, GPIO.IN)

def distance():

    GPIO.output(GPIO_TRIGGER, True)
    time.sleep(0.00001)
    GPIO.output(GPIO_TRIGGER, False)
    StartTime = time.time()
    StopTime = time.time()

    while GPIO.input(GPIO_ECHO) == 0:
        StartTime = time.time()

    while GPIO.input(GPIO_ECHO) == 1:
        StopTime = time.time()

    TimeElapsed = StopTime - StartTime
    distance = (TimeElapsed * 34300) / 2
    return distance

if __name__ == '__main__':
    try:
        while True:
            dist = distance()
            print ("Measured Distance = %.1f cm" % dist)
            time.sleep(1)

    except KeyboardInterrupt:
        print("Measurement stopped by User")
```

Figure 22: Python code for first circuit

In figure 22, it shows the python code that we implement in our system to operate the ultrasonic sensor and vibration.

For the final connections we have placed the electric components into the vest. We couldn't get a new vest because of delayed delivery time, so one of the team members volunteered to offer the closest cloth she has to the vest we wanted. We looked up the literal meaning of vest in online Cambridge dictionary to support our decision. We found this definition "a piece of clothing like a jacket without sleeves, that is worn over other clothes " that exactly describes the piece of cloth we have, so we decided to use it instead of wasting time waiting for a new vest's arrival [23].

First, two wholes were made in the chest area of the vest. Then, we pulled the camera from the inside to the outside, that the wires will be invisible in the front, and the camera body is fully visible on the front. After that, the camera was stitched by a needle and thread from three places, these are the holes that were already there in the camera board. To stitch the ultrasonic sensor, the same previous steps were made, opening a hole in fabric then stitching the sensor to fabric by thread and needle from four corners this time. Figure 23 shows the final front view of the vest.

Figure 23: System with vest connection

Figure 24. shows the vest from the inside. The breadboard was sticked in place by it's own glue that gets exposed ones the protective sticker layer is removed.

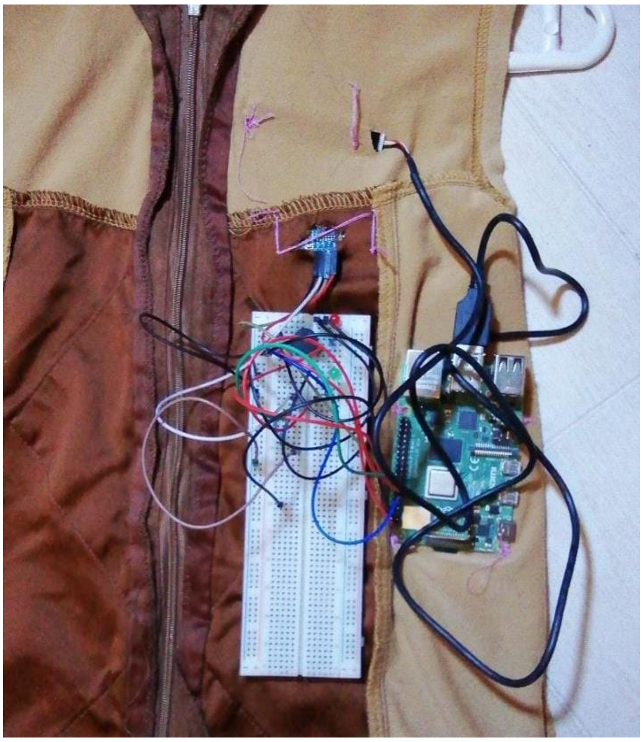

Figure 24: System with vest connection (in back)

Figure 25. shows how to remove the yellow protective sticker and the sticky gluey beadboard under it. The raspberry pi board was stitched next to it by thread and needle.

Figure 25: The yellow protective sticker

A fabric bag was made later and stitched to the vest from the inside to hold the Power bank in place as shown in figure 26.

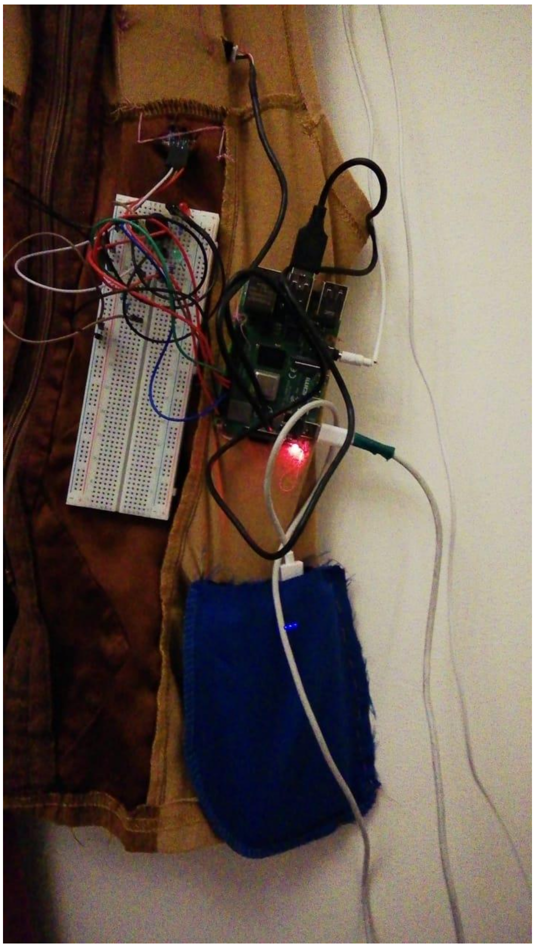

Figure 26: Vest after power bank added

The push button that we have, is a high low button. It is only ON as long as you continue pressing it as show in figure 27. and turns OFF when you stop pressing. We couldn’t find a switch early enough as there are very limited places that sells it in Saudi Arabia. So, we decided to use a switch from a child’s toy as shows in figure 28.

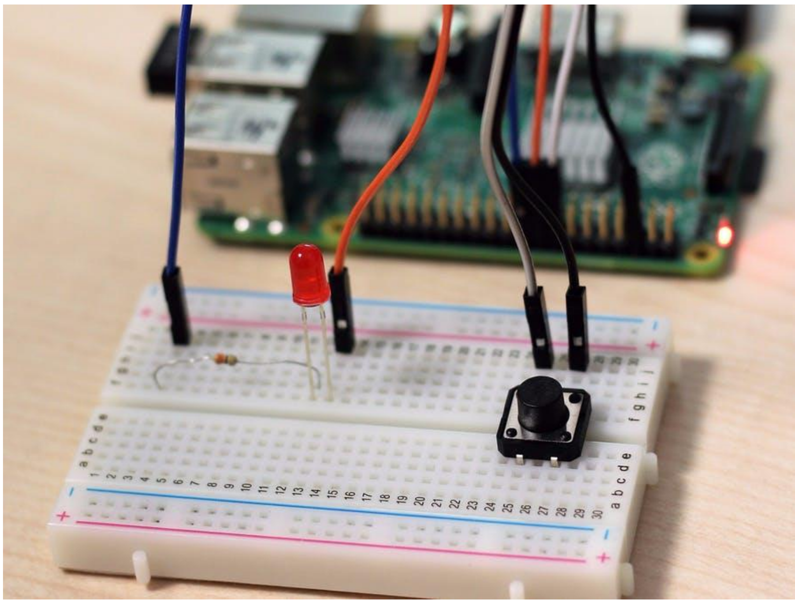

Figure 27: The push button

Figure 28: switch from child's toy

