Abstract

Braille characters are designed for the visually challenged which consist of six embossed points arranged in a standard character format. Finger sensitivity is crucial in reading Braille characters by touch, and learning them can be challenging as they need to be memorized. The present work aims to use image pre-processing techniques such as cropping, grayscale conversion, thresholding, erosion, and dilation to process Braille character images. These processed images will then be subjected to image processing and optical character recognition (OCR) methods to recognize the Braille characters and convert them into alphanumeric text. The implementation of this innovative idea could have a significant positive impact on the creation of educational materials for individuals with visual impairment and other related applications.

Keywords:-

Braille, grayscale, thresholding, erosion, dilation, image processing, OCR (Optical Character Recognition).

1. Introduction

The Braille system was invented to aid visually impaired individuals in reading and writing. However, there are still challenges associated with recognizing and translating Braille characters to understandable text. Braille characters are typically composed of six dots arranged in a rectangular grid of two dots wide and three dots in vertical column. Therefore, the system will need to accurately identify and separate the individual dots or dot patterns representing the Braille characters. The proposed system makes use of Image Processing and Optical Character Recognition (OCR) techniques to accurately recognize Braille characters from images.

People with vision impairments frequently read and write using the Braille system. It appears based on collection of raised dots that could be read with a touch. Anyone who is visually impaired can read the Braille character. The Braille character can be specified in any language, including English, even though the Braille character is a type of code rather than a language. Within a predetermined area known as a Braille cell, the symbols of Braille (which correspond to alphabets) are formed. The points where dots are elevated, which are uniformly numbered, can be used to describe a specific combination. For instance, dots representing 1-3-4 show a cell with three dots at the top, bottom, and top of the right column, which corresponds to the English letter m in the binary vector [101100].

The proposed framework will enable better accessibility to Braille literature and increase independence and literacy among visually impaired individuals. The approach used in this work involves pre-processing the input image, segmenting the Braille characters, and applying OCR techniques to recognize the characters. The proposed system will provide a user-friendly Graphical User interface (GUI), enabling easy and efficient conversion of Braille text into conventional text. The results of this work will have a significant impact on the visually impaired community and promote inclusive education and equality.

2. **Literature Survey**

A device has been developed in [1] that utilizes Optical Character Recognition (OCR) and Speech Recognition algorithms to convert printed materials into Braille alphabet for visually impaired individuals. This device aims to provide faster access to written culture and make documents, articles, and other products more accessible for visually impaired individuals.

Author of [2] proposed a novel approach for Braille character detection based on edge features is proposed, taking into account the small size of Braille characters in natural scene images. A data set called NSBD was created, which consists of natural scene Braille images.

A sample of the OCR tools that are accessible and instruct students on how to use the Tesseract tool, which is transcribed in either Latin male or female has been developed in [3]. Using the device learning concept, this paper aims to provide OCR software that recognizes text characters.

Author of [4] suggests a user-friendly smartphone app made specifically for those with visual impairments. The optical character recognition (OCR) framework's image-to-text conversion feature is then used to transform the acquired image to text. The volume down button on a smartphone, which is easily located by touching, can be used by a person who is blind to take a picture or they can allow the picture be taken automatically.

The fundamental concept proposed by the author in [5] is the creation of a text-dictation system for blind persons. The camera module attached to the Raspberry Pi allows the blind to interpret text from images without assistance from a human.
The process of converting Devanagari Hindi Braille first to text, then to voice has been examined in [6].

The author in [7] presents a design proposal for a low-cost braille printer that uses OCR technology (Optical Character Recognition) through a mobile application with an algorithm that allows identifying text and images for braille translation or tactile embossing.

In order to recognize Cyrillic alphabet letters in the Braille representation system, an artificial neural network has been developed in [8]. The network will receive training and testing for reading Braille that contains scanned Cyrillic letters.
The Author of [9] has converted the braille characters into alpha-numeric text and then has developed a method for recognizing them.

The Author of [10] proposed an approach that focuses on Gujarati Braille script for identifying braille characters. It also emphasizes conversion techniques and focuses on a variety of material for translating Braille into multiple languages.
Creating an intuitive application for Android phones that conducts image to speech conversion has been done in [11].

Author of [12] addressed a novel profile-based method for segmenting printed text that divides the text in document images into lines, words, and characters, as well as an algorithm for correcting the skew angle formed during scanning of the text document.
A low-cost Braille pad has been created in [13] to allow persons who are blind to read text files from a computer using their tactile sense.

Author of [14] proposed an approach for the creation of BrailLector, a system that can speak Braille writing using dynamic thresholding, an adaptable Braille grid, recovery dots, and text-to-speech software.

An optical character recognizer (OCR) for a Braille character, the system used for writing by the visually impaired are described in [15]. The finished system is a polished Windows application that functions as a standard text OCR and is quite appealing to blind people (in fact, they supported the project).

3. Positioning

3.1. Problem statement:

Braille character recognition system using image processing and OCR techniques to address this problem and improve accessibility for visually impaired individuals the system will allow users to take an image of a Braille character or text using a camera or scanner, and the system will recognize the Braille characters present in the image and convert them into text. The system will also be able to output the text in a format that is easy for users to read and understand, such as through text-to-speech or displaying the text on a screen.

