RASPBERRY PI BASED TRANSLATION COMPANION

By

Aneesh M (71382204006)

Deepa Arasu P (71382204008)

Saktheeswaran M J (71382204047)

Report submitted in partial fulfillment of the requirements for the Degree of Bachelor of Engineering in Electronics and Communication Engineering

Sri Ramakrishna Institute of Technology

Coimbatore – 641010

May, 2025

ACKNOWLEDGEMENT

Behind every achievement lies an unfathomable sea of gratitude to those who actuated it, without them it would never have into existence. To them we lay the word of gratitude imprinted within us.

We would like to express our gratitude to the SNR management Trust and the Principal **Dr. J. David Rathnaraj**, for giving us access to all the necessary technological resources.

We express our gratitude to **Dr. S. Anila**, Head of the Department of Electronics and Communication Engineering for her conscientious guidance and encouragement.

We would like to extend our sincere gratitude to our Design supervisor , **Mr.4S.Thiyagarajan**, Assistant Professor , Department of Electronics and Communication Engineering for his cordial support and guidance.

We would like to extend our sincere gratitude to our Design Project coordinator, **Ms. P. Devi**, Assistant Professor (Sr. Gr.), Department of Electronics and Communication Engineering for her cordial support and guidance.

We sincerely thank all the teaching and non-teaching staff of Electronics and Communication Engineering department for all what they have been for their enquiry about the project.

BONAFIDE CERTIFICATE

Certified that this Design Project report titled **"RASPBERRY PI BASED TRANSLATION COMPANION"** is the bonafide work of **ANEESH M (71382204006), DEEPA ARASU P (71382204008), SAKTHEESWARAN M J (71382204047)** who carried out the project work under my supervision.

SIGNATURE

Dr. S. ANILA

Professor and Head

Department of Electronics and Communication Engineering, Sri Ramakrishna Institute of Technology, Coimbatore – 10.

SIGNATURE

MR. S. THIYAGARAJAN

Project Supervisor
Assistant Professor

Department of Electronics and Communication Engineering, Sri Ramakrishna Institute of Technology, Coimbatore – 10

Submitted for the viva-voce examination held on ____________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

APPROVAL AND DECLARATION

This project report titled "Raspberry Pi Based Translation Companion" was prepared and submitted by Aneesh M (71382204006), Deepa Arasu P (71382204008), Saktheeswaran M J (71382204047) and has been found satisfactory in terms of scope, quality and presentation as partial fulfillment of the requirement for the Bachelor of Engineering (Electronics and Communication Engineering) in Sri Ramakrishna Institute of Technology, Coimbatore (SRIT).

Checked and approved by

Mr. S. Thiyagarajan
Project Supervisor
Assistant Professor

Department of Electronics and Communication Engineering

Sri Ramakrishna Institute of Technology
Coimbatore-10
May, 2025

RASPBERRY PI BASED TRANSLATION COMPANION

ABSTRACT

This project focuses on developing an end-to-end system that captures an image, extracts text from the image, translates the text into Tamil, and converts the translated text into speech. The process begins with capturing an image using a camera-enabled device, after which the system applies Optical Character Recognition (OCR) techniques, such as Tesseract OCR, to detect and extract any readable text present in the image. Once the text is successfully extracted, it is passed through a language translation module that translates the text from its original language into Tamil, enabling better understanding for Tamil-speaking users. To make the output more accessible, especially for visually impaired individuals or users who prefer auditory content, the translated Tamil text is further processed using a Text-to-Speech (TTS) engine, which reads the text aloud. By integrating image processing, language translation, and speech synthesis technologies into a unified application, this system serves as a practical tool for enhancing language accessibility and promoting inclusive in real-world environments where reading or understanding foreign-language text might be challenging.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

As of 2024, the global population is approximately **8.2 billion people**, with at least **2.2 billion** experiencing vision impairments, of which **1 billion cases** could have been prevented or are yet to be addressed, according to the World Health Organization (**WHO**). This system helps visually challenged and non-educated individuals access printed content using a Raspberry Pi-based assistive device. It employs a camera to capture text, processes it using Optical Character Recognition (**OCR**), and vocalizes it via a Text-to-Speech (**TTS**) engine. To enhance accessibility, an English-to-Tamil translator is integrated, translating the text into Tamil before it is read aloud through an earpiece or speaker. This compact system ensures efficient, inclusive, and user-friendly operation for Tamil-speaking users.

1.2 Applications

- Assistive Technology for Visually Impaired Individuals
  Helps blind or visually impaired users read printed documents through real-time text recognition and audio playback.
- Portable Text-to-Speech Devices
  Can be used in compact, low-cost mobile devices that convert printed text to speech using Raspberry Pi, useful in libraries, schools, or public places.
- Digitization of Printed Documents
  Converts physical text (books, papers, bills) into editable and searchable digital formats, reducing the need for manual data entry.
- Language Learning and Literacy Tools
  Aids in spelling, pronunciation, and reading practice, especially for children or individuals learning a new language.