We carefully took out the button to use it, but we faced a problem that the wires were so thin and fragile, we can’t directly connect them to our breadboard or they will melt.so, we decided to remove the wires from the button and attach to it stronger wires, jumper wires, like the ones we use in the breadboard. Here we faced a new problem that the toy wired were soldered to the button by melted materials. We looked online how to remove it. it turns out we needed soldering iron like the one shown in figure no.

We tried heating a knife in stove and use it for disordering, but it only melts away the wire, give bad smoke, and ruin the button.

In millstone 4, we successfully implemented the anti-bump. In the last millstone, after the semester was shortened, we did all our best to complete the project as much as we can. So, we focused on implementing the more complex part that is the image detection and recognition. In the proposed system, we planned to have audio output naming the detected objects. Now the detected output appears in GUI screen in figure 29 with audio.

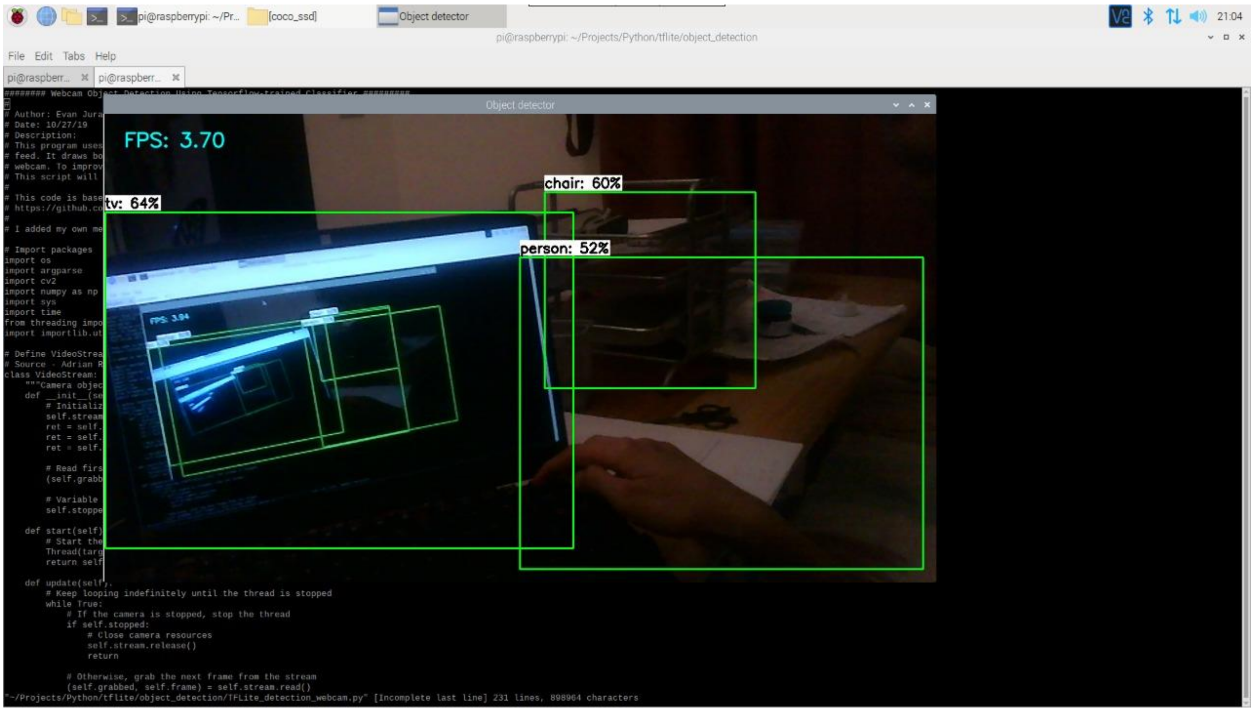

Figure 29: The output of USB camera

Then, we have used a tablet to operate the system using an app called ConnectBot as shown in figure 30. ConnectBot is an open-source Safe Shell client for the Android operating system. It lets users safely log in remotely to servers that run a stable shell daemon. This allows users to enter commands from their Android device and make them run on a remote server rather than on their local device.

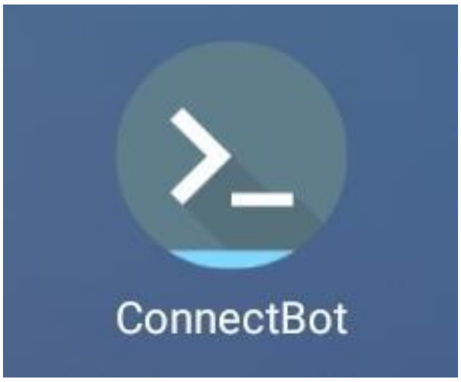

Figure 30: The tablet app used

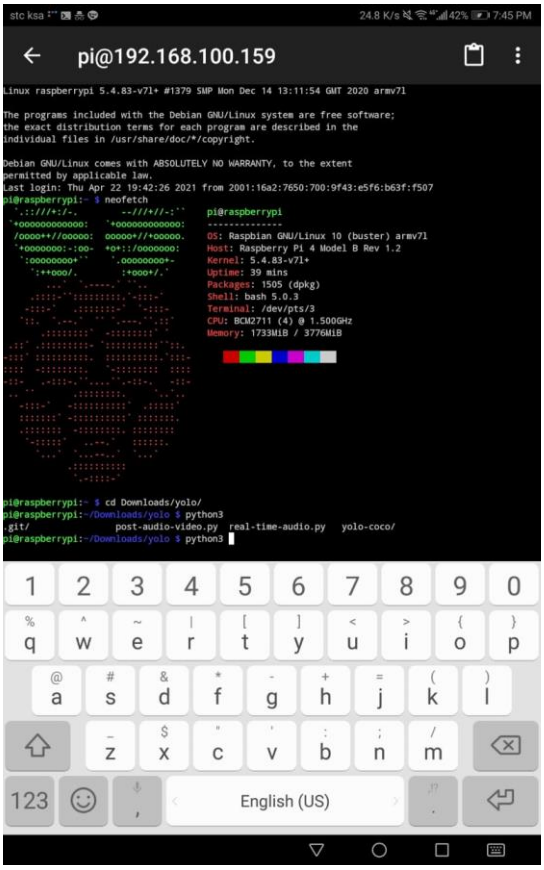

Figure 31: The command in ConnectBot

17 Analysis of Overall Result

17.1 Ultrasonic sensor function(input)

Completion percentage: 100%

The ultrasonic sensor is implemented successfully and functions as required. It detects the presence of an object at distance of 1 m.

17.2 buzzer(output)

Completion percentage: 100%

At the detection of an object at 1 meter distance, the buzzer works successfully and turns on making a sound.

17.3 ON/OFF

Completion percentage: 100%

The on/off button which is placed in the breadboard works successfully and an LED component turns on to indicate that current is flowing throw the circuit.

17.4 OS Raspberry pi

Completion percentage: 100%

The operating system of Raspberry was installed successfully on the microSD card by using the raspberry pi imager. Then, a microSD card was inserted into the Raspberry Pi device. After that, we connect the Raspberry Pi to a monitor to display its content and start coding, but we faced an issue with the HDMI cable as Raspberry Pi 4 required a special kind of cable (micro HDMI) which is not available at that time. We used another approach by connecting the Raspberry Pi 4 to the laptop using LAN (local area network) with help of a router after plugging the power supply into the Raspberry Pi 4.

17.5 Connecting the ultrasonic sensor

Completion percentage: 100%

We connected Raspberry Pi's ultrasonic sensor to the physical Pin 16 i.e. GPIO23 of the Raspberry Pi. We used a combination of 680Ω and 1.5 KΩ Resistor to convert the Echo pin to 3.3V Logic (approximately) and connected it to Physical Pin 18 i.e. GPIO24 of the Raspberry Pi. Finally, we provided the +5V and GND connections to the Ultrasonic Sensor from the Raspberry Pi Pins.

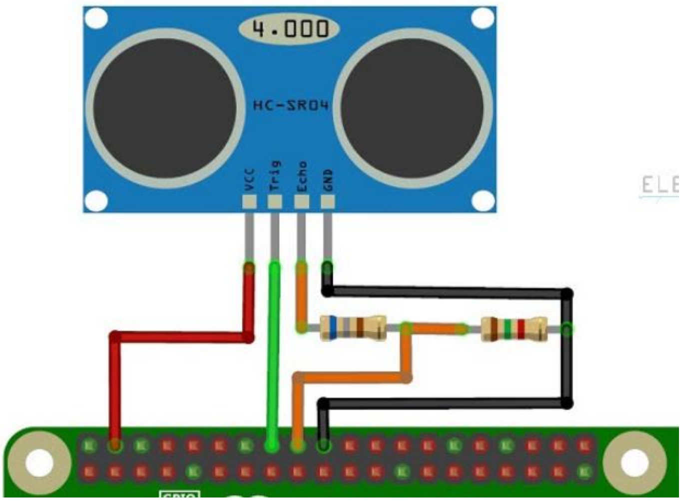

Figure 32: Connection of the ultrasonic sensor pins to the raspberry pi

