Counselling Code : VBIT

VIGNANA BHARATHI Institute of Technology

(A UGC Autonomous Institution, Approved by AICTE, Accredited by NBA & NAAC-A Grade, Affiliated to JNTUH)

A MAJOR PROJECT REPORT

ON

PiReader: A Voice-Assisted Reading Device for the low Vision Users

Submitted in partial fulfillment of the requirement
for the award of the degree of

BACHELOR OF TECHNOLOGY
IN

COMPUTER SCIENCE AND ENGINEERING
(Data Science)

BY

Katravath Venkatesh	**21P61A6794**
Lakavath Yaku	**21P61A67A9**
Kamuni Sanjay Kumar	**21P61A6784**

Under the esteemed guidance of
Mr.Krishna Chaitanya
Assistant Professor
Dept. of CSE(DS)

VIGNANA BHARATHI INSTITUTE OF TECHNOLOGY

(A UGC Autonomous Institution, Approved by AICTE, Affiliated to JNTUH,

Accredited by NBA & NAAC) Aushapur (V), Ghatkesar (M), Medchal(dist.)

April – 2025

Counselling Code : VBIT

VIGNANA BHARATHI Institute of Technology

(A UGC Autonomous Institution, Approved by AICTE, Accredited by NBA & NAAC-A Grade, Affiliated to JNTUH)

Aushapur (V), Ghatkesar (M), Hyderabad, Medchal – Dist, Telangana – 501 301.

DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
(Data Science)

CERTIFICATE

This is to certify that the major project titled ***"PiReader: A Voice-Assisted Reading Device for the low Vision Users"*** *submitted by* ***Katravath Venkatesh (21P61A6794), Lakavath Yaku (21P61A67A9), Kamuni Sanjay Kumar (21P61A6784)*** *in B. Tech. IV-II semester Computer Science & Engineering (Data Science) is a record of the bonafide work carried out by them. The results embodied in this report have not been submitted to any other University for the award of any degree*

INTERNAL GUIDE
Mr. Krishna Chaitanya
Assistant Professor

PROJECT CO-ODINATOR
Dr. P. Punitha
Associate Professor

HEAD OF THE DEPARTMENT
Dr. Y. Raju
Associate Profess

EXTERNAL EXAMINER

DECLARATION

We, **Katravath Venkatesh**, **Lakavath Yaku, Kamuni Sanjay Kumar** bearing hall ticket number **21P61A6794, 21P61A67A9, 21P61A6784** hereby declare that the major project report entitled "**PiReader: A Voice-Assisted Reading Device for the low Vision Users**" under the guidance of ***Mr. Krishna Chaitanya***, Assistant Professor, Department of Computer Science and Engineering (Data Science), Vignana Bharathi Institute of Technology, Hyderabad, have submitted to Jawaharlal Nehru Technological University Hyderabad, Kukatpally, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering (Data Science).

This is a record of bonafide work carried out by us and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

KATRAVATH VENKATESH (21P61A6794)

LAKAVATH YAKU (21P61A67A9)

KAMUNI SANJAY KUMAR (21P61A6784)

ACKNOWLEDGEMET

We are extremely thankful to our beloved Chairman, **Dr. N. Goutham Rao** and Secretar**y, Dr. G. Manohar Reddy** who took keen interest to provide us the infrastructural facilities for carrying out the project work, Self-confidence, hard work, commitment and planning are essential to carry out any task. Possessing these qualities is sheer waste, if an opportunity does not exist. So, we whole- heartedly thank **Dr. P. V. S. Srinivas,** Principal, and **Dr. Y. Raju**, Head of the Department, Computer Science and Engineering (Data Science) for their encouragement and support and guidance in carrying out the project.

We would like to express our indebtedness to the project coordinator, **Dr. P. Punitha**, Associate Professor, Department of CSE (Data Science) for her valuable guidance during the course of project work.

We thank our Project Guide, ***Mr. Krishna Chaitanya***, Assistant Professor, for providing us with an excellent project and guiding us in completing our major project successfully.

We would like to express our sincere thanks to all the staff of Computer Science and Engineering, VBIT, for their kind cooperation and timely help during the course of our project. Finally, we would like to thank our parents and friends who have always stood by us whenever we were in need of them.

ABSTRACT

Low vision significantly impacts individuals ability to access printed text, limiting their independence and access to information. *PiReader* is a voice-assisted reading device designed to empower users with visual impairments by enabling seamless interaction with printed materials. Built on the Raspberry Pi platform, PiReader integrates Optical Character Recognition (OCR) technology, text-to-speech (TTS) engines, and voice command interfaces to deliver an intuitive, cost-effective reading solution. The system captures images of printed documents using a camera module, processes the text through OCR, and vocalizes the content using a high-quality speech synthesis engine. Users can navigate through the text using simple voice commands, enhancing usability without the need for a screen-based interface. PiReader aims to bridge the accessibility gap by providing a portable, user-friendly, and affordable alternative to existing assistive reading technologies.

Keywords:

Low Vision, Raspberry Pi, Optical Character Recognition (OCR), Text-to-Speech (TTS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

VISION:

To be recognized as a Centre of Excellence in Data Science to meet the ever growing needs of Industry and Society.

