DRISHTI – SMART GLASSES FOR VISUALLY IMPAIRED

A Project Report Submitted by

Joswin Valerian Concessao	**J Arpana**
(4NM16CS056)	**(4NM16CS055)**

Nidhi Prakash

(4NM16CS079)

UNDER THE GUIDANCE OF

Dr. Sarika Hegde

Assistant Professor

Department of Computer Science and Engineering

in partial fulfilment of the requirements for the award of the Degree of

Bachelor of Engineering in
Computer Science & Engineering

from

Visvesvaraya Technological University, Belagavi

NITTE
EDUCATION TRUST

N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India

(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC
☎: 08258 - 281039 – 281263, Fax: 08258 – 281265

Department of Computer Science and Engineering

April 2020

NITTE
EDUCATION TRUST

N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India
(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC
☎: 08258 - 281039 – 281263, Fax: 08258 – 281265

Department of Computer Science and Engineering

B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

CERTIFICATE

Certified that the project work entitled

"Drishti – Smart Glasses for Visually Impaired"

is a bonafide work carried out by

Joswin Valerian Concessao (4NM16CS056) *J Arpana (4NM16CS055)*
Nidhi Prakash (4NM16CS079)

in partial fulfilment of the requirements for the award of ***Bachelor of Engineering***

Degree in Computer Science and Engineering

prescribed by ***Visvesvaraya Technological University,***
Belagavi *during the year 2019-2020.*

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

Signature of Guide **Signature of HOD** **Signature of Principal**

Semester End Viva Voce Examination

Name of the Examiners **Signature with Date**

1. ____

2. ____

ACKNOWLEDGEMENT

We believe that our project will be complete only after we thank the people who have contributed to make this project successful.

First and foremost, our sincere thanks to our beloved principal, **Dr. Niranjan N. Chiplunkar** for giving us an opportunity to carry out our project work at our college and providing us with all the needed facilities.

We express our deep sense of gratitude and indebtedness to our guide **Dr. Sarika Hegde,** Assistant Professor, Department of Computer Science and Engineering, for her inspiring guidance, constant encouragement, support and suggestions for improvement during the course for our project.

We express our gratitude to the project co-ordinators **Mrs. Pallavi K N**, **Mrs. Asmitha Poojary**, **Mr. Ranjan Kumar H S** and **Mr. Raju K** for their constant support.

We would like to thank **Mr. Rajeevan K**, System Analyst and **Dr. Ashwini B**, Associate Professor, for providing us access to the GPU server during the course for our project.

We sincerely thank **Dr. K.R. Udaya Kumar Reddy**, Head of Department of Computer Science and Engineering, Nitte Mahalinga Adyantaya Memorial Institute of Technology, Nitte.

We also thank all those who have supported us throughout the entire duration of our project.

Finally, we thank the staff members of the Department of Computer Science and Engineering and all our friends for their honest opinions and suggestions throughout the course of our project.

Joswin Valerian Concessao
J Arpana
Nidhi Prakash

ABSTRACT

It is difficult to imagine life without vision which leads one to be completely dependent on others. In today’s fast paced world, the difficult daily routines are still a struggle for visually impaired. People with visual impairment face various problems in their daily life as the modern assistive devices are often not meeting their requirements in terms of price and level of assistance. So by developing this product, we want to help the visually impaired and make them independent by adding the additional features to the product such as Reading a text characters using OCR API, Image Captioning and time and weather from weather Application Programming Interface. This project presents a new product to provide a smart assistance in the form of smart glasses for visually impaired.

Keywords: Application Programming Interface, Image Captioning, Optical Character Recognition API, Convolution Neural Network, Image Processing.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Many people suffer from serious visual impairments preventing them from travelling independently. People with visual impairment face various problems in their daily life as the modern assistive devices are often not meeting their requirements in terms of price and level of assistance. This project proposes a new design of assistive smart glasses for visually impaired people.

Speech recognition plays a big role in this project. Speech recognition is the ability of a machine or program to identify words and phrases in spoken language and convert them to a machine-readable format. Basic speech recognition software has a limited vocabulary of words and phrases, and it may only identify these if they are spoken very clearly. This is mostly used when the problem requires only certain words to be recognized. More sophisticated software can accept natural speech. Speech recognition works using algorithms through acoustic and language modelling. Acoustic modelling represents the relationship between linguistic units of speech and audio signals; language modelling matches sounds with word sequences to help distinguish between words that sound similar. Often, hidden Markov models are used as well to recognize temporal patterns in speech to improve accuracy within the system. The most frequent applications of speech recognition within the enterprise include call routing, speech-to-text processing, voice dialing and voice search. Here, we are using Speech recognition to convert the voice commands to text and provide the required assistance.

APIs help the product to get information from the internet such as Weather, News etc. API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other. Each time you use an app like Facebook, send an instant message, or check the weather on your phone, you’re using an API.

Reading anything other than braille is impossible for visually impaired people. This stops them from reading important documents, books and newspaper.

Character recognition is used in this project to eliminate this. Character recognition, usually abbreviated to optical character recognition or shortened OCR, is the mechanical or electronic translation of images of handwritten, typewritten or printed text (usually captured by a scanner) into machine-editable text. It is a field of research in pattern recognition, artificial intelligence and machine vision. Though academic research in the field continues, the focus on character recognition has shifted to implementation of proven techniques.

The goal of the project is to develop a product with features which can help the blind easily do their work without any dependence on another person.

1.1 OVERVIEW

This project presents a new design of assistive smart glasses which is to assist the visually impaired in multiple daily tasks using the advantage of wearable design format. We proposed new methods which combines the key aspects of some useful methods and added some extra capabilities for assisting the visually impaired. This project may solve some of major problems of visually impaired that are still existing.

1.2 PROBLEM STATEMENT

Drishti- A device that can assist the visually impaired perform their daily tasks with ease without being dependent on another person. The device accepts voice commands and aids the user. There are various common commands like any other smart assistant but with added commands specially for visually impaired. Character recognition is used to recognize the characters present in a book, document or newspaper and this will be converted to speech and sent as audio signals to the earphones. Drishti is a portable device which has a processing unit and required input-output devices attached to the glass that can be carried anywhere easily by the user. It is powered by a portable power source which can power it for a day. The processing unit Raspberry Pi and the portable power supply needs to be tied around the waist of the user. This device will be connected to the glasses which have a camera, earphones and microphone attached to it.

1.3 STUDY AREA

The visually impaired person gives verbal commands to the device which is then converted to machine understandable format. The device processes these commands and performs operations and speaks back to the user with results. These operations include reading a text characters, Face recognition, News and Weather updates, Current time and Image Captioning. This project helps visually impaired become self-reliance and independent.

1.4 OBJECTIVE

1. The main objective of this project is to provide assistance to do their daily tasks and know their surroundings better.
2. The smart glasses are affordable by using single board computer raspberry pi 4 and other required modules.
3. This project presents a new product to provide smart assistance in the form of smart glasses for visually impaired.

The scope of project is to help visually impaired to recognize the objects in front of them by performing the following operations:

- Speech to text.
- Read text.
- Weather and News.
- Wikipedia Search.
- Current Time.
- Face recognition.
- Image Captioning.
- Convert the information to voice to be understood by visually impaired via headphone

Smart glasses will have most of the smart features that will help the visually impaired.

1.5 MOTIVATION

Physical movement is one of the biggest challenges for visually impaired. Travelling or merely walking down a crowded street can be challenging. Because of this, many people with low vision will prefer to travel with a sighted friend or family member when navigating unfamiliar places. Blindness can make it difficult to use the internet for research, recreation, social media and shopping. For example, someone who is blind can't directly read the information on a web page. People with visual impairment face various problems in their daily life as the modern assistive devices are often not meeting their requirements in terms of price and level of assistance. This project presents a new product to provide a smart assistance in the form of smart glasses for visually impaired people. The smart glasses cost is kept less by using single board computer raspberry pi 4 and other required modules. The device can assist the visually impaired perform their daily tasks with ease without being dependent on another person. The device accepts voice commands and provides assistance to the user. There are various common commands similar to any other smart assistant but with added commands specially for visually impaired.

1.6 ORGANIZATION OF THE CHAPTERS

The project report has been organized under nine chapters, which are as follows:

Chapter I: Introduces to the main idea of the project. It gives a brief knowledge about the aim and methodology of the same.

Chapter II: It includes literature survey of related works.

Chapter III: Discusses the system requirements that are needed for the project. These include functional requirements, non-functional requirements, user requirements and hardware requirements.

Chapter IV: Includes various software approach.

Chapter V: Approach includes the system design details which includes sequence diagram and use case diagram.

Chapter VI: Includes the implementation details of the project, application is explained in detail.

Chapter VII: Deals with system testing concepts and the various test cases for the project.

Chapter VIII: Discuss the results of the project.