17.6 Connecting buzzer

Completion percentage: 100%

We plugged a piezo buzzer into the breadboard as shown on the figure 31 and 32.

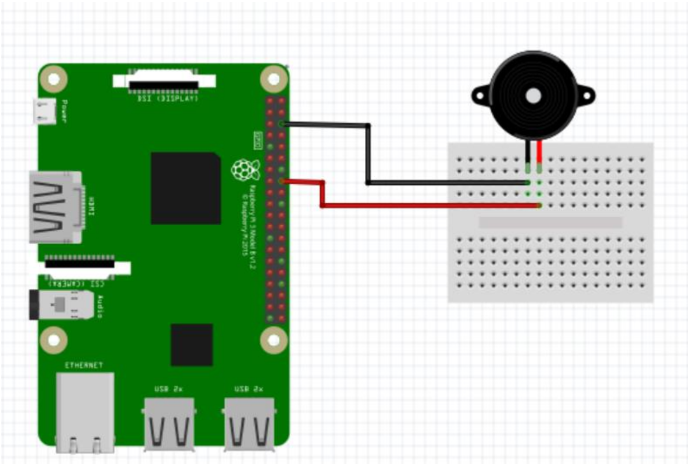

Figure 33: Circuit diagram of the buzzer

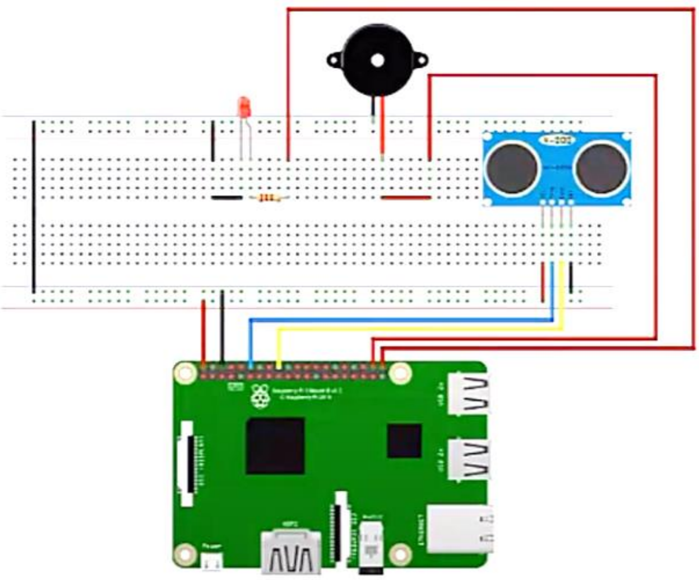

Figure 34: Circuit diagram of the ultrasonic sensor and the buzzer

17.7 Connecting vibration motor

Completion percentage: 0%

The vibration motor didn’t arrive yet so we replaced it temporarily with the buzzer mentioned above to get an output and make sure that the ultrasonic sensor works.

17.8 Connecting USB Camera

Completion percentage: 100%

We have connected the USB camera into the USB port of the Raspberry Pi. Then, we have installed the fswebcam and open-cv packages to control the camera. However, when we run the code, we faced some errors and we searched and tried to fix that issue. After that, we reinstall open-cv with its all libraries. Finally, we fixed the issue.

17.9 Object Detection

Completion percentage: 100%

We have installed YOLO and use Pre-Trained Model which called Common Objects in Context (COCO) . The COCO contains a large collection of images that have been tagged and labeled by researchers.

17.10 Text-to-Speech

Completion percentage: 100%

We had a short semester. So, we don’t have enough time for this technique to be implemented, but we have tried many times until we succussed.

17.11 Smart Vest Design

Completion percentage: 100%

We used a vest and put all the parts of the system (USB camera, ultrasonic, and the Raspberry Pi) in a suitable place as we planned.

18 Comprehensive Remarks on Overall Project Outcome and Achievements

In implementation phase, we have completed all the circuits (the ultrasonic circuit, mini-eye, and power) of our smart vest system. We were able to learn basic python skills that we need, prepare the work environment, and connect, code, and test the all the circuits as we plan in the work plan in section 12 table 6.

In the system circuit, we added two LEDs to test our circuit and to make sure that everything is working. One LED is red and the other is a green LED. When we run the project, the red LED will be on and if there are a near object and the distance between the device and the object is one meter or less the red LED will be off and the green LED will be on.

For the object detection technique, we choose the beginning Raspberry Pi camera. However, the wire of the camera was very short and not flexible. After that, we decided to change the type of camera from a Raspberry pi camera into a USB camera which contains a long wire, high accuracy, and flexibility. We connected the USB camera to the Raspberry Pi USB port. Then, we installed the fswebcam and open-cv packages to control and test the camera. When we wrote some commands for the camera in the CMD command of the Raspberry Pi, the camera opened and took a photo successfully. However, when we ran the code, we got some errors, and we couldn't control the camera from the code. We searched a lot to fix this problem and followed many steps which are checking the version of the python (it should be the last version), reinstalling the open-cv with all libraries especially the cv2 library which helps on controlling the camera. After that, we controlled the camera from the code successfully.

