Grenze International Journal of Engineering and Technology, June Issue

Braille Support System through ML

Michael Roshan J[1], Mohamed Abdul Fazil[2], and Dr. L Javid Ali[3]
[1-3]St. Joseph's Institute of Technology/Information Technology, Chennai, India
Email: micharlroshan675@gmail.com, {mohamedabdulfazil, javidinaug}@gmail.com

***Abstract*— This system proposes an effective way of communication for the unsighted by converting the braille script into a speech format through machine learning. This proposed system aims at achieving this in most languages available by understanding the various dialects of the braille system. The system provides the functionality of converting braille to speech. The braille text is first converted to recognizable text using an OCR device which enables the program to understand the input. The system includes a Braille input device that translates text input into Braille output and a Braille display that converts Braille input into text output. The machine learning model is trained on a dataset of Braille characters and their corresponding text representations to accurately recognize and interpret Braille characters. The proposed system is designed to improve Braille reading and writing efficiency and accuracy for visually impaired individuals, ultimately promoting greater independence and inclusivity. The results of the proposed system demonstrate its potential to significantly enhance the accessibility and usability of Braille for individuals with visual impairments.**

***Keywords*—Unsighted; Dialects; OCR device**

I. INTRODUCTION

Braille can be a sequence of raised dots that may be studied with the arms of blind or sullen people. Instructors, caregivers, and others who are not blind usually read Braille with their eyes. Braille is not a dialect. Or maybe code that can write and view many dialects such as English, Spanish, Arabic, Chinese, and a few other dialects. Braille is spoken as a local dialect by thousands of people around the world and provides insight into education for all. Visual impairment can significantly reduce a person's ability to analyze and write, limiting learning and working opportunities.

Braille can become a material construct for writing that enables people with external disabilities to learn and write by touch. Even so, reading and writing Braille can be difficult and time-consuming, especially for those familiar with the structure. Machine learning procedures can mechanize braille interpretation and provide systems that provide real-time feedback to users. Braille is specially designed for people with disabilities. Braille is made up of 6 dots. These characteristic dot patterns enable users with disabilities to learn and type different characters.

Braille has already been studied and composed with slates and styluses. Today's world is equipped with the latest technology to make people's lifestyles more comfortable. Touchscreen gadgets are used more often. Between 2009 and 2014, the popularity of portable screen readers increased exponentially from 12% to 82%. The use of touchscreen devices has changed the way people look, making them unable to live better, better, better, stronger, better, better lives. Touch screens have been an important development in recent years. This allows customers to complete a variety of tasks. These include e-learning, recovery resolutions, family responsibilities, games, online shopping, and more. A touch screen is a driving tool used in daily life because it is portable and easy to bargain.

Grenze ID: 01.GIJET.9.2.372

Blind people are an indispensable part of every society. Braille images are formed within spatial units known as braille cells. A complete Braille cell consists of six convex strokes arranged in two parallel lines of three strokes each. The speck positions are distinguished by numbers from one through six. Sixty-four combinations are conceivable utilizing one or more of these six specks. A single cell can be utilized to speak to a letter set letter, number, accentuation stamp, or indeed an entire word.
Braille to English character transformation has not been investigated significantly; to the leading of our information, no current application for the outwardly impeded recognizes a position-free touchscreen-based Braille information passage in real-time and provides both writings as well as voice input. Hence, there's a solid requirement for an application that's both available to and usable for outwardly impeded clients. The application proposed by the creators of Reference is special since outwardly impeded clients can tap any place on the screen and are not required to discover the exact area of the dabs. This work is an extension of previous work by Sana et al. The English Braille dataset used in this review was collected using an application proposed by Sana et al. The input power component has changed in this query. Creators have already been saving braille images and using them to wait for text. The account for each point entered here is stored in a content record. Authors physically validated numeric data sets obtained from touch screen applications. Discord was evacuated some time ago. Using this state-of-the-art dataset, braille-to-English character recognition was performed using machine learning methods such as DT, SVM, and KNN combined with RICA- and PCA-based embedding extraction strategies. In addition, random Timberland and Consecutive strategies were applied to comparative studies. Figure 1 shows a schematic of the study conducted.

