Journal for Re Attach Therapy and Developmental Diversities
eISSN: 2589-7799
2020 July; 3 (2): 113-120

A Unified Tesseract-Based Text-To-Braille Conversion System For Visually Impaired People

Ranjan K. Pradhan[1*], Siwani Agrawal[2]

[1*,2]Department of Electrical Engineering, College of Engineering and Technology, Biju Patnaik University of Technology, Bhubaneswar, Odisha, India
[1]Email: rkpradhan@cet.edu.in

***Corresponding Author:** Dr. Ranjan K. Pradhan
*Department of Electrical Engineering, College of Engineering and Technology, Biju Patnaik University of Technology, Bhubaneswar, Odisha, India
*Email: rkpradhan@cet.edu.in

ABSTRACT:

Background: Braille has been the most common means of reading and learning among visually challenged people for many years. Despite the importance of developing Braille-assistive technologies for blind people, the there exists a clear gap in the current Optical Character Recognition (OCR)-based conversion of Text-to-Braille documents and accuracy of translator systems. While the Long Short-Term Memory (LSTM)-based Tesseract OCR engine has been a popular tool in language processing, its ability in capturing mixed alphabet and mathematical symbols from a scanned document has not been explored earlier.
Methods: This work presents a novel Tesseract LSTM-based Text-to-Brielle conversion framework that converts English alphabets, mathematical symbols, punctuations, sentences, paragraphs in a scanned image to Grade-1 and Grade-2 Braille codes for visually challenged people. Using SimpleCV image pre-processing algorithms the image filtering, detection of edges and segmentation of mixed characters were accomplished, and the performance of the Tesseract LSTM model was examined by testing its accuracy in converting mixed fonts or symbol of various sizes, and converting whole document image to Braille codes.
Results: The results revealed that the efficacy of Tesseract OCR engine for Brielle conversion was better for large size font and symbol, showing 86% accuracy for fonts greater than 16 point size with an average accuracy of 88%. It was observed that the economic conversion of Text-to-Braille using open source LSTM network provide a powerful tool for fast and language-free translation of texts to both Grade 1 and 2 Braille.
Conclusion: The Tesseract OCR engine provide an efficient, cost-effective way of converting mixed text or document images to Braille codes, and can be extended easily to build real-time multilingual text-to-Braille translators for visually impaired.

Keywords*: Tesseract; Text-to-Braille Conversion; Visually Impaired; Image Processing; Optical Character Recognition.*

INTRODUCTION

Braille language has been the only medium of communication for blind people, and sign language is the major way through which deaf and dumb people interact with each other[1], [2]. A typical Braille system comprises of engraved dots, which are numbered from 1 to 6, arranged in 2x3 matrix. Different combination of dots indicate various characters of a language. A well designed Braille system must enable one to easily read, write, and study regardless of individual's age and health condition[3]. Therefore, developing a hybrid software-driven tool that effectively convert Text-to-Braille alphabet is crucial for translating written text in any language into raised dots, which can be easily felt with the fingertips of visually impaired people. In recent years, sophisticated machine learning algorithms were proposed, showing promising image segmentation and translation accuracies in representing Braille for various languages[4], [5]. However, the majority of these models are limited by conversion speed, hardware compatibility and validation with limited available data which often affects the system efficiency and cost of conversion from text to Braille .

Previously, a number of attempts have been made to design efficient OCR-based single to multi-lingual translators for conversion of numbers, consonants and vowels of printed text images into texts and Braille alphabets. For example, in a recent work, the Tamil text image data was used to verify the systems performance, showing good conversion accuracy[2]. Several algorithms have been proposed for the conversion of Kannada text image to Braille, which includes creation of six knowledge bases to minimize the time complexity[6]. Furthermore, attempts to analyze the conversion of scanned

Braille document to Indian Kannada text, where this language consists of compound (ottaksharas) and basic characters which results in different sized characters were made recently, making the translation more accurate[7].

