ASSISTIVE DEVICE DESIGNED FOR DEAF, DUMB, AND BLIND PEOPLE

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF TECHNOLOGY

IN

Department of Electronics and Communication

Submitted by:

Deepanshi Nagarwal (2K20/EC/076)

Bhanu Pratap (2K20/EC/065)

Under the supervision of

Prof.(Dr.) Rajeev Kapoor

DEPT. OF ELECTRONICS AND COMMUNICATION

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

MAY 2024

DEPT. OF ELECTRONICS AND COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

We, Deepanshi Nagarwal (2K20/EC/076), & Bhanu Pratap(2K20/EC/065) students of B.Tech in Electronics and Communication, hereby declare that the Project Dissertation titled — "Assistive Device Designed for Deaf Dumb and Blind people" which is submitted by us to the Department of Electronics and Communication, DTU, Delhi in fulfillment of the requirement for awarding of the Bachelor of Technology degree, is not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma, Fellowship, or other similar title or recognition.

Place: New Delhi	Deepanshi Nagarwal	Bhanu Pratap
Date: 31/05/2024	(2K20/EC/076)	(2K20/EC/065)

DEPT. OF ELECTRONICS AND COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project titled ”Assistive Device Designed for Deaf, Dumb and Blind People” which is submitted by Deepanshi Nagarwal (2K20/EC/076), & Bhanu Pratap (2K20/EC/065) for the fulfillment of the requirements for awarding of the degree of Bachelor of Technology (B.Tech) is a record of the project work carried out by the students under my guidance & supervision. To the best of my knowledge, this work has not been submitted in any part or fulfilments for any Degree or Diploma to this University or elsewhere.

Prof.(Dr) Name Of supervisor

Place: New Delhi

Dr Rajeev Kapoor

Date: 31/05/2024

Professor

Department of Electronics and Communication

Delhi Technological University

ABSTRACT

One of the most precious gifts to a human being is the ability to see, listen, speak, and respond according to the situation. But some unfortunate ones are deprived of this. Making a single compact device for people with Visual, Hearing and Vocal impairment is a tough job. Communication between deaf-dumb and normal people has been always a challenging task. This paper proposes an innovative communication system framework for deaf, dumb, and blind people in a single compact device. We provide a technique for a blind person to read a text and it can be achieved by capturing an image through a camera that converts a text to speech (TTS). It provides a way for deaf people to read a text by speech-to-text (STT) conversion technology. Also, it provides a technique for dumb people using text-to-voice conversion. The system is provided with four switches and each switch has a different function. The blind people can be able to read the words using by Tesseract OCR (Online Character Recognition), the dumb people can communicate their message through text which will be read out by espeak, and the deaf people can be able to hear other's speech from text. All these functions are implemented by the use of Raspberry Pi.
Keywords: Raspberry Pi, Tesseract OCR (Online Character Recognition), espeak, Speech to text (STT), Text to Speech (TTS).

ACKNOWLEDGEMENT

The successful completion of any task is incomplete and meaningless without giving any due credit to the people who made it possible without which the project would not have been successful and would have existed in theory.

First and foremost, we are grateful to Dr. O P Verma, HOD, Department of Electronics and Communication Engineering, Delhi Technological University, and all other faculty members of our department for their constant guidance and support, constant motivation, and sincere support and gratitude for this project work. We owe a lot of thanks to our supervisor, Dr. Rajeev Kapoor, Professor, Department of Electronics and Communication Engineering, Delhi Technological University for igniting and constantly motivating us and guiding us in the idea of a creatively and amazingly performed Major Project in undertaking this endeavor and challenge and also for being there whenever we needed his guidance or assistance.

We would also like to take this moment to show our thanks and gratitude to one and all, who indirectly or directly have given us their hand in this challenging task. We feel happy and content in expressing our vote of thanks to all those who have helped us and guided us in presenting this project work for our Major Project. Last, but never least, we thank our well-wishers and parents for always being with us, in every sense and constantly supporting us in every possible sense whenever possible.

Bhanu Pratap
(2K20/EC/065)

Deepanshi Nagarwal
(2K20/EC/076)

Contents

LIST OF TABLES

List of Figures

Chapter 1

INTRODUCTION

1.1 Overview

Approximately 285 million people are judged to be visually impaired worldwide of which 39 million are blind and 246 are said to have low vision. Approximately 90% of this world's visually impaired people the dispirited income people and 82% of people living with blindness aging persons and above. The number of people visually impaired from eye-related diseases has been brought down in the past 20 years according to global estimated work. In which 80% of all visual restitution can be prevented or cured. India is considered to be the home of the world's largest act of blind people. In this world, about 37 million are blind, of which 15 million are from India. So many
researchers have been getting along in this universe, but the visual impairment could not be broken for good. In lodge to facilitate these people we have developed an assistive device for blind people who do not want the assistance of other neighbours. The development of our project helps the multitude to experience loss and go independently.

All around the world, about 9.1 billion people are deaf and mute. In their daily life, they face plenty of problems with their communication. Sign language is a linguistic process that is employed for communication among normal people and handicapped people. Sign language relies on sign patterns such as the body language of the person and movements of the arm to facilitate the discernment between the great and unwashed. The deaf and vocally impaired people don't simply have to learn the customized sign language, but the core issue is that they can communicate with the usual sort of multitude in society. It is similarly impossible for all the masses to learn sign language to understand what is said through gestures. Therefore, the communication gaps still exist between the deaf and dumb people. Dumber people can simply tilt the message by sign language which could not be understandable by other people. In resolving these difficulties with visually and vocally impaired people we have used the tiny credit card-sized computer named Raspberry Pi. With this device, we provide the solution for blind, deaf, and dumb people. For blind people, the image is converted to voice by using Tesseract software, deaf people receive their content by message as soon as the opposite person speaks out it is displayed as a message. The dumb persons conveyed their message through text instead of sign language which is delivered via e-speak. We have provided the necessary steps to resolve the problems of those masses.

1.2 Introduction to the research topic

Visually impaired people require support with regular tasks including navigating, detecting obstacles, and maintaining safety, especially indoors and outdoors. As a result of the advancement of assistive technology, their lives have become substantially more convenient.

All of these assistance devices evolved from the original white cane. Even during the pandemic, assistive technology has continued to help visually impaired people. The gradual improvement of these visual impairment devices has made them a topic of conversation. This study examines contemporary assistive devices and technologies for the visually handicapped. These techniques are classified according to their functions and objectives. These are the key contributions:

- A thorough division of assistive technology into categories
- A detailed evaluation of every category of assistive devices
- A profound understanding of the features of assistive technology
- An exploration of the necessary tests and experiments on the development of assistive devices
- A discussion of the benefits of assistive devices powered by renewable energy sources
- An analysis of how the development of assistive equipment has changed throughout time, comparing the pre-and post-pandemic periods of COVID-19
- An overview of the creation of assistive technology for visually impaired people

PROBLEM STATEMENT

- In the earlier days blind people are catered to the basic learnings of the braille system.
- The language in the braille will go from left to right across the page, just like printed words

1.3 Background and Context

The term "assistive device" is a catch-all phrase that refers to both hardware and software that helps people with disabilities to use technology in a way that improves their quality of life. It is feasible to preserve a person's degree of functional competence by making use of this. These devices each have a sensor that collects data, which is then processed by a central processing unit (CPU) or a microcontroller. Depending on the device, the user could perceive either sound vibration, or both of these sensations.

Based on different functionalities, working principles, modalities, and uses, the current assistive devices can be classified into the four categories below:
1 Based on portability
2 Based on navigation
3 Based on detection
4 Based on smartphone assistance

. The four primary categories have been segmented further based on their functions. The first topic covered is the classification of assistive technology based on portability. Two sections examining the portability of wearable and nonwearable devices are devoted to wearable and non-wearable devices, respectively. Other technologies, such as those worn by blind or visually impaired people on their heads, ears, belts, feet, and hands, are wearable. These tools collect and process information from the environment for visually impaired people and then provide feedback via vibration, sound, or echo waves. The navigation category is shown here. These devices calculate the distance between a user and an impediment using non-vision sensors to direct the user to follow a secure path. In general, all indoor, outdoor, indoor and outdoor, and audio-tactile map navigation techniques fall under this category. A full smart city may be navigated via outdoor navigation, including the streets of the city, the airport, shopping malls, and pedestrians. Now it's time to classify things according to detection. The device may need to detect any impediments, such as a car, a staircase, a pedestrian, or an everyday object. However, this category also covers the recognition of gestures and additional items like characters, faces, emotions, and money

The factors that are essential for evaluating the utility and dependability of visual impairment assistive devices are referred to as features. This paper's features help users choose a suitable device. The performance of a device depends on:
1. Capturing Devices
2. Working hours

3. Response time
4. Coverage area
5. Feedback
6. Working range
7. Weight
8. Robustness
9. Cost

1.3.1 Background research

The physical details of assistive devices do not make up the full of the devices themselves; software plays a vital role in selecting how to develop assistive devices for a particular circumstance, cause the term “assistive device” is a catch-all phrase that refers to both hardware and software. Algorithms that are used for object detection, image classification, image segmentation, character recognition, feature extraction, and edge detection play an important role in making an assistive device used for fulfilling particular purposes.