II. Literature Review

Traditional Braille is a 6-dot code that can represent a maximum of 64 unique symbols with each braille cell. Since 8-dot braille can represent maximum um 256 unique symbols, it has huge potential to provide the ultimate solution to all woes faced by braille users while writing texts as well as mathematical and technical text. This paper presents a comprehensive unified braille Unicode system providing a detailed mapping of 8-dot braille Unicode pattern to represent the transcribing codes as well as the math, science, and computer symbols/characters.
A Braille-to-Chinese translation system composed of optical Braille recognition, Braille-to-pinyin conversion, and pinyin-to-character module. Experiments prove that our system can effectively and accurately translate Braille images into Chinese. Our system has broad application prospects, which can help sighted people read Braille and help visually impaired people in their studies and life.
The system introduces a system that will help visually impaired people to start their very basic education with a technology that is capable of recognizing Bangla characters. This voice recognition is done using the Visual Geometry Group model of Convolutional Neural Network (CNN) and Arduino Uno. A 16x4 LCD is used to generate the braille pattern for the recognized character.
The device takes voice as input from Android apps and can convert the voice signal into text format of braille and prints the braille letter on paper. EBCS is trained with 1 epoch and the accuracy is achieved at 97.6%. By implementing the considered Braille system, visually impaired people can entirely engage themselves in society.
The designed system takes the input through the Braille keyboard and produces the Braille output in Braille display; the corresponding English characters are also displayed on the LCD and also on the laptop if it is connected. It also has the capability of reading documents as text. By implementing the designed Braille system in schools and homes, Braille's literacy rate can be increased and visually impaired people can be employed and can fully participate the society.
We are introducing software for hearing and visually impaired people to transcribe video to printed Braille scripts. The implementation of such a system for visually handicapped people with a productive method of Real-time Braille transcription, text-to-speech, text-to-speech video-to-Braille conversion, work on commands, and flexibility will be of great use
Using the braille text written by the control system, its dot diameter, dot height, dot pitch, and square distance are in line with national standards. And the variance of each parameter obtained by calculation is smaller than that of the national braille standard, which verifies the feasibility of the designed braille machine control system.
One would no longer require knowing braille himself to be able to communicate with someone visually impaired. He could just simply type in normal text and those texts would get translated into Braille characters in real-time
Braille dots composing characters are identified on single-sided documents of average quality with maximum accuracy. Work presents an executable system including proposed methods to transcript the recognized braille

characters. Introduce new extraction methods of Braille characters from a Braille document and decode them into Sinhala characters. Then the decoded Sinhala characters are normalized to Sinhala text which is in a human-understandable form.

The Automation of the Braille reading and digitization process will help in preserving ng, distributing Braille over the network, and reproducing hard copy-only demands. The need for digitizing Braille has led many researchers in improvising and implementing optical Braille recognition systems.

Here we are implementing an enhanced Braille system that helps blind people to read text or content. We scanned information that came and the image-by-image processing techniques and the same will be converted into text using OCR. The detected text will be given to raspberry pi which recognizes every character and converts it into Braille code. With the help of solenoid, we are displaying that Braille code on Braille.

The algorithm applies two tools for pattern recognition and translation of Braille code, such as Mathematical Morphology and Alpha-Beta Bidirectional Associative Memories, respectively.

This describes the implementation of a prototype device for individuals dealing with both visual and hearing impairments to communicate. This is carried out with the help of a speech input system and a braille decoder. The input is processed through a machine learning model which recognizes the spoken words, converts spoken words to text capitalizes each letter in a word, and sends the respective ASCII code to a custom-designed decoder through the Raspberry Pi 4B module. The decoder is designed using logic gates which are further implemented using n-type and p-type MOSFETs. The proposed device provides an output in the form of braille codes corresponding to each letter which have been implemented using LEDs.

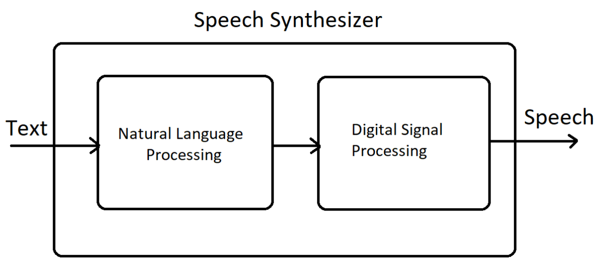
III. Proposed Work Description

The proposed system enhances as they build on to the general Braille script and can be converted to Speech format. This system is done to ease the process of converting a Braille to Text and then Reading it out to directly convert it from Braille to Speech and will be able to support multiple languages which support Braille Unicode.

First, the system would have to understand the braille Unicode which is the source language of the braille which has to be converted and this is identified using the input which is fed into the system. The input can be given digitally into the system or a braille paper can be scanned using an OCR device and can be converted into the input and fed into the system.