Chapter IX: Outlines conclusions and future work that can be done.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING SYSTEM

In order to design an assistive device for the visually impaired, information about their needs and their expectations should be gathered for a better design, it is easy to understand that the main problem is the sight loss, but to explore the situation without sight and to identify their needs is not an easy task for the sighted people. Some of the existing system are:

OrCam devices such as OrCam MyEye are portable, artificial vision devices that allow visually impaired to understand text and identify objects through audio feedback, describing what they are unable to see with a smart camera mounted on the frames of your eyeglasses. It reads printed text, such as books, newspapers, food labels, restaurant menus, and even street signs.

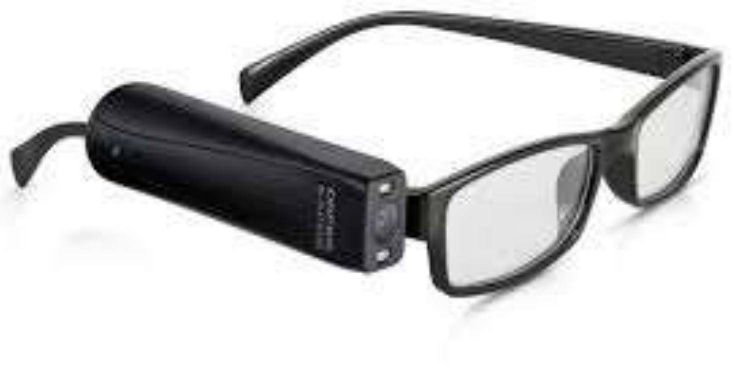


Figure 2.1.1: OrCam

Esight is another eyewear technology for people with low vision. It captures processes live scenes, and displays them back on a specialized screen in front of the user’s eyes. Esight yields immediate improvement in visual ability, with face recognition and ADLs showing a tentative benefit of further use. Overall the visual ability, reading and visual information showed greatest benefit with device use.

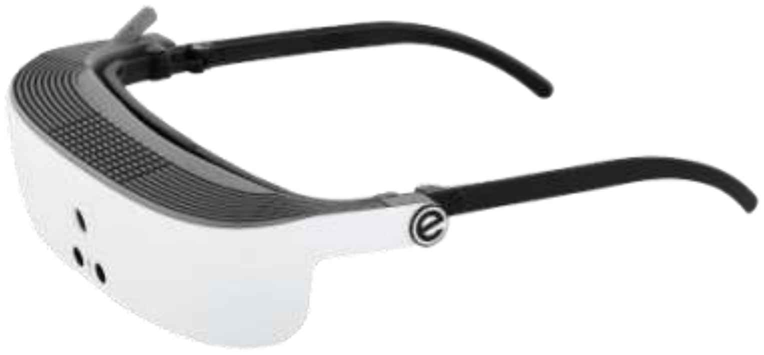


Figure 2.1.2: Esight

Aira are smart glasses that also use a camera and connectivity to bring assistance to people with a visual impairment. In this case, however, what you're connected to is a trained assistant who provides spoken feedback about what you are looking at. Useful for help with identifying objects, reading documents, menus or medication. These offer a pair of eyes to guide you through unfamiliar routes or indoor surroundings.

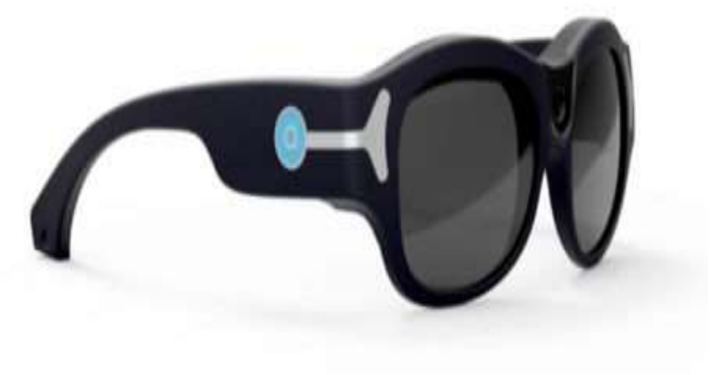


Figure 2.1.3: Aira

2.2 PROPOSED SYSTEM

Smart glasses for visually impaired is to assist the visually impaired to perform their daily tasks with ease without being dependent on another person. The device accepts voice commands and aids the user. There are various common commands like any other smart assistant but with added commands specially for visually impaired. Character recognition is used to recognize the characters present in a book, document or newspaper and this will be converted to speech and sent as audio signals to the earphones. The device can work online as well as offline but with restricted features. Image Captioning is present to help the visually impaired to safely move from one place to another. Our system doesn’t require the user to point his/her finger to trigger the operations rather it works on audio and can perform more tasks than just object detection. Information such as weather, time, news etc. can be also played back to the user.

The proposed system consists of various components such as Raspberry Pi 4 4GB Pi camera, Power-bank 5V 3A, Earphones and other required component. The device will be powered by a portable power bank which can power it for a day. The processing unit Raspberry Pi and the portable power supply needs to be tied around the waist of the user. This device will be connected with the glasses which have a camera, earphones and microphone attached to it. It helps visually impaired become self-reliance and independent and the building cost is kept low, so it turns to be an effective device for visually impaired. The goal of the project is to develop a product with features which can help the blind easily do their work without any dependence on another person.

CHAPTER 3

SYSTEM ANALYSIS AND REQUIREMENTS

3.1 SYSTEM ANALYSIS

3.1.1 Relevance of Platform

Linux is one of the most reliable, secure and worry-free operating systems available. Linux is the best-known and most-used open source operating system. It is software that sits underneath all of the other software on a computer. The code used to create Linux is free and available to the public to view and edit.

3.1.2 Relevance of Programming Language

Python is a powerful and object-oriented high-level programming language. It has very simple easy-to-use syntax. It works on cross-platform operating systems and can be used across to develop a wide range of applications including those intended for image processing, text processing, web, and enterprise level using scientific, numeric and data from the network.

Shell script is a computer program designed to be run by the unix shell a command-line interpreter. The various dialects of shell scripts are considered to be scripting languages. The term is also used more generally to mean the automated mode of running an operating system shell. Writing a shell script is much quicker than writing the equivalent code in other programming languages. The many advantages include easy program or file selection, quick start, and interactive debugging.

3.2 REQUIREMENT ANALYSIS

3.2.1 Scope and Boundary

The scope of project is to help visually impaired to get a sense of his/her surroundings and get few smart assistive help by performing the following operations:

1. Speech to text.
2. Read text.
3. Face Recognition.
4. Weather and News.
5. Wikipedia Search.
6. Current Time.
7. Image Captioning.
8. Convert the information to voice to be understood by visually impaired via headphone.

3.3 FUNCTIONAL REQUIREMENTS

3.3.1 Software Requirements:

1. Python 3.7

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python was designed to be highly readable which uses English keywords frequently whereas other languages use punctuation and it has fewer syntactical constructions than other languages. Python is a general-purpose language, which means it can be used to build just about anything, which will be made easy with the right tools/libraries. Professionally, Python is great for backend web development, data analysis, artificial intelligence, and scientific computing.

2. OpenCV

OpenCV is a library of programming functions mainly aimed at real-time computer vision.

The library is cross-platform and free for use under the open-source BSD license. It is developed for c++ and python. It supports windows, linux, mac, ios. cv2 is the name that OpenCV developers chose when they created the binding generators. This is kept as the import name to be consistent with different kind of tutorials around the internet.

3. Speech Recognition

Speech recognition is an interdisciplinary subfield of computational linguistics that develops methodologies and technologies that enables the recognition and translation of spoken language into text by computers. Speech recognition systems require “training” where an individual speaker reads text or isolated vocabulary into the system.

1.3.2 Hardware Requirements:

1. Raspberry Pi 4 4GB

Raspberry Pi is a small, powerful, cheap, hackable and education-oriented computer board. It operates in the same way as a standard PC, requiring a keyboard for command entry, a display unit and a power supply. Many people have used the Raspberry Pi to make things like cameras, gaming machines, robots, web servers and media centers.

2. Pi camera

The Raspberry Pi camera module can be used to take high-definition video, as well as stills photographs. It’s easy to use for beginners. The camera consists of a small (25mm by 20mm by 9mm) circuit board, which connects to the Raspberry Pi's Camera Serial Interface (CSI) bus connector via a flexible ribbon cable.

3. Power-bank 5V 3A

A power bank is a portable charging device which is used to power the Raspberry pi.

Power banks are not just a simple battery: they use sophisticated electronic circuitry to manage being charged, and then charging other devices.

4. Earphones and other required components

Earphones are a pair of small listening devices that are designed to be worn on or around the head over a user's ears.

5. GPU Server

GPU servers, as the name suggests, are servers packed with graphics cards, designed to harness this raw processing power. Using an offloading process, the CPU can hand specific tasks to the GPUs, increasing performance.

Specifications:

1. Dual Intel Xeon CPU
2. 128GB Memory
3. Dual Nvidia Tesla P100 GPU 12GB
4. 500GB HDD
5. A color monitor with a minimum resolution of 1024x786 pixels

Uses:

1. Training a simple machine learning model would take a lot of time without advanced GPUs. The GPU server helps in quick training of the model.
2. It can be remotely accessed using Secure
3. Shell from any terminal.