Similarly, various methods have been proposed for efficient conversion from Southern Indian Braille to respective language[1]. Most works described that certain amount of work is done in conversion of various languages to Braille. Many studies report that more than one fourth of people aged more than 50 are visually impaired[9]. In other hand, Ravi Kumar and Srinath studied a system for hand punched Braille character recognition which gives 98% accuracy[10]. Perera and Wanniarachchi [11]proposed a system which identifies Sinhala Braille characters and translate it to Braille. However, in all these systems the target output was well known. However, the complex script structure, lack of standardisation, inaccurate translation rules, successful training and awareness among users remain the key factors affecting the conversion performance of OCR engine.

Therefore, to overcome these issues, it is important to develop a cost-effective, accurate and language free transliteration framework based on simple image segmentation and machine learning method, validated by variety of complex data of different languages. LSTM-based neural network models have been extensively used in various fields for time-series forecasting detection, and classification and prediction applications[12], [13]. However, the amount of data available for these model training plays a crucial role in applicability of these models. In recent years, the idea of using a small dataset based on a trained model over a similar one, is thought to offer several advantages in complex machine learning applications. Nevertheless, the application of LSTM networks to convert real time scanned texts to Brielle codes are limited and yet to be explored for development of less resourced language technology.

In the present work, we have developed a unified text-to-Braille translator that processes both texts and symbols or equation using Tesseract open source OCR engine[14], which particularly uses LSTM Network for converting complex data from scanned documents in English test language to corresponding Grade 1 and Grade 2 braille. The efficacy of LSTM network is tested using variety form of images (from single character to single page format) as input for accurate conversion to braille. Various models and engines have been studied earlier to detect and recognize texts from both structured and unstructured data sources, and their accuracy and efficiency were compared with the present Tesseract-based model.

The outline of the paper is described as follows. Section II briefly describes the proposed image segmentation, character recognition, LSTM Network features, and conversion workflow used in this study for text-to-Brielle code conversion. This also presents an overview of different type of structured datasets used for training and testing of the modified tesseract LSTM model. Section III describes the results and performance analysis of scanned, PDF and blurred images to editable text document conversion method, and evaluation of Tesseract LSTM model performance for different types of inputs including, texts, symbols, equation, paragraph, single page conversion tasks. Section IV presents the conclusions drawn from this present study, and the scopes for future application of the current approach.

MATERIALS AND METHODS

System Description:

The proposed convertor is composed of two important modules, the image acquisition and pre-processing module, and the Tesseract-based braille convertor module as shown in Figure 1. The input text images are basically collected from the user using a scanner or camera system and are saved in JPEG (.jpg) format for pre-processing and segmentation. The image data are fed into a stand-alone program coded in SimpleCV where the image will undergo necessary pre-processing and fed to Tesseract OCR engine for character recognition. When the pre-processing is completed, the Tesseract engine start identifying the characters from the pre-processed image.

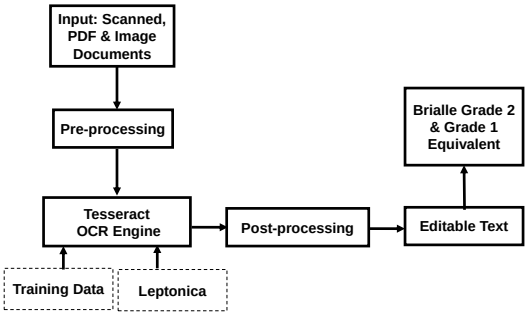

Fig. 1: The block diagram representation of the proposed Tesseract-based Text-to-Braille translation work flow for visually impaired.

The Braille coding system comprises of different types of characters which are also called “cells”. Each Braille cell is made up of six dot positions arranged as two columns of three dots to form a rectangular shape. A dot may be raised at any of these six positions to form sixty-four combinations including, the combination in which no dots are raised. Positions of these dots are universally numbered 1 to 3 from top to bottom on the left, and 4 to 6 from top to bottom on the right. Figure 2A shows the typical Braille cell pattern that the proposed converter may produce as output. Furthermore, in a Brielle translator, the recognized characters of an input image are in the form of ASCII equivalent codes and sent serially to the microcontroller for controlling the braille cell movement, which will be the tangible outputs (braille cell display board) for the blind.