Once the system identifies the source language, it can use existing Python libraries to convert the Braille text into a machine-readable format. This format is versatile and can be easily converted to other formats such as text or audio. To convert Braille to speech, the system will use a text-to-speech (TTS) engine that can take the machine-readable Braille text as input and produce audio output.

The Proposed system involves a Braille-to-language interpreter module that performs a series of steps to convert Braille characters into machine-readable text or speech output. It starts by processing the input Braille characters to identify individual Braille cells and their dot configurations. Then, it recognizes the Braille patterns by matching them against a database of known patterns for the specific language. The module translates the recognized patterns into textual or speech representation using a character mapping algorithm. Finally, post-processing is applied to correct errors and enhance the accuracy of the output. By following these steps, the module enables visually impaired individuals to access and understand Braille content effectively.



The process of converting Braille characters into machine-readable text or speech output involves several key steps. First, Braille input processing is carried out, which entails identifying individual Braille cells and analyzing the dot configurations within them. This step allows for the extraction of the Braille patterns encoded in the input. Next, Braille pattern recognition comes into play, where the recognized patterns are matched against comprehensive databases containing information about the specific dot positions and combinations used for each

Braille character. This enables the system to accurately identify the corresponding Braille characters based on the recognized patterns.

Once the Braille characters are identified, a character translation step is performed. This involves mapping the Braille characters to their corresponding textual or speech representations using character mapping algorithms or lookup tables specific to the language being interpreted. Finally, post-processing techniques are applied to refine the output. These techniques may involve error correction, spell-checking, grammar correction, and formatting to enhance the accuracy and coherence of the resulting machine-readable text or speech output. Together, these steps form a Braille-to-language interpreter module that facilitates the accessibility and understanding of Braille content for visually impaired individuals.

It's important to note that the accuracy of the OCR device and the TTS engine is critical to the success of the system. If the OCR device produces inaccurate or incomplete results, the system will not be able to accurately convert the Braille text to speech. Similarly, if the TTS engine produces poor-quality audio output, it may be difficult for the user to understand the converted text.

This system will use the available Python libraries which can understand the raw input from the OCR scanner and will make it into a system-understandable format. This new format is versatile and can simply be used to convert to text /audio formats.

The system mainly uses the TTS synthesizer as a primary algorithm to convert the braille text to speech format. Text-to-speech (TTS) is a technology that converts written text into spoken words. The algorithm for TTS generally involves the following steps:

1. Text preprocessing: The first step is to process the input text to remove any unwanted characters or symbols and convert it into a format that can be used by the TTS engine. This may involve tasks such as spell-checking, grammar-checking, and punctuation normalization.
2. Linguistic analysis: Once the text is preprocessed, it is analyzed by the TTS engine to identify the linguistic features such as the phonemes, stress patterns, and intonation of the words. This information is used to generate a phonetic representation of the text.
3. Acoustic modeling: The phonetic representation of the text is used by the TTS engine to generate a set of speech parameters such as pitch, duration, and amplitude. This involves selecting appropriate speech units (e.g., phonemes, syllables, or words) and assigning appropriate acoustic characteristics to them.
4. Synthesis: Finally, the speech parameters are used to synthesize the speech waveform that corresponds to the input text. The synthesized waveform can be outputted as digital audio or played through a speaker.

TTS algorithms can use different techniques for each of these steps, and the quality of the output can depend on the complexity of the algorithm and the quality of the input data. There are also many open-source and commercial TTS engines available that can be used to implement TTS functionality in software applications.

IV. System Architecture

The System Architecture of the paper is expressed in Fig 5, which shows the overall working of the system. It shows how the system interprets the user's input and gives the output.

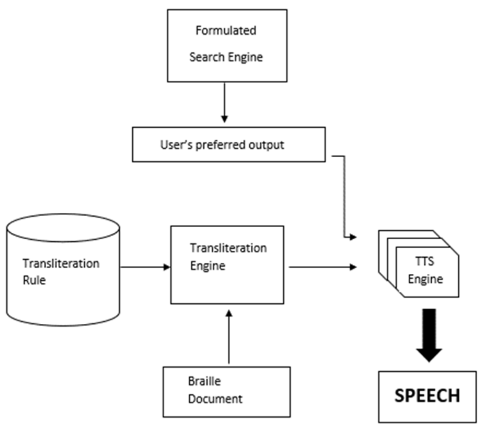


Fig 5: System Architectural Diagram

The architecture diagram maps out the braille system's physical implementation and describes the system's components. The proposed work can be described as the following modules

