Reading Assistance System for Visually Impaired Using Raspberry Pi

Samuel Santhosh K[1], Shanmitha S[2], Tarukesh Raj T J[3], Aaron Kevin Cameron[4]
{ 21ec70@kcgcollege.com[1], 21ec77@kcgcollege.com[2], 21ec307@kcgcollege.com[3], aaron.ece@kcgcollege.com[4] }

ECE Department, KCG College of Technology, Chennai, Tamil Nadu, India[1, 2, 3, 4]

Abstract. The lack of ability to see the objects and text in real-life situation is one of the most problems for visual impaired peoples which, as a result, create limitations in having access to information and communicating with others. Despite providing an effective, method of reading, the Braille system is slow and burdens the reader with considerable effort to generate text. Thus, we introduce a new solution, a reading companion using RMVA technology, which flows from smart reading to fast reading, reading in all directions and collision avoidance reading, based on the Raspberry Pi to help them to read by easier way. The system uses a camera (a webcam or a mobile camera which is connected to a Raspberry Pi) for the acquisition of the printed text images. These images are preprocessed with techniques such as skew correction, segmentation, and feature extraction, and then Optic Character Recognition is carried out with the help of Tesseract OCR engine. The obtained text content is further converted into machine readable form and synthesized in audio by TTS synthesis in order to make the discrete blind users easily comprehend the text messages by recognizing their text content. The system has three main functional modules: scene capture which basically involves having a camera module capture the text and and sending it to the processing platform, data processing (OCR software) extracts the connected text from the image and separate it from the image text and non-image text, speak output which involves the generating of audio from the elapsed text. The camera captures the information, uses OCR to process it and then generation Text to Speech (TTS) either on your PC/mobile phone by interfacing with Python (using PyTesseract and gTTS libraries) and provides it to the visually impaired person.

Keyword: Raspberry Pi, Optical Character Recognition (OCR), Pre-processing, Tesseract, Google Text-to-Speech (gTTS), Python Programming.2023

1 Introduction

There are an estimated 314 million blind or vision-impaired individuals worldwide, of whom approximately 45 million are totally blind (World Health Organization [WHO], 2021). In the United States, 25.2 million adults—over 8% of the population—were reported to have blindness or visual impairment as of 2008, according to the National Health Interview Survey (National Federation of the Blind [NFB], 2020). Vision loss significantly impacts daily life, complicating activities such as reading, mobility, and workplace participation. Research further indicates that individuals with visual impairment (VI) face additional barriers, including limited access to healthcare, high costs of adaptive equipment, and a lack of inclusivity in education and employment (International Agency for the Prevention of Blindness [IAPB], 2022; Royal National Institute of Blind People [RNIB], 2020).

ICITSM-Part II 2025, April 28-29, Tiruchengode, India

DOI 10.4108/eai.28-4-2025.2358084

In the U.S. alone, an estimated 7.6 million people live with varying degrees of visual impairment or blindness (NFB, 2020). Nine out of ten blind individuals face even more significant difficulties in low-income nations because of restricted healthcare and resourcing (World Blind Union [WBU], 2020). These challenges are further multiplied by economic factors, as some 70% of blind adults are jobless (European Blind Union [EBU], 2021), highlighting the imbalance related to educational and employment opportunities.

Table 1. Summarised Key Statistics (Numbers) Of Visually Impaired Individuals from Various Organizations.

Organization	Year	Key Statistics (Numbers)
WHO	2021	2.2 billion people with vision impairment
IAPB	2022	90% of visually impaired people live in low-income settings
NFB (USA)	2020	7.6 million visually impaired in the U.S.
RNIB (UK)	2020	72% unemployment rate among blind/partially sighted adults
WBU	2020	70% unemployment among blind adults globally
EBU	2021	30 million visually impaired in Europe
APH	2019	15% of visually impaired U.S. students lack access to adapted learning resources

Table 1 Overview of visual impairment by leading international and regional sources. The scale of these numbers underscores the general difficulties faced by people with visual impairment in achieving education and employment opportunities. These problems are even worse due to the absence of low-cost assistive computer aids, such as Optical Character Recognition (OCR) devices and screen readers. This highlights the critical need for engineering tools that enable significantly increased like spaciousness and better quality of life to those who are blind. In this work, we aim to advance assistive technology that makes consuming text-based content faster and cheaper.