CHAPTER 2

LITERATURE REVIEW

2.1 Technical Background

Optical Character Recognition (OCR) has become a foundational technology in the development of assistive systems, especially for visually impaired individuals. OCR is defined as the process of converting scanned or photographed documents into machine-readable text. The recognition process includes identifying character glyph s from images and translating them into text using machine learning algorithms and image processing techniques.

OCR systems generally involve multiple stages: image acquisition, pre-processing, segmentation, feature extraction, and classification. In the context of embedded systems, especially low-cost solutions, platforms like **Raspberry Pi** are increasingly used due to their affordability, size, and processing capability. Raspberry Pi 3 Model B, for instance, provides sufficient computational power to execute lightweight OCR algorithms along with text-to-speech (TTS) conversion modules.

The **Text-to-Speech (TTS)** component is crucial for reading out the recognized text. TTS engines convert machine-readable text into synthesized speech, aiding users who are blind or visually impaired. Open-source tools like **Tesseract OCR** and **eSpeak** are widely used in this domain due to their integration support with Python and OpenCV libraries.

In the presented system, image acquisition is handled via an HD webcam attached to a Raspberry Pi, which captures the textual content. The captured image is then subjected to pre-processing steps such as skew correction, binarization, and noise removal. Segmentation techniques using image histograms help isolate individual characters, while feature extraction aids in classifying each glyph.

The use of **OpenCV**, an open-source computer vision library, allows for efficient implementation of these stages. OpenCV supports Python and C++, which helps in achieving real-time processing. This setup ensures that visually impaired users can interact with physical text in a meaningful, audible manner, reducing dependence on others.

2.2 Related Works

A variety of systems and algorithms have been developed over the past decade aimed at improving text accessibility for the visually impaired using OCR technology:

Shalini Sonth and Jagadish S Kallimai (2017) proposed an OCR-based facilitator for the visually challenged, which emphasized mobile integration and audio feedback. Their work highlighted the necessity for improved accuracy in noisy image conditions and the challenge of background complexity . **Sanjana and Monisha (2016)** developed an automatic book reader using Raspberry Pi. Their system used a TTS engine for vocalizing the recognized text and focused on affordability and portability. They emphasized the importance of enhancing recognition quality under varied lighting conditions .

D. Velmurugan et al. (2016) implemented a smart reader using Raspberry Pi, specifically designed for visually impaired users. Their work included hardware-level optimizations and highlighted the role of pre-processing in OCR accuracy. Their architecture closely aligns with the current project's goals, especially the use of Python and Tesseract OCR .

Faiz Alotaibi and Muhamad Taufik Abdullah (2017) explored OCR for Quranic image similarity matching. While their focus was on religious text, they introduced advanced pattern recognition and template-matching algorithms to improve OCR performance on stylized fonts . **Michael D. Kim and Jun Ueda (2017)** investigated motion de-blurring techniques to enhance OCR performance during fast scanning in robotic systems. Although not directly targeted at assistive devices, their approach addressed a common issue in mobile OCR—image blur due to hand movement .

These works collectively contribute to a growing body of knowledge that integrates machine vision and speech synthesis to improve accessibility.

2.2.1 Inference from the Literature Review

Based on the literature survey, several key insights and inferences can be drawn to guide the work on “Raspberry Pi Based Translation Companion”.

1. **Accuracy in Varied Conditions**: A common challenge across all implementations is ensuring OCR accuracy under different lighting, background textures, and text styles. Techniques such as adaptive binarization, motion de-blur, and histogram-based segmentation are vital.
2. **Low-Cost Hardware Feasibility**: Raspberry Pi and similar single-board computers have proven sufficient for lightweight OCR-TTS systems. Their integration with Python and open-source libraries offers a flexible development platform for assistive applications.
3. **Open-Source Tools Advantage**: The wide use of Tesseract OCR and OpenCV demonstrates the community’s trust in open-source technologies, which offer adaptability, customization, and cost-effectiveness.
4. **Importance of TTS Integration**: Simply recognizing text is not sufficient for accessibility; its conversion to natural-sounding speech is equally important. Advanced TTS engines that simulate human-like voice add to the usability and comfort of such systems.
5. **Need for Real-Time Systems**: All reviewed systems aim for real-time or near-real-time processing to ensure a seamless user experience. This necessitates optimization in both hardware and software components.

The current project draws effectively from these insights by implementing an OCR-TTS pipeline using Raspberry Pi, Python, OpenCV, and Tesseract, specifically tailored for visually impaired users. The integration of these components into a handheld or wearable device can significantly enhance personal independence and access to printed information.

.

CHAPTER 3

PROPOSED METHODOLOGY

3.1 Problem Statement

Visually challenged individuals face significant barriers when trying to access printed text. Traditional screen readers can only read text already available in digital formats and are ineffective at interpreting images or printed documents. Moreover, current systems often lack natural speech output, are cumbersome, or fail to provide real-time, accurate recognition from varied backgrounds and formats.