MISSION:

- To empower students with innovative and cognitive skills to gain expertise in the field of Data science.
- To inculcate the seeds of knowledge by providing industry conducive environment to enable students excel in the field of Data Science.
- To provide an appropriate ambience to nurture the young Data Science professionals.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO 1: Domain Knowledge: Impart strong foundation in basic sciences, Mathematics, Engineering and emerging areas by advanced tools and Technologies.

PEO 2: Professional Employment: Develop Professional skills that prepare them for immediate employment in industry, government, entrepreneurship and R&D.

PEO 3: Higher Degrees: Motivation to pursue higher studies and acquire masters and research.

PEO 4: Engineering Citizenship: Communicate and work effectively, engage in team work, achieve professional advancement, exhibit leadership skills, and ethical attitude with a sense of social responsibility.

PEO 5: Lifelong Learning: Lead edge of the industrial engineering discipline and respond to challenges of an ever-changing environment with the most current knowledge and technology.

PROGRAM OUTCOMES (POs)

Engineering graduates will be able to:

1. **Engineering Knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem Analysis**: Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues, and the consequent responsibilities relevant to professional engineering practice.
7. **Environment and sustainability**: Understand the impact of professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective

Presentations, and give and receive clear instructions.

10. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary Environments.
11. **Life-long learning**: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: Understand fundamental concepts in statistics, mathematics and computer Science to gain an understanding and working knowledge of various tools for analysis.

PSO2: Represent the knowledge, predicate logic and then transform the real life information into visually appealing data using suitable tools.

PSO3: Get Expertise in different aspects and appropriate models of Data Science and use large data sets to cater to the growing demand for data scientists and engineers in industry.

Course Outcomes (COs)

CO1 - Identify the problem by applying acquired knowledge from survey of technical publications.

CO2 - Analyze and categorize identified problem to formulate and fine the best solution after considering risks.

CO3 - Choose efficient tools for designing project.

CO4 - Build the project through effective team work by using recent technologies.

CO5 - Elaborate and test the completed task and compile the project report.

Correlation Levels

Substantial/ High	3
Moderate/ Medium	2

CO – PSO Correlation Matrix

COs	PSOs		
	PSO1	PSO2	PSO3
CO1	2	2	3
CO2	3	2	2
CO3	2	3	
CO4	2	2	3
CO5		2	2
CO	1.8	2.2	2

CO – PO Correlation Matrix

COs	POs											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	2	2	2	2	3	2	2	2	3
CO2	2	3	3	3	2	2	2	3	3	3	3	2
CO3	3	2	2	2	3	3	3	3	2	2	2	2
CO4	2	3	3	2	2	2	2	3	3	3	3	2
CO5	2	2	2	2	3	3	3	3	2	2	2	2
	2.4	2.4	2.4	2.2	2.4	2.4	2.4	3	2.4	2.4	2.4	2.2

Project Outcomes (PROs)

1. High Accuracy in Hate Speech Identification: One of the primary outcomes is achieving a high level of accuracy in identifying hate speech within text data. This involves developing machine learning models or algorithms capable of accurately distinguishing between hate speech and non-hate speech content.

2. Reduced False Positive Rate: While accuracy is crucial, it's also important to minimize false positives, i.e., incorrectly flagging non-hate speech content as hate speech. Achieving a low false positive rate helps maintain the credibility and usability of the hate speech detection system.

3. Scalability and Efficiency: Hate speech detection systems often need to process large volumes of data from various sources such as social media platforms, online forums, or news articles. Scalability and efficiency outcomes focus on developing algorithms or systems that can handle this volume of data in a timely and resource-efficient manner.

4. Multilingual Support: Hate speech is not confined to a single language or region. Therefore, an important outcome for hate speech detection projects is the ability to detect hate speech across multiple languages. This involves either developing language-agnostic models or building separate models for different languages.

5. User-Friendly Integration: Hate speech detection tools may be integrated into various platforms such as social media websites, online forums, or content moderation systems. An essential outcome is ensuring that these tools are user-friendly and seamlessly integrate with existing platforms, allowing for easy deployment and adoption by platform administrators and users alike.

PRO – PSO Correlation Matrix

PROs	PSOs		
	PSO1	PSO2	PSO3
PRO1	3	2	3
PRO2	2	3	2
PRO3	2	2	3
PRO4	2	2	2
PRO5	2	2	3
	2.2	2.2	2.6

PRO – PO Correlation Matrix

PROs	POs											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
PRO1	2	2	3	2	3	3	3	3	2	3	2	2
PRO2	3	3	3	3	3	2	2	3	2	3	3	3
PRO3	2	2	2	2	2	2	2	3	2	2	2	3
PRO4	2	2	3	2	2	2	2	3	3	3	2	2
PRO5	3	2	2	3	3	3	3	3	2	2	2	2
	2.4	2.2	2.6	2.4	2.6	2.6	2.6	3	2.2	2.6	2.2	2.4

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER – 1

1. INTRODUCTION

1.1 INTRODUCTION

Reading is a fundamental part of daily life, enabling individuals to access information, communicate, and engage with the world. However, for people with low vision, reading printed text can be a significant challenge, often requiring specialized tools or assistance. To address this issue, **PiReader** has been developed as an innovative **voice-assisted reading device** designed specifically for **low vision users**.