Table 1

Table 2

Table 3

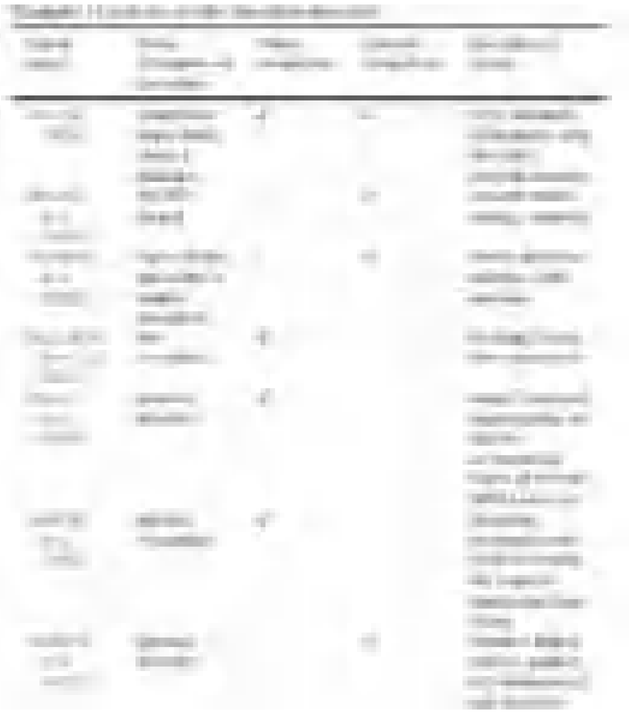


Table 4

1.3.2 User Survey

Making new assistive devices to help visually impaired people must be going on. Assistive devices are made by researchers who follow a definite work pathway. Despite developments in technologies and an extensive range of options for assistive devices, their widespread use and user approval remain limited. In the future, we should continue researching assistive devices. This section offers recommendations for future designers of assistive devices for the visually impaired:

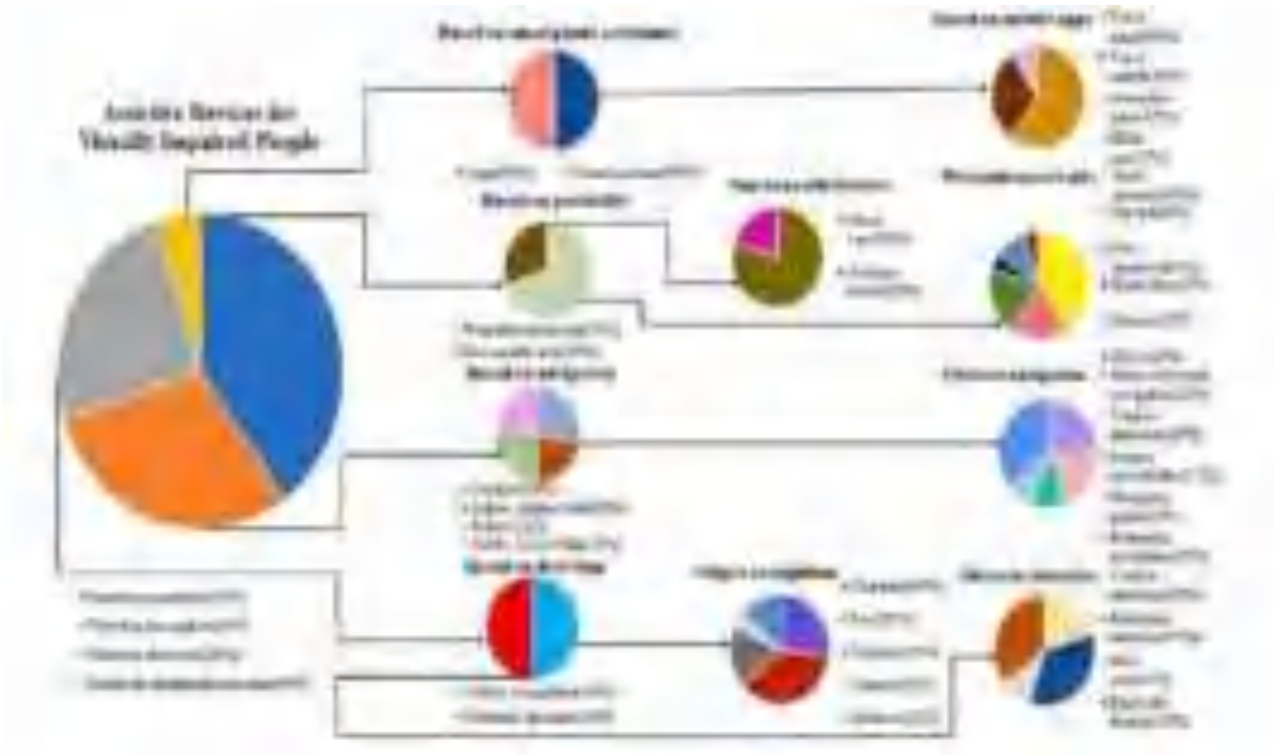


Fig 1 records all the assistive devices existing being used

1.4 Importance of Assistive Devices for Individuals with Sensory

As of late, people tend to use renewable energy sources instead of outdated, hazardous batteries. Such a renewable and sustainable energy source is solar energy. Any device that uses solar panels instead of or in addition to batteries is laudable.

Even during the pandemic, assistive technology has advanced at a sustainable rate, as evidenced by the creation of smart sanitizers for these individuals.

1.5 Organization of the Thesis

One of the most precious gifts to a human being is the ability to see, listen, speak, and respond according to the situation. But some unfortunate ones are deprived of this. Making a single compact device for people with Visual, Hearing, and Vocal impairment is a tough job. Communication between deaf-dumb and normal people has always been a challenging task. This paper proposes an innovative communication system framework for deaf, dumb, and blind people in a single compact device. We provide a technique for a blind person to read a text and it can be achieved by capturing an image through a camera that converts a text to speech (TTS). It provides a way for deaf people to read a text by speech-to-text (STT) conversion technology. Also, it provides a technique for dumb people using text-to-voice conversion. The system is provided with four switches and each switch has a different function. The blind people can be able to read the words using Tesseract OCR (Online Character Recognition), the dumb people can communicate their message through text which will be read out by speech, the deaf people can be able to hear others' speech from text. All these functions are implemented by the use of Raspberry Pi.

Fig 3: Step-by-step process to design an assistive device

2
State of art assistive technologies

2.1 Portable Camera-Based Assistive Text and Product Label Reading From Hand-Held Objects for Blind Persons

We propose a camera-based assistive text reading framework to help blind persons read text labels and product packaging from hand-held objects in their daily lives. To isolate the object from cluttered backgrounds or other surrounding objects in the camera view, we first propose an efficient and effective motion-based method to define a region of interest (ROI) in the video by asking the user to shake the object. This method extracts moving object regions by a mixture-of-Gaussians-based background subtraction method. In the extracted ROI, text localization and recognition are conducted to acquire text information. To automatically localize the text regions from the object ROI, we propose a novel text localization algorithm by learning gradient features of stroke orientations and distributions of edge pixels in an Adaboost model. Text characters in the localized text regions are then binarized and recognized by off-the-shelf optical character recognition software. The recognized text codes are output to blind users in speech. The performance of the proposed text localization algorithm is quantitatively evaluated on ICDAR-2003 and ICDAR-2011 Robust Reading Datasets. Experimental results demonstrate that our algorithm achieves the state of the arts. The proof-of-concept prototype is also evaluated on a dataset collected using ten blind persons to evaluate the effectiveness of the system's hardware. We explore user interface issues and assess the robustness of the algorithm in extracting and reading text from different objects with complex backgrounds.

2.2 Vision-Based Assistive System for Label Detection with Voice Output

A camera-based assistive text reading framework to help blind persons read text labels and product packaging from hand-held objects in their daily resides is proposed. To isolate the object from cluttered backgrounds or other surrounding objects in the camera view, we propose an efficient and effective motion-based method to define a region of interest (ROI) in the video by asking the user to shake the object. In the extracted ROI, text localization and recognition are conducted to acquire text information. To automatically localize the text regions from the object ROI, we propose a novel text localization algorithm by learning gradient features of stroke orientations and distributions of edge pixels in an Adaboost model. Text characters in the localized text regions are then binarized and recognized by off-the-shelf optical character recognition software. The recognized text codes are output to blind users in speech.