3.2 Objectives

1. Develop an assistive system that enables visually impaired individuals to read printed text using a camera-based setup.
2. Implement Optical Character Recognition (OCR) using OpenCV and Tesseract to extract text from captured images.
3. Integrate Text-to-Speech (TTS) technology to convert recognized text into speech output, making the content accessible via audio.
4. Use Raspberry Pi as a low-cost, portable, embedded platform to perform image capture, processing, OCR, and speech synthesis.
5. Optimize preprocessing and feature extraction techniques to enhance the accuracy and speed of OCR even with noisy or skewed inputs.
6. Enable user-friendly operation through a single-click button for document scanning and real-time voice output.

3.3 Methodology

Image acquisition: In this step, the inbuilt camera captures the images of the text. The quality of the image captured depends on the camera used. We are using the Raspberry Pi's camera which 5MP camera with a resolution of 2592×1944.

Image pre-processing: This step consists of color to gray scale conversion, edge detection, noise removal, warping and cropping and thresholding. The image is converted to gray scale as many OpenCV functions require the input parameter as a gray scale image. Noise removal is done using bilateral filter. Canny edge detection is performed on the gray scale image for better detection of the contours. The warping and cropping of the image are performed according to the contours. This enables us to detect and extract only that region which contains text and removes the unwanted background. In the end, Thresholding is done so that the image looks like a scanned document. This is done to allow the OCR to efficiently convert the image to text.

Image to text conversion: The above diagram(fig.5) shows the flow of Text-To-Speech. The first block is the image pre-processing modules and the OCR. It converts the preprocessed image, which is in .png form, to a .txt file. We are using the Tesseract OCR.

Text to speech conversion: The second block is the voice processing module. It converts the .txt file to an audio output. Here, the text is converted to speech using a speech synthesizer called Festival TTS. The Raspberry Pi has an on-board audio jack, the on-board audio is generated by a PWM output**.**

3.4 Advantages

- Enhanced Accessibility: Enables visually impaired users to read printed text independently through audio output, improving their access to information.
- Low-Cost Implementation: Uses affordable hardware components like the Raspberry Pi and open-source software (Tesseract, OpenCV), making it budget-friendly for widespread adoption.
- Portable and Compact: The system is small and lightweight, making it convenient for personal use in homes, schools, libraries, and public areas.

- Real-Time Processing: Offers quick text recognition and speech output, allowing users to receive instant feedback without delays.
- Minimal Power Consumption: The Raspberry Pi consumes less power compared to traditional computing systems, supporting extended use on portable power sources.
- Customizability and Upgradability: Being open-source and modular, the system can be easily modified to support new languages, handwriting recognition, or additional input methods.
- Simple User Interface: Operated with minimal interaction (e.g., button click), making it accessible for users without technical expertise.
- Supports Digital Archiving: Converts physical text into digital form, allowing for easy storage, editing, and document management.

3.5 Scope of the Project

This project aims to improve accessibility for visually challenged individuals by enabling them to read printed text independently using a compact embedded system. It utilizes a Raspberry Pi device with an HD camera, OCR software (Tesseract), and a Text-to-Speech (TTS) engine to convert printed content into audible speech. The system processes text in real-time and provides voice output, reducing users' reliance on others for reading tasks. Its compact and cost-effective design makes it suitable for use in homes, libraries, educational institutions, and public spaces. In addition, the system supports basic automation of document reading and can enhance user engagement through simple one-touch operation. Beyond assistive use, it can also aid in document digitization and audio-based learning, with potential for future expansion to multiple languages and mobile integration.

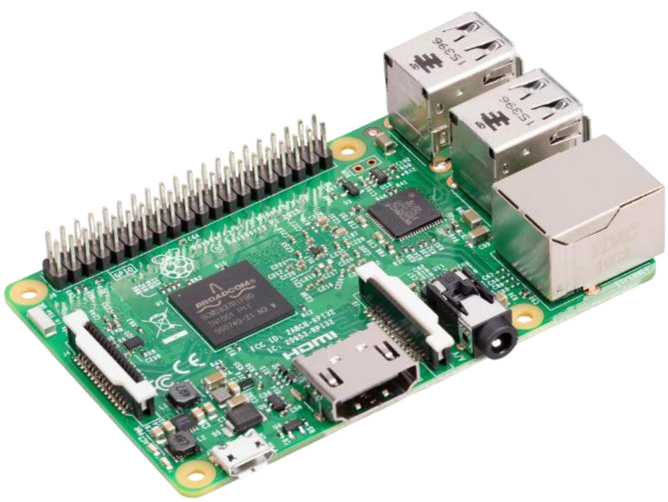
3.6 Hardware

3.6.1 Raspberry Pi

The Raspberry Pi 3 Model B is a single-board computer built around the Broadcom BCM2837 system-on-chip, featuring a 64-bit quad-core ARM Cortex-A53 processor running at 1.2 GHz. It includes 1 GB of LPDDR2 RAM, 40 GPIO (general-purpose input/output) pins, full-size HDMI output, a microSD card slot for storage and OS, and interfaces like CSI (Camera Serial Interface) and DSI (Display Serial Interface). It supports four USB 2.0 ports, a 10/100 Ethernet port, built-in 802.11n Wi-Fi, and Bluetooth 4.1. The board can be powered via a micro-USB power supply rated at 5V/2.5A and is capable of running a full Linux operating system, such as Raspbian. It can handle tasks ranging from media playback and simple computing to embedded and IoT projects. With HDMI and USB support, it connects easily to monitors, keyboards, and peripherals, making it a compact and powerful tool for prototyping and development work.