PiReader utilizes advanced **optical character recognition (OCR)** and **text-to-speech (TTS)** technologies to convert printed or digital text into clear, natural-sounding speech. By integrating these features into a compact and user-friendly device, PiReader empowers visually impaired individuals to read books, documents, signs, and other printed materials independently. With voice commands and tactile feedback, the device ensures accessibility, ease of use, and portability.

The development of PiReader aims not only to enhance reading accessibility but also to promote inclusivity, independence, and improved quality of life for low vision individuals. This paper explores the design, implementation, and impact of PiReader in addressing the reading challenges faced by visually impaired users.

1.2 PROBLEM STATEMENT

Individuals with low vision face significant barriers when accessing printed and written information in everyday life. Despite the availability of digital tools and screen readers, many of these solutions are either too complex, expensive, or limited in functionality when it comes to reading physical printed text. Traditional assistive technologies often lack user-friendliness, real-time feedback, and adaptability to various reading environments.

There is a pressing need for an affordable, portable, and intuitive device that can assist low vision users in reading printed materials independently and efficiently. Such a device must combine accuracy in text recognition, clear audio output, and ease of interaction through voice commands or tactile input. The lack of such a comprehensive and accessible tool continues to restrict the independence and quality of life of individuals with visual impairments.

PiReader aims to address this gap by providing a **voice-assisted reading device** that leverages

OCR and TTS technologies, enabling low vision users to seamlessly convert printed text into speech. The device is designed to be cost-effective, easy to use, and highly accessible, making reading more inclusive for visually impaired individuals.

1.3 OBJECTIVE

The primary objective of **PiReader** is to design and develop an affordable, user-friendly, and portable **voice-assisted reading device** that enables individuals with **low vision** to read printed text independently and with ease.

1.4 AIM OF THE PROJECT

The aim of the **PiReader** project is to develop a **voice-assisted reading device** that empowers individuals with **low vision** to independently access and understand printed text through real-time audio feedback.

By combining **optical character recognition (OCR)** and **text-to-speech (TTS)** technologies in a single, portable, and easy-to-use device, the project strives to enhance **reading accessibility**, promote **digital inclusion**, and improve the **daily living experience** of visually impaired users.

CHAPTER - 2

2. LITERATURE SURVEY

Access to printed information remains a significant challenge for individuals with visual impairments. Over the years, various technologies have been developed to assist low vision users, particularly in the domain of text recognition and audio feedback. This literature survey reviews key contributions in the field that have laid the foundation for the development of devices like **PiReader**.

1. **Optical Character Recognition (OCR) Systems**
   OCR technology has evolved significantly, with systems like **Tesseract OCR** and **ABBYY FineReader** leading in accuracy and language support. These tools can extract text from printed documents, but their standalone use often requires additional software or devices for audio conversion and accessibility.
2. **Text-to-Speech (TTS) Engines**
   TTS technologies such as **Google Text-to-Speech**, **Amazon Polly**, and **eSpeak** have enabled machines to vocalize textual content. Research shows that natural-sounding TTS engines greatly enhance user comprehension and comfort for low vision users.
3. **Existing Assistive Devices**
   Devices like the **OrCam MyEye**, **KNFB Reader**, and **Seeing AI** by Microsoft offer portable solutions for reading. However, studies highlight limitations such as high cost, limited offline capabilities, and complexity in usage [Ref: Singh & Kumar, 2020, *Journal of Accessibility Studies*]. These constraints restrict adoption among economically disadvantaged users.
4. **Voice-Activated Interfaces**
   With the rise of virtual assistants like Alexa and Google Assistant, voice commands have proven to be an effective input method for people with visual impairments. Integrating voice interfaces into assistive devices improves usability and independence.
5. **Low-Cost Hardware Platforms**
   Raspberry Pi and Arduino platforms have made it possible to prototype affordable assistive technologies. Research projects using Raspberry Pi for OCR and TTS processing show promising results in reducing cost while maintaining functional effectiveness.

2.1 EXISTING SYSTEM

Several assistive technologies currently exist to aid visually impaired individuals in reading printed text. These systems primarily rely on a combination of **Optical Character Recognition (OCR)** and **Text-to-Speech (TTS)** to convert printed material into audible speech. While these solutions have made significant strides in accessibility, they also come with notable limitations.

2.2 PROPOSED SYSTEM

The proposed system, **PiReader**, is a **voice-assisted reading device** designed specifically to meet the needs of individuals with **low vision**. It aims to overcome the limitations of existing solutions by offering an affordable, portable, and user-friendly alternative that does not require continuous internet access or advanced digital skills.