3.2. Product position statement:

Our Braille character recognition system is a cutting-edge technology that uses image processing and OCR techniques to recognize and convert Braille characters into text, providing visually impaired individuals with improved accessibility to written materials. This innovative system is accurate, efficient, and user-friendly, allowing users to easily capture images of Braille characters and convert them into text in a format that is easy to understand. Our system can have a significant impact on the lives of visually impaired individuals, empowering them to participate in educational, professional, and social activities with greater ease and independence.

4. Project overview

The objective of the Braille Character Recognition using Image Processing project is to develop a system which can accurately convert Braille characters into readable text using image processing techniques. This will involve developing algorithms for processing images of Braille documents, training and testing the system using a Braille character dataset, and integrating the system into an easy-to-use interface. The ultimate aim of the project is to provide a more efficient and accessible method of reading and writing for individuals who are blind or visually impaired.

4.1. **Objectives:**

The project has three main objectives: First, to design and create an OCR engine capable of accurately identifying Braille characters from images of Braille documents. Second, to incorporate this OCR engine into a Braille Character Recognition system that can convert the recognized Braille characters into readable text. Lastly, to develop a user-friendly interface that can be easily navigated by individuals who are blind or visually impaired, allowing them to use the Braille Character Recognition system with ease.

4.2. **Goals:**

The primary goal of this project is to create a Braille Character Recognition system that utilizes OCR (Optical Character Recognition) technology to accurately identify and convert Braille characters into readable text.

5. **Project Scope**

Blind individuals face challenges when it comes to reading regular text, which is why braille letters were created as a tactile system of raised dots. A braille letter consists of six dots arranged in various ways to form different letters.
However, learning to read braille requires a significant amount of time and the development of finger sensitivity to recognize and remember the different dot combinations. As a result, not many people are able to read braille. Assistive technology is necessary to convert braille into text and make it easier to read, reducing the time required to read it using fingers.

6. **Methodology**

The proposed system takes an input from a camera or a scanned image of a braille text, then converts it into English text using OCR technique. The extracted text is further converted to speech using TTS (Text-To-Speech) software as shown in Figure 1 and 2.

The overall method for Braille character's pattern recognition is segmented into two steps i.e. Pre-Processing and Pattern recognition.

6.1. **Pre-Processing Steps:**

6.1.1. **Scanning of the Braille code** - a digital image of the Braille characters is captured using a scanner or a camera, and image processing algorithms are used to analyze the image and determine the arrangement of dots.

6.1.2. **Conversion of RGB image to binary image** - Thresholding is a simple and commonly used technique for converting an RGB image to a binary image. If the intensity value of a pixel is higher than the threshold, it is set to white (1), otherwise, it is set to black (0).

6.1.3. **Removing of noise and unwanted dots** - Mathematical morphology is a set of image processing operations that can be used for various tasks, including noise removal.

Operations such as erosion and dilation can be used to remove small isolated noise pixels or fill in gaps in larger dots.

6.1.4. **Resizing of the binary image** - Morphological operations, such as erosion and dilation, can be used for resizing binary images. Erosion can be used to shrink or downscale a binary image, while dilation can be used to expand or upscale a binary image.

6.1.5. **Converting the image (bit 1 to 0 and bit 0 to 1)** - Many image processing libraries, such as OpenCV, provide built-in functions for image complementing.

6.1.6. **Removal unwanted edges of the image** - Morphological operations, such as erosion and dilation, can be used for image edge pruning. Erosion is a morphological operation that can be used to shrink or thin edges in an image by iteratively removing pixels from the edges of objects. Dilation, on the other hand, can be used to expand or thicken edges by adding pixels to the edges of object.