The GPIO pins allow direct control of electronic components like LEDs, sensors, and motors, making it suitable for hardware interfacing. Its versatility has made it popular in educational, industrial, and assistive technology applications. The onboard wireless capabilities allow it to be used in networked environments without additional adapters. It also supports Python and other programming languages, enabling easy integration with computer vision, automation, and machine learning applications. Due to its low power consumption and high flexibility, it is widely used in smart home systems, robotics.

Figure 3.6.1: Raspberry Pi



Web Camera

The webcam features automatic focus adjustment, which allows it to maintain image clarity without manual calibration, especially important when scanning documents at varying distances or angles. Additionally, auto white balance and low-light compensation are supported to adapt to different lighting environments, making the system usable indoors or in dim conditions.

It connects to the Raspberry Pi via USB 2.0, which is natively supported by the Pi's four USB ports, eliminating the need for external drivers. The webcam is fully compatible with OpenCV, the open-source computer vision library used in the project, enabling real-time image capture and processing using Python scripts.

For OCR accuracy, the camera must provide images with high contrast and minimal motion blur. Therefore, webcams with a CMOS sensor and a frame rate of at least 30 fps are preferred, as they can provide smoother and more detailed frames for processing.

The webcam used in this project is not only cost-effective and compact but also reliable for continuous image acquisition. Its features ensure that the printed text is captured with enough quality for successful segmentation, feature extraction, and recognition using the Tesseract OCR engine.

Product Features

- Auto Focus Adjustment Maintains sharp image clarity without manual calibration, ideal for scanning documents at varying distances or angles.
- Auto White Balance Automatically adjusts color tones for natural-looking images in different lighting environments.
- Low-Light Compensation Enhances visibility and performance in dimly lit or indoor settings.
- USB 2.0 Connectivity Plug-and-play connection
- CMOS Image Sensor Captures high-quality, low-noise images with better light sensitivity, essential for clear text recognition.

- 30 FPS Frame Rate Delivers smooth and detailed video frames, reducing motion blur during image capture.
- OpenCV Compatibility Fully supported by OpenCV for real-time image processing and integration with Python-based applications.
- Compact and Lightweight Design Space-efficient and easy to mount or reposition for flexible project setups.
- Cost-Effective and Reliable Affordable without compromising on continuous performance, making it ideal for long-duration tasks like OCR scanning.

Key Specifications:

- **Sensor Type**: CMOS
- **Resolution**: 720p / 1080p (based on model)
- **Frame Rate**: 30 frames per second (fps)
- **Focus**: Auto Focus
- **White Balance**: Automatic
- **Low-Light Performance**: Supported (with compensation)
- **Interface**: USB 2.0
- **Compatibility**: Raspberry Pi, OpenCV, Python
- **Operating System Support**: Linux (Raspberry Pi OS), Windows
- **Power Supply**: USB-powered (5V)

Figure 3.6.2: Web Camera

3.6.2 Memory Card

The SD card is used as the primary storage medium for the Raspberry Pi, housing the operating system (Raspbian) and all necessary software components including the OCR engine, TTS engine, and Python scripts. It also stores the captured images and the extracted text data for further processing. The SD card ensures that all the system files and user data are readily accessible during operation. Additionally, it provides a compact and portable solution for saving processed outputs and maintaining the system's functionality.

Features:

1. Resolution: High-definition (typically 720p or 1080p) for clear image capture.
2. Interface: USB or CSI (Camera Serial Interface) compatible with Raspberry Pi.
3. Frame Rate: Supports real-time image capture, usually 30 frames per second (fps).
4. Focus Type: Fixed or auto-focus to maintain clarity of text in various lighting conditions.
5. Plug-and-Play Support: Easily connects to Raspberry Pi without requiring special drivers.
6. Low Power Consumption: Suitable for embedded systems with limited power supply.
7. Compact Design: Small and lightweight, making it easy to integrate into portable assistive devices.
8. Wide Compatibility: Works with image processing libraries like OpenCV and OCR tools like Tesseract.

Figure 3.6.3: R307 – Micro SD Card

Specifications:

- **Type**: microSD card
- **Capacity**: Minimum 8 GB (16 GB or higher recommended)
- **Speed Class**:
  - Class 10 or
  - UHS-I (Ultra High Speed Class 1)
- **Format**: FAT32 for cards ≤32 GB or exFAT for larger cards

A high-speed and high-capacity microSD card ensures smooth operation of Raspbian

OS and storage for OCR and image processing tasks.