During the changes that occurred in the recent period we got a short semester, we have some changes in our plan to complete the project. First, we decide to use another algorithm that pre-trained model instead of using Faster R-CNN which required more time to train the model. We have chosen TensorFlow Lite as the first option because this technique is useful for tracking people at a crosswalk, tracking faces for facial recognition, looking for obstacles in a self-driving car, and so on and using it to perform object detection with a pre-trained Single Shot MultiBox Detector (SSD) model [24]. The problem in this algorithm was heating the raspberry pi which injured one of our team members when she wore to test the system. So, we have chosen another different algorithm which is the You Only Look Once (YOLO) algorithm. YOLO is using the pre-trained model. Second, the dataset we have used is Common Objects in Context (COCO). COCO contains a lot of images for common kinds of objects which is more useful than our previous dataset that was detecting just three types of objects (person, car, wall).

For the last technique which is converting the Text-to-Speech, we used Google API that used the gTTS package. It applies DeepMind's groundbreaking research in WaveNet and Google's powerful neural networks to deliver the highest fidelity possible [25]. Then, we wrote the code of Text-to-Speech to our previous code and we connected the earphone to the raspberry pi audio port. Then, we connected the hardware pieces to the vest and test our system in a standard way that we follow from the trusted Internet resources that we planned in the proposal phase.

Finally, the system able to open USB camera and implement object detection and give a percentage of accuracy in the top of each object and return voice feedback.

19 Project Enhancement and Future Work

Our smart vest be improved to perform better in many ways. The first is by powering it by solar cells to make it environment friendly. The second is to have a pocket in it that can count the money and say it's amount via a speaker. Adding this feature can protect blind people from being scammed while shopping alone. The third is to use eco-friendly fabric for the vest because they are safe for the environment, recycled, and biodegradable. The fourth improvement may sound very impossible, but we believe that it's possible at some point of the future. This improvement is to rate the clothes that the blind person is wearing under the vest, like shirt and jeans for example and tell the person if they are matching in color, fashion, and style. This is done placing little camera at the inside side of the vest. Then train a learning model by feeding it many pictures of the latest fashion trends. The last improvement is to let the user ask the vest verbally and the vest replies. For example, “is there an empty chair in this room?”, or “is my shoe in front of the door?”.

20 Conclusion