2 Literature Survey

1. Assistive Reading Landscape and Design Goals

Assistive reading technologies have been designed to allow visually impaired people to read printed material in real time [1] [13] [14]. Traditionally such systems involve text image acquisition, its recognition via OCR and extraction of the result for a sound output: to synthesize speech [15]. Study highlights the need for a low-cost system that is portable in nature and also offline working to enable global reach independent of internet connectivity [16] [17]. On-device solutions, for instance the ones based on a Raspberry Pi system that provide privacy, inexpensive and independence, are clearly advantageous especially to blind users [19] [20] [21].

2. Embedded Platform: Raspberry Pi as the Core

Raspberry Pi became a popular choice for assistive applications because of its cost-effectiveness, computational power and large community support [3] [11]. Following the launch of Raspberry Pi 4 and 5, it is possible to run lightweight deep learning frameworks such as TensorFlow Lite and ONNX [12]. Options are available like a Raspberry Pi AI Kit, accelerated by accelerators such as Hailo-8L which increases processing speed to deliver real time inference for requested tasks on text detection and recognition [6] [7]. This renders Raspberry Pi a viable platform for an assistive reading system [2].

3. Image Capture and Pre-Processing for OCR

The reading accuracy of the OCR operation is directly related to image pre-processing quality. Denoising, illumination correction, contrast increase, deskew, and binarization are identified through studies as the critical steps to enhance recognition accuracy [15]. For images taken by hand, such as a skewed and poorly lit ones, these processes greatly reduce recognition errors[5]. Methods including, thresholding [8] have been used to adjust the input images for more reliable recognition on embedded devices [6].

4. Text Detection in Natural Scenes

Conventional techniques like Stroke Width Transform (SWT) and Regions (MSER) are supplanted by deep learning-based detectors [4]. Since models such as the Efficient and Accurate Scene Text (EAST) detector [12] and DB networks provide real-time performance in handling multi-oriented and curved text. Such lightweight models can be plugged into the OpenCV deep neural network (DNN) module, allowing text regions to be efficiently extracted at reliable quality prior to OCR [3].

5. Text Recognition Techniques

Once text areas are located, recognition is usually performed by OCR engines. Regarding OCR, Tesseract OCR (and especially its LSTM-based version v4+) continues to represent a popular and lightweight application for the Raspberry Pi [15]. For more complicated cases like curved or styled texts, Convolutional Recurrent Neural Networks (CRNN) and transformer-based recognizers (e.g., TrOCR) would achieve better recognition accuracy [4]. But CRNN and other deep learning-based recognizers are too computationally intensive, and have to be optimized or installed in Raspberry Pi as another esoteric accelerator [12].

6. Multilingual and Low-Resource Script Challenges

A significant challenge in reading assistance is support for multilingual text, especially in low-resource scripts such as many Indian languages. While OCR systems like Tesseract cover a wide range of scripts, accuracy varies significantly depending on the language and quality of training data [8]. Recent research explores script-agnostic detection and recognition methods, enabling systems to generalize better to unseen languages [1] [13]. Such advancements are crucial for developing inclusive reading assistance tools in multilingual societies [14].

7. Text-to-Speech (TTS) Integration

Converting recognized text into speech is the final stage of the reading pipeline [9]. Lightweight TTS engines such as eSpeak NG are widely adopted on Raspberry Pi due to their compact size and multilingual support [10]. More recent advancements such as Piper TTS provide neural voice synthesis with improved naturalness while still being optimized for Raspberry Pi hardware [12]. For higher performance systems, models like Tacotron 2, FastSpeech, or VITS provide human-like voices but often require greater computational power than a Raspberry Pi can provide without acceleration [9] [10].

4 Proposed System

Fig 1 shows that the proposed system is designed to capture and process printed text using a Pi camera connected to a Raspberry Pi. When a document is placed in front of the camera, it takes an image and processes it using OCR technology powered by the Tesseract library. This allows the scanned text to be converted into editable digital content. A text-to-speech library then transforms the extracted text into audio, making it accessible for listening. OpenCV is used to analyse the image, extract labels, and recognize text for voice output. The generated audio can be heard through a headset connected via a 3.5mm audio jack or wirelessly through Bluetooth speakers.

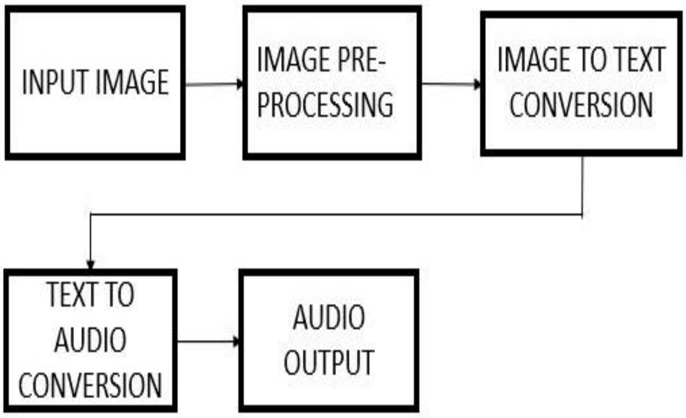


Fig. 1. Proposed System Block-Diagram.