6.2. **Pattern Recognition Steps:**

6.2.1. Segmentation of the Braille characters Image into lines.

6.2.2. Segmentation of the Braille character lines into alphabet.

6.2.3. Dividing alphabets into grids of size 3 x 2.

6.2.4. Checking the threshold criterion, count the number of white pixels that satisfy the condition.

6.2.5. Generating pattern vector based on the result in terms of 0 and 1.

6.2.6. Link the pattern vectors with alphabets that correspond to it.

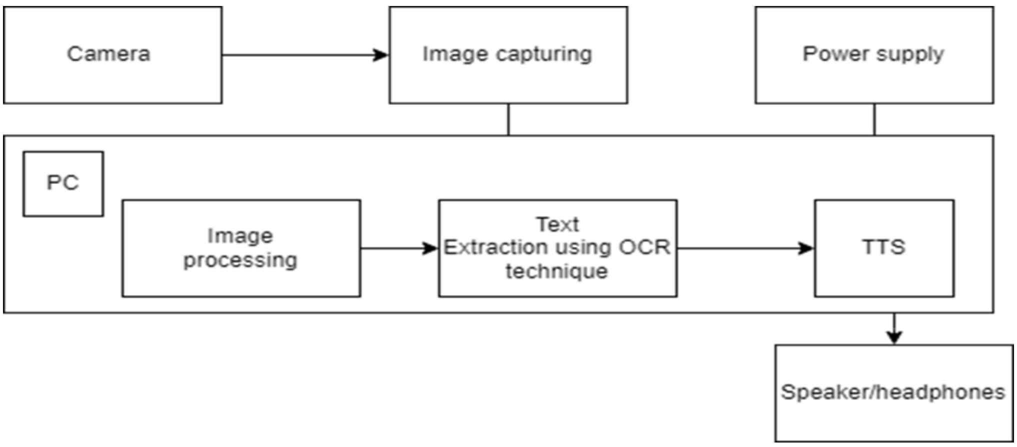

Figure 6.1: Translation of Braille Characters

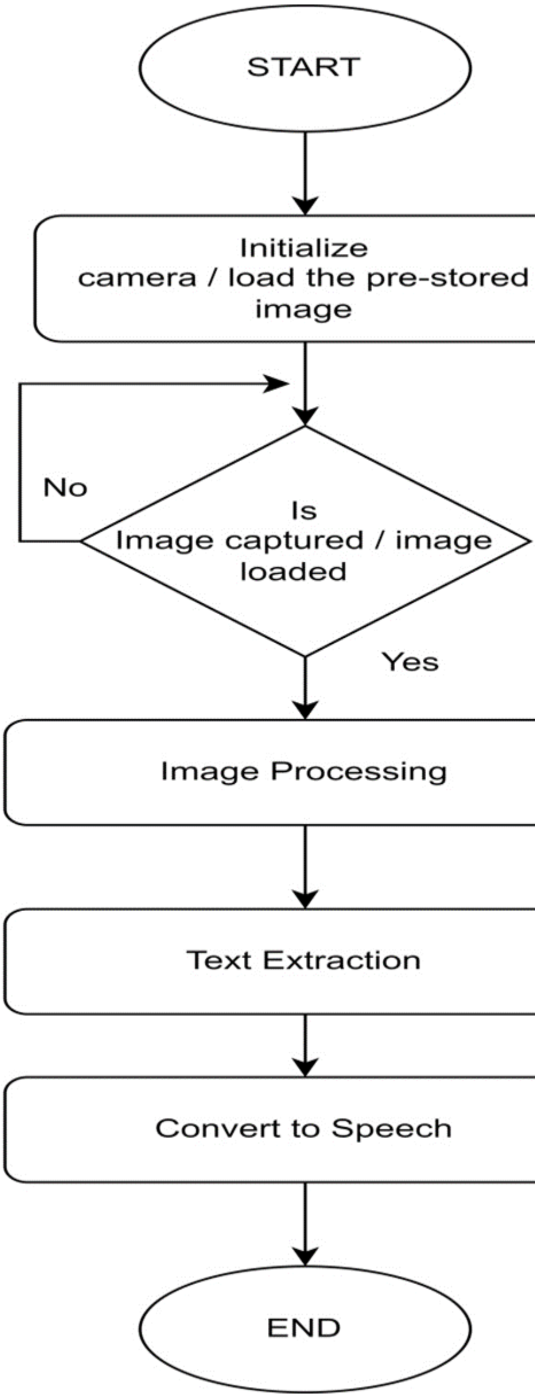

Figure 6.2 : Flow Chart of Complete Process of proposed project

Image Capturing: First step is to capture an image from the document or book which is placed under the camera to capture an image from the document or book. The camera used to capture an image is PC camera.

Image Pre-processing: Image pre-processing is to remove unwanted noise in the image by applying appropriate threshold. It is used for correcting skew angles, sharpening of image, thresholding and segmentation.

Text extraction: The developed system makes use of modules that are used to extract the recognized text. The extracted text is then converted into speech using a text-to-speech synthesizer. Finally, the synthesized speech output is obtained as the end result of the process.

7. Modules identified

7.1. **Image Acquisition Module:** Responsible for capturing Braille characters image using a camera or scanner.

7.2. **Pre-processing Module:** This module will perform pre-processing operations on the acquired image, such as noise reduction, image enhancement, and binarization to prepare the image for recognition.

7.3. **Segmentation Module:** This module will identify individual Braille cells within the pre-processed image and separate them for individual recognition.

7.4. **Feature Extraction Module:** This module will extract features from the segmented cells, such as the location and number of raised dots, to enable the recognition of the Braille characters.

7.5. **Classification Module:** This module will use machine learning or pattern recognition algorithms to classify the Braille cells into corresponding characters.

7.6. **Text-to-Speech Module:** This module will convert the recognized Braille characters into speech output for the user.

7.7. **User Interface Module:** This module will provide a user-friendly interface that helps users to interact with the system, including inputting images, viewing the output, and adjusting system settings.

8. Project Implementation

8.1. Architectural Design:

The Braille Character Recognition system consists of the following components:

User Interface: This component provides a user-friendly interface for capturing images, displaying output, and adjusting system parameters. It is implemented using the PyQt5 library for Python.

Image Processing: This component is responsible for preprocessing the captured images to enhance their quality and extract the Braille character cells. It is implemented using the OpenCV library for Python.

Feature Extraction: This component extracts the specific dot patterns from each Braille cell using image processing techniques. It is also implemented using the OpenCV library for Python.

Character Recognition: This component recognizes the Braille characters from the extracted features using a template matching technique. It is implemented using Python's built-in libraries.

OCR: This component uses the Tesseract OCR engine to recognize the recognized Braille characters. It is implemented using the pytesseract library for Python.

Text-to-Speech: This component converts the recognized characters to speech using the eSpeak TTS engine. It is implemented using the py-espeak-ng library for Python.