SPEAKER \ EARPHONE

Figure 3.6.4 : speaker and 3.5 mm jack

The hardware model consists of a Raspberry Pi, HD camera, and speaker or Bluetooth headset. The camera captures images of printed text and sends them to the Raspberry Pi. The Pi processes the images using Python and Tesseract OCR to extract text. The extracted text is converted to speech using a Text-to-Speech (TTS) module. The audio is then played through the output device, allowing the user to hear the content.

Specifications:

For this OCR-based assistive system, the **speaker specifications** should ensure clear and audible sound output suitable for spoken text in various environments. Recommended specifications include:

- **Type**: Portable USB or 3.5mm jack speaker / Bluetooth speaker (if wireless is preferred)
- **Power Output**: 3W to 5W RMS (sufficient for clear voice output)
- **Frequency Response**: 100 Hz – 18 kHz (optimized for speech clarity)
- **Connectivity**: Compatible with Raspberry Pi via USB, 3.5mm audio jack, or Bluetooth
- **Power Supply**: USB-powered or battery-operated for portability

3.6.3 Power Cable

Figure 3.6.5: Power Cable

The power cable is used to supply electrical power to the Raspberry Pi, which is the core processing unit of the system. Typically, a micro-USB or USB-C power cable (depending on the Raspberry Pi model) is connected to a 5V, 2.5A–3A power adapter plugged into a standard wall socket. This stable power supply is essential for running the Raspberry Pi, camera, OCR processing, and Text-to-Speech functions without interruption. A reliable power cable ensures continuous operation of the hardware and prevents unexpected shutdowns during image processing or speech output.

3.6.4 Software

Image Processing

Books and papers have letters. Our aim is to extract these letters and convert them into digital form and then recite it accordingly. Image processing is used to obtain the letters. Image processing is basically a set of functions that is used upon an image format to deduce some information from it. The input is an image while the output can be an image or set of parameters obtained from the image. Once the image is being loaded, we can convert it into gray scale image. The image which we get is now in the form of pixels within a specific range. This range is used to determine the letters. In gray scale, the image has either white or black content; the white will mostly be the spacing between words or blank space.

Feature Extraction

In this stage we gather the essential features of the image called feature maps. One such method is to detect the edges in the image, as they will contain the required text. For this we can use various axes detecting techniques like: Sobel, Kirsch, Canny, Prewitt etc. The most accurate in finding the four directional axes: horizontal, vertical, right diagonal and left diagonal is the Kirsch detector. This technique uses the eight point neighborhood of each pixel.

Optical Character Recognition

Optical character recognition, usually abbreviated to OCR, is the mechanical or electronic conversion of scanned images of handwritten, typewritten or printed text into machine encoded text. It is widely used as a form of data entry from some sort of original paper data source, whether documents, sales receipts, mail, or any number of printed records. It is crucial to the computerization of printed texts so that they can be electronically searched, stored more compactly, displayed on-line and used in machine processes such as machine translation, textto- speech and text mining. OCR is a field of research in pattern recognition, artificial intelligence and computer vision.

Tesseract

Tesseract is a free software optical character recognition engine for various operating systems. Tesseract is considered as one of the most accurate free software OCR engines currently available. It is available for Linux, Windows and Mac OS.

An image with the text is given as input to the Tesseract engine that is command based tool. Then it is processed by Tesseract command. Tesseract command takes two arguments: First argument is image file name that contains text and second argument is output text file in which, extracted text is stored. The output file extension is given as .txt by Tesseract, so no need to specify the file extension while specifying the output file name as a second argument in Tesseract command. After processing is completed, the content of the output is present in .txt file. In simple images with or without color (gray scale), Tesseract provides results with 100% accuracy. But in the case of some complex images Tesseract provides better accuracy results if the images are in the gray scale mode as compared to color images. Although Tesseract is command-based tool but as it is open source and it is available in the form of Dynamic Link Library, it can be easily made available in graphics mode.

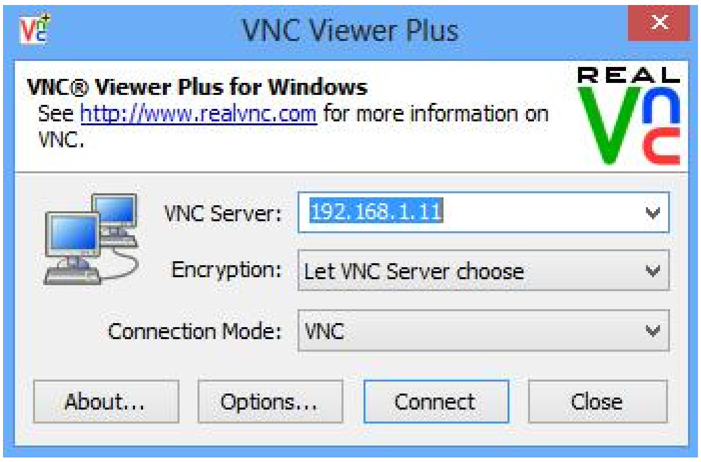


Figure 3.6.6: VNC software

Thonny

A beginner-friendly Integrated Development Environment (IDE) designed specifically for learning and writing **Python** code. It comes pre-installed with Raspberry Pi OS and is ideal for projects like yours that involve OCR and Text-to-Speech, as it provides a simple and clean interface to develop and run Python scripts.