- OCR Recognition of Braille Text Module
- Source Language Identification Module
- Braille to Language Interpreter Module
- Language to Speech Converter

A) OCR (Optical Character Recognition) technology can be used to recognize Braille text and convert it into machine-readable text that can be further processed by software applications. The OCR recognition of the Braille text module generally involves the following steps:

- *Image preprocessing*: The first step is to preprocess the input image to remove any noise or distortion that may affect the accuracy of the OCR process. This may involve tasks such as binarization, noise reduction, and contrast enhancement.
- *Segmentation*: Once the image is preprocessed, the Braille characters in the image are segmented and separated from each other. This is important to ensure that the OCR algorithm can recognize each character individually.
- *Character recognition*: The segmented Braille characters are then passed through an OCR algorithm that uses machine learning techniques to recognize the characters and convert them into machine-readable text. The OCR algorithm may use different techniques such as template matching, neural networks, or statistical models to recognize the characters.
- *Post-processing*: The machine-readable text outputted by the OCR algorithm may contain errors or inaccuracies, so it needs to be post-processed to improve its accuracy. This may involve tasks such as spell-checking, error correction, and formatting.

B) Source language identification (SLI) module is a component of natural language processing systems that are used to identify the language of the input text. The SLI module generally involves the following steps:

- *Preprocessing*: The input text is preprocessed to remove any non-textual elements such as images, tables, or symbols that may affect the accuracy of the SLI process.
- *Feature extraction*: The preprocessed text is then passed through a feature extraction algorithm that identifies the distinctive characteristics of the text that are unique to each language.
- These characteristics may include the frequency of certain letters or word structures, the presence of certain grammatical structures or punctuation, and the overall vocabulary of the text.
- *Language identification*: The extracted features are then compared with the features of known languages in a language model database. The language model database contains statistical information about the distinctive features of each language, which can be used to match the features of the input text with the features of the known languages. The most likely language of the input text is then determined based on the best match with the features of the known languages.
- *Confidence scoring*: The SLI module may also output a confidence score that indicates the degree of certainty with which the input text was identified as a particular language. This score can be used to assess the accuracy of the SLI process and to determine whether further processing or verification is needed.

C) A Braille-to-language interpreter module is a software component that converts Braille characters into machine-readable text or speech output. The Braille to language interpreter module generally involves the following steps:

- *Braille input processing*: The input Braille characters are processed to identify individual Braille cells and the corresponding Braille patterns.
- *Braille pattern recognition*: The Braille patterns are then matched against a pre-defined database of Braille patterns for the specific language being interpreted. The database contains information about the patterns used in the language, such as the dot positions and combinations used for each character.
- *Character translation*: The recognized Braille patterns are then translated into machine-readable text or speech output using a character mapping algorithm. The character mapping algorithm maps the Braille patterns to the corresponding textual or speech output for the language being interpreted.
- *Post-processing*: The outputted text or speech may contain errors or inaccuracies due to the complexity of the Braille script or the variations in how it is written. Therefore, post-processing may be needed to correct errors, format the output, or improve the accuracy of the output

D) Alana language-to-speechwriter is a software component that converts machine-readable text into speech output in a specific language. The language-to-speech converter generally involves the following steps:

- *Text input processing*: The input text is preprocessed to remove any non-textual elements such as images, tables, or symbols that may affect the accuracy of the speech output.
- *Text normalization*: The preprocessed text is then normalized to improve the accuracy of the speech output. This may involve removing any punctuation or special characters, expanding abbreviations, or converting numbers to their spoken equivalents.
- *Language modeling*: The normalized text is then passed through a language model that contains statistical information about the pronunciation and intonation patterns of the language being converted. The language model may be based on phonemes, diphones, or other speech units that are specific to the language
- *Speech synthesis*: The language model generates a sequence of phonemes or speech units that are combined to produce the speech output. The speech synthesis process may involve selecting appropriate intonation patterns, applying prosody rules, and adding pauses or other vocalizations to improve the naturalness of the speech output.
- *Post-processing*: The speech output may contain errors or inaccuracies due to the complexity of the language or the limitations of the speech synthesis technology. Therefore, post-processing may be needed to correct errors, improve the naturalness of the speech output, or customize the output to the user's preferences.