2.3 Assistive Translator for Deaf & Dumb People

Communication between a deaf-mute and a normal person has always been a challenging task. The project aims to facilitate people using a glove-based deaf-mute communication interpreter system. The glove is internally equipped with five flex sensors, tactile sensors and accelerometer. For each specific gesture, the flex sensor produces a proportional change in resistance and accelerometer measures the orientation of hand. The output from the sensor is analog values it is converted to digital. The processing of these hand gestures is in ARM processor. Processor compares the input signal with predefined voltage levels stored in memory. According to that required output sound is produced which is stored is SPI memory with the help of speaker. In such a way it is easy for deaf and dumb to communicate with normal people

2.4 Voice Assisted Text Reading System for Visually Impaired Persons Using TTS Method

In the survey American Foundation for Blind 2014, is observed that there are 6.8 trillion people are visually impaired and they still find difficult to roll their day today life it is important to take necessary measure with the emerging technologies to help them to live the current world irrespective of their impairments.In the motive of supporting the visually impaired, a method is proposed to develop a self-assisted text to speech module in order to make them read and understand the text in an easier way. It is not only applicable for the visually impaired but also to any normal human beings who are willing to read the text as a speech as quickly as possible.A finger mounted camera is used to capture the text image from the printed text and the captured image is analyzed using optical character recognition (OCR). A predefined dataset is loaded in order to match the observed text with the captured image. Once it is matched the text is synthesized for producing speech output. The main advantage of proposed method is that, it reduces the dataset memory required for the comparison since only character recognition is being done. The same work is simulated using Matrix laboratory (MATLAB) simulator software for the performance analysis of proposed work for various input sets

2.5 Portable Camera-Based Assistive Text and Product Label Reading From Hand-Held Objects for Blind Persons

We propose a camera-based assistive text reading framework to help blind persons read text labels and product packaging from hand-held objects in their daily lives. To isolate the object from cluttered backgrounds or other surrounding objects in the camera view, we firs propose an efficient and effective motion-based method to define a region of interest (ROI) in the video by asking the user to shake the object. This method extracts moving object region by a mixture-of-Gaussians-based background subtraction method. In the extracted ROI, text localization and recognition are conducted to acquire text information. To automatically localize the text regions from the object ROI, we propose a novel text localization algorithm by learning gradient features of stroke orientations and distributions of edge pixels in an Adaboost model. Text characters in the localized text regions are then binarized and recognized by off-the-shelf optical character recognition software.The recognized text codes are output to blind users in speech. Performance of the proposed text localization algorithm is quantitatively evaluated on ICDAR-200 3and ICDAR-2011 Robust Reading Datasets. Experimental results demonstrate that our algorithm achieves the state of the arts. The pro concept prototype is a ls evaluated on a dataset collected using ten blind persons to evaluate the effectiveness of the system’s hardware. We explore user interface issues and assess robustness of the algorithm in extracting and reading text from different objects with complex backgrounds.

2.6 Assistive Translator for Deaf & Dumb People

Communication between a deaf-mute and a normal person has always been a challenging task. The project aims to facilitate people using a glove-based deaf-mute communication interpreter system. The glove is internally equipped with five flex sensors, tactile sensors, and an accelerometer. For each specific gesture, the flex sensor produces a proportional change in resistance and the accelerometer measures the orientation of the hand. The output from the sensor is analog values it is converted to digital. The processing of these hand gestures is in the ARM processor. The processor compares the input signal with predefined voltage levels stored in memory. According to that required output sound is produced which is stored is SPI memory with the help of the speaker. In such a way it is easy for deaf and dumb to communicate with normal people.

2.7 Design of Communication Interpreter for Deaf and Dumb Person

In this paper, we describe a gesture-based device for a deaf and dumb person as communication for a person, who cannot hear is visual, not auditory. Generally, dumb people use sign language for communication, but they find difficulty in communicating with others who don't understand sign language. So there is a barrier to communication between these two communities. This work aims to lower this barrier in communication. The main aim of the proposed project is to develop a cost-effective system that can give voice to voiceless people with the help of Smart Gloves. With the proposed work sign language is converted into text and speech using a flex sensor and microcontroller. It means that by using smart gloves communication will not be a barrier between two different communities

Table 5

Sign language teaching technologies

2.8 Gesture Recognition System for Physically Challenged Persons

The physically challenged persons are one of the excluded sections of society and the present study is conducted among the physically challenged persons about their problems in their daily life. disable people are people with physical or mental incapacities. There are four types of physically challenged, they are: physically handicapped, blind, deaf, and dumb. Disabled people sometimes have difficulty in doing things that other people may take for granted, such as traveling on public transport, climbing stairs are even using some household appliances to communicate with others. Communication barriers are experienced by people who have disabilities that affect hearing, speaking, or understanding There are different ways to communicate with those people, hence our paper aims to reduce barriers to communication by developing an assistive device for the physically challenged using Bluetooth-based android application and image processing.

Technologies based on gesture recognition for the translation of sign language into speech and vice versa.

2.9 Existing system

In the earlier days, blind people catered to the basic learnings of the Braille system. Braille is a scheme of rising symbols that the great unwashed who are blind or partially- sighted have been employed worldwide for over 150 years. The language in the Braille will go from left to right across the page, just like printed words.

Nowadays Braille users can read 1 computer screens and other electronics support using refreshable braille displays. Traditionally, the gesture recognition method was divided into two categories namely vision-based and sensor-based methods. In the vision-based method, the computer camera is an input device for various gestures of hands and figures. In sensor-based systems, gloves are used which can achieve the accurate positions of hand gestures. Lots of studies have been done on sensor-based approaches like gloves, helmets, etc. But wearing it continuously is not possible. Therefore further work is concentrated on image-based approaches. The paper "Intelligent Sign Language Recognition Using Image Processing" deals with the computer system in which sign language is captured processed and translated to speech. For deaf people, speech is analyzed and converted to sign language on screen. Sign language can be understood through devices like sign language translators and electronic gloves.

DIS ADVANTAGES

- Accury will be less
- Speed is low
- Approximate output will be printed.

2.10 Evolution of Assistive Device

We sought to include only research articles on deafness as a condition to be studied and on the use of assistive technologies to improve the interpretation, communication, and social interaction between Deaf and hearing communities.

We did not consider publications other than articles because, as they might be in the prototype phase and lack validity if they are not tested in daily care settings. The publication period and language were not taken into account.

Table 6 exclusive and inclusive criteria to do an analysis indepth.

we were able to determine that most of the proposed technologies were based on *gesture recognition for the translation of sign language into speech and vice versa*. These tools favor the communication between Deaf people and hearing people who do not use sign language, so they can be implemented in educational, work, and social contexts. This type of technology has received particular interest from researchers because it addresses an essential need of the Deaf community; consequently, there is a current trend to develop similar too

3 Design and development of a multimodal assistive device

3.1 Design specifications

3.1.1 Functional Requirements

The particular necessities are user interfaces. The outside clients are the customers. Every one of the customers can utilize this product for ordering and looking.

- Hardware Interfaces: The outside equipment interface utilized for ordering and looking is PCs of the customers. The PC's might be portable PCs with remote LAN as the web association gave will be remote.
- Software Interfaces: The working Frameworks can be any rendition of windows.
- Performance Prerequisites: The PC's utilized must be at least Pentium 4 machine with the goal that they can give ideal execution of the item.

3.1.2 Non-Functional Requirements

Non-utilitarian necessities are the capacities offered by the framework. It incorporates time imperative and requirements on the advancement procedure and models. The non-useful prerequisites are as per the following:

- Speed: The framework ought to prepare the given contribution to yield inside fitting time.
- Ease of utilization: The product is thought to be easy to understand. At that point, the clients can utilize it effortlessly, so it doesn't require much preparation time.
- Reliability: The rate of disappointments ought to be less than just the framework is more solid.
- Portability: It was thought to be anything but difficult to actualize in any framework

3.1.3 Hardware requirements

The most widely recognized arrangement of prerequisites characterized by any working framework or programming application is the physical PC assets, otherwise called equipment, An equipment necessities list is frequently joined by an equipment similarity list, particularly if there should be an occurrence of working frameworks. HCL records tried, perfect, and now and then incongruent equipment gadgets for a specific working framework or application. The accompanying sub-segments examine the different parts of equipment prerequisites.

All PC working frameworks are intended for a specific PC design. Most programming applications are restricted to specific working frameworks running on specific structures. In spite of the fact that engineering-free working frameworks and applications exist, most should be recompiled to keep running on another design.

The energy of the focal preparing unit (CPU) is a central framework necessity for any product. Most programming running on x86 engineering characterize preparing power as the model and the clock speed of the CPU. Numerous different highlights of a CPU that impact its speed and power, similar to transport speed, storage, and MIPS are frequently overlooked. This meaning of energy is regularly wrong, as AMD Intel Pentium CPUs at comparative clock speeds frequently have distinctive throughput speeds.

- 10GB HDD(min)
- 128 MB RAM(min)
- Pentium P4 Processor 2.8Ghz(min)

3.1.4 SOFTWARE REQUIREMENTS

Programming necessities manage characterizing programming asset necessities and requirements that should be introduced on a PC to give ideal working of an application.

These necessities or requirements are for the most part excluded in the product establishment bundle and should be introduced independently before the product is introduced.

- Python 2.7 or higher
- Pycharm
- Opencv

3.2 APPLICATIONS OF PYTHON

3.2.1 GUI-BASED DESKTOP APPLICATION

Python has simple syntax, modular architecture, rich text processing tools and the ability to work on multiple operating systems which make it a desirable choice for developing desktop-based applications. There are various GUI toolkits like wxPython, PyQt or PyGtk available which help developers create highly functional Graphical User Interface (GUI). The various applications developed using Python includes:

- **Image Processing and Graphic Design Applications:**
  Python has been used to make 2D imaging software such as Inkscape, GIMP, Paint Shop Pro and Scribus. Further, 3D animation packages, like Blender, 3ds Max, Cinema 4D, Houdini, Lightwave and Maya, also use Python in variable proportions.

- **Scientific and Computational Applications:**

  The higher speeds, productivity and availability of tools, such as Scientific Python and Numeric Python, have resulted in Python becoming an integral part of applications involved in computation and processing of scientific data. 3D modeling software, such as FreeCAD, and finite element method software, such as Abaqus, are coded in Python.