Key Features of Thonny:

- Simple Interface: Minimal clutter, making it perfect for beginners and educational use.
- Built-in Python Shell: Allows interactive testing and debugging of Python code.
- Debugger: Step-by-step execution helps in understanding how the code works.
- Package Manager: Easy installation of Python libraries such as pytesseract, opencv, and pyttsx3.
- Error Highlighting: Helps in quickly spotting syntax or runtime errors.

Use in This Project:

In OCR-based assistive system, Thonny is used to write and test the Python code that captures the image, processes it using Tesseract OCR, and converts it to speech. Its straightforward environment supports rapid development and debugging, making it ideal for deploying and tweaking code on the Raspberry Pi.

OpenCV

- Preprocessing images to enhance OCR accuracy.
- opencv-python (commonly imported as cv2)
- OpenCV provides tools for image enhancement like grayscale conversion, thresholding, noise reduction, and edge detection. These are crucial for improving the input quality before feeding the image into Tesseract.

Translation Module

- PurposeTranslates extracted text into Tamil.
- Library Options
    - Online: google trans (unofficial Google Translate API)
    - Offline (advanced): argos-translate or custom trained models using transformers + ONNX

.

Block Diagram

The block diagram outlines the process flow of an **Optical Character Recognition (OCR) system with Text-to-Speech (TTS)** functionality designed for visually challenged individuals. It begins with a **Camera (HD)** that captures high-resolution images of printed or handwritten text. These images are sent to the **Raspberry Pi**, which serves as the processing unit, running Python scripts and managing each stage of the system. Inside the Pi, the image undergoes **segmentation**, where it is broken into lines, words, and characters, followed by **feature extraction**, which analyzes character shapes for accurate recognition. The processed data is then fed into the **Tesseract OCR engine**, which converts the visual input into editable text.

In the final phase, the extracted text is passed to a **Text-to-Speech (TTS)** module, which transforms the text into audible speech using Python libraries such as pyttsx3 or espeak. The resulting audio is output through a **speaker or Bluetooth headset**, enabling visually impaired users to hear the text content. This end-to-end process—from image capture to speech output—makes the system a practical and accessible solution for real-time reading assistance.

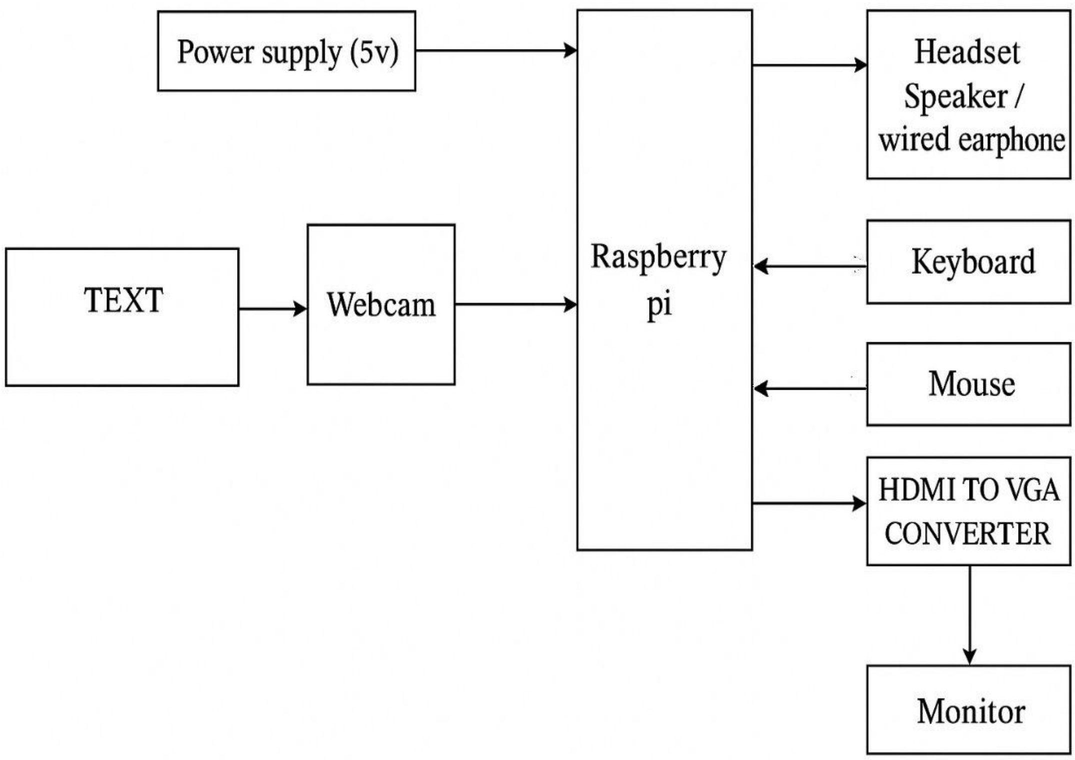


Figure 3.7: Block Diagram of Raspberry Pi Based Translation Companion

3.8 Flow Chart

The flowchart illustrates the step-by-step process of an English-to-Tamil Text-to-Speech Conversion System that assists users—particularly the visually impaired—by converting written English text from an image into spoken Tamil.