In conclusion, we believe that our system is a very good system to help blind people, although we couldn’t fully implement it right now due to time limitations, but we believe a bright future is waiting ahead for this smart vest. By the end this semester, we are so satisfied that we chose this topic as we came out with so many learning outcomes. For example, we learned how to use raspberry pi, and we got so comfortable using the sensors after it was so challenging. Whenever we heard the word raspberry pi, we thought of it as a very complex thing that is impossible for us to learn by ourselves, but doing this project we learned that with self-learning nothing is impossible and all it takes is enough patience, time, dedication, and effort. We also got to deal with a new programming language, python, that we didn’t study in college. It was not easy to do debugging and deal with it, but we learned that there are unlimited resources in the internet, and that with good collaboration and team work, the one who feels weak and unable to do a task, will be pulled and supported by her team. Moreover, we feel really proud for trying to help in solving real life problems that blind people face. We were so lucky to work in this project and we would highly recommend for next year graduate students to implement a project using raspberry pi. And regarding us, we will continue working with raspberry pi, and internet of things, to implement a model of full smart house for blind people for our master’s degree project. We pray that god blesses all the teachers, instructors, supervisor, who helped us in this project, and we pray that we be effective members of the society and make our beloved parents and university proud of us.

References

[1] Organization, U. (1970, January 01). WHO: Global data on visual impairment. Retrieved November 01, 2020, from https://www.mendeley.com/catalogue/44522b71-66a8-3a5f-9dfb-32321e7feac0/?utm_source=desktop

[2] Gandhi, R. (2018, July 09). R-CNN, Fast R-CNN, Faster R-CNN, YOLO - Object Detection Algorithms. Retrieved November 01, 2020, from https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

[3] Real-time Human Detection with OpenCV. (n.d.). Retrieved November 01, 2020, from https://thedatafrog.com/en/articles/human-detection-video/

[4] Pulkit SharmaMy research interests lies in the field of Machine Learning and Deep Learning. Possess an enthusiasm for learning new skills and technologies. (2020, October 19). Image Segmentation: Types Of Image Segmentation. Retrieved November 01, 2020, from https://www.analyticsvidhya.com/blog/2019/04/introduction-image-segmentation-techniques-python/

[5] A. Roselin Berna, R. Suganya, N. Surya, A.Suvetha, K. Manju Priya. (2018, April). Multifunctional Device for Blind People. Retrieved from https://grdjournals.com/uploads/conference/GRDCF/007/017/GRDCF007017.pdf

[6] Esra Ali, Tong Boon Tang. (2016, July). Smart Glasses for the Visually Impaired People. Retrieved from https://www.researchgate.net/publication/304802688_Smart_Glasses_for_the_Visually_Impaired_People

[7] How Microcontrollers Work. (2018, September 26). Retrieved November 22, 2020, from https://www.intervalzero.com/embedded/how-microcontrollers-work/

[8] Microcontroller. (2020, November 22). Retrieved November 22, 2020, from https://en.wikipedia.org/wiki/Microcontroller

[9] Miguel Gudino Arrow Electronics Miguel Gudino is an electrical engineer that specializes in electronic passive components and computer organization. He believes t... Read more, Miguel Gudino Arrow Electronics, Gudino, M., Electronics, A., & Miguel Gudino is an electrical engineer that specializes in electronic passive components and computer organization. He believes t... Read more. (2020, March 06). Arduino Uno or Raspberry Pi 3 - How to Decide.