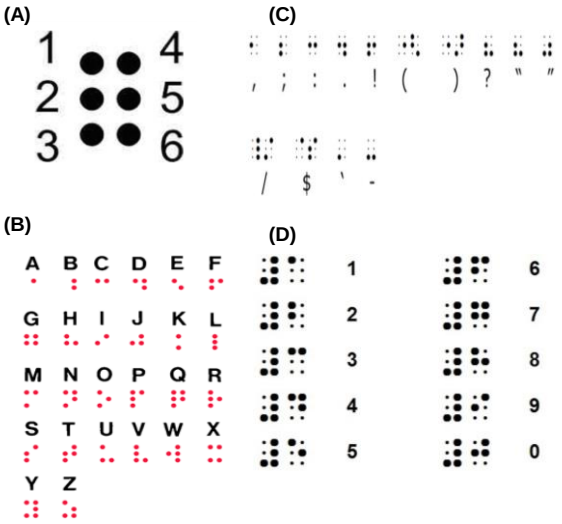

Fig. 2: Shown here are (A) the representative Braille cells, (B) Grade 1 Braille alphabets for English language, (C) Braille alphabets for different symbols and punctuations, and (D) Grade 2 Brielle alphabets for English language.

The system was also verified by accuracy testing where the output braille and the scanned line of text in Arial font were compared. Various font the Arial font of 12, 14, 16, 18 and 20 sizes the target accuracy was achieved depending on the model used and its parameter values. Higher accuracy for a larger font size was expected since the character features are more detailed on those larger fonts.

The present framework was applied to test Grade 1 Braille and Grade 2 Braille conversion by evaluating and comparing the images through various steps in Tessaract engine, which includes i) text extraction, ii) text segmentation, iii) feature extraction, iv) classification, v) Grade 1 Braille script conversion. After capturing an image of the desired text document, the different image pre-processing techniques will be used by the system that will extract the texts. The individual data will be used for processes of acquisition, binarization, noise filtering, edge detection, character segmentation, and text recognition as part of Tesseract image pre-processing. Specifically, image acquisition serves as the first step in order to obtain the image of any tests. Here the iImage binarization converts the colored levels of the image through RGB to grayscale conversion. The thresholding process will convert the gray image into a black and white image in order to have a definite differentiation between the letter pixels and background pixels. The binary image is then utilized for noise filtering in order to remove unwanted pixels in the image.

The next step is the edge detection where it is used to find boundaries between lines of objects and determines discontinuities in the image. For text image segmentation, a clear distinction was made between letter spaces and word spaces. Then the segmented characters were detected using character recognition methods of Tesseract engine. To verify the accuracy of the proposed system, the actual outputs were compared to the expected outputs. The OCR must recognize different fonts. All 26 letters of the alphabet for both uppercase and lowercase, numbers 0-9, and basic punctuation marks of Figure 2 were tested under a series of trials.
ASCII code equivalent to a particular letter or number will be the input to the braille platform and the Braille output should represent the expected equivalent character. The overall system accuracy represents an acceptable accuracy rate of 85 % above in every trial. Each word output of the refreshable braille will be compared to its corresponding word printed on the document. Simulations were performed with an actual document that contains different font and font size to which the braille output of the whole system will be compared. Each word and paragraph output of the refreshable braille will be compared to its corresponding word and paragraph printed in the document. The number of correct output words will determine the accuracy of the whole system.

LSTM-driven Tesseract Engine

Tesseract optical character recognition (OCR) engine is one of the most widely used OCR engines that provides a high accuracy rate compared with other available engines and hence selected here for the analysis of mixed-data [11-12]. Specifically, Tesseract version 4.0 is a new open source platform that has been adapted for approximately 140 different languages and it works based on Long Short-Term Memory (LSTM) neural network. LSTM Network is a specific form of Artificial Recurrent Neural Network that is known to give higher accuracy on several image recognition applications that that of the earlier versions of Tesseract. For Braille translation it needs to be trained from scratch or be fine-tuned using already trained language databases. Different simulation experiments were planned to explore the performance of Tesseract LSTM to capture a number of complex, noisy text image data written in english language and converted to editable text for Braille codes.