5 Preliminaries

A. Optical Character Recognition (OCR)

Optical Character Recognition (OCR) is a technology that enables automatic recognition of characters through an optical mechanism. Just as the human eye functions as an optical system to capture visual inputs, OCR processes images to extract text, mimicking the human ability to read. However, while OCR technology offers substantial capabilities, it does not yet match the nuanced reading abilities of humans. OCR systems can identify both printed and handwritten text, though their performance is heavily influenced by the quality of the input documents. For instance, OCR works best with images that contain mostly text, with minimal clutter, such as those captured by a mobile camera. In recent developments, Fig 2 shows OCR Block Diagram.

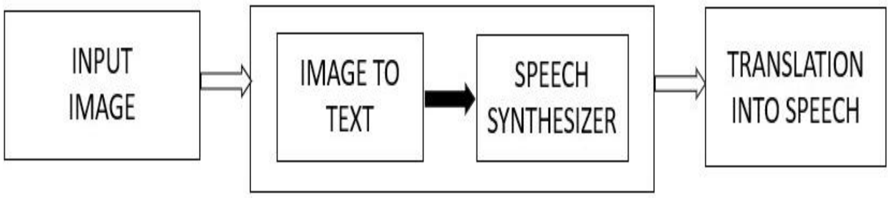


Fig. 2. OCR Block Diagram.

OCR is now also easily available on mobile platforms, especially on Android devices, leveraging open-source solutions such as Google's Tesseract OCR engine. This application enables users of mobile devices to photograph text and have the results translated into audio via text-to-speech synthesizers. The usefulness is further augmented by means of Google's language translation services, which turns OCR into a potential application for translating and reading printed text out loud right from images shot in mobile devices. Conventional Chinese character recognition systems typically are implemented with the assistance of scanner or digital camera to capture the characters and computer software to process the captured images. But a major drawback to these types of systems is that they require additional hardware such as the scanners or cameras, which takes up a lot of space and requires a fair share of time to set up. To mitigate these drawbacks, character recognition systems on mobile devices, such as Android smart phones, have been suggested. Other than reducing costs, these systems require no proprietary or expensive hardware, they may be used in a remote environment and from the textual library, and they can be more flexible in a portable sense.

B. Tesseract

Tesseract is perceived as one of the most powerful open-source OCR engines in the world because of its high performance, extensibility, and active community of developers. It is easy to use and can even be used as a drop-in replacement, making it the perfect choice for researchers or developers. In order to recognise the characters properly, an OCR system (using Tesseract) includes the following major steps: Segmentation, Feature extraction and Character classification. Segmentation is the process of determining the individual glyphs or characters on the binary image. This is needed to distinguish between the text and background. Then, glyph analysis is carried out and and a numeric vector representing features is constructed. This step is usually difficult as there is no straightforward approach to choose the most suitable feature from the glyphs. Finally, classification provides the character associated to the feature, i.e. performs recognition as last step of an OCR system.

These processes attempt to simulate up to a point, the complex process of reading, of course, in a human way not free of limitations. Tesseract is based on its Page Layout Analysis (PLA) technology developed independently and it can accept input images as binary images. It is adaptive to view both regular black-on-white text and inverse white-on-black text. It begins with the description of components, and is preserved in the course of connected component analysis. The nested outlines form blobs, and the latter are used to form lines of text. Text lines are examined for constant pitch versus proportional text. Fixed-pitch text is segmented by character cell and proportional text is segmented using a combination of definitive and blurred spaces. Tesseract's recognition process is divided into two main passes. The first pass tries to identify the words. Below this level, when a frame is recognized correctly, the words that match it contribute to the training of acceptance, this being from the recognition to recognition. The second pass processes the incorrectly recognized words in the first pass by reading across the line to enhance accuracy. In the last stage the fuzzy spaces are cleaned up by Tesseract after which it can handle text layouts. Tesseract also analyzes if it can recognize lowercase and uppercase letters of that specific font by considering the alternate hypotheses for the x-height when processing the checker.

