Grenze International Journal of Engineering and Technology, June Issue

Developing Systems for Translating Braille Documents into Natural English Text to Improve Accessibility

Vishwanath Venkatesh Murthy[1], Sahitya K[2] and Moulya Vishwanath[3]
[1]Associate Professor, MCA, RNSIT, Bengaluru
Email: vishwanathmurthy@rnsit.ac.in
[2]Assistant Professor, MCA, RNSIT, Bengaluru
Email: sahithya.k@rnsit.ac.in
[3]CS, SJBIT, Bengaluru.
Email: mona21vishu@gmail.com

***Abstract*— Braille documents serve as a vital educational resource for visually impaired individuals. However, their traditional form, often created on heavy and cumbersome metallic plates, presents significant difficulties in terms of handling, maintenance, and transportation. Additionally, these documents are prone to wear and tear, especially in unfavourable environmental conditions, which further hinders their usability and longevity. This study seeks to overcome these limitations by converting Braille documents into digital text. Digitized documents can be effortlessly shared across the globe via the internet and reproduced in Braille whenever necessary. This approach simplifies storage, ensures better preservation, and facilitates broader accessibility. The primary focus of this research is on extracting Braille character patterns from physical documents for accurate translation into English text. The paper examines and evaluates numerous tools utilized for this Braille-to-text conversion process.**

***Index Terms*— Braille, CNN, CV2, Deep learning Feature extraction, Keras, Tensor Flow, Matplotlib, YOLO.**

I. INTRODUCTION

Braille, a tangible writing approach designed for persons with visual impairments, has been a crucial tool for accessible communication and education since its creation by Louis Braille in the 19th century [21]. Despite its significance, traditional Braille documents come with challenges due to their physical format, which tends to be bulky, heavy, and difficult to manage. Issues like wear and tear, especially in unfavourable environmental conditions, further complicate their storage and usability [2, 6]. Converting Braille into digital text provides a practical solution to these problems, enabling better preservation, easier sharing, and wider accessibility in a digitally interconnected world.

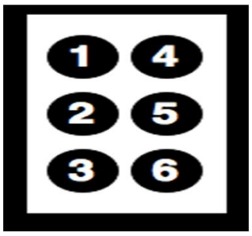
The process of translating Braille into text not only enhances accessibility but also integrates traditional Braille with contemporary digital systems [2, 6]. Digitized Braille content allows for efficient storage, easy distribution through digital platforms, and the ability to recreate Braille materials as needed. This process involves analyzing and extracting Braille characters from physical documents and converting them into readable text formats.

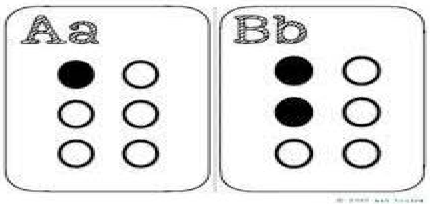
Grenze ID: 01.GIJET.11.2.560_26

This paper examines the tools and techniques employed in the Braille-to-text translation process, focusing on their effectiveness, limitations, and potential to develop ease of access for visually weakened individuals in the digital age [2].

A. Challenges

Braille plates, usually metallic in nature, are heavy and bulky. Moving these documents from one place to another is a difficult process and storing them necessitates considerable space. Furthermore, environmental conditions can cause wear and tear, which makes preservation challenging. To overcome these drawbacks, the transformation of Braille materials into an electronic form, for example, digital text, is necessary.
The translation procedure starts with the scanning of Braille documents to create images. This process can cause problems such as non-uniform lighting, skewing, or other kinds of noise during this step. The quality of the original Braille document also affects the scanned image by incorporating more noise. Preprocessing the images to remove noise and make them clearer is an important and demanding step in the translation procedure. Additionally, the Braille coding scheme is distinct, with each character represented by a unique combination of six dots arranged in a 3x2 matrix as shown in figure-1.





The Braille cell Cell representing alphabets aA and bB Braille word

Figure-1: Braille cell arrangement and alphabets in Braille code

B. Challenges in preprocessing the noise in Scanned Images

Braille images obtained through scanning frequently include noise caused by factors such as uneven lighting, shadows, or smudges on the document surface. To attend these concerns, preprocessing techniques are most crucial to enhance image clarity, eliminate noise, and ensure uniformity in the representation of Braille cells before they are analysed and recognized [17].