Text to Braille Conversion

Using a simple python script the model converts the content of the text file to braille. The script covers the conversion of the numeric, alphabet, and special characters. It can be expanded to grade 2 braille's conversion by directly encrypting some words to braille's instead of character wise encryption of a word as done in grade 1. It has been further extended to multi-language translation to Braille by including French and Spanish along with English. The input text file is converted to a braille's text file accurately. Python being an open-source language provides ample free libraries that are being regularly maintained and can be used to further improve the script to make the system more efficient.

Performance Evaluation and Testing

The performance of the whole system can be evaluated by computing the accuracy of the conversion process under various operating conditions assumed in this study. For instance, the actual Braille system must be actuate in such a way that the pins rise and retract depends only on the input state. Thus the contents of the input image are selected to contain all the characters which are expected to be recognized by the convertor system. The overall accuracy of the system for each single trial is determined using the following formula as:

% Accuracy = (Number of times a character is correctly recognized /Total number of characters) × 100

RESULTS

This section describes a systematic analysis of a wide variety of characters, numbers, symbols, punctuations with varying degree of noises in the scanned documents that were collected from various sources for training and testing of the proposed Tesseract-LSTM model for the proposed Brielle convertor. The LSTM based Tesseract 4.0 OCR engine is trained on the Linux machine to detect and recognize text from an image. The model is trained in a multi-core machine, with OpenMP and Intel Intrinsics support for SSE/AVX extensions. For English and French alphabets, the dataset consisted of characters mixed with numbers and symbols, the file had 250K characters and 20K words. The model data provided has been trained on about 400000 text lines spanning about 4500 fonts. The image can be a scanned document or color picture. The input is a "JPG" or "PNG" format image file and the output will be the text identified, stored in a "txt" file. The accuracy and speed of the model have been altered and improved with better training by changing the weight variables to minimize the loss function error and by improving the dataset by adding text lines containing a mixture of characters and numbers along with symbols. Structured data without noise has provided better accuracy than unstructured data. Initially, the Tesseract 4.0 engine had an error of 2-4 percent for words mixed with symbols which have been reduced significantly below 1 percent by improving the training dataset and optimizing the loss function.

A set of tested images were identified with all in Arial font and with varying font sizes like 14, 16, 18, 20, 22 and 24. These images were called individually in the program and the results were displayed after the execution in order to track the changes for every algorithm applied. The system was systematically evaluated starting with font size 12 and the system was insensitive to recognise the lower character sizes (< 14) for Arial font. Table1 summarizes the computed model accuracies with different sizes of Arial fonts. Figure 4 shows the changes in average accuracy values as a function of font size.

The simulation and training of the Tesseract LSTM network was further carried out on the mixed dataset consisting of 50 various images and manually prepared training dataset with different font style and size. Conversion results of trained Tesseract engine for few sample text images are shown in the Figure 5. Similarly the ability of the Tesseract engine was tested by applying to a randomly selected paragraph for possible translation to editable texts. Figure 6 shows the output of a trained Tesseract model for converting a randomly selected sample paragraph. The result gives 0.1% error on the total word counts for Trained model as the weight has been increased by 5.6% on layer 1 output.

Journal for Re Attach Therapy and Developmental Diversities
eISSN: 2589-7799
2020 July; 3 (2): 113-120

Table 1: Summary of computed model accuracy for different trials and with different sizes of Arial font used in this study.

Trial	Accuracy				
	Arial Font Size				
	16	**18**	**20**	**22**	**24**
1	63.89	88.88	90.72	94.23	97.33
2	53.89	82.12	89.32	96.32	98.12
3	61.45	78.77	90.22	93.62	96.66
4	53.89	88.88	90.72	92.25	97.53
5	66.89	89.88	92.72	94.72	98.23
6	69.89	87.88	91.72	96.29	99.73
7	59.89	77.88	91.72	95.29	99.73
8	66.89	89.88	92.72	94.72	98.23
9	61.45	78.77	90.22	93.62	96.66
10	71.45	88.77	90.22	96.62	97.66
Avg.	**62.85**	**85.17**	**91.03**	**94.76**	**97.98**