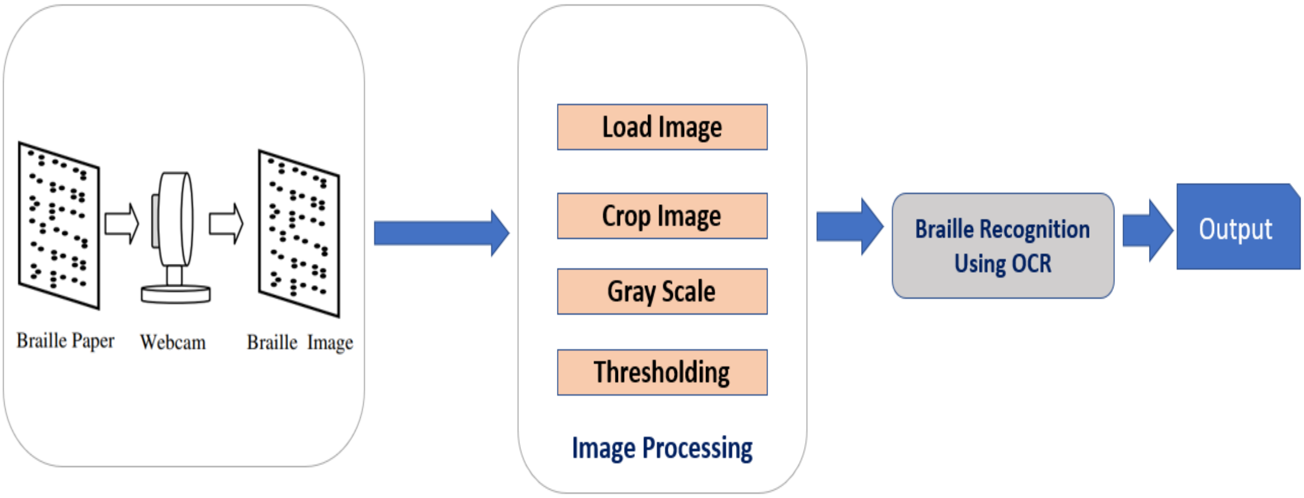

Figure 8.1 : Architecture Diagram of Braille Character Conversion

8.2. Class Diagram:

The major classes/modules in the Braille Character Recognition system include:

Main Window: This class defines the main window of the user interface.

Image Processor: This class provides methods for pre-processing the captured images.

Feature Extractor: This class extracts the dot patterns from the pre-processed image.

Character Recognizer: This class performs the Braille character recognition using template matching.

OCR Processor: This class uses the Tesseract OCR engine to recognize the recognized Braille characters.

Text To Speech Converter: This class converts the recognized characters to speech using the eSpeak TTS engine.

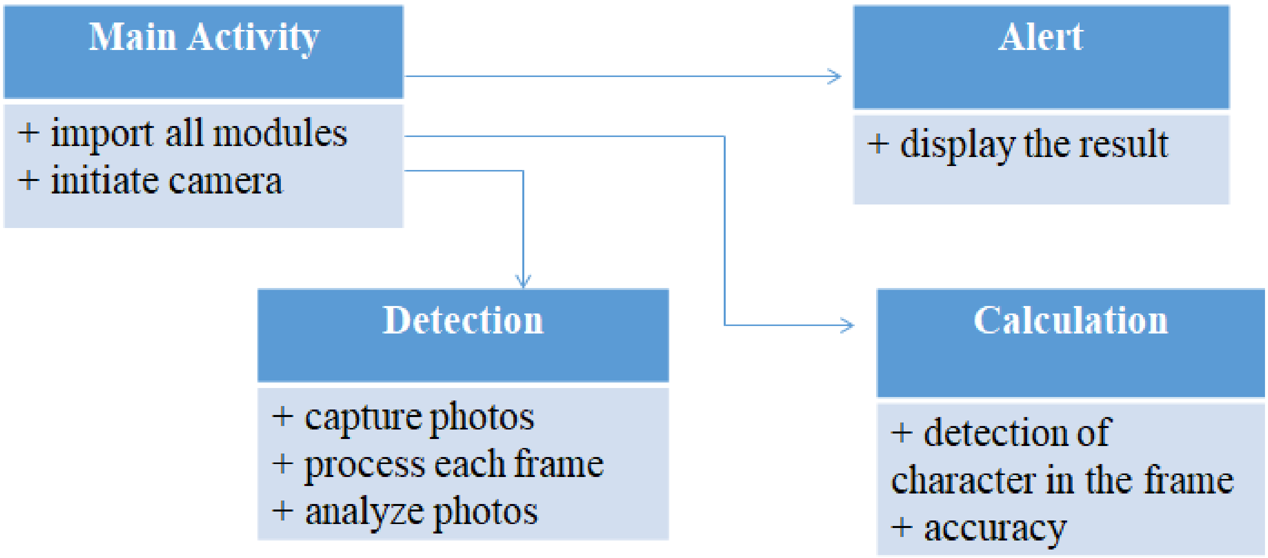

Figure 8.2 : Class Diagram of the proposed system

8.3. Entity Relationship Model:

As this is not a database-related project, there is no entity relationship model to be developed.

8.4. Sequence Diagram:

A sample sequence diagram for the major use case scenario of capturing and recognizing a Braille character is as follows:

- User opens the Braille Character Recognition system.
- User selects the camera option from the menu.
- User captures an image of the Braille character.
- The image is sent to the Image Processor component for pre-processing.
- The preprocessed image is sent to the Feature Extractor component for feature extraction.
- The extracted features are sent to the Character Recognizer component for character recognition.
- The recognized characters are sent to the OCR Processor component for OCR.
- The recognized text string is sent to the Text To Speech Converter component for speech conversion.
- The output speech is played through the computer speakers.