- **Games:**

  Python has various modules, libraries and platforms that support development of games. For example, PySoy is a 3D game engine supporting Python 3, and PyGame provides functionality and a library for game development. There have been numerous games built using Python including Civilization-IV, Disney's Toontown Online, Vega Strike etc.

3.2.2 Web Frameworks and Web Applications:

Python has been used to create a variety of web-frameworks including CherryPy, Django, TurboGears, Bottle, Flask etc. These frameworks provide standard libraries and modules which simplify tasks related to content management, interaction with database and interfacing with different internet protocols such as HTTP, SMTP, XML-RPC, FTP and POP. Plone, a content management system; ERP5, an open source ERP which is used in aerospace, apparel and banking; Odoo – a consolidated suite of business applications; and Google App engine are a few of the popular web applications based on Python.

3.2.3 Enterprise and Business Applications:

With features that include special libraries, extensibility, scalability and easily readable syntax, Python is a suitable coding language for customizing larger applications. Reddit, which was originally written in Common Lips, was rewritten in Python in 2005. Python also contributed in a large part to functionality in YouTube.

3.2.4 OPERATING SYSTEMS:

Python is often an integral part of Linux distributions. For instance, Ubuntu's Ubiquity Installer, and Fedora's and Red Hat Enterprise Linux's Anaconda Installer are written in Python. Gentoo Linux makes use of Python for Portage, its package management system.

3.2.5 LANGUAGE DEVELOPMENT:

Python's design and module architecture has influenced development of numerous languages. Boo language uses an object model, syntax and indentation, similar to Python. Further, syntax of languages like Apple's Swift, CoffeeScript, Cobra, and OCaml all share similarity with Python.

3.3 PROTOTYPING:

Besides being quick and easy to learn, Python also has the open source advantage of being free with the support of a large community. This makes it the preferred choice for prototype development. Further, the agility, extensibility and scalability and ease of refactoring code associated with Python allow faster development from initial prototype.Since its origin in 1989, Python has grown to become part of a plethora of web-based, desktop-based, graphic design, scientific, and computational applications. With Python available for Windows, Mac OS X and Linux / UNIX, it offers ease of development for enterprises. Additionally, the latest release Python 3.4.3 builds on the existing strengths of the language, with drastic improvement in Unicode support, among other new features.

Versions of python

- Python 1.0 Introduced in jan 1994
- Python 2.0 Introduced in oct 2000
- Python 3.0 introduced in dec 2008

latest version
python 3.6.3 ➔ 2016
python 3.7
Any new version should provide support for old version programs

- There is no- backward compatibility support
- Python 3 is not support to python 2 program

3.3.1 PYTHON MACHINE LEARNING

Python is a popular platform used for research and development of production systems. It is a vast language with number of modules, packages and libraries that provides multiple ways of achieving a task.

.

Python and its libraries like NumPy, SciPy, Scikit-Learn, Matplotlib are used in data science and data analysis. They are also extensively used for creating scalable machine learning algorithms. Python implements popular machine learning techniques such as Classification, Regression, Recommendation, and Clustering.

Python offers ready-made framework for performing data mining tasks on large volumes of data effectively in lesser time. It includes several implementations achieved through algorithms such as linear regression, logistic regression, Naïve Bayes, k-means, K nearest neighbor, and Random Forest.

Python in Machine Learning

Python has libraries that enables developers to use optimized algorithms. It implements popular machine learning techniques such as recommendation, classification, and clustering. Therefore, it is necessary to have a brief introduction to machine learning before we move further.

HARDWARE & SOFTWARE REQUIRMENT

H/W System Configuration:-

Processor	-	Dual Core
Speed	-	1.1 G Hz
RAM	-	4 GB (min)
Hard Disk	-	20 GB
Key Board	-	Standard Windows Keyboard
Mouse	-	Two or Three Button Mouse
Monitor	-	SVGA

S/W System Configuration:-

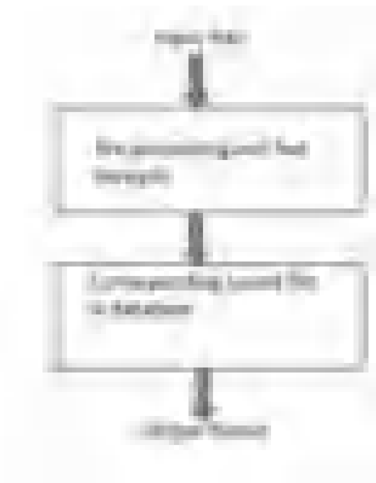
Operating System	:	Windows xp,7,8
Technology	:	Python
Front End	:	Tkinter
IDLE	:	Python 2.7 or higher
Database	:	MySQL

3.4 EXPLANATION OF THE ALGORITHMS

3.4.1 METHDOLOGY

3.4.1.A TEXT TO SPEECH

Text-to-speech (TTS) is a type of speech synthesis application that is used to create a spoken sound version of the text in a computer document, such as a help file or a Web page. TTS can enable the reading of computer display information for the visually challenged person, or may simply be used to augment the reading of a text message.
Current TTS applications include voice-enabled e-mail and spoken prompts in voice response systems. TTS is often used with voice recognition programs.
Like other modules the process has got its own relevance on being interfaced with, where Raspberry Pi finds its own operations based on image processing schemes. So once image get converted to text and there by it could be converted from text to speech.Character recognition process ends with the conversion of text to speech and it could be applied at any where.



Another method for converting the text into speech can be through the ASCII values of English letters. By using this method the coding length can be decreased. There are many Text to Speech converters are there but there performance depends on the fact that the output voice is how much close to the human natural voice. For example, consider a name pretty, it can be a name of a person as well as it can be defined as beautiful. Thus it depends on how the words are pronounced. Many text to speech engines does not give the proper pronunciation for such words thus combining some voice recordings can give more accurate result. The TTS system converts an English text into a speech signal with prosodic attributes that improve its naturalness. There are many systems which include prosodic processing and generation of synthesized control Parameters. The proposed system provides good quality of synthesized speech.

3.4.1.B Speech to text Converter

The process of converting spoken speech or audio into text is called speech to text converter. The process is usually called speech recognition. The Speech recognition is used to characterize the broader operation of deriving content from speech which is known as speech understanding. We often associate the process of identifying a person from their voice that is voice recognition or speaker recognition so it is wrong to use this term for it.

As shown in the above block diagram speech to text converters depends mostly on two models

1.Acoustic model and

2. Language model. Systems generally use the pronunciation model. It is really imperative to learn that there is nothing like a universal speech recognizer.

If you want to get the best quality of transcription, you can specialize the above models for the any given language communication channel. Likewise another pattern recognition technology, speech recognition can also not be without error. Accuracy of speech transcript deeply relies on the voice of the speaker, the characteristic of speech and the environmental conditions. Speech recognition is a tougher method than what folks unremarkably assume, for a personality's being. Humans are born for understanding speech, not to transcribing it, and solely speech that's well developed will be transcribed unequivocally. From the user's purpose of read, a speech to text system will be categorized based in its use.

3.4.1.C SPEECH SYNTHESIS (TTS)

Speech synthesis is the synthetic production of speech. A automatic data handing out system used for this purpose is called as speech synthesizer, and may be enforced in software package and hardware product. A text-to-speech (TTS) system converts language text into speech, alternative systems render symbolic linguistic representations.

Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database. Systems differ in the size of the stored speech units; a system that stores phones or diphones provides the largest output range, but may lack clarity. For specific usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely "synthetic" voice output.

The quality of a speech synthesizer is judged by its similarity to the human voice and by its ability to be understood clearly. An intelligible text to speech program permits individual with ocular wreckage or reading disabilities to concentrate to written words on a computing device. Several computer operational systems have enclosed speech synthesizers since the first nineteen nineties years.

The text to speech system is consist of 2 parts:-front-end and a back-end. The front-end consist of 2 major tasks. Firstly, it disciple unprocessed text containing symbols like numbers and abstraction into the equivalent of written out words. This method is commonly known as text, standardization, or processing. Front end then assigns spoken transcriptions to every word and divides and marks the text into speech units, like phrases, clauses, and sentences.

The process of assigning phonetic transcriptions to words is called *text-to-phoneme* or *grapheme-to-phoneme* conversion. Phonetic transcriptions and prosody information together make up the symbolic linguistic representation that is output by the front end. The back-end—often referred to as the *synthesizer*—then converts the symbolic linguistic representation into sound. In certain systems, this part includes the computation of the *target prosody* (pitch contour, phoneme durations), which is then imposed on the output speech.

Text-to-speech (TTS) is a type of speech synthesis application that is used to create a spoken sound version of the text in a computer document, such as a help file or a Web page. TTS can enable the reading of computer display information for the visually challenged person, or may simply be used to augment the reading of a text message. Current TTS applications include voice-enabled e-mail and spoken prompts in voice response systems. TTS is often used with voice recognition programs. There are numerous TTS products available, including Read Please 2000, Proverbe Speech Unit, and Next Up Technology's TextAloud. Lucent, Elan, and AT&T each have products called “Text-to-Speech”.