RESULTS

The braille system has been effective in identifying the braille characters and the execution time is comparatively less when compared to other braille converting algorithms The Braille system's effectiveness in identifying Braille characters and its relatively fast execution time can be attributed to several factors. One factor is the use of OCR technology, which allows the system to accurately scan and interpret the physical Braille characters on the paper.

The OCR technology uses image processing algorithms to convert the Braille characters into a digital format that can be analyzed by the system. Another factor is using machine learning algorithms specifically designed to recognize and interpret Braille patterns.

These algorithms use a combination of image recognition techniques and statistical analysis to accurately identify the corresponding Braille characters.

Finally, the use of Python libraries and other programming tools allows the system to efficiently process and translate the Braille characters into machine-readable text and then into speech output. Overall, the Braille system's effectiveness and efficiency in identifying and processing Braille characters can be attributed to the use of advanced technologies and programming techniques, which allow for accurate and efficient processing of Braille content.

CONCLUSION

According to the results, the proposed Braille system is an innovative solution that aims to enhance the accessibility of written information for visually impaired individuals. The system also achieves minimizing the execution time by close to a second and with 97% percent accuracy among the trained dataset.

REFERENCES

[1] Archana A, Shinde D. Text pre-processing and text segmentation for OCR. International Journal of Computer Science Engineering and Technology. 2012:810-12

[2] Archana A, Shinde D. Text pre-processing and text segmentation for OCR. International Journal of Computer Science Engineering and Technology. 2012:810-12

[3] G. Devi Priya, N. Indumathi, N. Kalaimagal, A. Suriya, JT. Vasuki, Hardware-Based Braille Pad on Mobile Phone, vol. 3, 2015.

[4] Paul H. Schimpf, "A Detailed Explanation of Solenoid Force", Eastern Washington University Cheney WA USA, 2013.K. Elissa, “Title of paper if known,” unpublished.

[5] P. Smithmaitrie, J. Kanjantoe, P. Tandayya, "Touching force response of the piezoelectric braille cell", Disability and Rehabilitation: Assistive Technology, vol. 3, no. 6, pp. 360-365, 2008

[6] T. Volkel, G. Weber, U. Baumann, "Tactile graphics revised:” the novel braille dis 9000pin matrix device with multitouch input”, International Conference on Computers for Handicapped Persons, pp. 835-842, 2008.

[7] A.Andoni and K. Onak, “Approximating edit distance in near-linear time,” Proc. Forty-First Annual ACM Symposium on Theory of Computing, Bethesda, MD, pp. 199-204, June 2009. DOI: 10.1145/1536414.1536444.

[8] E. Rex, A. Koenig, R. Baker, "Foundations of braille literacy", American Foundation for the Blind, 1994.

[9] A. Ray, S. Ghosh, B. Neogi, "An Overview on Tactile Display Haptic Investigation towards Beneficial for Blind Person", International Journal of Scientific & Engineering Research, vol. 6, no. 5, pp. 88-95, 2015.
[10] Bigham, J. P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R. C., ... & White, S. (2010). VizWiz: nearly real-time answers to visual questions. In Proceedings of the 23rd annual ACM symposium on User interface software and technology (pp. 333-342).
[11] Kim, D. H., Kim, H., Lee, D., Lee, J., & Lee, S. (2017). Braille learning using a smartphone and a portable wireless refreshable braille display. Journal of Visual Impairment & Blindness, 111(1), 69-78.
[12] Oh, J. S., Lee, J. H., Park, S., & Bae, C. (2017). Development of a Korean Braille input system based on a neural network. Neural Computing and Applications, 28(1), 157-163.
[13] Purohit, P. (2018). Language-independent text-to-Braille conversion for visually impaired people. International Journal of Scientific and Research Publications, 8(10), 92-95.
[14] Ramakrishnan, I., & Liu, R. R. (2018). Text-to-Braille conversion using a deep neural network. In Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference (pp. 712-717).
[15] Ramakrishnan, I., & Liu, R. R. (2018). Text-to-Braille conversion using a deep neural network. In Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference (pp. 712-717).
[16] Zhang, Y., & Li, Y. (2019). A design of English-Chinese Braille conversion system. Journal of Physics: Conference Series, 1153(1), 012083.
[17] S. Agarwal, S. Godbole, D. Punjani, and S. Roy, "How Much Noise Is Too Much: A Study in Automatic Text Classification," Proc Seventh IEEE International Conference on Data Mining (ICDM 2007), 2007, pp. 3-12.