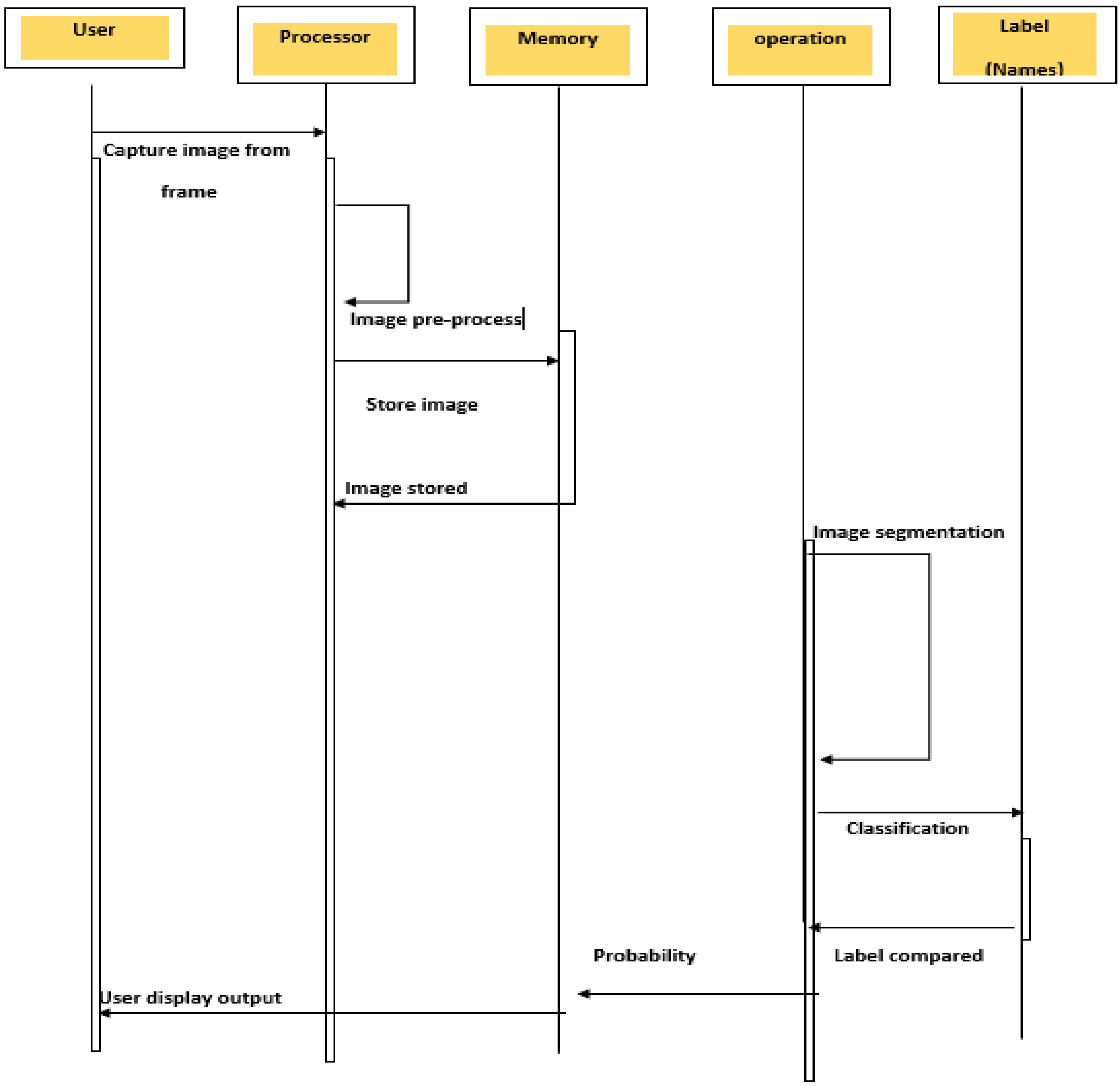

Figure 8.4 : Sequence Diagram of the proposed system

8.5. Description of Technology Used:

The Braille Character Recognition system is implemented using the following technologies:

Hardware Devices: A camera is used for capturing images

.

Software Products: Python 3.8 is used as the main programming language. The OpenCV library is used for image processing and feature extraction. The Tesseract OCR engine and pytesseract library are used for OCR. The eSpeak TTS engine and py-espeak-ng library are used for text-to-speech conversion. The PyQt5 library is used for developing the user interface.

Operating System: The system is developed and tested on Windows 10.

9. Findings / Results of Analysis

The proposed framework was able to successfully recognize and convert Braille characters in an image into text using image processing techniques and OCR. The system was tested on various Braille images and obtained accurate results. Graphical User Interface (GUI) based user-friendly system developed, to enable the user to upload an image and receive the corresponding text output as shown in Figure 6. Additionally, text-to-speech functionality is implemented, which allows the user to hear the recognized text as an audio output. Figure 7 represents the input image of Braille character. Upon loading the input image, the image is processed, resulting in the processed image, as illustrated in Figure 8. The processed image undergoes extraction of Braille characters, which are then represented as text for further processing as illustrated in Figure 9.