Retrieved November 22, 2020, from https://www.arrow.com/en/research-and-events/articles/comparing-arduino-uno-and-raspberry-pi-3

[10] Java Introduction. (n.d.). Retrieved November 22, 2020, from https://www.w3schools.com/java/java_intro.as

[11] Understand 9 features of Java programming language. (n.d.). Retrieved November 22, 2020, from https://www.codejava.net/java-core/features-of-the-java-programming-language

[12] C++ Introduction. (n.d.). Retrieved November 22, 2020, from https://www.w3schools.com/cpp/cpp_intro.asp

[13] Python Introduction. (n.d.). Retrieved November 22, 2020, from https://www.w3schools.com/python/python_intro.asp

[14] Shawn 11 months ago, & Shawn. (2019, December 25). Types of Distance Sensor and how to select one? Retrieved November 27, 2020, from https://www.seeedstudio.com/blog/2019/12/23/distance-sensors-types-and-selection-guide/

[15] Margaret Rouse. (2019, November 07). What is a Microcontroller and How Does it Work? Retrieved from https://internetofthingsagenda.techtarget.com/definition/microcontroller

[16] Pulkit SharmaMy research interests lies in the field of Machine Learning and Deep Learning. Possess an enthusiasm for learning new skills and technologies. (2020, May 24). Introduction to Object Detection Algorithms. Retrieved December 10, 2020, from https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

[17] Yadav 17/11/2019, R., Yadav, R., Rohit Yadav Rohit is a technology journalist and technophile who likes to…, Rohit Yadav Rohit is a technology journalist and technophile who likes to communicate the latest trends around cutting-edge technologies in a way that is straightforward to assimilate. In a nutshell, & Yadav, R. (2020, November 21). Top 7 Image Processing Libraries In Python. Retrieved December 10, 2020, from https://analyticsindiamag.com/top-8-image-processing-libraries-in-python/

[18] Sd standard overview: Sd association. (2020, December 25). Retrieved March 11, 2021, from https://www.sdcard.org/developers/sd-standard-overview/

[19] Witman, E. (2021, March 04). What is an Ethernet cable? Here's how to connect to the internet without Wi-Fi and get a Speedier connection. Retrieved March 11, 2021, from https://www.businessinsider.com/what-is-an-ethernet-cable

[20] Mott, N. (2020, March 05). Raspberry Pi imager makes it easier than ever to install raspbian. Retrieved March 11, 2021, from

https://www.tomshardware.com/news/raspberry-pi-imager-install-raspbian-os#:~:text=Raspberry%20Pi%20Imager%20automatically%20downloads,according%20to%20the%20Pi%20Foundation.

[21] Noobs. (n.d.). Retrieved March 11, 2021, from https://www.raspberrypi.org/documentation/installation/noobs.md

[22] https://www.instructables.com/Raspberry-Pi-Tutorial-How-to-Use-a-Buzzer/

[23] *vest*. Cambridge Dictionary. (n.d.). https://dictionary.cambridge.org/dictionary/english/vest.

[24] *How to Perform Object Detection with TensorFlow Lite on Raspberry Pi*. DigiKey Electronics - Electronic Components Distributor. (n.d.). https://www.digikey.com/en/maker/projects/how-to-perform-object-detection-with-tensorflow-lite-on-raspberry-pi/b929e1519c7c43d5b2c6f89984883588.

[25] Google. (n.d.). *Text-to-Speech: Lifelike Speech Synthesis | Google Cloud*. Google. https://cloud.google.com/text-to-speech.

21 Appendix

21.1 The source codes.