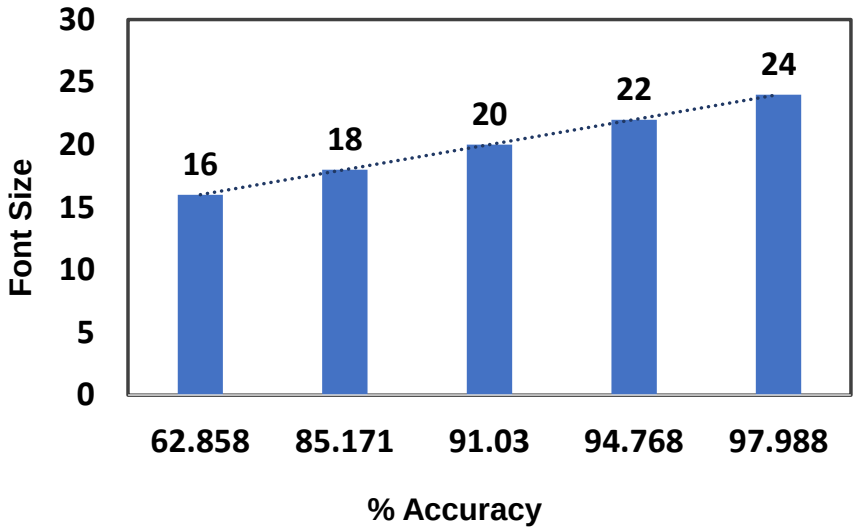

Fig. 4. Shown here is the predicted model accuracy (%) by Tesseract LSTM for different sizes of Arial fonts.

Figure 7 shows a sample page translated by the Tesseract OCR engine after successful training. Then the performance of the system was evaluated by comparing the accuracies for italic and Arial font conversion, which was demonstrated in Figure 8. It was observed that for both types of characters the system shows more than 87% accuracy, which is also consistent with the results obtained with a paragraph or page as input. Figure 9 demonstrates the final stage of conversion from text to Braille by a stand-alone python script for few sample texts as input.

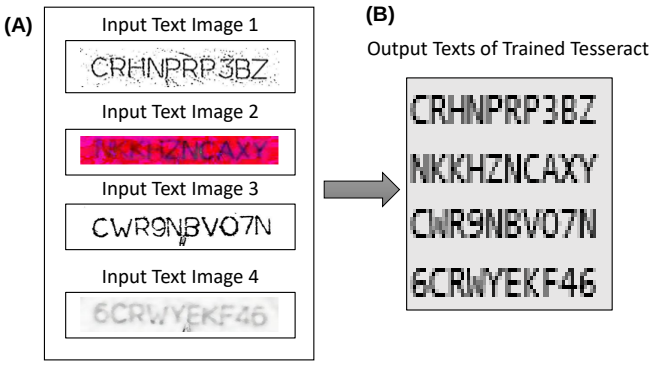

Fig. 5. Shown here is the image-to-text conversion by Tesseract LSTM model for different types of mixed numeric and text data.

Journal for Re Attach Therapy and Developmental Diversities
eISSN: 2589-7799
2020 July; 3 (2): 113-120

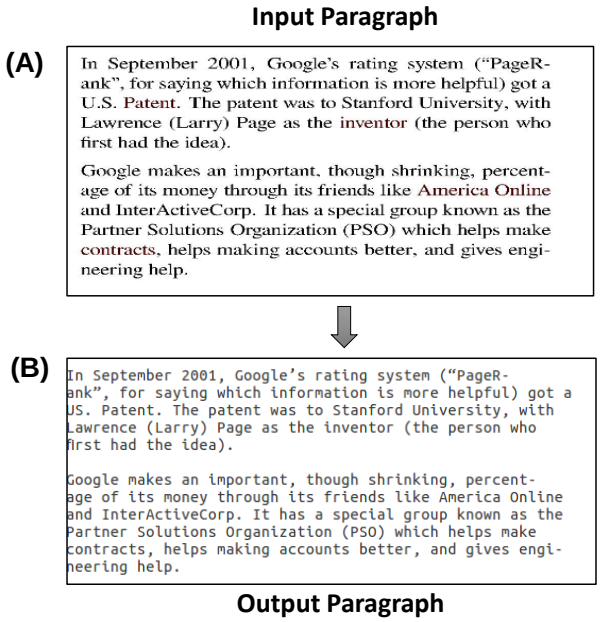

Fig. 6. Shown here is the image-to-text conversion by Tesseract LSTM model for a sample paragraph