2.2.1 SYSTEM ARCHIRECTURE:

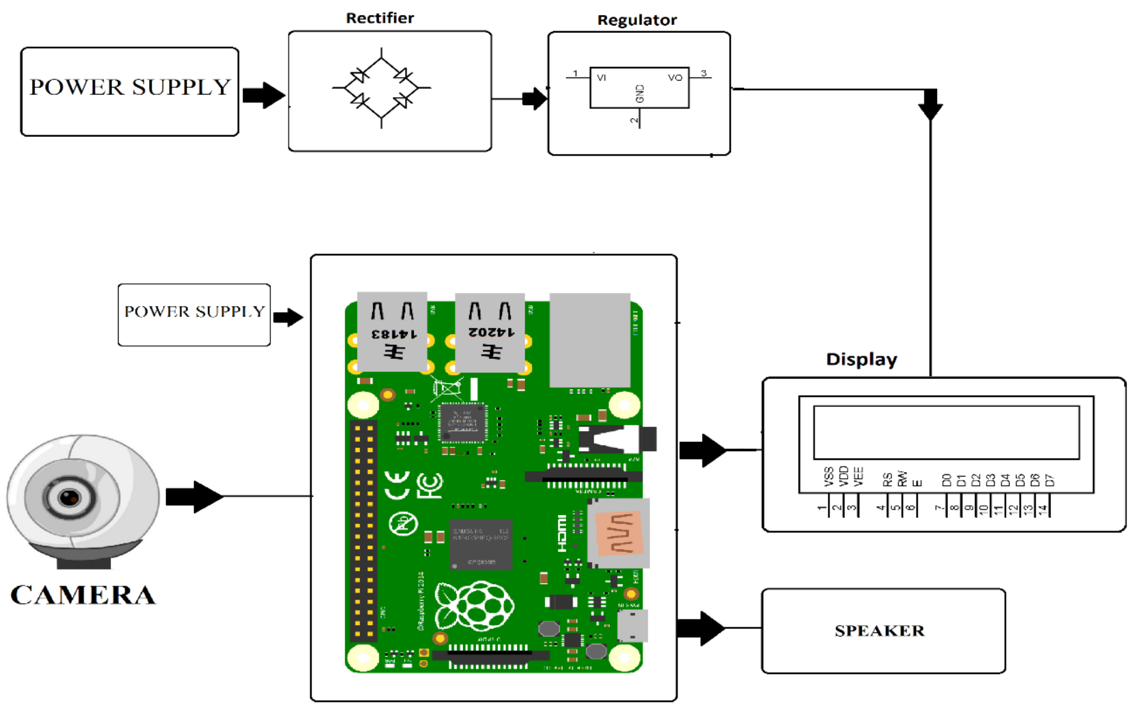

Fig 2.2.1 Architectural design

2.2.2 BENFITS OF PROPOSED SYSTEM

The PiReader offers several distinct advantages over existing systems, specifically designed to meet the needs of low vision users in an accessible, affordable, and user-friendly manner. The key benefits include:

Affordability: it leverages low-cost hardware like the Raspberry Pi and open-source software, making it accessible to a wider population, including those from economically disadvantaged backgrounds. **Portability and Convenience:** Compact and **lightweight design** makes it easy to carry and use anywhere at home, school, work, or in public places.

Offline Functionality:
No internet required for text recognition or speech generation and Reliable in remote areas or regions with limited connectivity.

Voice-Assisted Control: The integration of voice-assisted control enables hands-free operation, enhancing usability for individuals with motor impairments or limited technical experience.

Enhanced Independence: Allows low vision users to read printed materials such as books, newspapers, medicine labels, and official documents without assistance.

Customizability and Scalability: Built on open-source platforms, making it **easy to upgrade**, localize for multiple languages, or customize for specific user needs and that can be enhanced with additional features like Braille output or cloud backup if needed.

Inclusive Design: Includes **tactile buttons** for users who may prefer physical controls over voice commands and Designed for **simplicity**, ensuring accessibility even for elderly or non-tech-savvy individuals.

Multilingual Support: Supports text recognition and speech in **multiple languages**, including regional or local dialects.

2.3 SCOPE OF THE PROJECT

The **PiReader** project focuses on the development of a cost-effective, voice-assisted reading device specifically designed to aid individuals with **low vision** in accessing printed text. The project aims to bridge the accessibility gap by leveraging **optical character recognition (OCR)** and **text-to-speech (TTS)** technologies to provide real-time audio feedback from printed materials.

The scope includes the **design, development, and testing** of a standalone device that functions **offline**, eliminating the dependency on internet access and ensuring usability in both urban and rural environments. The system will be developed using **low-cost, open-source hardware and software**, such as Raspberry Pi, a camera module, and lightweight OCR/TTS libraries, ensuring affordability and ease of replication.

CHAPTER – 3

3. FEASIBILITY STUDY

3.1 FEASIBILITY STUDY

The feasibility study for **PiReader** aims to assess whether the development and deployment of a voice-assisted reading device for low vision users is viable in terms of technology, cost, usability, and societal impact.

Four key considerations involved in the feasibility analysis are

- Technical Feasibility
- Operational Feasibility
- Economical Feasibility
- Social Feasibility

3.2 Technical Feasibility

The development of PiReader is **technically feasible** using existing hardware and open-source software. Components like the Raspberry Pi, camera modules, and speaker systems are readily available and easy to integrate. OCR engines such as **Tesseract** and TTS engines like **eSpeak** or **Festival** provide accurate and real-time performance for converting text to speech. The system can also operate offline, ensuring stable performance without reliance on internet connectivity. Given the availability of development tools and community support, the technical implementation is straightforward and sustainable.

3.3 Operational Feasibility

The operational model of PiReader is simple and effective. It requires **minimal training**, and the inclusion of **voice commands and tactile buttons** ensures accessibility for users with varying levels of digital literacy. The device is **portable** and can be used independently without technical support. Maintenance is minimal, and software updates can be easily deployed if needed. Since it operates offline, it is also highly reliable in environments with unstable or no internet access.