```
# MODULES FOR GP.py
import RPi.GPIO as GPIO#module to control Raspberry Pi GPIO channels
            from gpiozero import Buzzer #To control the Buzzer sensor

from time import sleep
import time

# MODULES FOR yolo
import numpy as np
import time
import cv2
import os
import imutils
import subprocess
from gtts import gTTS
from pydub import AudioSegment


# INITIALIZE yolo

# load the COCO class labels our YOLO model was trained on
LABELS = open("yolo-coco/coco.names").read().strip().split("\n")

# load our YOLO object detector trained on COCO dataset (80 classes)
print("[INFO] loading YOLO from disk...")
net = cv2.dnn.readNetFromDarknet("yolo-coco/yolov3.cfg", "yolo-
coco/yolov3.weights")

# determine only the *output* layer names that we need from YOLO
ln = net.getLayerNames()
ln = [ln[i[0] - 1] for i in net.getUnconnectedOutLayers()]

cap = cv2.VideoCapture(0)#to start first camera or webcam
start = time.time()#takes the time in “seconds since the epoch” as
input
# and translates into a human readable string value as per the local
time.
```

```
# If no argument is passed, it returns the current tim
first = True
frames = []#emoty array
AudioSegment.converter = "/usr/bin/ffmpeg"#o convert the audio
files to the desired format

# INIT GP
buzzer = Buzzer(17) #assign pin 17 to buzzer


TRIG=23 #assign pin 23 to TRIG
ECHO=24 #assign pin 24 to ECHO
GPIO.setmode(GPIO.BCM)#apply BCM mode
GPIO.setwarnings(False)
GPIO.setup(27,GPIO.OUT)#Red LED
GPIO.setup(22,GPIO.OUT)#Green LED