3.4 NON-FUNCTIONAL REQUIREMENTS:

In systems engineering and requirements engineering, a non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. Non-functional requirements are conditions under which the system must be able to function and the quality the system must have. It defines how a system is supposed to be.

➢ Performance:

- We have chosen raspberry pi 4 because the new 1.5 GHz 4-core ARM chip is more than three times faster than the raspberry pi 3.

➢ Reliability:

- The system should be made fully functional at least 95% of the time
- In the event of crash, the system should maintain full functionality after it is restored.

➢ Maintainability:

- The design should be simple enough that they are easy to modify.
- The system should be designed such a way that it is easy to test.

➢ Usefulness:

- It will be great help for visually impaired and it will make their life bit easier.

➢ Scalability:

- Our device is suitable in terms of adding extra features and raspberry pi 4 supports many modules which can be attached to it to perform many operations.

➢ Portability:

- The device is a raspberry pi 4 which can run using power bank.

CHAPTER 4
SOFTWARE APPROACH

4.1 PYTHON 3.7

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python was designed to be highly readable which uses English keywords frequently whereas other languages use punctuation and it has fewer syntactical constructions than other languages. Python is a general-purpose language, which means it can be used to build just about anything, which will be made easy with the right tools/libraries. Professionally, Python is great for backend web development, data analysis, artificial intelligence, and scientific computing. Python can serve as a scripting language for web applications. Many developers have also used Python to build productivity tools, games, and desktop apps, so there are plenty of resources to help you learn how to do those as well. Python 3.7 is easy to learn, powerful programming language. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

4.2 OPENCV

OpenCV is a library of programming functions mainly aimed at real-time computer vision. The library is cross-platform and free for use under the open-source BSD license. It is developed for c++ and python. It supports windows, Linux, mac, iOS. cv2 is the name that OpenCV developers chose when they created the binding generators. This is kept as the import name to be consistent with different kind of tutorials around the internet. OpenCV-Python is a library of Python bindings designed to solve computer vision problems. OpenCV-Python makes use of Numpy, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays. In simple language it is library used for Image Processing. It is mainly used to do all the operation related to images.

4.3 SPEECH RECOGNITION

Speech recognition is an interdisciplinary subfield of computational linguistics that develops methodologies and technologies that enables the recognition and translation of spoken language into text by computers. Speech recognition systems require “training” where an individual speaker reads text or isolated vocabulary into the system. The system analyses the person’s specific voice and uses it to fine-tune the recognition of that person’s speech, resulting in increased accuracy.

Speech recognition applications include voice user interfaces such as voice dialing, call routing, domestic appliance control, search key words, preparation of structured determining speaker characteristics, speech-to-text processing.

CHAPTER 5

SYSTEM DESIGN

5.1 HIGH LEVEL DESIGN ARCHITECHTURE

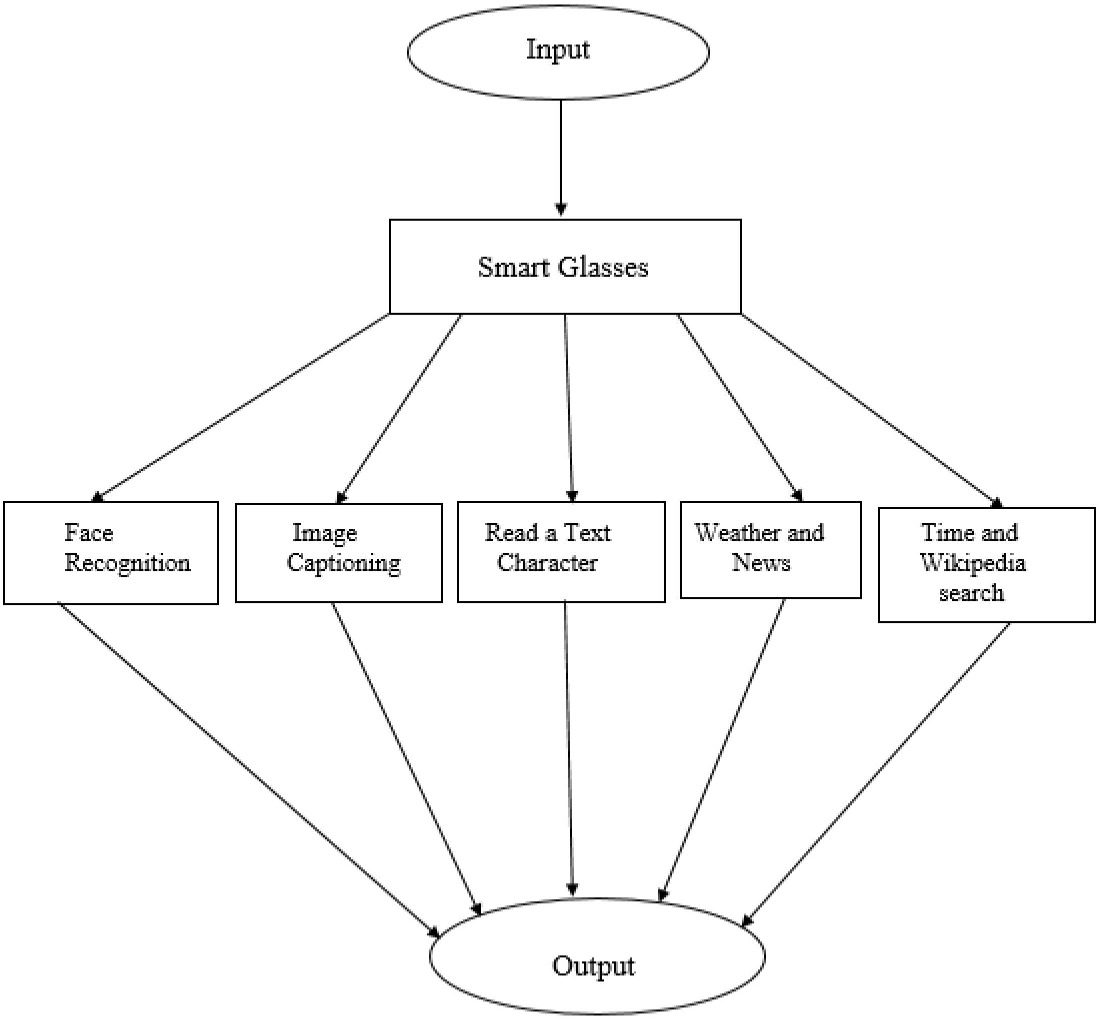


Figure 5.1: High Level Design Architecture

5.2 LOW LEVEL DESIGN ARCHITECHTURE

5.2.1 Sequence Diagram /DFD

A sequence diagram shows object interaction arranged in time sequence. It describes interactions among classes in terms of an exchange of messages over time. It is also called as event diagram. A sequence diagram is a good way to visualize and validate various run time scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modelling the new system. Messages are arrows that represent communication between the objects. Lifelines are vertical dashed lines that indicate the object presence over time.