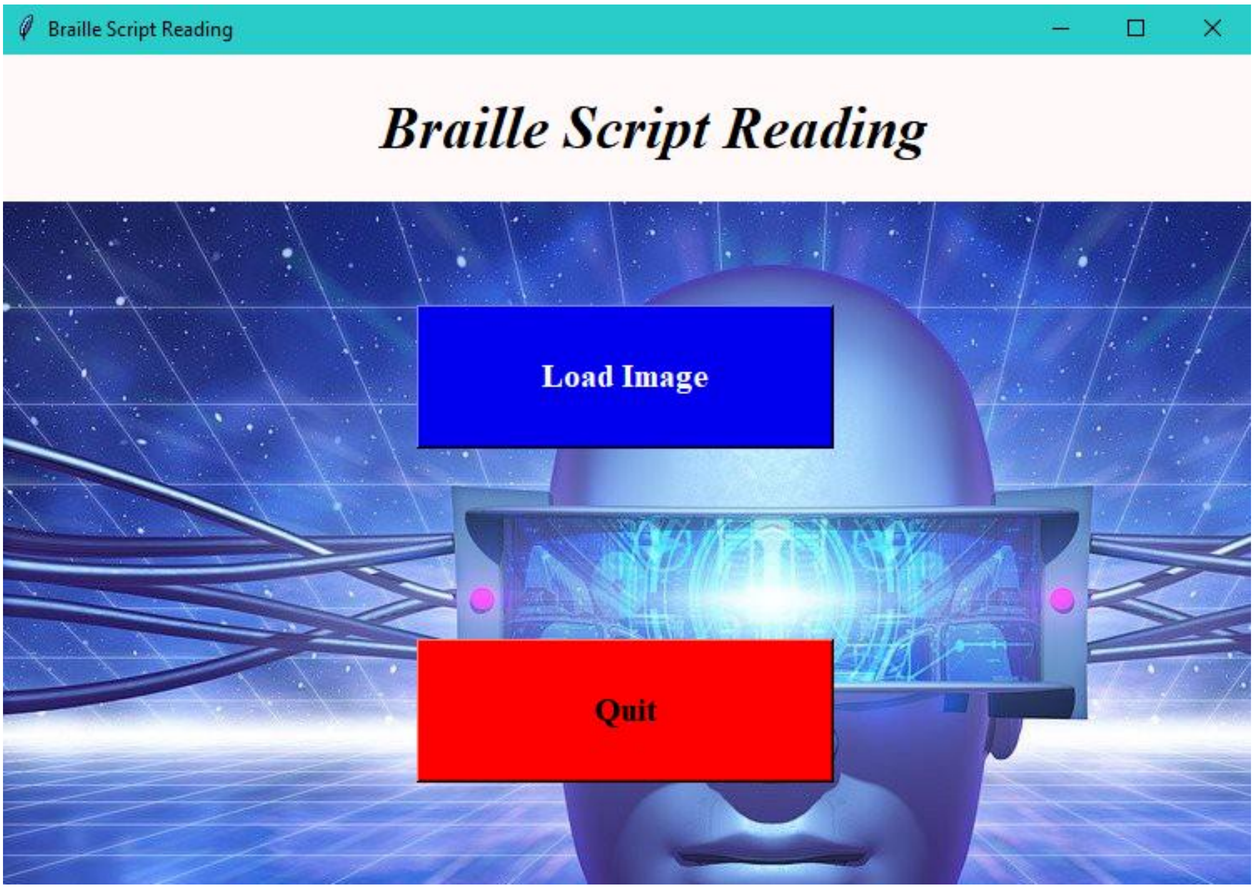

Figure 9.1 : Initial GUI after running the code

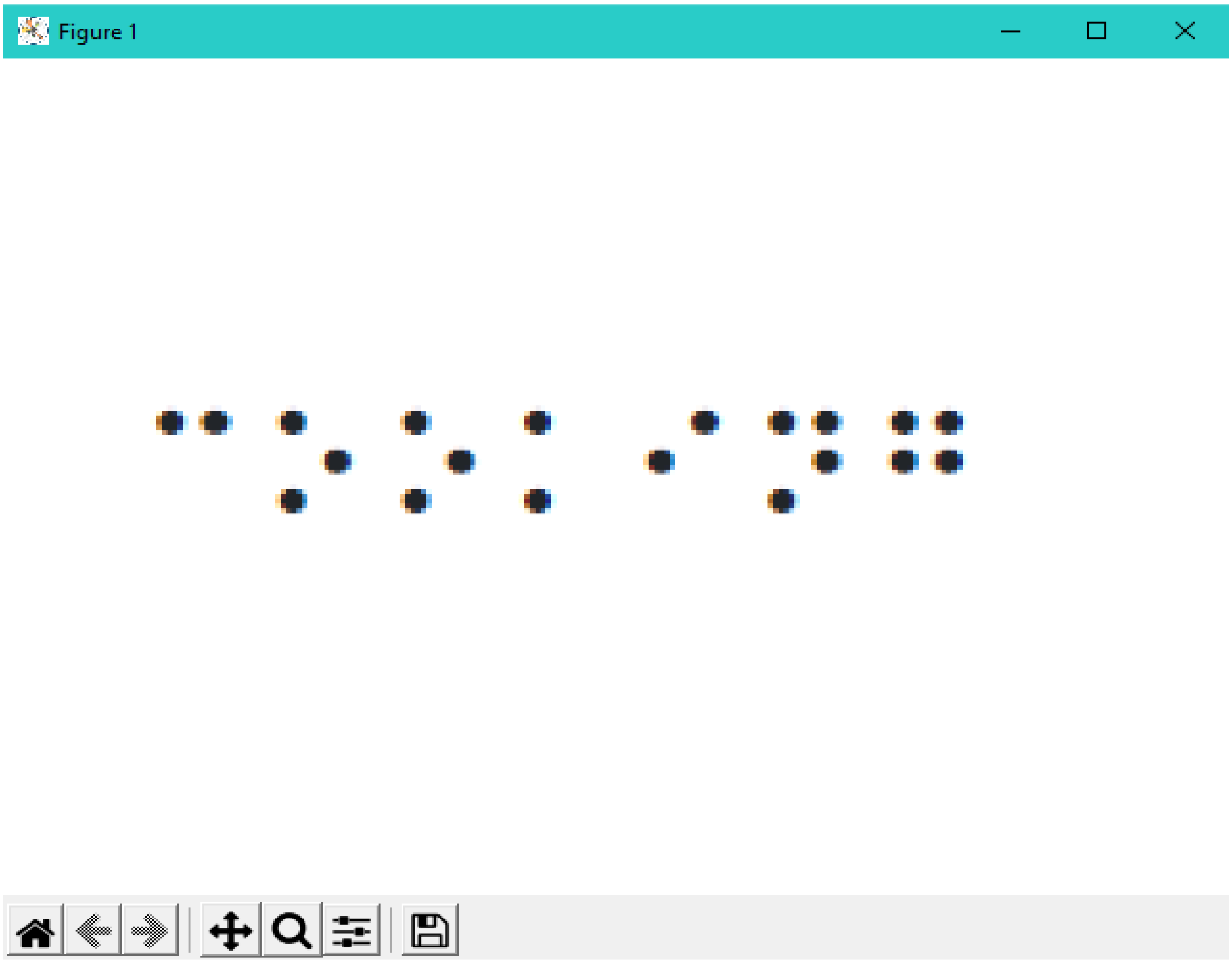

Figure 9.2 : Braille Character input image loaded - Cooking

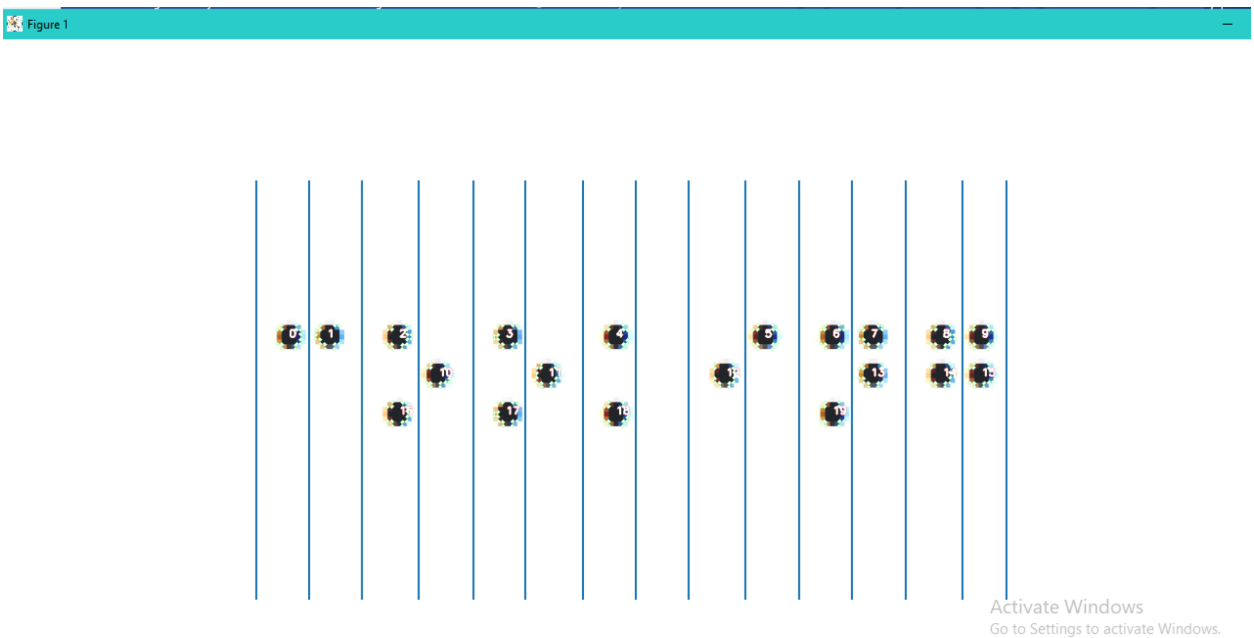

Figure 9.3 : Segmented Image of the Braille Character

```
*Python 3.7.8 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.8 (tags/v3.7.8:4b47a5b6ba, Jun 28 2020, 08:53:46) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: E:\RBITS__FILE\Braille\Braille-master\braille_GUI.py ========

[1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1]
[0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1]
[0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0]

[[1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1], [0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1
, 1, 1], [0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0]]
cooking

[1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1]
[0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1]
[0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0]

[[1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1], [0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1
, 1, 1], [0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0]]
cooking
```

Figure 9.4 : Conversion of Braille Characters into text

Overall, the findings and results show that the system is effective and efficient in recognizing Braille characters and converting them to text. Currently, the proposed system focuses on recognizing Braille characters in a specific language. As a future work, the system could expand to support multiple languages, as Braille is used in various languages around the world. This can involve training the OCR algorithms on diverse Braille character datasets from different languages and incorporating language-specific features and patterns.