3.4.1.D Speech Recognition

Speech recognition is the inter-disciplinary sub-field of computational linguistics that develops methodologies and technologies that enables the recognition and translation of spoken language into text by computers. It is also known as "automatic speech recognition" (ASR), "computer speech recognition", or just "speech to text" (STT). It incorporates knowledge and research in the linguistics, computer science, and electrical engineering fields. Some speech recognition systems require "training" (also called "enrollment") where an individual speaker reads text or isolated vocabulary into the system. The system analyzes the person's specific voice and uses it to fine-tune the recognition of that person's speech, resulting in increased accuracy. Systems that do not use training are called "speaker independent" systems. Systems that use training are called "speaker dependent".
Speech recognition applications include voice user interfaces such as voice dialing (e.g. "Call home"), call routing (e.g. "I would like to make a collect call"), domotic appliance control, search (e.g. find a podcast where particular words were spoken), simple data entry (e.g., entering a credit card number), preparation of structured documents (e.g. a radiology report), speech-to-text processing (e.g., word processors or emails), and aircraft (usually termed Direct Voice Input).
The term *voice recognition* or *speaker identification* refers to identifying the speaker, rather than what they are saying. Recognizing the speaker can simplify the task of translating speech in systems that have been trained on a specific person's voice or it can be used to authenticate or verify the identity of a speaker as part of a security process.
From the technology perspective, speech recognition has a long history with several waves of major innovations. Most recently, the field has benefited from advances in deep learning and big data. The advances are evidenced not only by the surge of academic papers published in the field, but more importantly by the worldwide industry adoption of a variety of deep learning methods in designing and deploying speech recognition systems.

Speech recognition works using algorithms through acoustic and language modeling. Acoustic modeling represents the relationship between linguistic units of speech and audio signals; language modeling 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.

- **MODULE**

- **PYTTSX**

Pyttsx is platform independent that is it is compatible with Windows, Linux, and MacOS speech library. This offers a great set of functionality and features.
The user can set their voice metadata that is information about the data such as gender male or female, pitch of the voice, age, name and language. It supports large set of voices.
So to install it in windows platform depending upon which version of python you are using.
For example if you are using python3 so you need to install pyttsx3.

- **GTTS**

This module is Google text to speech library in python.
To install this API in windows platform

- **SPEECH RECOGNITION**

Speech Recognition is the library which is compatible with Python 2.6, 2.7 and 3.3+, but it will require some additional installation steps for Python v2.0. For our project we have used Python v3.0+.
Speech Recognition will work very good if you need to work with existing audio files. The audio package comes in play when you need to capture microphone input.
The main class which is used in this package is Recognizer class. The use of recognizer instance is obviously to recognize the speech. Every instance of this class comes with various settings and functionality for recognizing speech from the speaker.

- **LCD DISPLAY:**

  - LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications.
  - A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits.
  - These modules are preferred over Seven Segments and other multi segment LED.

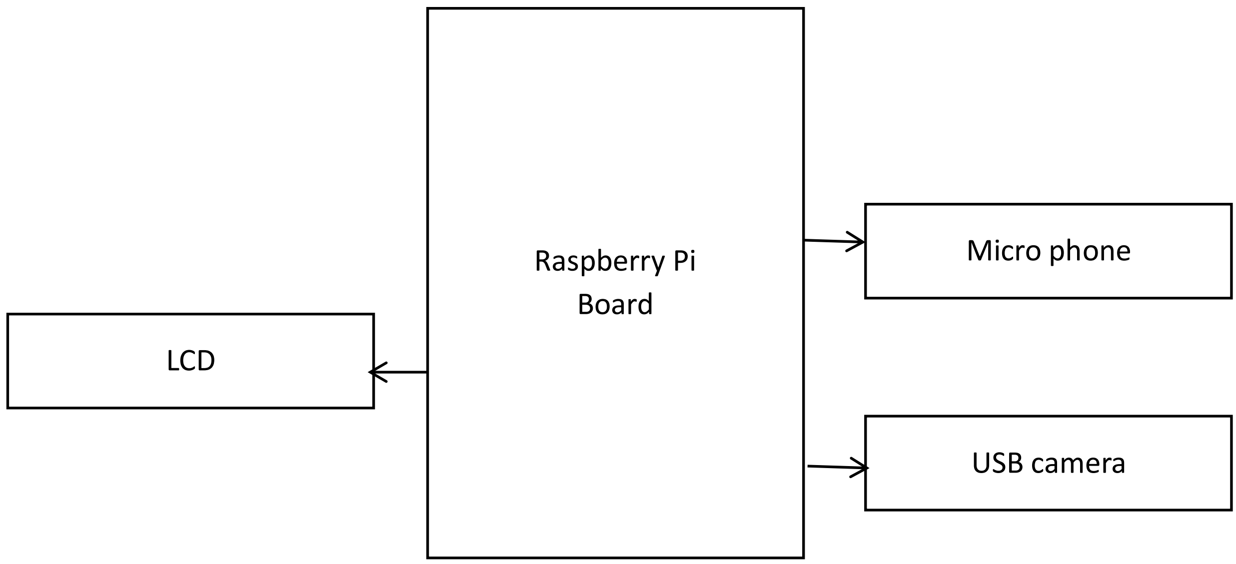
- A 16x2 LCD means it can display 16 characters per line and there are 2 such lines.

- **OPENCV PACKAGE**

- OpenCV: is a cross-platform library using which we can develop real-time computer vision applications.
- It mainly focuses on image processing, video capture and analysis.
- It is including features like face detection and object detection.

3.5 SYSTEM DESIGN

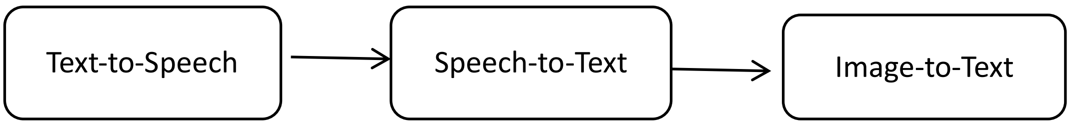
System design is the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. If the broader topic of product development "blends the perspective of marketing, design, and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user.



3.6 DATA FLOW DIAGRAM

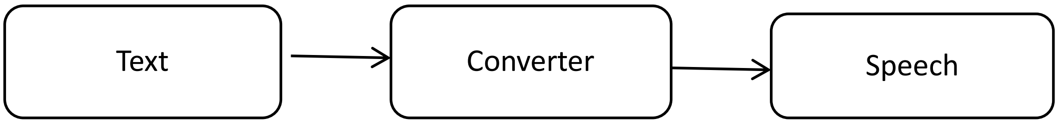
A data flow diagram is a graphical representation of the "flow" of data through an information system, modeling its *process* aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system.

LEVEL-1



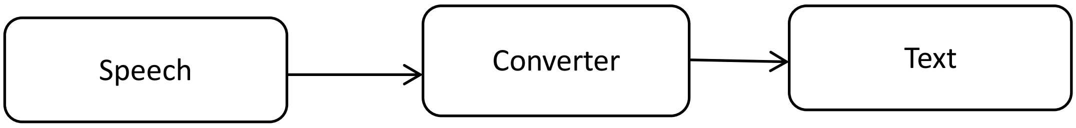
The first process text to speech conversion is done for the dumb masses who cannot speak. The Dumb people convert their thoughts to text which could be transferred to a voice signal. The converted voice signal is speak out by espeak synthesizer. After pressing the switch SW0 the OS and sub process imported. Call text to speech function and enter the text as input. Enter The Text" is speak out by espeak for raw input. After entering the text from keyboard, the espeak synthesizer converts text to speech. The process also provided with the keyboard interrupt ctrl+C

LEVEL-2



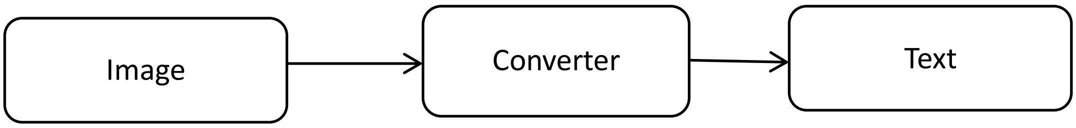
The second process is developed for blind people who cannot read normal text. In order to help blind people, we have interfaced the Logitech camera to capture the image by using OPENCV tool. The captured image is converted to text using Tesseract OCR and save the text to file out1.txt.Open the text file and split the paragraph into sentences and save it. In OCR, the adaptive thresholding techniques are used to change the image into binary images and the are transferred to character outlines. The converted text is read out by the espeak