C. Challenges in Identifying Cell Boundaries

Correct identification of embossed Braille dots on scanned images is one of the principal problems. This identification process can be inconsistent as a result of variation in dot height, spacing, and wear and tear over time, especially if the Braille document is old or not well-maintained [22].
Evaluating the precise boundaries of each Braille cell is additional challenge. Parting detached Braille cells can become more complicated due to biased scans, intersecting cells or scan distortions created during the scanning process [17]. Precise recognition depends on exact segmentation [3].

Figure-2: Shows the presence of different Variance in Sample of Braille Alphabet 'A'

Supervising Variance in identified Cells is another issue. Recognized Braille cells can be highly diverse due to factors like uneven dot spacing, inaccurate embossing, or noise injected while scanning [3, 7]. One of the most important challenges is creating a system that can handle these differences while still being very accurate [3]. Presence of some of the types of variances representing the same alphabet is shown in figure 2.
Challenges in adapting to different Braille Standards:
Depending on the area or goal, Braille systems may differ slightly in relationship of dot spacing or formatting variance [7]. The recognition process becomes more sophisticated when the translation procedure is modified to

account for these variances. A robust recognition algorithm is required to transform Braille cells into their corresponding alphabets once they had been identified and separated. Since every Braille character consists of a unique combination of six dots in a 3x2 grid, even minor differences can lead to misreading [16].

II. Tools Used in Braille-to-Text Translation

Several tools and techniques had been established to facilitate the translation of Braille into text, each catering to different stages of the process. Such tools always put together image processing, machine learning, and specified algorithms to boost accuracy and efficiency.

- Optical Braille Recognition (OBR) Tools: These tools use scanned images of Braille documents and employ image recognition algorithms to detect and interpret Braille dots [4]. They preprocess the scanned images to remove noise and segment Braille cells for accurate mapping to text. Examples include software like DotsPlus and Ophthalmic Braille Translators [5].
- Machine Learning-Based Models: Machine learning models, remarkably those leveraging convolutional neural networks (CNNs) [4], had been employed to identify Braille patterns [4]. These models are exercised on significant datasets of Braille images to recognize variations in dot patterns and map them to their corresponding text [19].
- OCR Adaptations for Braille: While traditional optical character recognition (OCR) tools are designed for printed text, adaptations have been made to handle Braille by customizing detection algorithms to interpret the unique 3x2 matrix format of Braille cells [17].
- Hardware Scanners and Embossers: Specialized Braille scanners equipped with high-resolution cameras can capture fine details of Braille dots. These scanners are combined with software packages to convert Braille material into digital form. Embossers can print text back into Braille for checking. [11, 15, 18].
- Open-Source Solutions: Certain open-source libraries and frameworks provide customizable alternatives for Braille-to-text translation, enabling researchers to test different algorithms and refine the recognition process. Some examples include Python reference library intended for image processing and pattern recognition [17].

These tools collectively address challenges like noise removal, cell segmentation, and character mapping, contributing to more efficient and accessible Braille-to-text translation systems.

III. Techniques used in Various stages of Braille-to-Text Translation

The Braille-to-Text Translation system for converting Braille doc to English text is split in 5 different stages as represented by figure-1.2 includes image procurement, image processing, segmentation and finally feature extraction with classification. After going through all these stages finally Braille document is translated to English sentence.

The various Braille to text translation stages are:

- Image acquisition – Braille images are obtained using a standard scanner.
- Image Pre-processing: Scanned image is pre-processed to remove noise.
- Image segmentation: Row and column segmentation to locate the Braille cells.
- Feature extraction: Perform Braille cell and Braille dots identification.
- Translate Braille cell to English alphabet.