C. Preprocessing

Once an image is captured, it goes through a preprocessing phase to enhance its quality, ensuring better accuracy during text extraction. This crucial step involves multiple techniques to refine the image before optical character recognition (OCR) is applied. First, noise removal is performed to eliminate any unwanted distortions that may interfere with text readability. Fig 3 shows Grayscale Conversion of RGB Image

Fig. 3. Grayscale Conversion of RGB Image

The visible characters are then adjusted with brightness and contrast to enhance their visibility against the background. Furthermore, if the text is skewed or misaligned, skew correction is performed so that the text is correctly aligned during the extraction. The image is additionally transformed into a binary (black and white) image, simplifying the processing of separating the text from the shape. These preprocessing methods are very important in enhancing the quality and the accuracy of OCR system and in guaranteeing the accuracy of text recognition results.

6 System Design Specification

A. Hardware Description

1) Raspberry Pi: The Raspberry Pi is a low-cost, credit cardsized computer used by hobbyists, educators and makers to experiment and build prototypes. Background: Great for media centers, robotics, and the Internet of Things because of its support for a lot of peripherals and programming languages. It's been done, and is doable for software development, electronics projects, etc., with its GPIO and USB and network ports and whatnot. What is recycled the recycled component in fig 4 the Pi4 model B is, replacing features used in the project:

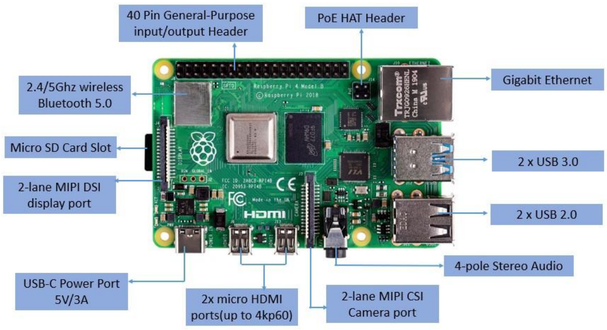


Fig. 4. Raspberry Pi 4 Model B

Processor: Broadcom BCM2711, 64-bit, Quad-core Cortex-A72 (ARM v8) processor at 1.8GHz

- Memory: Available in 1GB, 2GB, 4GB, or 8GB LPDDR4-3200 SDRAM configurations
- Wi-Fi Connectivity: Dual-band Wi-Fi (2.4 GHz & 5.0 GHz) with Bluetooth 5.0 and BLE support
- Networking: Gigabit Ethernet for wired networking at high speeds
- USB Ports: Two USB 3.0 and two USB 2.0 ports for external device connections
- GPIO: Standard 40-pin GPIO header with back-compatibility with previous models
- Display Output: Two micro-HDMI output terminals with support for 4Kp60 resolutions
- Camera & Display Interface: Includes a 2-lane MIPI DSI display port and a 2-lane MIPI CSI camera port
- Audio & Video Output: 4-pole stereo audio jack and composite video output
- Video Processing: Supports H.265 decoding (4Kp60) and H.264 encoding/decoding (1080p60/1080p30)
- Graphics: OpenGL ES 3.1 and Vulkan 1.0 support for enhanced rendering
- Storage: Micro-SD card slot for OS installation and storage
- Power Supply: It can be powered either by the 5V DC USB-C connector or GPIO header (minimum 3A required)
- Power over Ethernet (PoE): Supported with an additional PoE HAT
- Operating Temperature: It can operate at ambient temperatures between 0°C to 50°C2)

2) Camera Module: It is an essential hardware for photo enthusiasts and it is also combine the official camera board of raspberry Pi, support all revisions of the Pi. Through the use of image processing and OCR technology, it reads recognisable text out loud with a human-like voice. It is suitable for low-light environments – Offers efficiency and accuracy. Based on the processing power of Raspberry Pi, it can provide fast results, so that it is an assistive tool. Zebronics Web Crystal Pro camera is used in the project for OCR and TTS purposes. Featuring high resolution, autofocus, noise-reduction and a wide-angle lens to provide high-quality pictures that are clear enough for text recognition. Some characteristics of the webcam utilized for this project:

- It is a web camera powered by USB with a 3P lens that gives clear video.
- It comes with an integrated mic, night vision feature, and clip-on feature for easy mounting.

- Interface: USB, Image Sensor: CMOS, Lens: 3P quality lens
- Resolution: 640 x 480 (30 FPS), Cable Length: 1.2 meters.

Fig. 5. Camera Module.

This compact and lightweight 3P camera module can record 1080p video and capture photos. It interfaces directly with the Raspberry using USB. Once connected, simply boot the latest Raspbian OS to utilize it. Optimized for integration, this module is commonly used in image processing, machine learning, and surveillance applications. Fig 5 shows Camera Module.

B. Software Description

Fig. 6. Raspbian OS.