LEVEL-3



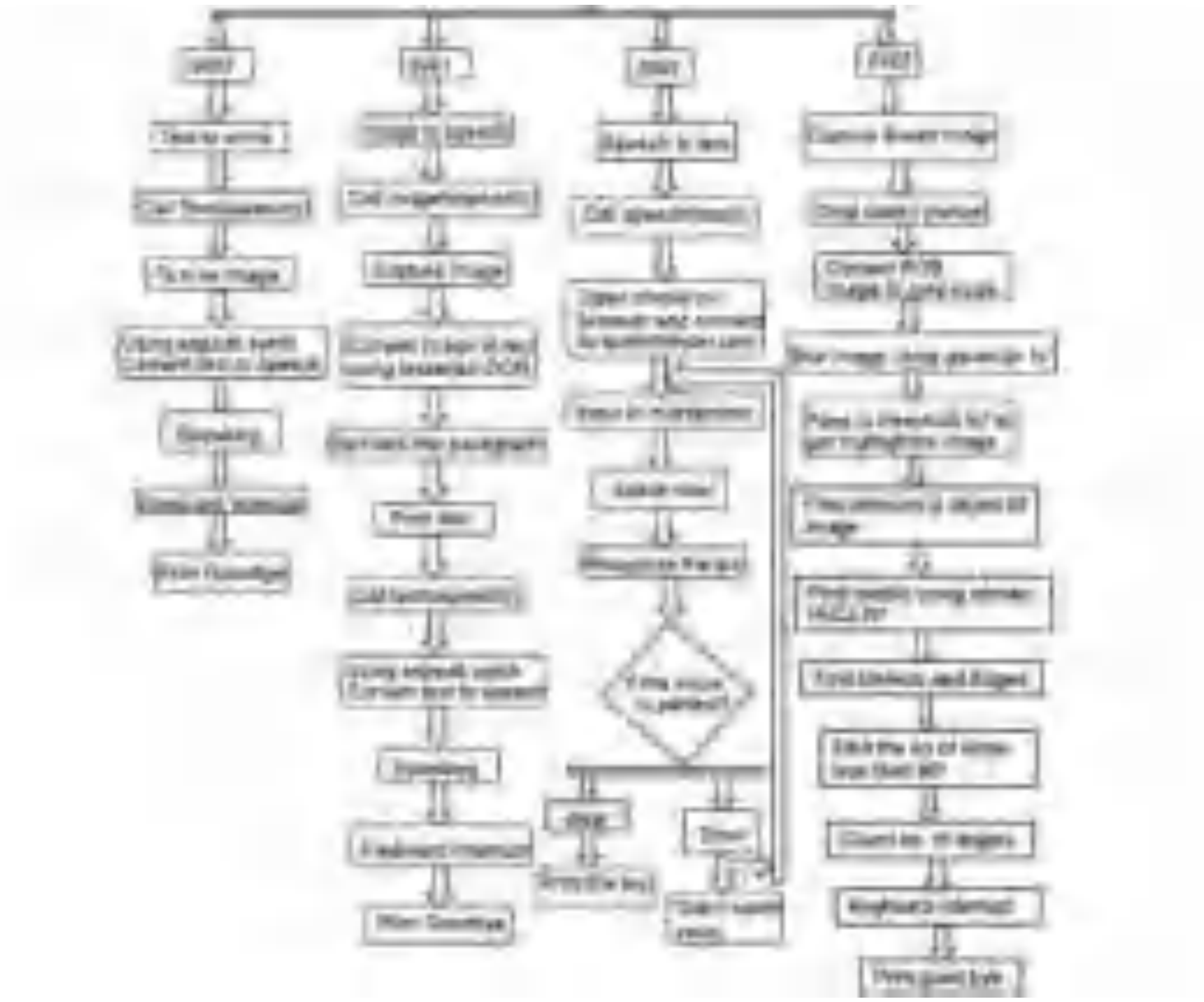
: The third process is developed for the hearing impairment people who cannot understand the words of normal people. In order to help them, our project is provided with a switch which is used to convert the voice of the normal people text. We have used a chromium browser which is automatically connected to URL speechtexter.com. The process is performed by assigning a minimum threshold voltage to recognize the voice signal. The input is given through a microphone which is converted into a text format. The URL supports a variety of languages. If the voice signal recognizable it will print the text else it gives the error signal

LEVEL-4



The fourth process is developed for vocally impaired people who cannot exchange the thoughts to the normal people. Dumb people uses gesture to communicate with normal people which are majorly cannot be understandable by normal people. The process started with the capturing the image and cropped the useful portion. Convert the RGB image into gray scale image for better functioning, Blur the cropped image through Gaussian blur function and pass it to the threshold function to get the highlighted part of the image. Find the contours and an angle between two fingers. By using convex hull function we can implement the finger point. Count the number of angles which is less than 90 degree which gives the number of defects. According to the number of defects, the text is printed on display and read out by the Speaker.

3.7 FLOWCHART OF THE ENTIRE SYSTEM



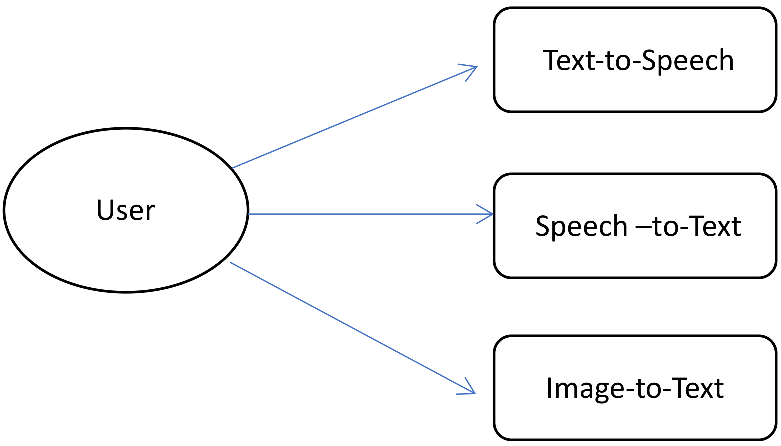
3.8 USE CASE DIAGRAM

3.8.1 UML DIAGRAMS

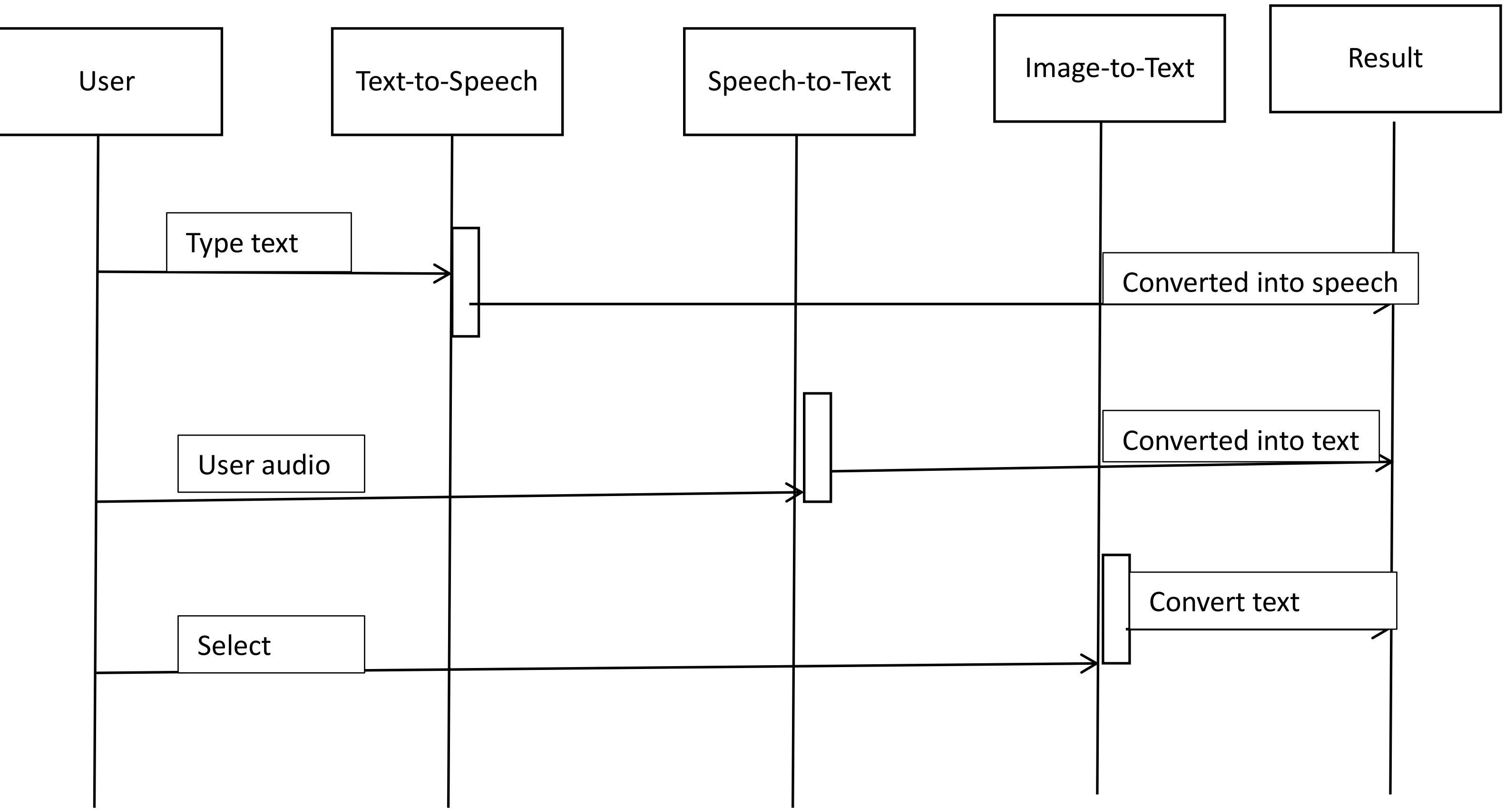
Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created, by the Object Management Group.

- **Use Case Diagrams**

A use case diagram at its simplest is a graphical representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system.