It begins with image capture using a camera, followed by image preprocessing to enhance quality. The Tesseract OCR engine then extracts English text from the image. This text is translated into Tamil using a Translator API. The translated Tamil text is then sent to a Tamil Text-to-Speech engine, which generates audio output that is played through a speaker or headset. The process ends after the Tamil speech is delivered.

The flowchart defines a linear, automated pipeline combining OCR, translation, and TTS to make printed English content audible in Tamil, highlighting each functional block in the system from start to finish.

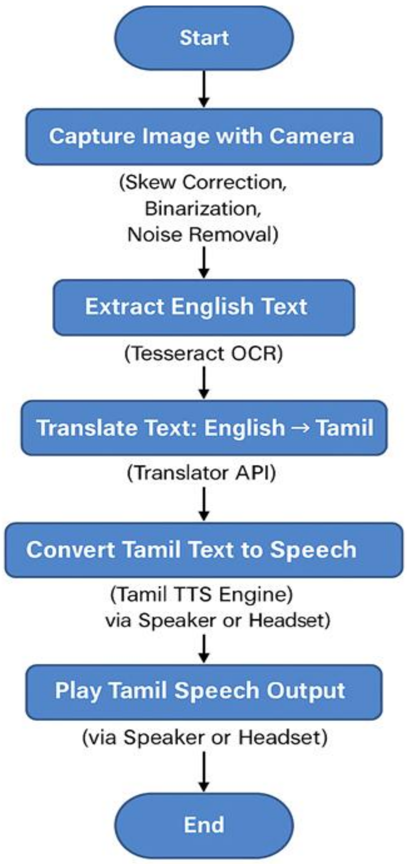


Figure 3.8: Flow Chart of Raspberry Pi Based Companion Translation

CHAPTER 4

RESULTS AND DISCUSSION

The proposed OCR system using Raspberry Pi successfully assists visually challenged individuals by converting printed text into audible speech. The system employs a high-definition camera to capture printed documents, processes them using Python-based OCR (Tesseract), and outputs the recognized text through a text-to-speech (TTS) engine.

Experimental Results:

- Image Acquisition: The HD camera captures clear, high-resolution images of the text (Figure 4).
- Pre-processing: The images undergo skew correction, binarization, and noise removal to improve OCR accuracy (Figure 5 - grayscale and edge-detected image).
- Text Recognition: Tesseract OCR engine extracts characters from processed images effectively (Figure 6).
- Spell Correction: A spell-check module enhances the accuracy of the recognized text (Figure 7).

System Performance:

- The recognition accuracy is acceptable for most printed texts.
- The use of spell correction further improves output readability.
- A higher resolution camera significantly enhances OCR accuracy.
- The system operates efficiently on Raspberry Pi 3 Model B with minimal delay in text recognition and speech output.

User Benefit:

- The system allows blind users to independently read printed material without reliance on Braille or digital formats.
- Output via Bluetooth headset or speaker enables discreet and convenient usage.

Limitations and Future Scope:

- Recognition accuracy may reduce for handwritten or stylized fonts.
- OCR performance can degrade under poor lighting conditions or low image quality.
- Future improvements could include better language models, neural OCR, and real-time camera feedback.

Steps involved in this project:

- Image Capturing
    - A high-definition camera connected to Raspberry Pi captures an image of the printed text.
    - Ensures clear and focused input for processing.
- Pre-Processing of Image
    - Skew Correction: Adjusts tilted images to align with the horizontal axis.
    - Binarization: Converts the image to black and white for better text recognition.
    - Noise Removal: Cleans the image by removing unwanted spots and distortions.
- Segmentation
    - Divides the processed image into:
        - Lines
        - Words
        - Characters
    - Uses histogram-based methods for accurate spacing and alignment detection.
- Feature Extraction
    - Identifies and extracts features from each character:
        - Width, height
        - Arcs and lines
        - Circles
        - Centroid positions
- Text Recognition (OCR)
    - Uses Tesseract OCR Engine to recognize and extract characters from segmented images.
    - Outputs machine-readable text from the captured image.
- Spell Correction
    - Applies a spell-check module to improve the accuracy of the recognized text.
- Text-to-Speech Conversion (TTS)
    - Converts the final corrected text into speech.
    - Outputs through a Bluetooth headset or speaker.
- User Interaction
    - User activates the system with a simple button press.
    - The system reads the document aloud, aiding visually challenged users.