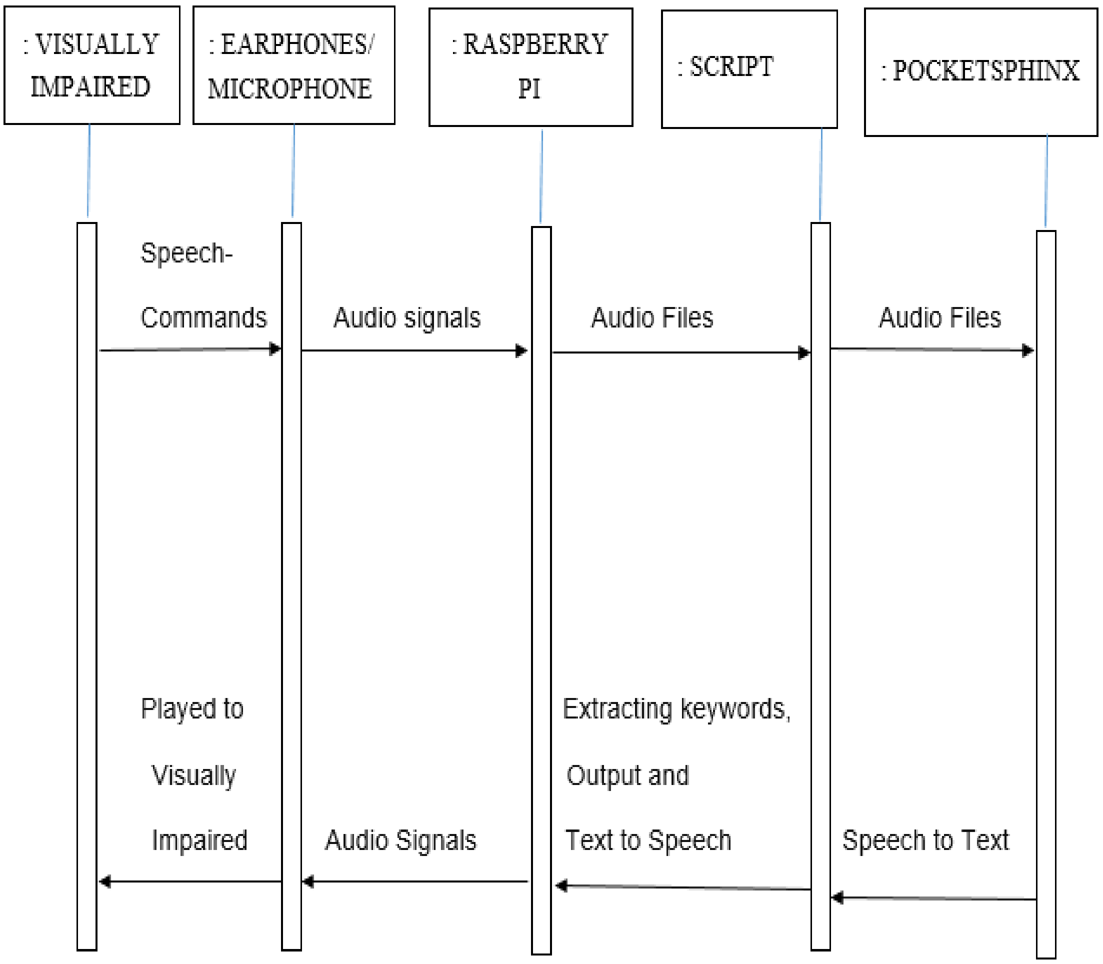


Figure 5.2: Sequence diagram

5.2.2 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques.

5.2.3 Use case Diagram

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform.

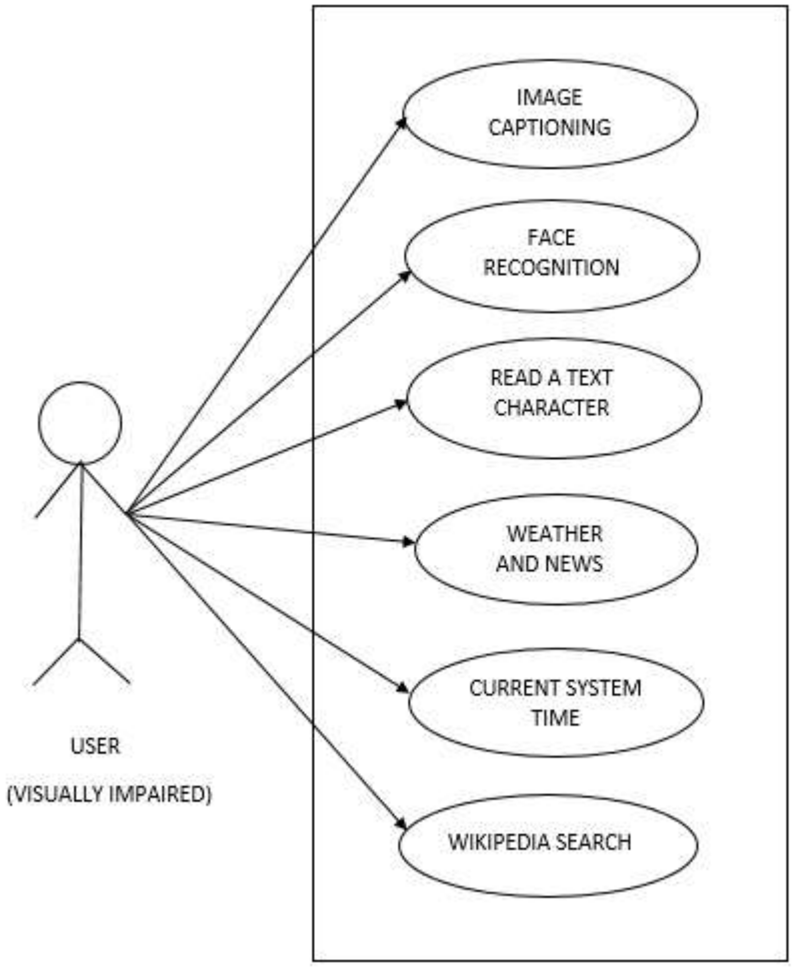


Figure 5.3: Use Case diagram

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 SOFTWARE APPROACH

6.1.1 Components Used

- Raspberry Pi 4 4GB
- Pi camera
- Power-bank 5V 3A
- Earphones and other required components

- **Raspberry Pi 4 4 GB**

Raspberry Pi is a small, powerful, cheap, hackable and education-oriented computer board. It operates in the same way as a standard PC, requiring a keyboard for command entry, a display unit and a power supply. Many people have used the Raspberry Pi to make things like cameras, gaming machines, robots, web servers and media centers.

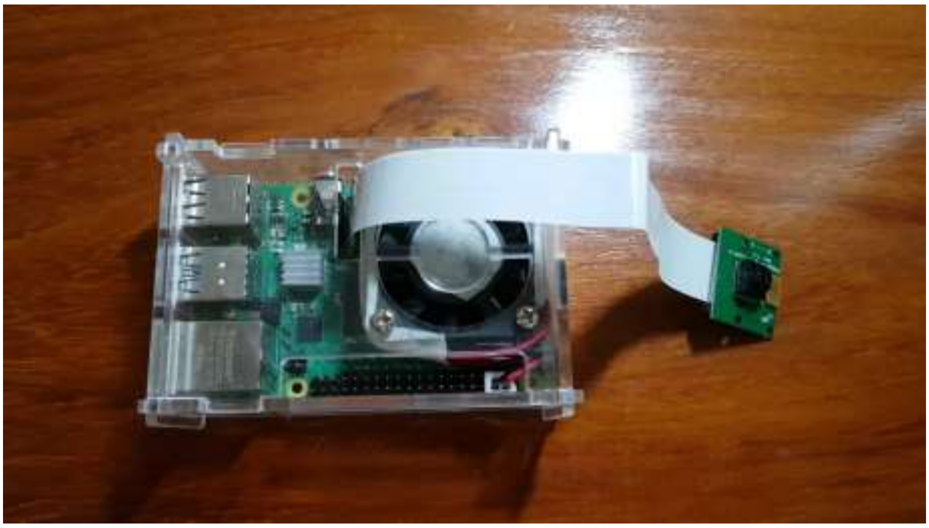


Figure 6.1.1.1 Raspberry pi 4 and Pi Camera

- **Pi camera**

The Raspberry Pi camera module can be used to take high-definition video, as well as stills photographs. It's easy to use for beginners.

The camera consists of a small (25mm by 20mm by 9mm) circuit board, which connects to the Raspberry Pi's Camera Serial Interface (CSI) bus connector via a flexible ribbon cable. The camera's image sensor has a native resolution of five megapixels and has a fixed focus lens.

- **Power-bank 5V 3A**

A power bank is a portable charging device which is used to power the Raspberry pi. Power banks are not just a simple battery they use sophisticated electronic circuitry to manage being charged, and then charging other devices.

Figure 6.1.1.2 Power Bank

- **Earphones**

Earphones are a pair of small listening devices that are designed to be worn on or around the head over a user's ears. They are electroacoustic transducers, which convert an electrical signal to a corresponding sound in the user's ear.