3.9 SEQUENCE DIAGRAM



3.10 PYTHON CODE FOR EACH ALGORITHM

3.10.1 TEXT TO SPEECH

```
#!/usr/bin/env python3
import argparse
import logging
from collections import defaultdict
from dataclasses import dataclass
from pathlib import Path


@dataclass
class Voice:
    lang_family: str
    lang_code: str
    dataset: str
    quality: str
    model_url: str
    config_url: str


@dataclass
class Language:
    native: str
    english: str
    country: str


_LANGUAGES = {
    "ar_JO": Language("العربية", "Arabic", "Jordan"),
    "ca_ES": Language("Català", "Catalan", "Spain"),
    "cs_CZ": Language("Čeština", "Czech", "Czech Republic"),
    "da_DK": Language("Dansk", "Danish", "Denmark"),
    "de_DE": Language("Deutsch", "German", "Germany"),
    "el_GR": Language("Ελληνικά", "Greek", "Greece"),
    "en_GB": Language("English", "English", "Great Britain"),
    "en_US": Language("English", "English", "United States"),
    "es_ES": Language("Español", "Spanish", "Spain"),
    "es_MX": Language("Español", "Spanish", "Mexico"),
    "fa_IR": Language("فارسی", "Farsi", "Iran"),
    "fi_FI": Language("Suomi", "Finnish", "Finland"),
    "fr_FR": Language("Français", "French", "France"),
    "is_IS": Language("íslenska", "Icelandic", "Iceland"),
```

```
    "it_IT": Language("Italiano", "Italian", "Italy"),
    "hu_HU": Language("Magyar", "Hungarian", "Hungary"),
    "ka_GE": Language("ქართული ენა", "Georgian", "Georgia"),
    "kk_KZ": Language("қазақша", "Kazakh", "Kazakhstan"),
    "lb_LU": Language("Lëtzebuergesch", "Luxembourgish", "Luxembourg"),
    "ne_NP": Language("नेपाली", "Nepali", "Nepal"),
    "nl_BE": Language("Nederlands", "Dutch", "Belgium"),
    "nl_NL": Language("Nederlands", "Dutch", "Netherlands"),
    "no_NO": Language("Norsk", "Norwegian", "Norway"),
    "pl_PL": Language("Polski", "Polish", "Poland"),
    "pt_BR": Language("Português", "Portuguese", "Brazil"),
    "pt_PT": Language("Português", "Portuguese", "Portugal"),
    "ro_RO": Language("Română", "Romanian", "Romania"),
    "ru_RU": Language("Русский", "Russian", "Russia"),
    "sk_SK": Language("Slovenčina", "Slovak", "Slovakia"),
    "sl_SI": Language("Slovenščina", "Slovenian", "Slovenia"),
    "sr_RS": Language("srpski", "Serbian", "Serbia"),
    "sv_SE": Language("Svenska", "Swedish", "Sweden"),
    "sw_CD": Language("Kiswahili", "Swahili", "Democratic Republic of the Congo"),
    "tr_TR": Language("Türkçe", "Turkish", "Turkey"),
    "uk_UA": Language("украї́нська мо́ва", "Ukrainian", "Ukraine"),
    "vi_VN": Language("Tiếng Việt", "Vietnamese", "Vietnam"),
    "zh_CN": Language("简体中文", "Chinese", "China"),
}

_QUALITY = {"x_low": 0, "low": 1, "medium": 2, "high": 3}


_LOGGER = logging.getLogger()


def main() -> None:
    parser = argparse.ArgumentParser()
    parser.add_argument(
        "--piper-voices", required=True, help="Path to piper-voices root"
    )
    parser.add_argument(
        "--model-url-format",
        default="https://huggingface.co/rhasspy/piper-
voices/resolve/v1.0.0/{lang_family}/{lang_code}/{dataset}/{quality}/{lang_code}-{dataset}-
{quality}.onnx?download=true",
        help="URL format for models with lang_family, lang_code, dataset, and quality",
    )
    parser.add_argument(
        "--config-url-format",
```

```
        default="https://huggingface.co/rhasspy/piper-voices/resolve/v1.0.0/{lang_family}/{lang_code}/{dataset}/{quality}/{lang_code}-{dataset}-{quality}.onnx.json?download=true",
        help="URL format for configs with lang_family, lang_code, dataset, and quality",
    )
    args = parser.parse_args()
    logging.basicConfig(level=logging.DEBUG)

    voices_by_lang_code = defaultdict(list)
    piper_voices = Path(args.piper_voices)
    for onnx_path in piper_voices.rglob("*.onnx"):
        _LOGGER.debug(onnx_path)
        parts = onnx_path.stem.split("-")
        if len(parts) != 3:
            _LOGGER.warning("Skipping %s", onnx_path)
            continue

        lang_code, dataset, quality = parts
        assert lang_code in _LANGUAGES, f"Missing {lang_code}"
        lang_family = lang_code.split("_")[0]
        model_url = args.model_url_format.format(
            lang_family=lang_family,
            lang_code=lang_code,
            dataset=dataset,
            quality=quality,
        )
        config_url = args.config_url_format.format(
            lang_family=lang_family,
            lang_code=lang_code,
            dataset=dataset,
            quality=quality,
        )

        voices_by_lang_code[lang_code].append(
            Voice(
                lang_family=lang_family,
                lang_code=lang_code,
                dataset=dataset,
                quality=quality,
                model_url=model_url,
                config_url=config_url,
            )
        )

    print("# Voices")
    print("")
```

```
    for lang_code in sorted(voices_by_lang_code):
        lang_info = _LANGUAGES[lang_code]
        if lang_code.startswith("en_"):
            print("*", lang_info.english, f"({lang_code})")
        else:
            print("*", lang_info.english, f"(`{lang_code}`, {lang_info.native})")

        last_dataset = None
        for voice in sorted(
            voices_by_lang_code[lang_code],
            key=lambda v: (v.dataset, _QUALITY[v.quality]),
        ):
            if voice.dataset != last_dataset:
                print("    *", voice.dataset)
                last_dataset = voice.dataset

            print(
                "        *",
                voice.quality,
                "-",
                f"[[model]({voice.model_url})]",
                f"[[config]({voice.config_url}.json)]",
            )


if __name__ == "__main__":
    main()
```

3.10.2 TEXT TO SPEECH THROUGH IMAGE

```
google-cloud-vision==1.0.0
google-cloud-texttospeech==2.2.0
picamera==1.13
import picamera
import time
import os
from google.cloud import vision
from google.cloud import texttospeech

# Needs permission for Cloud Vision API and Cloud Text-to-Speech API

os.environ["GOOGLE_APPLICATION_CREDENTIALS"]="YourServiceAccount.json"
client_vision = vision.ImageAnnotatorClient()
client_tts = texttospeech.TextToSpeechClient()
def takephoto():
  camera = picamera.PiCamera()
  camera.resolution = (1024, 768)

  # Show me a quick preview before snapping the photo (If you have a monitor)

  camera.start_preview()
  time.sleep(1)

  # Take the photo
  camera.capture('image.jpg')
  def main():
  takephoto()

  with open('image.jpg', 'rb') as image_file:
      content = image_file.read()

  image = vision.types.Image(content=content)

  response = client_vision.label_detection(image=image)

  response = client_vision.label(image=image)
    labels = response.label_annotations
    print('Labels:')
```

```
    synthesis_input = ''

    # Make a simple comma delimited string type sentence.
    for label in labels:
        print(label.description)
        synthesis_input = label.description + ', ' + synthesis_input

    synthesis_in = texttospeech.SynthesisInput(text=synthesis_input)

# Let's make this a premium Wavenet voice in SSML
    voice = texttospeech.VoiceSelectionParams(
        language_code="en-US",
        name="en-US-Wavenet-A",
        ssml_gender=texttospeech.SsmlVoiceGender.MALE
    )

# Select the type of audio file you want returned
    audio_config = texttospeech.AudioConfig(
        audio_encoding=texttospeech.AudioEncoding.MP3
    )

# Perform the text-to-speech request on the text input with the     selected
    # voice parameters and audio file type
    response = client_tts.synthesize_speech(
    input=synthesis_in, voice=voice, audio_config=audio_config
    )

# The response's audio_content is binary.
    with open("output.mp3", "wb") as out:
        # Write the response to the output file.
        out.write(response.audio_content)

print('Audio content written to file "output.mp3"')

file = "output.mp3"
    # apt install mpg123
    # Save the audio file to the dir
    os.system("mpg123 " + file)

if __name__ == '__main__':
    main()
```

3.10.3 SPEECH TO TEXT

```
sudo apt update -y && sudo apt upgrade -y
sudo apt install python3-pip flac ffmpeg -y
pip3 install SpeechRecognition
import sys, os
from pathlib import Path
import speech_recognition as sr
from subprocess import Popen, DEVNULL, STDOUT
def silent_google_recognition(audio_file):
   sys.stdout = open(os.devnull, 'w')
   result = r.recognize_google(audio_file)
   sys.stdout = sys.__stdout__
   return result
   def console(cmd):
   p = Popen(cmd, shell=True, stdout=DEVNULL, stderr=STDOUT)
   p.wait()
   #out, err = p.communicate()
   #return out.decode('ascii').strip()
   def convert_to_wav(file):
   file_name = str(Path(file).resolve())
   file_type = Path(file).suffix
   name=file_name.split(file_type)[0]
   output_wav = name + "-converted.wav"
   console("ffmpeg -i " + file + " -b:a 192k " + output_wav)
   return output_wav
   input_file = sys.argv[1]
   file_type = Path(input_file).suffix
if file_type != ".wav": input_file = convert_to_wav(input_file)
r = sr.Recognizer()
with sr.AudioFile(input_file) as source:
    audio = r.record(source)
    try:
   print("Google Speech Recognition thinks you said:\n\n" + silent_google_recognition(audio))
except sr.UnknownValueError:
   print("Google Speech Recognition could not understand audio")
```

```
except sr.RequestError as e:
    print("Could not request results from Google Speech Recognition service; {0}".format(e))
```