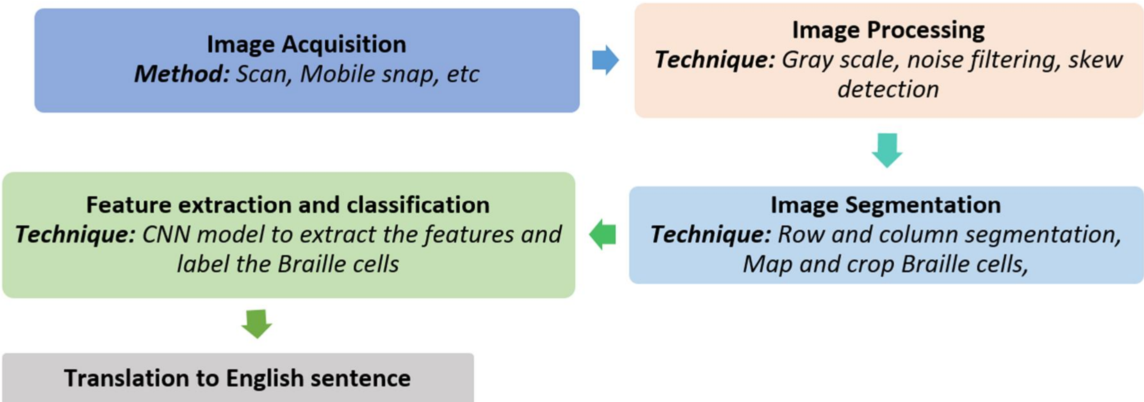


Figure 2: Braille to text system framework

A. Image Acquisition

The Braille images are obtained by scanning the physical Braille documents using a standard scanner. Many times, Braille images are trapped employing mobile devices by taking documents snaps [2, 6].

B. Image Pre-Processing Stage

The image preprocessing stage is crucial for enhancing the quality of scanned Braille images and preparing them for accurate recognition.

Important techniques employed in this stage include:

- Noise Reduction: It uses methods like Gaussian blur, morphological operations and median filtering to eliminate noise caused by irregular lighting, shadows, or blurs on the scanned document [16, 20].
- Binarization: Translating the Gray-scale image into a binary format (black and white) facilitates to isolate the Braille embossed dots from the background, streamlining further processing. Techniques such as Otsu's thresholding or adaptive thresholding are commonly used [5].
- Skew Correction: Images may become skewed during scanning, which can distort the arrangement of Braille cells. Algorithms like Hough Transform or projection-based methods correct these distortions by aligning the image [18].
- Contrast Enhancement: Enhancing the contrast between the Braille embossed dots and the background improves the visibility of embossed patterns, aiding in their detection. Histogram equalization or contrast stretching methods are typically applied [14].
- Edge Detection: Techniques like the Canny or Sobel edge detection algorithms are used to identify the boundaries of Braille dots and cells, facilitating accurate segmentation [16].
- Morphological Processing: Morphological operations such as dilation, erosion, opening, and closing are applied to refine dot shapes, bridge gaps, and remove small artifacts from the image [15].

C. Image Segmentation Stage

The goal of this process is to convert the tactile Braille dots into readable text that can be interpreted by screen readers or other assistive technologies [16].

Image Segmentation Techniques Used in Braille-to-Text Translation

- Thresholding: uses Global Thresholding or Adaptive Thresholding to convert grayscale images into binary images [16]. Global Thresholding: This method sets a global threshold value to transform grayscale images into binary images. White pixels are those that are brighter than the threshold, whereas black pixels are those that are darker than the threshold [1, 3]. When the dots and backdrop in a Braille image contrast well, this method works well. Adaptive Thresholding: This method modifies the threshold value according to specific areas of the image. It helps distinguish Braille dots from the backdrop in situations where the document's illumination or image quality varies [17].
- Edge Detection: uses Canny Edge Detection or Sobel Operator technique to detect edges in an image. Canny Edge Detection: This technique detects edges in an image by identifying areas where there is a rapid change in intensity [16]. It helps to identify the contours of Braille characters and their boundaries. Once the edges are detected, segmentation techniques can be applied to isolate individual Braille characters [19]. Sobel Operator: This is another edge-detection method used to highlight the borders of Braille dots. It helps in defining the exact region of interest for character extraction.
- Connected Component Labelling (CCL): In this technique, the image is processed to identify distinct connected regions (i.e., clusters of pixels) that represent the Braille dots. Once the dots are isolated, the connected components are labelled, allowing for the identification of each Braille character or symbol 20].
- Morphological Operations: This includes Erosion and Dilation operations. These are used to remove noise (erosion) or close gaps between connected dots (dilation) [19]. Erosion reduces the size of objects in a binary image, while dilation enlarges the features. These operations are used to enhance the segmentation quality by removing extraneous noise or ensuring that small gaps between Braille dots are closed [1, 3].
- Region-based Segmentation: This approach involves separating the image into discrete regions based on the similarity of pixel values [13, 19]. After the image is divided into smaller regions, the algorithm identifies areas that correspond to Braille dots and then proceeds to isolate them [16]. Watershed Algorithm is one of the popular region-growing technique for region-based Segmentation [3].
- Hough Transform: This technique is often used for detecting lines, circles, or other shapes within an image [13]. In Braille-to-text translation, it is used for detecting the circular or semi-circular Braille dots, especially when they appear in a regular grid pattern [6].