1) Raspbian OS: Fig 6 illustrates Raspberry Pi OS, formerly called Raspbian, a free operating system designed specifically for Raspberry Pi devices. Built on Debian Linux, it is optimized for the ARM architecture, ensuring efficiency, stability, and ease of use. Available in both a full desktop version and a Lite command-line edition, it caters to different user needs. With an intuitive interface, it is beginner-friendly and comes pre-installed with essential software like Python (Fig 7), Scratch, GPIO libraries, and office tools. Raspberry Pi OS supports various applications, including educational programming, DIY electronics, server setups, and media centers. It also provides essential libraries and drivers for seamless hardware and software integration, enabling smooth operation and real-time processing.

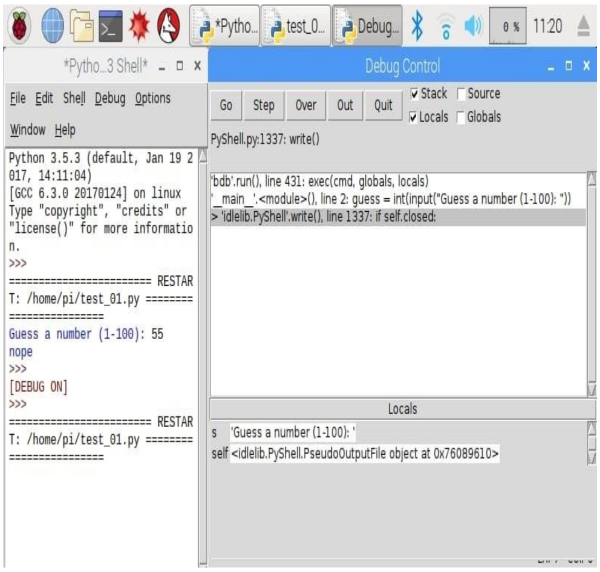


Fig. 7. Python IDE.

2) OCR: python-tesseract is an optical character recognition (OCR) tool for python. It is a wrap which provides access to the Tesseract-OCR engine It serves as a wrap for tesseract and adding an additional protection to import and use it into your project (even all the tesseract source files). It works with the Pillow and Leptonica imaging libraries to provide a powerful open-source OCR capability. Python-Tesseract is not only a Python library, but it can be used as a standalone script as well, to print text from the recognized images isrinkt of saving it to a file. The important functions are in the following table 2.

Table 2. Key Functions of Tesseract in Python.

get tesseract version	Tesseract version.
image to string	text from an image.
image to data	detailed OCR results
run and get output	raw output from Tesseract OCR

3) TTS: gTTS is a Python-based library that converts text into speech using Google's Text-to-Speech technology. In this application, it helps transform extracted text from images into spoken audio. After the OCR process identifies text from a document image, gTTS generates a corresponding voice output. This feature is especially useful for individuals with visual impairments, allowing them to listen to printed content instead of reading it. Fig 8 illustrates the flow process of the gTTS.

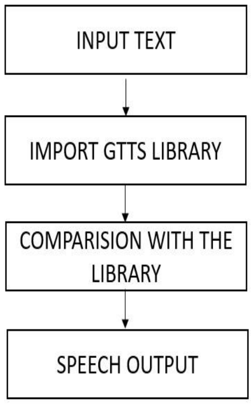
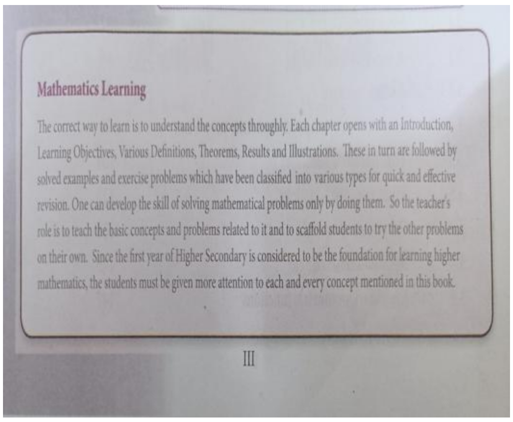


Fig. 8. gTTS Flow Chart.

7 System Implementation

A. Image Acquisition and Pre-Processing

Capturing a photograph of a text document using the Raspberry Pi camera, the process is illustrated in fig 9. The subsequent quality of OCR is very dependent on the quality of this image. Lighting is essential to avoid shadows and glare, which can make text illegible. The camera should be in a good focus setting not only because of aesthetic reasons (blurry pictures are unappealing), but also because of how unclear images might cause problems when extracting text. And positioning the document properly flat and with the camera parallel to it reduces distortion. Using a high-resolution camera helps with the clarity of the text and minimizes the risk of OCR processing errors.