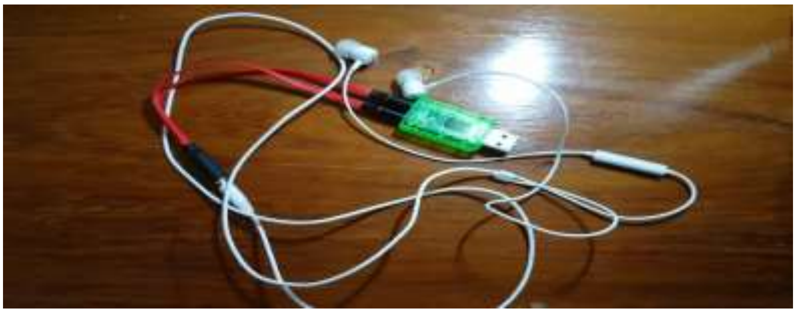


Figure 6.1.1.3 Earphones

6.1.2 Final Prototype

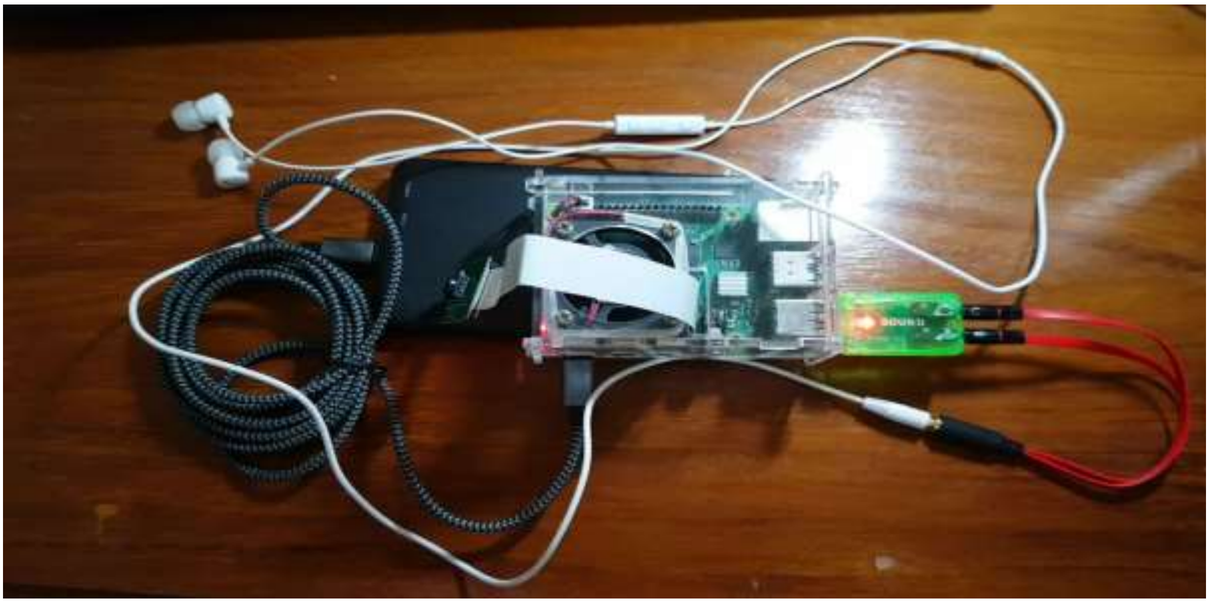


Figure 6.1.2 Prototype1

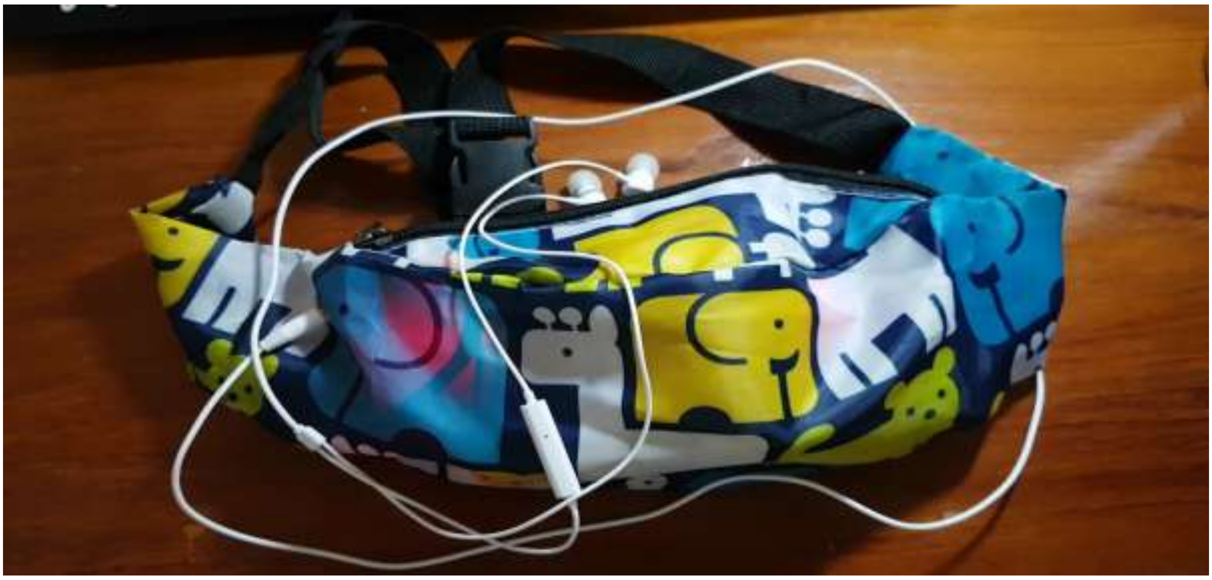


Figure 6.1.2.1 Final Prototype

6.2 MODULES

6.2.1 Accessing the Raspberry Pi from VNC Server and ssh:

The need list: Raspberry Pi, Ethernet cable, Your computer, Wireless Modem to provide your computer an internet access, USB WiFi antenna dongle.

Remote Raspberry Pi Access using SSH (command line)

The first option is to connect to the Raspberry Pi over SSH (Secure Shell). SSH allows you to remotely gain access to the command line of a Raspberry Pi from another computer over the network or the internet (if your network is setup for it). SSH only allow command line, not a full desktop. For a full graphical desktop, VNC is what you want. SSH is enabled by default on Raspbian and can be disabled using Raspi-config. SSH client is built into Linux distributions and Mac OS. For Windows system, the most popular SSH client is PuTTY (putty.exe).
To connect using PuTTY to your Raspberry Pi, you'll need to know its IP address. This can be found from directly from the Pi if you have access to it or from the router if you are too lazy to move.

The default SSH port is **22**
User name is: **pi**
Password: **raspberry**

Remote Raspberry Pi Access using VNC (graphical desktop)

VNC (Virtual Network Computing) is a graphical desktop sharing system that allows you to remotely control the desktop interface of a computer from another. You can use it to share just the graphic or transmit keyboard and mouse events as well, very similar to Windows' Remote Desktop but very difference in the underlying protocol.

First, you'll need to install a VNC package using a monitor or over SSH.

```
sudo apt-get install tightvncserver
```

Once installed, start a TightVNC Server

```
sudo vncserver
```

It will prompt you to enter a password and an optional view-only password. Once the display is created, make a note of the raspberry:#. The # is the display number. Such as:

```
New 'X' desktop is raspberry:1
```

Now we’ll need to install a VNC client. For Linux machine, you can use the xtightvncviewer by

```
sudo apt-get install xtightvncviewer
```

6.2.2 Installing the required Packages

Following command is used to install packages:
pip3 install <packagename>

6.2.3 Features

Convert Speech to Text

We are using two methods, offline and online for speech to text conversion. Offline speech to text is used for operations that have specific words in commands. Online speech to text is used for operations that has nouns like city name, during Wikipedia search or any command that requires a word to be recognized from a large set of words.

Convert Text to Speech

Text to speech conversion is performed using an eSpeak command which is a linux command. eSpeak is a compact open source software speech synthesizer for English and other languages, for Linux and Windows. eSpeak uses a "formant synthesis" method. This allows many languages to be provided in a small size. The speech is clear, and can be used at high speeds, but is not as natural or smooth as larger synthesizers which are based on human speech recordings.

Reading text

Image is captured using pi camera and then characters are identified and is played back to the visually impaired. Once the characters are identified it will be converted from text to speech.

Face Recognition

A face recognition system is a technology capable of identifying or verifying a person from an image. Image is captured using pi camera and saved as the person name. It is done using face recognition packages.

Weather

It will ask which city's weather we want to know. City name will be extracted and sent to the weather API from where we get the weather information of a particular city.

Time

It will tell the current system time.

News

Tells today's news to the user (such as news in India). It will get the top 5 news headlines which will be played back to visually impaired.

Wikipedia Search

When the Wikipedia operation is invoked, the system requests the user to tell the topic on which information is needed. E.g. "tell me about India". It uses regular expression to find out the topic and a search is performed to get information on that particular topic. Information will be played back to the visually impaired.

Image Captioning

For an Input image a combination of convolution neural network is applied to obtain vectoral representation of images. Recurrent neural networks decode those representations word by word into natural language sentences and it will be played back to the visually impaired.

6.2.4 Speech Recognition

Speech recognition is the ability of a machine or program to identify words and phrases in spoken language and convert them to a machine-readable format.

Speech Recognition API is used for both online and offline. Since the surrounding noise varies, we allow the program for a second to adjust the energy threshold of recording so it is adjusted according to the external noise level. There are certain offline Recognition systems such as PocketSphinx which is a lightweight speech recognition engine, specifically tuned for handheld and mobile devices. Commands are recognized and performs the required operations and if the commands are not recognized it will loop back to continue to listen to commands.

6.2.5 Text to Speech Conversion

Text-to-Speech (TTS) refers to the ability to read text and convert it into speech. We have used eSpeak offline text to speech conversion. Text to speech conversion is performed using a eSpeak package. eSpeak converts text to phonemes with pitch and length information. Any text given as argument to the eSpeak command is converted to audio signals which is then played to the user through the earphones.

6.2.6 Character Recognition

We need to load the image using openCV, which is imported as cv2. We import 'Image' from the Pillow library and PyTesseract library. Python-tesseract is an optical character recognition (OCR) tool for python. That is, it will recognize and read the text embedded in images. We then have a user defined function, ocr_core function that takes in a file name and returns the text contained in the image. This function will handle the core OCR processing of images. An image containing text is captured using the pi camera and once the image is captured it will be saved as "ocr.jpg". Characters are identified. Upon identification, the character is converted to text. Text will be played back to the visually impaired through earphones.