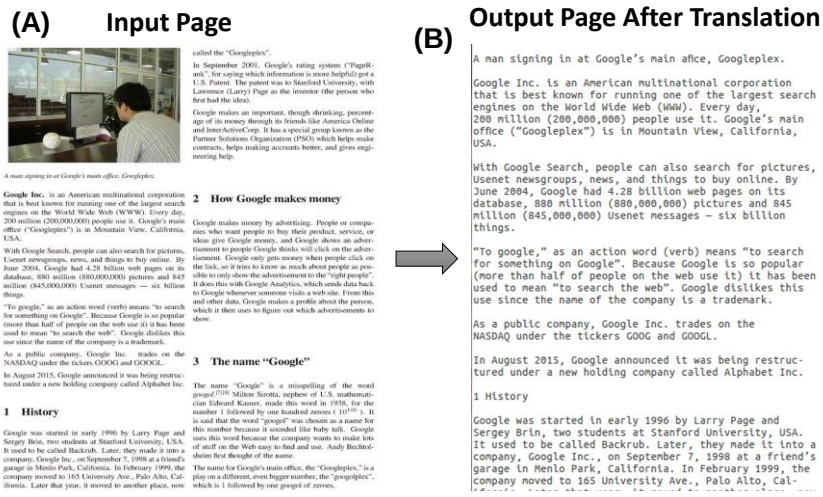

Fig. 7. Shown here is the image-to-text conversion by Tesseract LSTM model for a sample page.

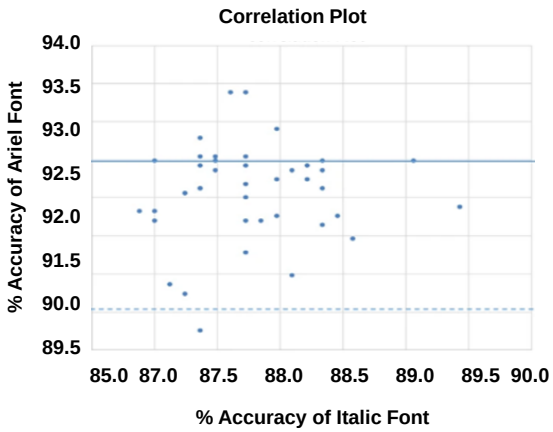

Fig. 8. Shown here is the correlation among the accuracies of two types of fonts (regular vs italic).

Fig.9. Shown here is the (A) different types of input data comprising characters of english, french, spanish, numbers, special character and symbol, (B) a sample Python script for converting text-to-Braille and (C) their corresponding braille's representation as output of the present convertor system.

CONCLUSION

The general motives behind this study was to develop a unified OCR platform using Tesseract LSTM network for efficient text-to-Braille conversion for visually impaired. Using a wide variety of complex structured datasets as image inputs, the LSTM network parameters are fine-tuned and the network was trained to obtain better accuracy compared to existing engines. The proposed framework shows significant improvement in terms of recognizing, single texts, symbols or punctuations, numbers, lines, paragraph and whole document page for english alphabets with minimal conversion time. This provides the blind people an alternative access to reading variety of English language materials by converting printed text into their equivalent braille script.

This study demonstrates the ability of Tesseract engine in recognizing mixed text and mathematical symbols in a document more efficient conversion into Brialle characters and provides a preliminary insights into developing microcontroller-based intelligent braille device for visually impaired. Systematic analysis of existing data reveals that he trained LSTM model of Tesseract depicts 0% average accuracy when tested for small size Arial and italic font (font size less than 14) and exhibits higher accuracy (greater than 86%) when used with large size Arial fonts (font size above 16). It was shown that by carefully modifying the network parameters and using suitable image data the Tesseract engine can be applied for quick text-to-Brialle conversion for visually impaired. The trained model can be effectively embedded into hardware (e.g., Raspberry Pi) for real time application.