Mathematics Learning

The correct way to learn is to understand the concepts throughly. Each chapter opens with an Introduction, Learning Objectives, Various Definitions, Theorems, Results and Illustrations. These in turn are followed by solved examples and exercise problems which have been classified into various types for quick and effective revision. One can develop the skill of solving mathematical problems only by doing them. So the teacher's role is to teach the basic concepts and problems related to it and to scaffold students to try the other problems on their own. Since the first year of Higher Secondary is considered to be the foundation for learning higher mathematics, the students must be given more attention to each and every concept mentioned in this book.

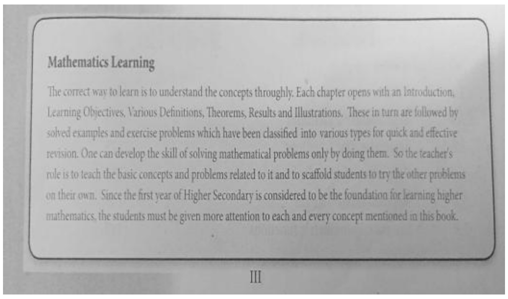
III

Fig. 9. Sample Image Captured by Camera Module.

After taking the image, pre-processing method is use to enhance the image quality for better text extraction as shown in fig 10. It also involves several stages starting from removal of the noise and unnatural visual attributes, like dust or wrinkles, and ending with processing of digitizing artefacts. The text is still legible at other lighting conditions provided that brightness and contrast are adjusted. Adaptive Thresholding: Applying dynamic-binarization for the better visibility. Skew correction is another significant preprocessing that aligns the text to improve OCR. This is achieved by determining a text angle and compensating for this. Then the binary version of image is threshold to its high contrast value for character recognition. All those steps pave the way to optimal image quality, which means there's even more that the text can be recognized correctly.

B. Text Extraction

After preprocessing, Optical Character Recognition (OCR) technology is used to scan images and extract text by identifying characters and converting them into an editable, machine readable format.



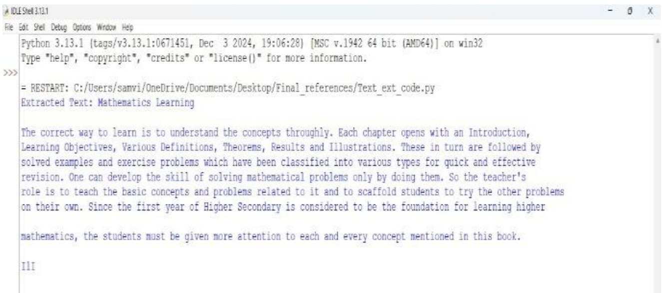
Mathematics Learning

The correct way to learn is to understand the concepts throughly. Each chapter opens with an Introduction, Learning Objectives, Various Definitions, Theorems, Results and Illustrations. These in turn are followed by solved examples and exercise problems which have been classified into various types for quick and effective revision. One can develop the skill of solving mathematical problems only by doing them. So the teacher's role is to teach the basic concepts and problems related to it and to scaffold students to try the other problems on their own. Since the first year of Higher Secondary is considered to be the foundation for learning higher mathematics, the students must be given more attention to each and every concept mentioned in this book.

III

Fig. 10. Sample after Grayscale Conversion.

This project utilizes the Tesseract library, which efficiently detects and processes printed text. Once the text is extracted, it undergoes further refinement to enhance accuracy. This involves error correction and text recognition techniques to resolve inaccuracies caused by unclear characters or formatting issues. The fig 11 below shows the extracted texts from the fig 9 after it is pre-processed.



```
Python 3.13.1 (tags/v3.13.1:0671451, Dec  3 2024, 19:06:28) [MSC v.1942 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/samvi/OneDrive/Documents/Desktop/Final_references/Text_ext_code.py
Extracted Text: Mathematics Learning

The correct way to learn is to understand the concepts throughly. Each chapter opens with an Introduction,
Learning Objectives, Various Definitions, Theorems, Results and Illustrations. These in turn are followed by
solved examples and exercise problems which have been classified into various types for quick and effective
revision. One can develop the skill of solving mathematical problems only by doing them. So the teacher's
role is to teach the basic concepts and problems related to it and to scaffold students to try the other problems
on their own. Since the first year of Higher Secondary is considered to be the foundation for learning higher

mathematics, the students must be given more attention to each and every concept mentioned in this book.

III
```

Fig. 11. Extracted Text using Pytesseract.