3.4 Economical Feasibility

PiReader is designed with **cost-effectiveness** as a core priority. By using affordable, widely available components and free open-source software, the total cost of development remains significantly lower than commercial alternatives like OrCam or KNFB Reader. The device can be assembled and deployed at a **fraction of the cost** of premium reading aids, making it a viable solution for NGOs, educational institutions, and individuals in low-income or rural communities. The **return on investment** is high in terms of social value and impact, especially for educational and accessibility-focused programs.

3.4 Social Feasibility

There is a strong **social demand** for affordable assistive technology for the visually impaired. PiReader has the potential to **improve quality of life**, **enhance educational opportunities**, and **increase independence** for low vision users. It promotes **digital inclusion** and can be adopted in schools, libraries, community centers, and homes. By addressing accessibility challenges, it contributes to the broader goals of **inclusive development and disability empowerment**.

CHAPTER – 4

4. SYSTEM REQUIREMENTS

4.1 Hardware Requirements

1. Raspberry Pi 4 Model B:

- **Processor :** ARM Cortex-A72 (Quad-Core) 1.5 GHz
- **Storage:** MicroSD card (32 GB or higher for OS and software)
- **Power Supply:** 5V/3A power adapter
- **Display (Optional):** 7” touchscreen display for visual feedback (can be excluded for purely audio-based operation)
- **RAM :** 4 GB or 8 GB RAM

2. Camera Module:

- **Resolution:** Minimum 5 MP (for clear image capture of printed text)
- **Interface:** Raspberry Pi Camera Module or compatible USB webcam

3. Audio Output:

- **Speaker:** Built-in speaker or external speakers with 3.5mm audio jack or Bluetooth support
- **Microphone (optional for voice control):** USB microphone or Raspberry Pi-compatible mic for voice commands

4.2 Software Requirements

- **Operating System:** Raspberry Pi OS
- **OCR Software:** Tesseract OCR
- **Programming Languages:** Python
- **Software Libraries and Frameworks:** OpenCV and PyAudio
- **Text-to-Speech (TTS) Engine:** eSpeak or Festival TTS

CHAPTER – 5

5.METHODOLOGY

The development of **PiReader** follows a structured and iterative approach, combining both hardware and software components to build a reliable, user-friendly voice-assisted reading device. The methodology can be broken down into the following key phases:

5.1 Requirement Analysis

- Identify the specific needs of low vision users through research, user interviews, and consultation with accessibility experts.
- Define hardware, software, and functional requirements.
- Establish goals like offline operation, multi-language support, voice and tactile input, and affordability.

5.2 Hardware Selection and Setup

- Choose suitable hardware components:
- Raspberry Pi 4 as the main processing unit.
- Camera module for capturing printed text.
- Speaker for audio output.
- Microphone and tactile buttons for voice and manual control.
- Assemble and configure the hardware for optimal integration and portability.

5.3 Software Development

- Install and configure **Raspberry Pi OS** and required drivers.
- Develop software modules using **Python**:
- OCR Module: Implemented using Tesseract OCR to extract text from images.
- Image Preprocessing: Use OpenCV to enhance image clarity for accurate OCR.
- TTS Module: Integrate eSpeak or Festival to convert recognized text to speech.
- Voice Control Module (optional): Use SpeechRecognition or PocketSphinx for voice command functionality.

5.4Integration of Modules

- Connect and synchronize OCR, TTS, and control inputs to create a seamless reading flow:

  Capture → Recognize → Convert → Read.

- Enable real-time response and user feedback.

5.5 User Interface Design

- Design a simple, intuitive interface with:
- Tactile buttons for physical navigation.
- Audio prompts to guide the user through each step.
- Voice control support for hands-free operation (optional).

5.6 Testing and Evaluation

- Test the device in various lighting and text conditions to evaluate**:**
- OCR accuracy, Speech clarity and System response time
- Ease of use for the target audience
- Conduct user trials with visually impaired individuals to gather feedback.

5.7 Optimization and Finalization

- Refine hardware placement and software logic based on testing outcomes.
- Add support for additional languages if needed.
- Finalize a low-power, portable, and user-friendly version of the PiReader device.

CHAPTER – 6

6. IMPLEMENTATION

Steps in building PiReader: A Voice-Assisted Reading Device for the low Vision Users

Before jumping straight to the implementation part, let's dive into the steps involved in building a hate speech detection project using Python.

- Development environment overview
- System Setup and Configuration (if applicable)
- Code Overview and Key Modules
- Challenges and Solutions During Implementation
- Integration of Components
- Testing and Debugging
- Testing and Evaluation

6.1 Development Environment Overview

The development environment for **PiReader: A Voice-Assisted Reading Device for Low Vision Users** is centered around the Raspberry Pi platform, chosen for its affordability, compact size, and flexibility. The Raspberry Pi 4 Model B, with its quad-core ARM Cortex-A72 processor and 4GB or 8GB of RAM, serves as the main processing unit. It supports a variety of peripherals, including a Raspberry Pi Camera Module or a USB webcam for capturing images of printed text, a microphone for receiving voice commands, and speakers or headphones for providing audio feedback. Optionally, physical buttons can be integrated using the GPIO pins for tactile interaction.