Another important fact is that most of the existing OCR systems require extensive training and skills to apply for diverse structured and unstructured data, showing their limited applications in real world. In contrast, the present open source Tesseract-based framework provides a cost-effective, accurate and use friendly tool that can be made available to the common mass for learning and professional purposes in language-free manner. Overall, the present analysis showed that using a training dataset that has a script similarity with a less-resourced language could increase the overall accuracy of the LSTM-based Tesseract engine. Post-processing and fine-tuning the engine parameters may be required for analysis of other language datasets.

The present analysis concluded that the scanned text data of any language could be effectively translated to Braille using LSTM-based platforms, regardless of heterogeneity in the associated features in low resolution text images. It was also observed that, in the absence of fine-tuning, the output could be improved by post-processing. However, if fine-tuning using different font styles is possible, it could eliminate the post-processing phase and improve the conversion rate in real time conversion of image (text)-to-Braille.

REFERENCES

[1] L. Yobas, D. M. Durand, G. G. Skebe, F. J. Lisy and M. A. Huff, "A novel integrable microvalve for refreshable braille display system", J. Microelectromechanical Syst., vol. 12, no. 3, pp 252-263, Jun.2003

[2] G. G. Devi and G. Sathyanarayanan, "Braille Document Recognition in Southern Indian Languages–A Review," in 2018 Fourth International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), Feb. 2018, pp 1–4.

[3] G. C. Bettelani, G. Averta, M. G. Catalano, B. Leporini and M. Bianchi, "Design and Validation of the Readable Device: A Single-Cell Electromagnetic Refreshable Braille Display", IEEE Trans. Haptics vol. 13, no. 1, pp. 239-245, Jan. 2020.

[4] P. Blenkhorn, “A system for converting braille into print,”IEEE Transactions on Rehabilitation Engineering, vol. 3, no. 2, pp. 215–221, Jun. 1995.

[5] P. Blenkhorn and G. Evans, “Automated Braille production from word-processed documents,”IEEE Transactions Neural Syst Rehabil Eng. 2001 Mar;9(1):81-5

[6] S. Shetty, S. Shetty, S. Hegde, and K. Pandit, “Transliteration of text input from Kannada to Braille and vice versa,” in 2019 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), Aug. 2019, pp. 1–4.

[7] M. V.Umarani and R. P.Sheddi, “A Review of Kannada Text to Braille Conversion,” 2018.https://www.semanticscholar.org/paper/A-Review-of-Kannada-Text-to-Braille-Conversion-V.Umarani-P.Sheddi/dfeabc9d90f38821b11017bda01cd9266f3e84d8

[8] Evans DG, Pettitt S, Blenkhorn P. A modified Perkins, Brailler for text entry into windows applications. IEEE Trans Neural Syst Rehabil Eng. 2002 Sep;10(3):204-6.

[9] Blenkhorn P, Evans DG, Baude A. Full-screen magnification for windows using DirectX Overlays. IEEE Trans Neural Syst Rehabil Eng. 2002 Dec;10(4):225-31..

[10] C. N. R. Kumar and S. Srinath, “A Novel and Efficient Algorithm to Recognize Any Universally Accepted Braille Characters: A Case with Kannada Language,” 2014 Fifth International Conference on Signal and Image Processing, pp. 292–296, Jan. 2014.

[11] Vidal-Verdú F, Hafez M. Graphical tactile displays for visually-impaired people. IEEE Trans Neural Syst Rehabil Eng. 2007 Mar;15(1):119-30

[12] L. Hakobyan, J. Lumsden, D. O’Sullivan, and H. Bartlett, “Mobile assistive technologies for the visually impaired,” Surv Ophthalmol, vol. 58, no. 6, pp. 513–528, 2013.

[13] D. T. V. Pawluk, R. J. Adams, and R. Kitada, “Designing Haptic Assistive Technology for Individuals Who Are Blind or Visually Impaired,” IEEE Trans Haptics, vol. 8, no. 3, pp. 258–278, 2015

[14] “Tesseract: an Open-Source Optical Character Recognition Engine | Linux Journal.” 2016. https://www.linuxjournal.com/article/9676.