C. Audio Output

The processed text is then fed into a Text-to-Speech (TTS) engine, such as gTTS, which converts it into spoken words. The generated audio output can be played through speakers or headphones, offering a voice-based representation of the document for easier accessibility and comprehension.

8 Python Scripts

The following are the python scripts used in the system and their respective outputs (except speech) are shown in the above fig 9,10,11. (shown in fig 11).Fig 12 and 13 are the python script used.

```
text_ext_code.py - C:/Users/samvi/OneDrive/Documents/Desktop/Final_references/text_ext_code.py (3.13.1)
File Edit Format Run Options Window Help
import cv2
import pytesseract
from PIL import Image

def capture_from_camera():
    cap = cv2.VideoCapture(0)

    if not cap.isOpened():
        print("Error: Unable to access the camera.")
        return

    while True:
        ret, frame = cap.read()
        if not ret:
            print("Failed to grab frame.")
            break

        cv2.imshow('Raspberry Pi Camera Feed', frame)

        extracted_text = extract_text_from_frame(frame)
        print("Extracted Text:", extracted_text)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()

def extract_text_from_frame(frame):
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    extracted_text = pytesseract.image_to_string(gray)

    return extracted_text

capture_from_camera()
```

Fig. 12. Python Script for Extraction of text from the Pre-Processed Image.

```
*speech_code.py - C:/Users/samvi/OneDrive/Documents/Desktop/Final_references/speech_code.py (3.13.1)*
File Edit Format Run Options Window Help
import cv2
import pytesseract
from gtts import gTTS
import os

def capture_from_camera():
    cap = cv2.VideoCapture(0)
    if not cap.isOpened():
        print("Error: Unable to access the camera.")
        return
    while True:
        ret, frame = cap.read()
        if not ret:
            print("Failed to grab frame.")
            break
        cv2.imshow('Raspberry Pi Camera Feed', frame)
        extracted_text = extract_text_from_frame(frame)
        print("Extracted Text:", extracted_text)
        if extracted_text.strip():
            text_to_speech(extracted_text)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    cap.release()
    cv2.destroyAllWindows()
def extract_text_from_frame(frame):
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    extracted_text = pytesseract.image_to_string(gray)
    return extracted_text
def text_to_speech(text):
    tts = gTTS(text=text, lang='en')
    tts.save("output.mp3")
    os.system("mpg321 output.mp3")

capture_from_camera()
```

Fig. 13. Python Script for Text-to-Speech Conversion.

9 Overall System Implementation

Fig. 14. Overall Implementation of the System.

This is a prototype of a blind reader project designed to assist visually impaired individuals by scanning and interpreting text. The webcam, mounted on a transparent stand at an angle, is positioned to efficiently capture text placed on the plate below. The surrounding red LED lights indicate built-in illumination, ensuring readability even in low-light conditions. The text is placed on a flat plate as shown in fig 14, providing a stable surface for Optical Character Recognition (OCR). The small red and white box is the Raspberry Pi, which functions as the central processing unit, managing image processing, OCR, and textto-speech conversion. The gold-colored cylindrical speaker serves as the audio output, allowing the system to read aloud the recognized text for the user. It is likely either wireless or connected via a cable for smooth audio transmission.

10 Conclusion

The proposed system, implemented using a Raspberry Pi, efficiently processes input images to generate clear audio output. This cost-effective, portable and compact device aids visually impaired individuals by converting text into speech, enhancing accessibility. While it cannot fully replace Braille, it provides an alternative for reading non-Braille text. Traditional text-to-speech systems may struggle with pronunciation and spelling clarity, which can be mitigated by character by-character recognition though at the cost of processing time. An alternative approach involves Braille displays, where servo motor-based designs offer a more affordable solution compared to conventional models. Future improvements could include a voice-controlled interface to enhance usability and convenience.

References

[1] R. Král, P. Jacko and T. Vince, "Low-Cost Multifunctional Assistive Device for Visually Impaired Individuals," IEEE Access, vol. 13, pp. 56326–56337, 2025, doi: 10.1109/ACCESS.2025.3554366.

[2] Krishnavajjala, S. H. Mansur, J. Jose, and K. Moran, "Motorease: Automated detection of motor impairment accessibility issues in mobile app UIs," in Proc. IEEE/ACM 46th Int. Conf. Softw. Eng. (ICSE), 2024, pp. 2580–2592, doi: 10.1145/3597503.3639167.

[3] G. C. Duarte, D. S. Loubach, and I. Sander, "High-level reconfigurable embedded system design based on heterogeneous models of computation," IEEE Access, vol. 13, pp. 63918–63934, 2025, doi: 10.1109/ACCESS.2025.3559695.