6.2.7 Face Recognition

A facial recognition system is a technology capable of identifying or verifying a person from a digital image. Image is captured using pi camera. Until the face is seen by the camera, camera will be on. Face recognition code performs one of the two operations, identifying the face or saving the face. When the command is 'who is this', face recognition package is used to recognize the face. Captured image will be resized so that it will be easy for face recognition. It will be then converted to RGB color from BGR. If new faces are added, then we need to retrain it and find the face encodings or else no training is required. It works by comparing facial features from the captured image with faces stored. Distance from the captured image to all the saved images is found. Minimum distance value will be given to best match index using which we find out the person's name. When the command is 'Save this face', a response will be played back to the visually impaired through headphone, 'Tell the name'. Image is captured using pi camera. Image is resized and converted to RGB color. If the person's name already exists, it will return file already exist. If there are multiple faces detected, the device responds "Multiple faces detected and won't save the face data. If the image has only one face and face data of that person doesn't exist then the face data is stored with the name of the person and return as 'face saved'. It will be played back to visually impaired through earphones.

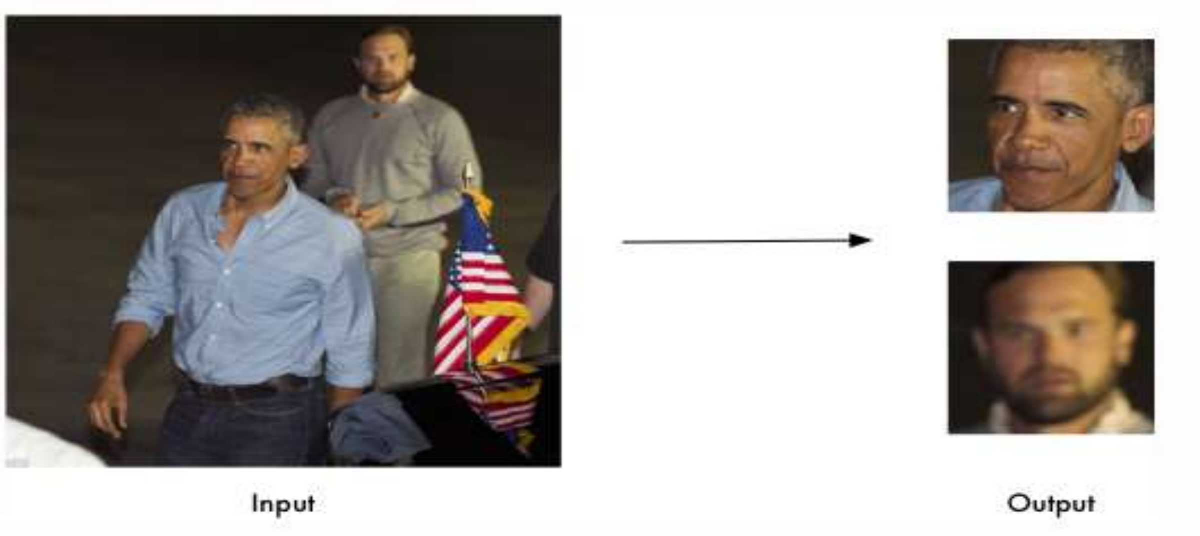


Figure 6.2.7.1 Find all the faces that appear in a picture

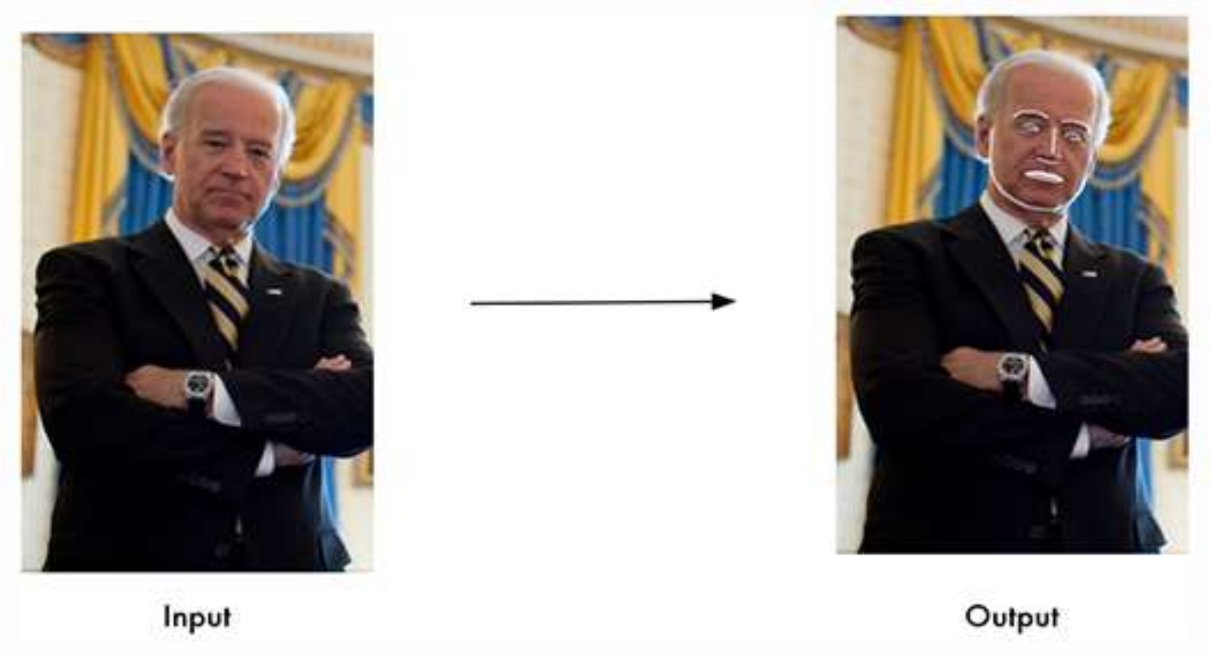


Figure 6.2.7.2 Get the locations and outlines of each person's eyes, nose, mouth and chin.

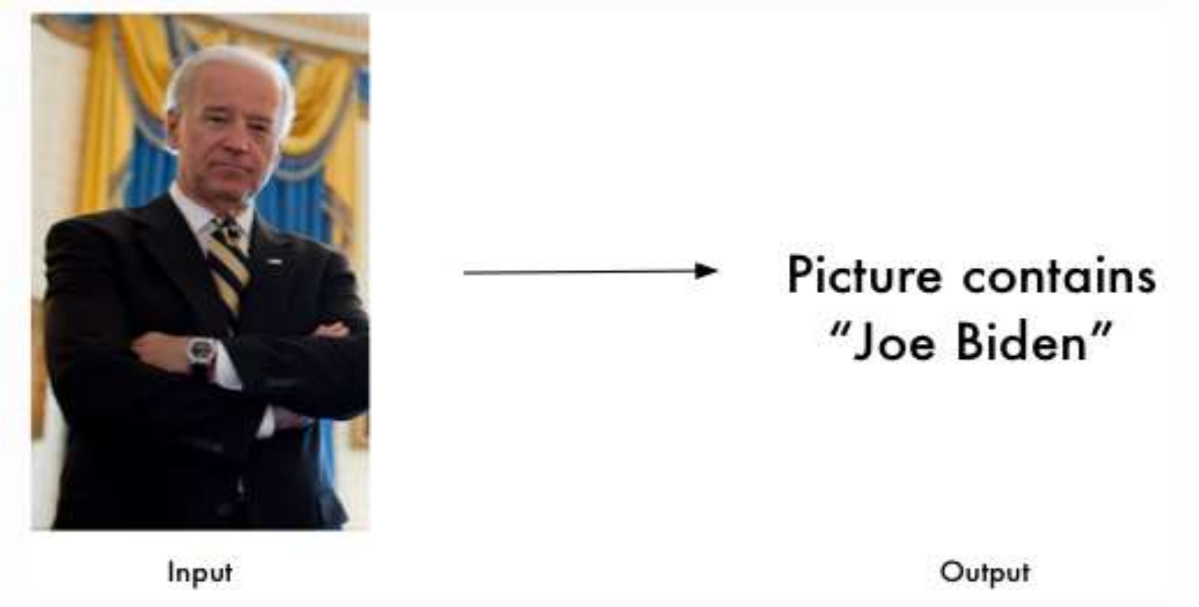


Figure 6.2.7.3 Recognize who appears in each photo.

6.2.8 Weather

First, we need to import the re (Regular Expression) library into the script with import re. When the command given invokes weather operations, a response “Which City” will be played back through earphones to the visually impaired. If the user tells “Udupi”, here whichever word spoken is taken as a keyword. We are using web API called OpenWeatherMap to which the city name will be sent as an input. City name is sent to API and then we get all the weather information of that city. We get maximum and minimum temperature of city in Celsius along with the current weather which will be played back to the visually impaired through earphones.

6.2.9 News

When the command given is “news”. The urllib.request package helps in opening urls and it opens ‘https://news.google.com/rss’ website. Content of the news will be in xml format. We are converting the information received to xml file using BeautifulSoup. Later we are extraction value from the key "item" which gives us the headlines of all the news articles extracted from the website. The code extracts only the top 5 news headlines from a list of headlines and plays them to the user.

6.2.10 Wikipedia

First, we need to import the Regular Expression package into the script with import re. ‘Wikipedia’ is a Python library that makes it easy to access and parse data from Wikipedia. When the command given is to perform a Wikipedia Search, A response will be played back through earphones which is “What do you want to search in Wikipedia” to the visually impaired. If the user tells “tell me about India”, here whichever word comes after “about” is taken as a keyword. ‘India’ will be taken as a topic to be searched. The information will be given to AIResponse which will be played back to the visually impaired.