The software stack is primarily built using Python due to its simplicity and wide range of libraries. Key components include OpenCV for image processing, Tesseract OCR (accessed via the pytesseract library) for converting captured text into readable content, and speech libraries such as pyttsx3 or gTTS for text-to-speech conversion. Voice commands are processed using the SpeechRecognition library. If a graphical or web interface is needed, lightweight frameworks such as Flask can be employed. For handling physical components like buttons, libraries such as RPi.GPIO or gpiozero are utilized.

Development is typically carried out using editors like VS Code or Thonny, either directly on the Raspberry Pi or remotely via SSH. Version control is managed with Git, supporting collaboration and code tracking. Testing is done on-device to ensure real-time performance, with debugging facilitated through terminal access and Python’s logging capabilities. Accessibility considerations are integrated during development, such as using high-contrast modes and screen readers, to ensure the system remains user-friendly for those with visual impairments. Optional enhancements include Wi-Fi-based updates and the use of cloud-based OCR services for improved accuracy. Overall, the PiReader development environment balances simplicity, functionality, and accessibility to deliver an effective assistive reading solution.

6.2 System Setup and Configuration (if applicable)

The PiReader system setup starts with installing Raspberry Pi OS on a microSD card and booting up the Raspberry Pi. SSH or VNC is enabled for remote access, and system packages are updated. The camera module is connected and activated using raspi-config. Essential Python libraries such as OpenCV, Tesseract OCR (pytesseract), speechrecognition, pyttsx3, and gpiozero are installed. Microphone and speaker settings are configured for clear voice input and output. Text-to-speech functionality is enabled using pyttsx3 (offline) or gTTS (online). GPIO pins are optionally set up for button inputs. Audio and camera modules are tested with sample scripts to verify functionality. A Flask-based interface can be optionally configured for local control. The main application is set to auto-start on boot, allowing easy, hands-free use. This setup ensures PiReader operates smoothly as an assistive device for low vision users.