10. Cost of the Project

Requires no cost.

11. Conclusion

In conclusion, our Braille character recognition work successfully achieved its objective of recognizing Braille characters in an image and converting them into text using image processing techniques and OCR. GUI and text-to-speech functionality has been added to improve user experience and accessibility. The developed system has the potential to make a significant impact on the visually impaired community by providing an efficient and accurate method of recognizing Braille characters. It can also assist in digitizing and preserving Braille literature, which would make it more widely available and accessible to those who rely on Braille for reading and writing. In future, this work can be extended to multiple languages.

12. Project Limitations and Future Enhancements

The accuracy of the OCR can be affected by factors such as image quality, lighting conditions, and noise, which can limit the system's performance.

Currently, proposed system focuses on recognizing Braille characters in a specific language. Future work can involve expanding the system to support multiple languages, as Braille is used in various languages around the world. This can involve training the OCR algorithms on diverse Braille character datasets from different languages and incorporating language-specific features and patterns.

Use template matching techniques to recognize well-defined Braille characters with consistent shapes. Apply feature extraction techniques to identify important features from the image of the Braille character, such as its shape, size, and position of raised dots, and use them for pattern recognition.

Utilize image segmentation, morphological operations, and noise reduction techniques to improve the image's quality and reduce noise impact and other factors on recognition accuracy.

Recognize and convert complete sentences or paragraphs, which would be particularly useful for digitizing and preserving Braille literature.

13. References

[1] Dalip, Filiz, et al. "Raspbraille: Conversion to Braille Alphabet with Optical Character Recognition and Voice Recognition Algorithm." Hittite Journal of Science and Engineering 9.4: 253-261.

[2] Lu, Liqiong, et al. "Anchor-Free Braille Character Detection Based on Edge Feature in Natural Scene Images." *Computational Intelligence and Neuroscience* 2022.

[3] Sharma, Sanchit, et al. "Optical Character Reader." *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2022, Volume 2*. Singapore: Springer Nature Singapore, 2022. 333-344.

[4] Ab Wahab, Mohd Nadhir, et al. "Text reader for visually impaired person." *Journal of Physics: Conference Series*. Vol. 1755. No. 1. IOP Publishing, 2021.

[5] Nag, B. Hemanth, et al. "DEVELOPMENT OF REAL TIME TEXT RECOGNITION SYSTEM FOR VISUALLAY IMPAIRED PEOPLE." (2020).

[6] Kaur, Parmesh, et al. "Conversion of Hindi Braille to speech using image and speech processing." 2020 IEEE 7th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON). IEEE, 2020.

[7] Encinas, Israel Durán, et al. "Low-cost Braille printer prototype design with OCR technology." *2019 International Conference on Inclusive Technologies and Education (CONTIE)*. IEEE, 2019.

[8] Smelyakov, Kirill, et al. "Braille character recognition based on neural networks." *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*. IEEE, 2018.

[9] Subur, Joko, Tri Arief Sardjono, and Ronny Mardiyanto. "Braille character recognition using find contour and artificial neural network." *JAVA Journal of Electrical and Electronics Engineering* 14.1 (2016).

[10] Patel, Nancy K., et al. "A Review on Gujarati Braille Character Recognition." *International Journal of Innovations & Advancement in Computer Science IJIACS ISSN* (2015): 2347-8616.

[11] Mithe, Ravina, Supriya Indalkar, and Nilam Divekar. "Optical character recognition." *International journal of recent technology and engineering (IJRTE)* 2.1 (2013): 72-75.

[12] Shinde, Archana A., and D. G. Chougule. "Text pre-processing and text segmentation for OCR." *International Journal of Computer Science Engineering and Technology* 2.1 (2012): 810-812.

[13] Supriya, S., and A. Senthilkumar. "Electronic braille pad." 2009 international conference on control, automation, communication and energy conservation. IEEE, 2009.

[14] Falcón, Néstor, et al. "Image processing techniques for braille writing recognition." Computer Aided Systems Theory–EUROCAST 2005: 10th International Conference on Computer Aided Systems Theory, Las Palmas de Gran Canaria, Spain, February 7–11, 2005, Revised Selected Papers 10. Springer Berlin Heidelberg, 2005.

[15] Hermida, Xulio Fernández, A. Corbacho Rodriguez, and F. Martin Rodriguez. "A Braille OCR for Blind People." *Proceedings of ICSPAT-96. Boston (USA)* (1996).

14. Copies of articles:

14.1 Conference papers published (Certificate with Published Paper):

ISSN: 2583-6129

ISJEM e-Journal

ISSN: 2583-6129

INTERNATIONAL SCIENTIFIC JOURNAL OF ENGINEERING & MANAGEMENT

An International Scholary || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

ACCEPTANCE CERTIFICATE

This is to certify that the submitted Paper titled

Braille Character Recognition Using Image Processing

has been accepted for publication in

INTERNATIONAL SCIENTIFIC JOURNAL OF ENGINERING AND MANAGEMENT

ISSN: 2583-6129

Volume 02 Issue 04 April, 2023

Editor-in-Chief

e-Mail: editor@isjem.com

www.isjem.com