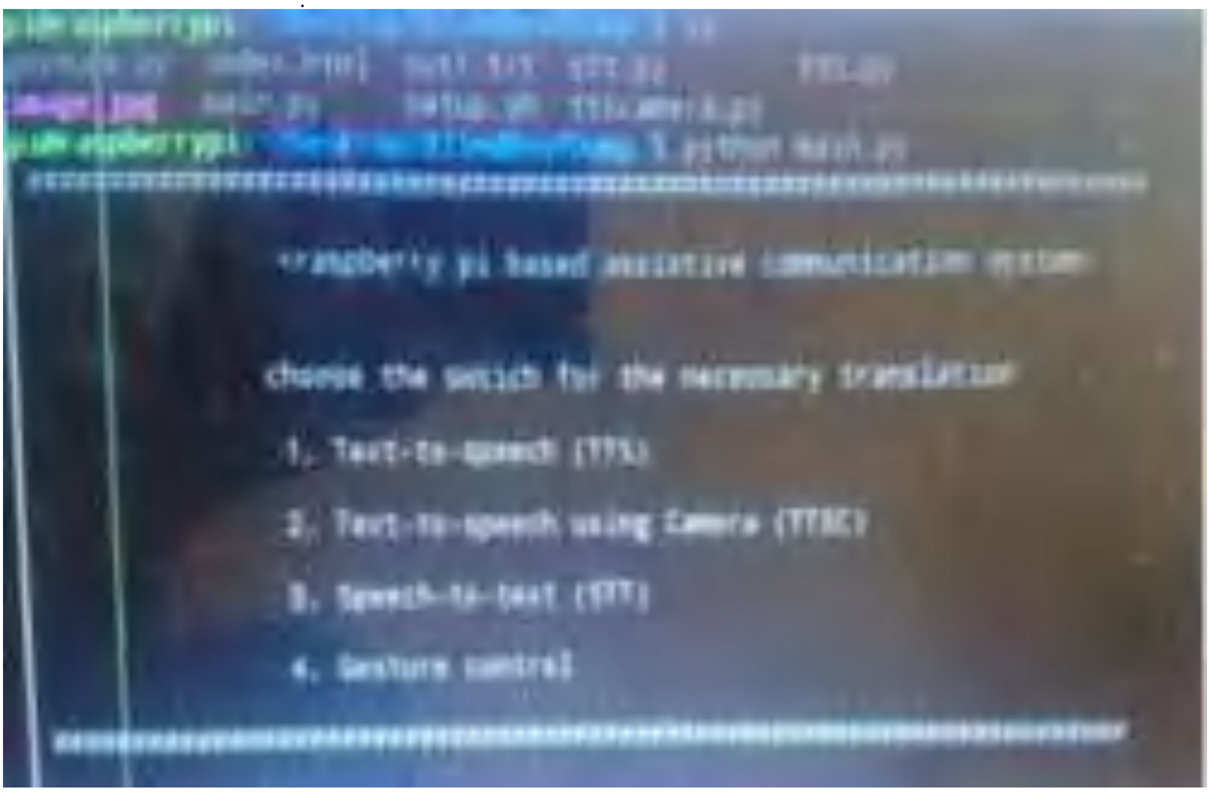
3.10.4 CAPTURING AND ANALYSING IMAGE

```
import sys
import cv2
import numpy as np
import time


def find_depth(right_point, left_point, frame_right, frame_left, baseline,f, alpha):

    # CONVERT FOCAL LENGTH f FROM [mm] TO [pixel]:
    height_right, width_right, depth_right = frame_right.shape
    height_left, width_left, depth_left = frame_left.shape

    if width_right == width_left:
        f_pixel = (width_right * 0.5) / np.tan(alpha * 0.5 * np.pi/180)

    else:
        print('Left and right camera frames do not have the same pixel width')

    x_right = right_point[0]
    x_left = left_point[0]

    # CALCULATE THE DISPARITY:
    disparity = x_left-x_right      #Displacement between left and right frames [pixels]

    # CALCULATE DEPTH z:
    zDepth = (baseline*f_pixel)/disparity           #Depth in [cm]

    return zDep
```

3.11 PROJECT OVERVIEW

We aim to develop a prototype model for blind dumb and deaf people by employing in a single compact device. The project provides a unique solution for these people to manage their sites by themselves. The project is catered with the source code of Python. It is the easiest programming language to interface with the Raspberry Pi. The project is run by the source code of Python to assist blind dumb and deaf people in a single device which is so compact and easy for them to manage. The system is provided with 4 switches. Each switch has different functions. We have chosen the switch for necessary conversion.

1) Text-to-speech (TTS) using (SW0)
2) Text-to-speech using camera (TTSC) using (SW1)
3) Speech to text(STT) using (SW2)
4) Gesture control using(SW3)



3.12 PROTEUS SIMULATION

4 CONCLUSION

The conclusion is that no device is regarded to be an optimal device. Therefore, there is a requirement for designing an intelligent system capable of covering essential characteristics for supporting visually impaired people. This paper may help researchers and scientists enthusiastic about developing the device for visually impaired people. Future researchers must step forward to work on this because, as we all know, there aren't many economically successful devices. Additionally, we anticipate that this paper will serve as motivation for future developers to enhance their interactions with medical professionals, their knowledge of the system's medical requirements, and their comprehension of the visually impaired as the product's target market. We propose that future assistive systems employ technological advances to provide a solution that is globally accessible and responds to emergencies. We believe that this paper's analysis of the systems and recommendations can serve as a springboard for further research in the field.

By this paper, we have designed the prototype model for blind, deaf and dumb people by employing a single compact device. The important key factor of this project to facilitate these people and to fix them more confident to manage their sites by themselves. The primary advantage is that the device can be taken away easily and is of about less weight.

4.1 FUTURE WORK

To further this project can be followed out with any other advanced devices by using simple coding language to get it less complicated. The complication can be reduced by a tiny gadget which could be more useful those people in this electronic world.

REFERENCES

[1] Chucai Yi, Student Member, IEEE, Yingli Tian, Senior Member, IEEE, and Aries Arditi "Portable Camera-Based Assistive Text and Product Label Reading From Hand-Held Objects for Blind Persons" 2013 IEEE.

[2] Vasanthi.G and Ramesh Babu.Y Department of ECE, DMI College of Engineering, Chennai, India. "Vision Based Assistive System for Label Detection with Voice Output"Jan-2014.

[3] Dharanikumar Chowdary. M, M. Narasimhulu, G. Subrahmanya Sharma, "Advanced Embedded Chatter Box For Physically Challenging Persons" in Jan 2012.

[4] Bhavina Patel,Vandana Shah, Ravindra k Shirsagar, "Microcontroller Based gesture Recoginition Syatem For the Handicap People"in Dec-2011.

[5] Pallavi Varma, Shimi.S.L, Richapriyadarshini, "Design Of Communication Interpreter For Deaf and Dumb person" in Jan-2015.

[6] S.B.Shroke, Nandar Dehpande, Prashant Deshmukh, SA=njay Kumar Mathapati. "Assistive Transulator For Deaf And Dumb People"-in July2014.

[7] Michael McEnancy "Finger Reader Is audio reading gadget for Index Finger " in July 2014.

FIGURE SIZE	Disparity (Block Size)	Depth (Block Size)
10,5	25	5
15,10	45	10
10,10	30	10
15,5	35	5
5,5	15	5

Appendix

Bibliography

[1] Shiyam Raghul M, Surendhar K, Suresh N, R. Hemalatha, Sri Ramakrishna Enineering College, Tamil Naidu,"Raspberry Pi Based Assistive Device For Deaf, Dumb and Blind people", IJSRSET 2016

[2] Ashish S. Nikam, Aarti G. Ambekar, "Sign Language Recognition Using Image Based Hand Gesture Recognition Techniques", IC-GET 2016.

[3] Shraddha R. Ghorpade, Surendra K. Waghamare, "Full Duplex Communication System for Deaf & Dumb People," International Journal of Emerging Technology and Advanced Engineering (IJETAE), Volume 5, Issue 5, May 2015, ISSN 2250-2459.

[4] Rohit Rastogi, Shashank Mittal, Sajan Aggrawal, "A Novel Approach for Communication among Blind, Deaf and Dumb people", IEEE 2015.

International Journal of Humanities Social Science and Management (IJHSSM)
Volume 4, Issue 2, Mar.-Apr., 2024, pp: [illegible] www.ijhssm.org

ACCEPTANCE LETTER

Date of Submission: [illegible] Date of Acceptance: [illegible]

Dear Authors,
Bhanu Pratap and Deepanshi Nagarwal,

Congratulation! As a result of reviews and revisions, we are pleased to inform you that your following Manuscript has been formally accepted for publication in the ***International Journal Of Humanities Social Science and Management.***

TITLE: "ASSISTIVE DEVICE DESIGNED FOR DEAF, DUMB, AND BLIND PEOPLE"

MANUSCRIPT ID: 2190

Dr. M. Kaalappan

Editor-in-Chief
International Journal Of Humanities Social Science and Management
HTTPS://IJHSSM.ORG/EDITORIAL-BOARD.PHP