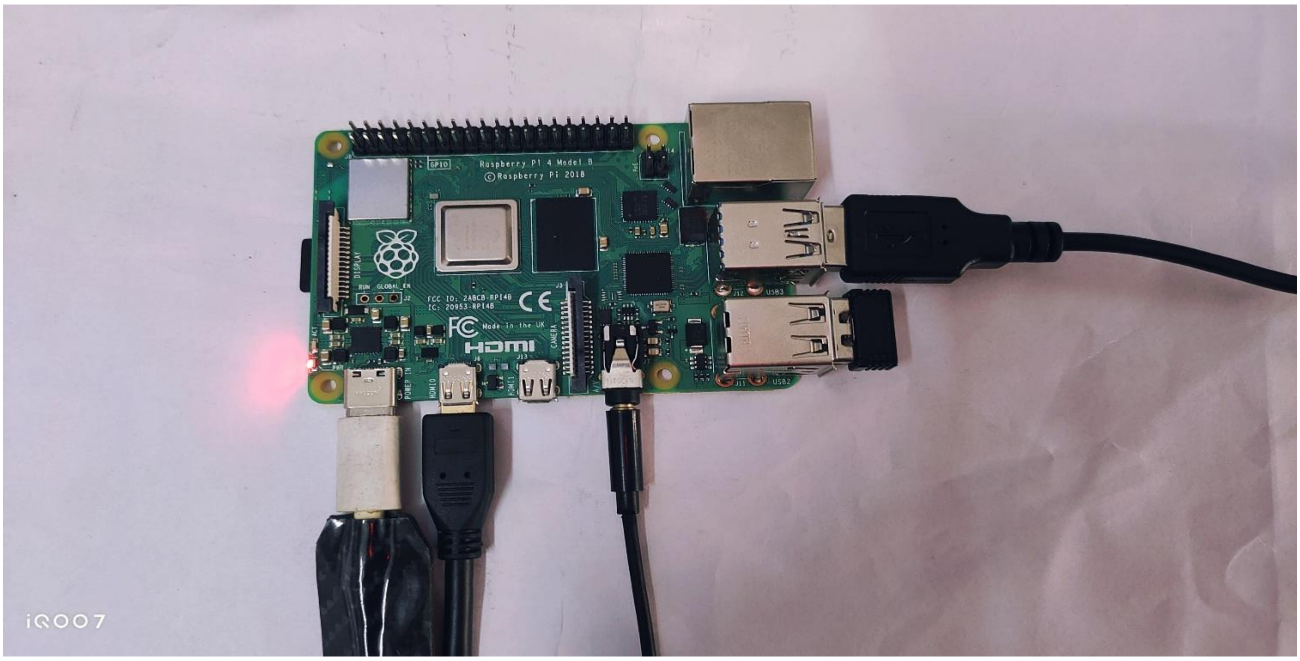

Figure 6.2.1: Raspberry Pi connections

6.3 Code Overview and Key Modules

As The PiReader code is written in Python and structured into key modules for clarity and efficiency. The **Image Capture Module** handles camera input, while the **OCR Module** uses Tesseract to extract text. The **Voice Command Module** processes user commands using the speechrecognition library. The **Text-to-Speech Module** uses pyttsx3 for reading text aloud. A **Button Control Module** (optional) uses gpiozero for physical input. The **Main Controller Script** links all modules, managing the workflow from voice input to text output. Logging and a simple config file are used for settings. An optional Flask-based interface can also be added.

```
import pytesseract
import cv2
from PIL import Image
import os
import pyttsx3
language = 'en'
engine = pyttsx3.init()
engine.setProperty('rate', 100)
key = cv2. waitKey(1)
webcam = cv2.VideoCapture(0)
while True:
  try:
      check, frame = webcam.read()
      cv2.imshow("Capturing", frame)
      key = cv2.waitKey(1)
      if key == ord('z'):
        cv2.imwrite(filename='saved_img.jpg', img=frame)
        webcam.release()
        string = pytesseract.image_to_string('saved_img.jpg')
        print(string)
        engine.setProperty('rate', 125)
        engine.say("hi")
        engine.say(string)
```

```
        engine.runAndWait()
        print("Image saved!")
        cv2.destroyAllWindows()
        break

    except(KeyboardInterrupt):
      print("Turning off camera.")
      webcam.release()
      print("Camera off.")
      print("Program ended.")
      cv2.destroyAllWindows()
      break
```

This Python program captures an image from the webcam when the user presses the 'z' key, saves the image, and then uses Tesseract OCR to extract any text from the image. The extracted text is read aloud using the pyttsx3 text-to-speech engine. The webcam feed is displayed continuously until the 'z' key is pressed. If the user interrupts the program, it gracefully shuts down the webcam and terminates the process. The program is designed to assist visually impaired users by allowing them to take pictures of text and have it read aloud to them.

1. sudo apt-get update
2. sudo apt-get upgrade
3. sudo apt-get install build-essential cmake pkg-config libjpeg-dev libtiff5-dev libjasper-dev libpng-dev libavcodec-dev libavformat-dev libswscale-dev libv4l-dev libxvidcore-dev libx264-dev libfontconfig1-dev libcairo2-dev libgdk-pixbuf2.0-dev libpango1.0-dev libgtk2.0-dev libgtk-3-dev libatlas-base-dev gfortran libhdf5-dev libhdf5-serial-dev libhdf5-103 python3-pyqt5 python3-dev -y
4. pip3 install opencv-python==4.5.3.56
5. sudo apt-get install python-opencv
6. pip install -U numpy
5. sudo apt-get install tesseract-ocr
6. pip3 install pytesseract
7. pip3 install pyttsx3

The commands update the system, install essential libraries, and set up tools for image processing, OCR, and text-to-speech. They install dependencies for OpenCV, Tesseract OCR, NumPy, and pyttsx3 for text-to-speech functionality. This prepares the system for capturing images, extracting text, and reading it aloud.

6.4 Challenges and Solutions During Implementation

- **Limited Processing Power of Raspberry Pi**

  Optimizing the code for performance is key. Using lightweight libraries like OpenCV and Tesseract with appropriate configurations (e.g., reducing image resolution or limiting OCR processing to relevant regions) can improve processing speed. Also, minimizing background tasks to prioritize the core functionality helps.

- **Low-Quality Camera Input**

  Ensuring proper camera positioning and lighting is essential for accurate text recognition. Using software adjustments, such as enhancing image contrast and resolution, can improve OCR accuracy. In low-light conditions, adding external lighting or using better-quality cameras might help.

- **Inaccurate Text Recognition**

  Tesseract may struggle with noisy or distorted images. Pre-processing the image (e.g., grayscale conversion, thresholding, noise removal) before passing it to Tesseract can improve text recognition. Additionally, training Tesseract with specific fonts or custom configurations can enhance accuracy.

- **Voice Feedback Timing and Clarity**

  Fine-tuning the pyttsx3 engine's speech rate and volume settings can improve the clarity and timing of the voice feedback. Pre-recording certain responses (e.g., "Image captured") can reduce the load during real-time feedback.

- **Limited User Interface for Low Vision Users**

  The user interface should be designed to be as simple and intuitive as possible. Large buttons, high-contrast themes, and voice commands for navigation can improve accessibility. If a graphical interface is used, it should be designed with accessibility guidelines in mind.

- **Hardware Integration (Buttons, Microphone, etc.)**

  Properly configuring GPIO pins and integrating buttons or microphones can sometimes be tricky. Ensuring correct wiring, testing with simple scripts, and using libraries like gpiozero or RPi.GPIO for hardware interaction helps streamline integration.

- **Connectivity and Software Dependencies**

  Ensuring all required libraries and software packages are installed and compatible is crucial. Using pip and apt-get to manage dependencies and performing regular updates can help avoid versioning issues. In case of network connectivity issues, providing offline functionality for OCR and TTS ensures reliable operation.

6.5 Integration of Components

PiReader integrates several components to provide a voice-assisted reading experience for low-vision users. The camera captures images, which are processed by Tesseract OCR for text recognition. Voice commands are recognized using the speechrecognition library, enabling hands-free operation. Extracted text is read aloud using pyttsx3, with adjustable speech settings. Physical buttons connected to GPIO pins offer tactile control, while an optional simple interface can be created with Flask. The main script coordinates these functions, ensuring smooth operation from image capture to speech output.