[4] Palanisamy et al., "Transformers for vision: A survey on innovative methods for computer vision," IEEE Access, vol. 13, pp. 95496–95523, 2025, doi: 10.1109/ACCESS.2025.3571735.

[5] Y. Shimoda, T. Sugino, M. Okui, R. Nishihama, and T. Nakamura, "Knee joint assist in seated motion focusing on variable viscosity property in exoskeleton assistive device: Comparison of variable and constant viscosity coefficients," IEEE Access, vol. 12, pp. 144208–144218, 2024, doi: 10.1109/ACCESS.2024.3471701.

[6] N. Falih, S. H. Supangkat, F. F. Lubis, and O. M. Prabowo, "Revolutionizing process automation: The synergy of low-code development platforms, robotic process automation, and integrated smart system platform," IEEE Access, vol. 12, pp. 118694–118706, 2024, doi: 10.1109/ACCESS.2024.3447882.

[7] Arora, A. Jain, D. Yadav, V. Hassija, V. Chamola, and B. Sikdar, "Next generation of multi-agent driven smart city applications and research paradigms," IEEE Open J. Commun. Soc., vol. 4, pp. 2104–2121, 2023, doi: 10.1109/OJCOMS.2023.3310528.

[8] R. Buoy, M. Iwamura, S. Srun, and K. Kise, "Toward a low-resource non-Latin-complete baseline: An exploration of Khmer optical character recognition," IEEE Access, vol. 11, pp. 128044–128060, 2023, doi: 10.1109/ACCESS.2023.3332361.

[9] D.-H. Cho, H.-S. Oh, S.-B. Kim, and S.-W. Lee, "EmoSphere++: Emotion-controllable zero-shot text-to-speech via emotion-adaptive spherical vector," IEEE Trans. Affect. Comput., 2025, doi: 10.1109/TAFFC.2025.3561267.

[10] W.-C. Huang, Y.-C. Wu, and T. Toda, "Multi-speaker text-to-speech training with speaker anonymized data," IEEE Signal Process. Lett., vol. 31, pp. 2995–2999, 2024, doi: 10.1109/LSP.2024.3482701.

[11] D. Sanjaya, R. Setiawan, and N. F. Hikmah, "RNA-BioLens: A novel Raspberry Pi-based digital microscope with image processing for acute lymphoblastic leukemia detection," IEEE Access, vol. 13, pp. 23618–23628, 2025, doi: 10.1109/ACCESS.2025.3532305.

[12] M. B. Musthafa, S. Huda, T. T. Nguyen, Y. Kodera, and Y. Nogami, "Optimized ensemble deep learning for real-time intrusion detection on resource-constrained Raspberry Pi devices," IEEE Access, vol. 13, pp. 113544–113556, 2025, doi: 10.1109/ACCESS.2025.3584373.

[13] Q. Na, H. Zhou, H. Yuan, M. Gui, and H. Teng, "Improving walking path generation through biped constraint in indoor navigation system for visually impaired individuals," IEEE Trans. Neural Syst. Rehabil. Eng., vol. 32, pp. 1221–1232, 2024, doi: 10.1109/TNSRE.2024.3375944.

[14] Z. Yu and M. Hu, "Real environment warning model for visually impaired people in trouble on the blind roads based on wavelet scattering network," IEEE Access, vol. 12, pp. 82156–82167, 2024, doi: 10.1109/ACCESS.2024.3412328.

[15] F. Drira, F. LeBourgeois, and H. Emptoz, "OCR accuracy improvement through a PDE-based approach," in Proc. 9th Int. Conf. Document Analysis and Recognition (ICDAR), 2007, pp. 1068–1072, doi: 10.1109/ICDAR.2007.4377079.

[16] World Health Organization, World Report on Vision. Geneva: WHO, 2021. [Online]. Available: https://www.who.int/publications/i/item/world-report-on-vision

[17] International Agency for the Prevention of Blindness (IAPB), Vision Atlas 2022. [Online]. Available: https://www.iapb.org/vision-atlas

[18] National Federation of the Blind (NFB), Blindness Statistics, 2020. [Online]. Available: https://nfb.org/resources/blindness-statistics

[19] World Blind Union (WBU), Annual Report, 2020. [Online]. Available: https://worldblindunion.org

[20] European Blind Union (EBU), Facts and Figures on Blind and Partially Sighted People in Europe, 2021. [Online]. Available: https://www.euroblind.org

[21] American Printing House for the Blind (APH), Annual Report 2019. [Online]. Available: https://www.aph.org