6.2.11 Image Captioning

We used an attention-based model that automatically learns to describe the content of an images.

- For an Input image a combination of convolution neural network is applied to obtain vectoral representation of images. Recurrent neural networks decodes those representations word by word into natural language sentences. Model learns a word/ image alignment.
- As the model generates each word, its attention changes to reflect the relevant parts of the image.
- “soft” deterministic attention mechanism trainable by standard back-propagation methods and “hard” stochastic attention mechanism trainable by maximizing an approximate variational lower bound.
- We can gain insight and interpret the results of this framework by visualizing “where” and “what” the attention focused on.
- We quantitatively validate the usefulness of attention in caption generation. Our model takes a single raw image and generates a caption y encoded as a sequence of 1-of-K encoded words. $y = \{y1, . . . , yC \}$, $yi \in R$, where K is the size of the vocabulary and C is the length.
- We use a long short-term memory (LSTM) network that produces a caption by generating one word at a time.

6.2.12 Implementation

```
r = sr.Recognizer()
with sr.Microphone() as source:
        print('Please Wait....Sensing Ambient Noise!')
        #r.pause_threshold =  1
        r.adjust_for_ambient_noise(source, duration=2)

def subCommand():
    "listens for commands"
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print('Say something...')
        r.pause_threshold = 1
        r.adjust_for_ambient_noise(source, duration=1)
        audio = r.listen(source)
    try:
        command = r.recognize_google(audio).lower()
        print('You said: ' + command + '\n')
    #loop back to continue to listen for commands if unrecognizable speech is received
    except sr.UnknownValueError:
        print('....')
        command = myCommand();
    return command

def AIResponse(audio):
    "speaks audio passed as argument"
    newaudio=""
    if audio!='':
        if audio[0]=='b':
            audio=audio[1:]
    for char in audio:
        if char!='\'' and char!='"' and char!="\\":
newaudio+=char
    print(newaudio)
    for line in newaudio.splitlines():
        os.system("espeak '" + line+"'")
```

```
def myCommand():
    "listens for commands"
    model_path = get_model_path()
    #print(model_path)
  try:
    speech = LiveSpeech(
        verbose=False,
        sampling_rate=16000,
        buffer_size=2048,
        no_search=False,
        full_utt=False,
        hmm=os.path.join(model_path, 'en-us'),
        lm='./speech/5564.lm',
        dic='./speech/5564.dic'
    )
except:
        print("Error occured in LiveSpeech")
        speech=''
    for phrase in speech:
        print(str(phrase).lower())
        print(len(str(phrase).lower()))
        #print(str(phrase).lower())
        try:
            assistant(str(phrase).lower())
        except:
            print("Error occured in commands!")
        return

def assistant(command):
    "if statements for executing commands"
     print(command)
    if 'shutdown' in command:
        AIResponse('Bye bye Sir. Have a nice day')
       sess.close()
        sys.exit()
```

```
#greetings
    elif 'hello' == command:
        day_time = int(strftime('%H'))
        if day_time < 12:
            AIResponse('Hello Sir. Good morning')
        elif 12 <= day_time < 18:
            AIResponse('Hello Sir. Good afternoon')
        else:
            AIResponse('Hello Sir. Good evening')
    elif 'help me' == command:

        AIResponse(
        You can use these commands and I'll help you out:
        1. Can you tell the weather : Tells you the current condition and temperature
        2. Can you tell me the news : Tells you the news in India
        3. What is the time : Tells you the system time
        4. Wikipedia Search : Gives you information from wikipedia
        5. Save this face : Saves the face image for face recognition
        6. Who is this : Tells you the persons name
        7. What is in front of me : Tells you what is in front of you
        )


#top stories from google news
    elif 'news' in command and 'tell' in command:
        try:
            news_url="https://news.google.com/news/rss"
            Client=urlopen(news_url)
            xml_page=Client.read()
            Client.close()
            soup_page=soup(xml_page,"xml")
            news_list=soup_page.findAll("item")
            for news in news_list[:5]:
                AIResponse(str(news.title.text.encode('utf-8')))
        except Exception as e:
                print(e)
```

```
#current weather
    elif 'the weather' in command and 'tell' in command:
        AIResponse('Which City')
        scommand = subCommand()
        try:
            reg_ex = re.search('(.*)', str(scommand))
            if reg_ex:
                city = reg_ex.group(1)
                owm = OWM(API_key='ab0d5e80e8dafb2cb81fa9e82431c1fa')
                obs = owm.weather_at_place(city)
                w = obs.get_weather()
                k = w.get_status()
                x = w.get_temperature(unit='celsius')
                AIResponse('Current weather in %s is %s. The maximum temperature is %0.2f and the minimum temperature is %0.2f degree celcius' % (city, k, x['temp_max'], x['temp_min']))
        except Exception as e:
            print(e)
#time
    elif 'what is the time' in command or 'what is the current time' in command:
        import datetime
        now = datetime.datetime.now()
        AIResponse('Current time is %d hours %d minutes' % (now.hour, now.minute))
#askme anything
    elif 'wikipedia search' == command:
        AIResponse('Want do you want to search in wikipedia')
        scommand=subCommand()
        reg_ex = re.search('about (.*)', scommand)
        try:
            if reg_ex:
                topic = reg_ex.group(1)
                ny = wikipedia.page(topic)
                AIResponse("\""+str(ny.content[:500].encode('utf-8'))+"\"")
        except Exception as e:
                print(e)
                AIResponse(str(e))
```

```
#face save
    elif 'save this face' in command:
        AIResponse('Tell a name')
        scommand = subCommand()
        reply = face_store(str(scommand))
        AIResponse(reply)

    elif 'who is this' in command:
        face_names = face_recog()
        if len(face_names)>0:
            for face in face_names:
                AIResponse(face)
    elif 'read this' in command:
        reply = ocr_core()
        AIResponse(reply)
#AIResponse('Hi User, I am Drishti and I am your personal voice assistant, Please give a command or say "help me" and I will tell you what all I can do for you.')
#loop to continue executing multiple commands
AIResponse('Hi')
while True:
    myCommand()
```

CHAPTER 7

SYSTEM TESTING

7.1 INTRODUCTION

System testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system. Testing phase is performed after coding to detect all the errors and provide quality assurance and ensure reliability of the software. Testing is vital to the success of the system. During testing, the software to be tested is executed with a set of test cases, and the behavior of the system for the test cases is evaluated to determine if the system is performing as expected. Clearly the success of testing in revealing errors depends critically on the test cases.

7.2 UNIT TESTING

Unit testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

- Unit testing increases confidence in changing/ maintaining code. If good unit tests are written and if they are run every time any code is changed, we will be able to promptly catch any defects introduced due to the change.

- Also, if codes are already made less interdependent to make unit testing possible, the unintended impact of changes to any code is less.

- Codes are more reusable. In order to make unit testing possible, codes need to be modular. This means that codes are easier to reuse.

Development is faster. If you do not have unit testing in place, you write your code and perform that fuzzy ‘developer test’ (You set some breakpoints, fire up the GUI, provide a few inputs that hopefully hit your code and hope that you are all set.) But, if you have unit testing in place, you write the test, write the code and run the test. Writing tests takes time but the time is compensated by the less amount of time it takes to run the tests; You need not fire up the GUI and provide all those inputs. And, of course, unit tests are more reliable than ‘developer tests’. Development is faster in the long run too. How? The effort required to find and fix defects found during unit testing is very less in comparison to the effort required to fix defects found during system testing or acceptance testing.

- The cost of fixing a defect detected during unit testing is lesser in comparison to that of defects detected at higher levels. Compare the cost (time, effort, destruction, humiliation) of a defect detected during acceptance testing or when the software is live.
- Debugging is easy. When a test fails, only the latest changes need to be debugged. With testing at higher levels, changes made over the span of several days/weeks/months need to be scanned.

7.3 INTEGRATION TESTING

Integration Testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

Need of Integration Testing:

Although each software module is unit tested, defects still exist for various reasons like:

- A Module, in general, is designed by an individual software developer whose understanding and programming logic may differ from other programmers. Integration Testing becomes necessary to verify the software modules work in unity.
- At the time of module development, there are wide chances of change in requirements by the clients. These new requirements may not be unit tested and hence system integration Testing becomes necessary
- Interfaces of the software modules with the database could be erroneous
- External Hardware interfaces, if any, could be erroneous
- Inadequate exception handling could cause issues.

7.4 ACCEPTANCE TESTING

Acceptance testing is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system’s compliance with the business requirements and assess whether it is acceptable for delivery.