6.6 Testing and Debugging

Testing and debugging PiReader involves checking each component for functionality and accuracy. First, the image capture process is tested to ensure the camera properly detects and saves images when triggered by the user, whether through a button or key press. The OCR accuracy is verified by ensuring Tesseract extracts text correctly from various printed materials, adjusting pre-processing methods as needed to improve accuracy. Voice command recognition is tested by confirming that the system responds accurately to spoken commands like "scan" or "read" in different environments. Additionally, the text-to-speech output is checked to ensure the extracted text is read aloud clearly with appropriate speech rate and volume.

Hardware testing is also crucial, particularly for the physical buttons connected to GPIO pins, ensuring they trigger the correct actions. The microphone's sensitivity and the quality of the speech output are checked to ensure reliable voice input and clear audio output. Debugging involves logging system activity to track errors, using debugging tools like pdb or print statements to pinpoint issues in image capture, OCR, or speech synthesis. Proper error handling is implemented to manage issues such as no text detected or misinterpreted commands, ensuring the system fails gracefully.

Finally, user testing with low-vision users is conducted to gather feedback on usability, voice clarity, and accessibility. This feedback is used to refine the system, improving overall user experience and ensuring PiReader meets the needs of its intended audience.

6.7 Testing and Evaluation

Testing and evaluation of PiReader focus on ensuring that the system provides accurate and reliable performance for low-vision users. The first step is functional testing, where the camera's image capture capability is verified by ensuring it properly detects and saves images upon user command. OCR accuracy is evaluated by testing the system with different types of text (e.g., printed books, documents, etc.) in various conditions, and adjustments to image pre-processing techniques are made if needed to improve text recognition. The voice command functionality is thoroughly tested by checking that spoken commands like "scan" or "read" are consistently recognized and trigger the correct actions.

In addition, text-to-speech performance is evaluated to ensure that the extracted text is read aloud clearly, with appropriate adjustments to speech rate and volume based on user preferences. Hardware components like buttons connected to GPIO pins are tested for responsiveness, and the microphone and speaker quality are assessed for clear input and output. Debugging is carried out to identify and fix any errors, ensuring smooth image capture, OCR processing, and voice output. Logging tools are used to track any failures or issues, allowing developers to address them quickly. User testing with low-vision users is crucial for assessing the system's accessibility, usability, and overall effectiveness. Feedback from these users helps refine the interface, voice command recognition, and speech clarity. This iterative process of testing and evaluation ensures that PiReader meets the needs of low-vision users by providing a functional, reliable, and user-friendly assistive device.

CHAPTER – 7

7. CONCLUSION &FUTURE SCOPE

The development of **PiReader** demonstrates the feasibility and effectiveness of creating a low-cost, voice-assisted reading device tailored specifically for individuals with low vision. By integrating open-source technologies such as **Tesseract OCR** and **eSpeak TTS** on a **Raspberry Pi** platform, the device successfully enables users to read printed text independently, without requiring internet connectivity. Its intuitive interface, portability, and affordability make it a practical and empowering tool for visually impaired users in both urban and rural settings. The project not only highlights the importance of inclusive design but also shows how accessible technology can significantly improve quality of life.

Looking ahead, there is significant potential to enhance **PiReader** further. Future improvements could include **handwriting recognition**, **object detection**, and **document translation** for broader usability. Incorporating **AI-based speech enhancement** and **natural language processing (NLP)** could improve voice clarity and comprehension. The device can also be expanded to support **cloud connectivity** for data storage or software updates and to integrate **Braille output devices** for users with complete vision loss. With ongoing development, PiReader could evolve into a multifunctional assistive device, helping bridge more accessibility gaps and supporting a larger community of users with visual impairments.

CHAPTER – 8

8. REFERENCES

[1] Bhattacharya, A., Ghosh, S., & Saha, S. K. (2020). A Review of Speech Synthesis Techniques for Text-to-Speech Conversion in Assistive Technology. International Journal of Speech Technology, 23(2), 263-281.

[2] Bokhari, M. B., Iqbal, M., Anjum, F., & Rauf, F. (2020). Design and Implementation of Voice-Operated Text Reading System for Visually Impaired. In Proceedings of the International Conference on Advanced Communication Technologies and Networking (pp. 435- 442).

[3] Xie et al. "Assistive Technology for Blind and Visually Impaired People: An Overview of Text-to-Speech Technology."2020

[4] Mishra, A., Dubey, A., & Shukla, R. K. (2019). Development of Voice Assisted Reading System for Visually Impaired People. In Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (pp. 65-71).

[5] Singh et al. "Voice-based Text Reader for Visually Impaired People." (2019)

[6] Sharma et al. "Text to Speech for Visually Impaired: A Survey." (2019)

[7] "Design and Evaluation of a Voice- Activated Text Reader for Visually Impaired Individuals." (2019)

[8] Vijayarani, S., & Saranya, P. (2018). Voice-Assisted Reading System for Visually Impaired. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics (pp. 1323-1326).

[9] Thakur et al. "Design and Development of an Assistive Text Reading System for the Blind and Visually Impaired." by Thakur et al. (2018)

[10] Kwon et al. "A study on the Development of Voice-Enabled Text Reading System for Visually Impaired Persons." (2017)