Figure 4.1: Working Hardware Model

The hardware model consists of a Raspberry Pi, HD camera, and speaker or Bluetooth headset. The camera captures images of printed text and sends them to the Raspberry Pi. The Pi processes the images using Python and Tesseract OCR to extract text. The extracted text is converted to speech using a Text-to-Speech (TTS) module. The audio is then played through the output device, allowing the user to hear the content.

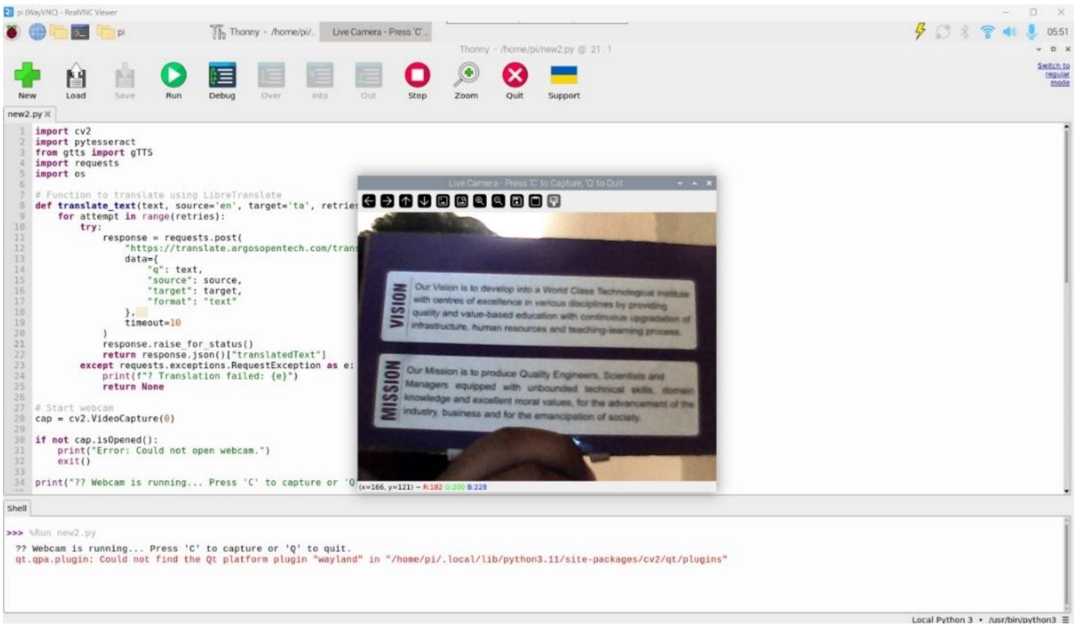


Figure : Image Captured by Web Camera

CHAPTER 5

CONCLUSION

This project successfully demonstrates the implementation of an assistive system that enables visually challenged individuals to access printed text using optical character recognition (OCR) and text-to-speech (TTS) technologies. By integrating a Raspberry Pi, HD camera, and open-source libraries like Tesseract and OpenCV, the system captures, processes, and converts printed text into audible speech in real-time. The compact, cost-effective design ensures portability and ease of use, making it practical for everyday applications in homes, libraries, and public spaces. The solution not only enhances independence and accessibility for the visually impaired but also opens up opportunities for broader applications such as document digitization and audio-based education. Future improvements may include multi-language support, better accuracy for handwritten text, and integration with mobile or cloud platforms to expand usability and functionality

REFERENCE

[1] Faiz Alotaibi, Muhamad Taufik Abdullah, "Optical Character Recognition for Quranic image similarity matching,"IEEE,2017.

[2] Shalini Sonth, Jagadish S kallimai, "OCR based facilitator for the visually challenged,"IEEE,2017.

[3] Michael D.kim and Jun Ueda, "Dynamics-based motion de-blurring improves the performance of Optical Character Recognition during fast scanning of a robotic eye,"IEEE,2017.

[4] Sanjana, Monisha, "OCR based automatic book reader for the visually impaired using raspberry pi,"IEEE,2016.

[5] Velumurugan D. et al, "Hardware Implementation of Smart Reader for Visually Impaired using Raspberry Pi", IJAREEIE, March 2016.

[6] N. Venkata Rao et al, " Optical Character Technique Recognition Algorithms, Journal of Theoretical and Applied Information Technology",20th January 2016.

[7] K. Nirmala Kumari et al, "Image Text to Speech Conversion using OCR Technique in Raspberry Pi", IJAREEIE, May 2016.

[8] Seiichi Uchida, Koichi Kise, "OCR fonts revisited for camera-based character recognition,"IEEE,2016.

[9] D.Velmurugan et al, " A Smart Reader for Visually Impaired People Using Raspberry PI",IJSEC, Vol 6 ,pp 2997-3001

[10] Nagaraja L. et al, " Vision Based Text Recognition using Raspberry Pi, International Journal of Computer Applications", NCPSIA 2015.

[11] Kwang in Kim, Keechul Jung, "Texture-based approach for text detection in images,"IEEE,2013.

[12] https://raspberry.piaustralia.com.au/raspberry-pi-3-model-b