# FUNCTION THAT CONVERTS VIDEO TO IMAGE WITH  AI
def yolo():
   # Capture frame-by-frameq
      ret, frame = cap.read()#video frame will be returned here, if there
no frames grabbed the image will be empty
      frame = cv2.flip(frame,1)# to flip a 2D array left-right
(horizpntally), return image
      frames.append(frame)# append this image to the frames array

      if ret:
         key = cv2.waitKey(1)#will display a frame for 1 ms, after which
display will be automatically closed.
         end = time.time()
         # grab the frame dimensions and convert it to a blob
         (H, W) = frame.shape[:2]# H -> height, W-> weight
         # construct a blob from the input image and then perform a
forward
         # pass of the YOLO object detector, giving us our bounding
boxes and
         # associated probabilities
         blob = cv2.dnn.blobFromImage(frame, 1 / 255.0, (416, 416),
               swapRB=True, crop=False)# arg 1-> image, arg2-> scale
factor, arg3-> size
         #swapRB flag which indicates that swap first and last channels
```

```
	#crop flag equal false which indicates image will not cropped
after resized
	net.setInput(blob)
	layerOutputs = net.forward(ln)

	# initialize our lists of detected bounding boxes, confidences,
and
	# class IDs, respectively
	boxes = []
	confidences = []
	classIDs = []
	centers = []

	# loop over each of the layer outputs
	for output in layerOutputs:
	    # loop over each of the detections
	        for detection in output:
	            # extract the class ID and confidence (i.e., probability) of
	                # the current object detection
	                scores = detection[5:]# elements after index 5
	                classID = np.argmax(scores)
	                confidence = scores[classID]

	                # filter out weak predictions by ensuring the detected
	                # probability is greater than the minimum probability
	                if confidence > 0.5:
	                    # scale the bounding box coordinates back relative
to the
	                        # size of the image, keeping in mind that YOLO
actually
	                        # returns the center (x, y)-coordinates of the
bounding
	                        # box followed by the boxes' width and height
	                        box = detection[0:4] * np.array([W, H, W, H])
	                        (centerX, centerY, width, height) =
box.astype("int")

	                        # use the center (x, y)-coordinates to derive the
top and
	                        # and left corner of the bounding box
	                        x = int(centerX - (width / 2))
```

```
                    y = int(centerY - (height / 2))

                    # update our list of bounding box coordinates,
confidences,
                    # and class IDs
                    boxes.append([x, y, int(width), int(height)])
                    confidences.append(float(confidence))
                    classIDs.append(classID)
                    centers.append((centerX, centerY))

        # apply non-maxima suppression to suppress weak, overlapping
bounding
        # boxes
        idxs = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.3)

        texts = []

        # ensure at least one detection exists
        if len(idxs) > 0:
            # loop over the indexes we are keeping
                for i in idxs.flatten():#this is used to get a copy of an given
array collapsed into one dimensi
                    # find positions
                    centerX, centerY = centers[i][0], centers[i][1]

                    if centerX <= W/3:
                        W_pos = "left "
                    elif centerX <= (W/3 * 2):
                        W_pos = "center "
                    else:
                        W_pos = "right "

                    if centerY <= H/3:
                        H_pos = "top "
                    elif centerY <= (H/3 * 2):
                        H_pos = "mid "
                    else:
                        H_pos = "bottom "

                    texts.append(H_pos + W_pos + LABELS[classIDs[i]])
```

```
        print(texts)

        if texts:
            description = ', '.join(texts)
            tts = gTTS(description, lang='en')
            tts.save('tts.mp3')
            tts = AudioSegment.from_mp3("tts.mp3")
            subprocess.call(["ffplay", "-nodisp", "-autoexit", "tts.mp3"])
        else:
            tts = gTTS("Object undefined", lang='en')
            tts.save('tts.mp3')
            tts = AudioSegment.from_mp3("tts.mp3")
            subprocess.call(["ffplay", "-nodisp", "-autoexit", "tts.mp3"])



while True:
    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")
    sleep(0.2)
    GPIO.output(TRIG, True)
    sleep(0.00001)
    GPIO.output(TRIG,False)

    while GPIO.input(ECHO)==0: #wait till echo is low
        pulse_start=time.time()
    while GPIO.input(ECHO)==1: #wait till echo is high
        pulse_end=time.time()
    pulse_duration=pulse_end-pulse_start
    distance= pulse_duration*17150
    distance=round(distance,2)
    print("distance:",distance,"cm")
    sleep(2)

    if distance <= 100.00:
        GPIO.output(27,GPIO.LOW)
        GPIO.output(22,GPIO.HIGH)#Green LED ON
```

```
        buzzer.beep()
        yolo()


    else:
        GPIO.output(27,GPIO.HIGH)#Red LED ON
        GPIO.output(22,GPIO.LOW)
        buzzer.off()

# When everything done, release the capture
cap.release()
cv2.destroyAllWindows()
os.remove("tts.mp3")
```