There are various forms of acceptance testing:

- User acceptance Testing
- Business acceptance Testing
- Alpha Testing
- Beta Testing

CHAPTER 8

RESULTS AND DISCUSSSIONS

8.1 RESULTS

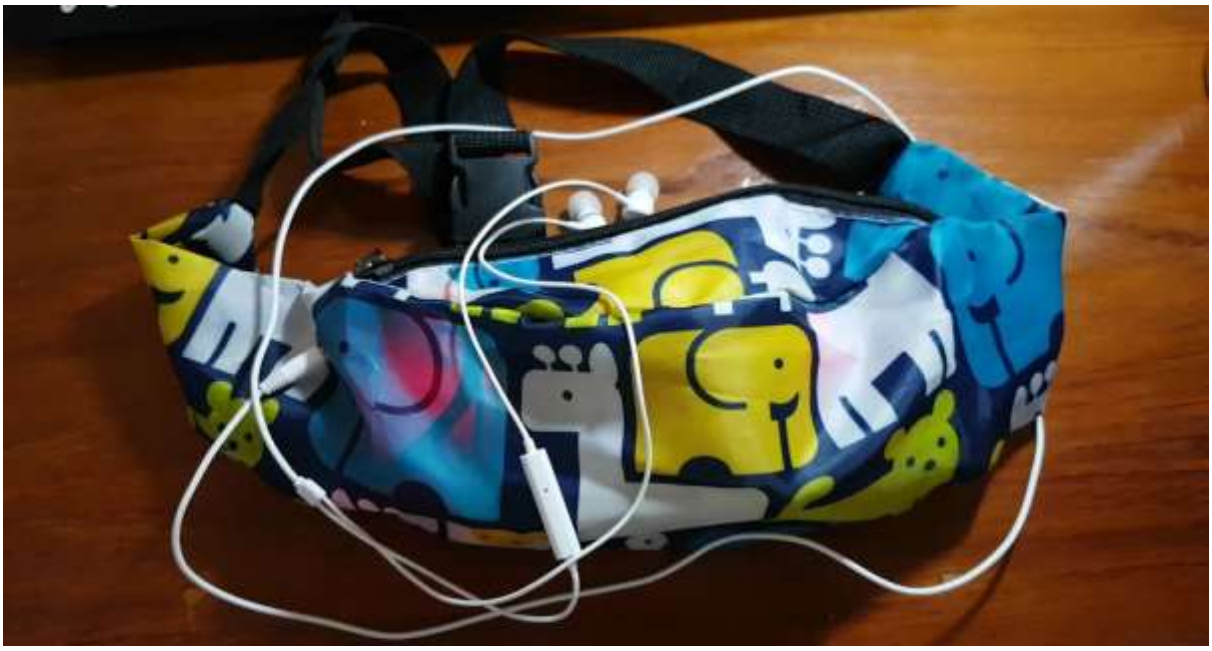
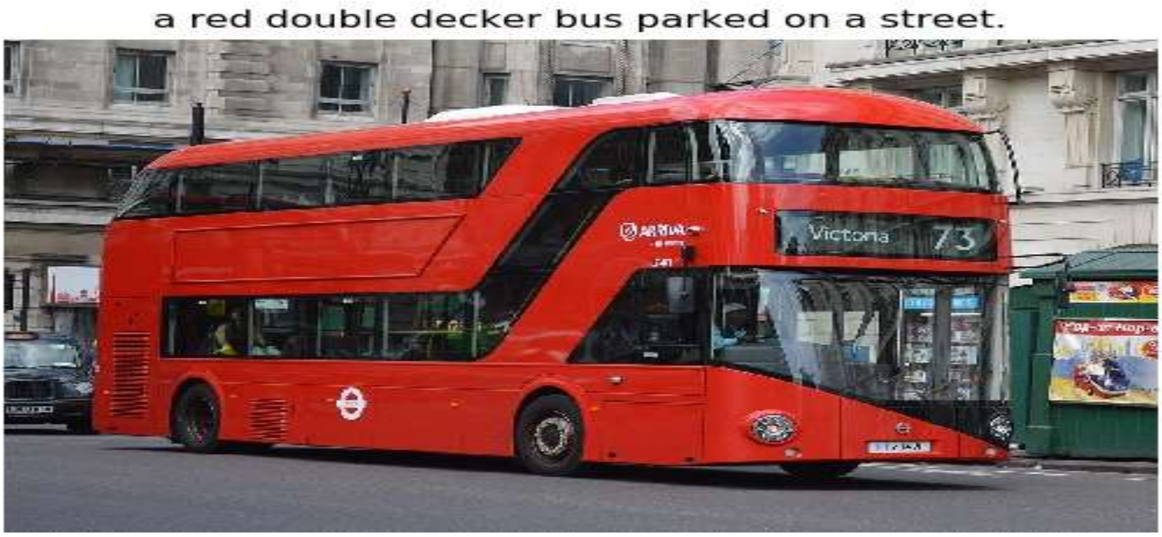
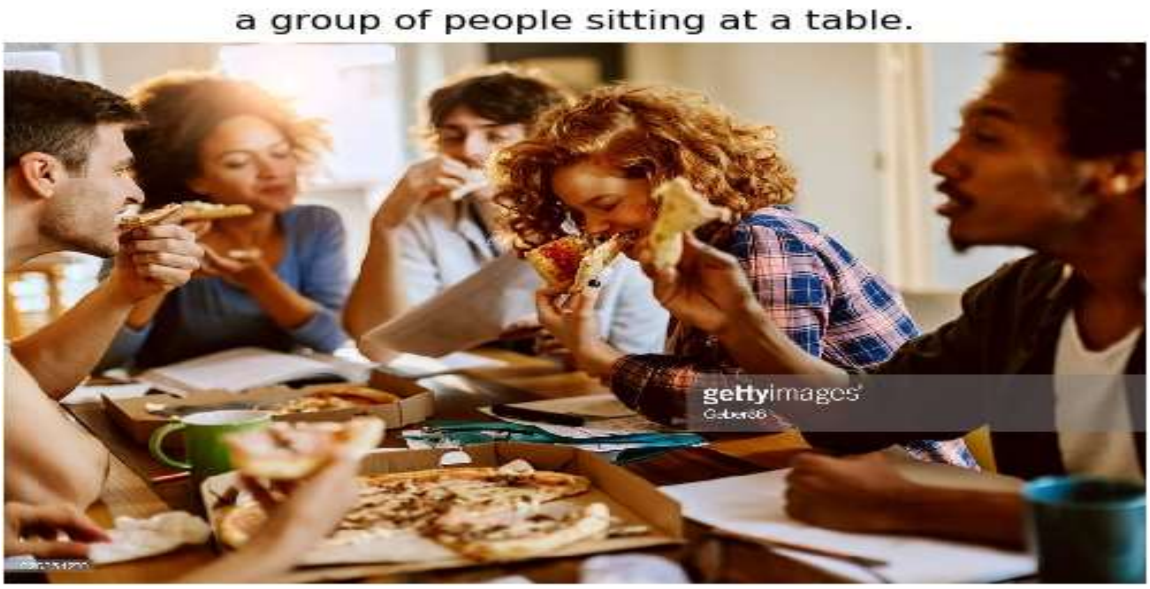


Figure 8.1.1 Final Prototype



a red double decker bus parked on a street.

Figure 8.1.2 Output1 of Image Captioning



a group of people sitting at a table.

Figure 8.1.3 Output2 of Image Captioning

8.2 DISCUSSIONS

In this project, we proposed new methods which combines the key aspects of some useful methods and added some extra capabilities for assisting the visually impaired. This project may solve some of major problems of visually impaired that are still existing. Many of them can’t even walk without the help of others. Their life always depends upon their caregivers and can be quite difficult for them alone. People with visual impairment face various problems in their daily life as the modern assistive devices are often not meeting the consumer requirements in terms of price and level of assistance. This project presents a new design of assistive smart glasses for visually impaired.

The main purpose of the project is to help visually impaired to recognize the objects in front of them through Speech to text, Convert the information to voice to be understood by visually impaired via headphone, Read a book, Face recognition. This project helps visually impaired become self-reliance and independent.

CHAPTER 9

CONCLUSION AND FUTURE WORK

9.1 CONCLUSION

Visually impaired people are those who are either totally blind or having a very low vision that is legally considered as blindness. This project is intended to help this type of people to widen their scope of independence by giving them a description of the live scenes delivered in an audio format using an earpiece. The project is implemented using Python 3.7 as a main software program, and the low cost single board computer the raspberry pi 4 as a platform. The project is also using the raspberry pi camera to capture real time images and an earpiece to voice out the descriptions. This smart glasses are practically feasible device and can be conveniently carried/used by any visually impaired.

9.2 FUTURE WORK

As a recommendation for future work, more additional assistance feature should be added. As the camera quality is a very important component, therefore, as a recommendation a better camera can be added to the system to improve the results accuracy. To deliver the final product which can be worn by a visually impaired person.

REFERENCES

1. Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, Yoshua Bengio. ICML. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

2. Nickolay Shmyrev. (2009). CMUSphinx Open Source Speech Recognition Toolkit

3. Raspberry Pi Documentation

4. Orcam MyEye 2 Documentation. (2020)

5. Tesseract OCR Documentation

6. Adam Geitgey. (2020). Face Recognition Documentation

7. David Huggins-Daines. (2015). PocketSphinx API Documentation