D. Feature Extraction Stage

Feature extraction is essential for converting Braille images into machine-readable text [20]. By capturing the unique patterns of Braille symbols (dot positions, shapes, textures, etc.).

Geometric Features

Uses various features of Braille information like Dot Positioning, Dot Count and Grid Structure.

- Dot Positioning: Since Braille characters are represented by a 2x3 grid of raised dots, the relative positions of the dots within the grid are key features. By capturing the precise spatial location of each dot, the algorithm can determine which specific character is being represented [13].
- Dot Count: Braille symbols are made up of dots arranged in different patterns. The number and positions of the dots can serve as distinctive features. For example, a Braille character with one raised dot in the top left corner is different from one with three dots in the left column [14].
- Grid Structure: The 2x3 grid structure of Braille can be used to define the character. The relative position of dots in this grid is an essential feature to extract, as it corresponds directly to the Braille character being represented [20].

Shape Features

Braille dots are typically circular, so analyzing the roundness or circularity of the dots can provide important features. Additionally, the area of each dot can be used as a feature, as variations in size can help distinguish different symbols, especially in cases of low-quality images or embossing [20].

Histogram of Oriented Gradients (HOG)

HOG is a feature extraction system that captures the distribution of gradients (changes in intensity) in different orientations [16]. For Braille, this can help in detecting the contours and the arrangement of the Braille dots. The HOG method is effective for capturing the texture of the Braille pattern and can be robust to variations in lighting and noise.

Zoning Techniques

Zoning is a technique where the image is divided into several regions or zones, and features are obtained from each zone individually [12]. For Braille, zoning can be applied to the grid structure of the Braille symbols (e.g., dividing the 2x3 grid into smaller sections and analyzing the presence or absence of dots in each zone). Each zone's feature set may include simple properties like the number of raised dots in a zone or more complex statistics (e.g., density, orientation) based on pixel information in each zone [6].

Fourier Transform Features

The Fourier Transform is a technique used to analyze the frequency components of an image. By transforming the Braille image into the frequency domain, it's possible to extract frequency features that represent the periodicity and regularity of Braille dots [13]. For example, the regular spacing between Braille dots can be captured by the frequency components. This method is particularly useful for recognizing the spatial frequency of Braille patterns, which is important for distinguishing between different Braille characters based on their dot arrangements [14].

Principal Component Analysis (PCA)

PCA is a dimensionality reduction technique used to reduce the complexity of the feature space while preserving as much variance as possible. In Braille-to-text translation, PCA can be applied to the extracted features of Braille characters, reducing the number of features used by classifiers while still retaining key distinguishing information [21].

Wavelet Transform Features

The Wavelet Transform is a mathematical technique used for analyzing signals at different scales or resolutions [12].

In Braille images, the wavelet transform can capture both high-frequency (edge) and low-frequency (smooth) components of Braille symbols [12]. This helps in identifying both the fine details of Braille dots and the overall structure of the characters.

Deep Learning-based Features

- Convolutional Neural Networks (CNNs): CNNs can be trained directly on raw Braille images to learn spatial hierarchies of features. These networks automatically extract relevant features at multiple levels (edges, textures, shapes) without the need for hand-crafted feature extraction [7]. The learned features are then used for classification, significantly improving accuracy, especially for complex Braille patterns and noisy images.

- Transfer Learning: In some cases, pre-trained deep learning models (e.g., trained on large datasets like ImageNet) are fine-tuned on Braille images to extract features that help classify Braille symbols efficiently [1]. This method leverages the power of large-scale image recognition models and adapts them to the Braille context.

IV. Conclusion

In conclusion, the digitization of Braille documents represents a significant advancement in improving availability for visually impaired individuals. By leveraging tools such as image preprocessing, segmentation, and feature extraction, researchers can effectively convert Braille into readable text, thus facilitating easier storage, sharing, and reproduction of Braille materials. Despite challenges like noise, cell boundary identification, and dot variations, recent advancements in machine learning, particularly CNNs and deep learning, hold great promise in overcoming these barriers. Effective segmentation and feature extraction techniques, including geometric and shape-based methods, further enhance accuracy in translation. The integration of modern technologies ensures that Braille can seamlessly connect with digital systems, improving accessibility on a global scale. Future research should focus on refining these methods and exploring new algorithms to address diverse Braille standards, as well as scalability in processing large volumes of documents. Such innovations will be pivotal in promoting equality and inclusion for individuals with visual impairments. As digital Braille systems evolve, their widespread adoption will contribute to a more accessible and inclusive world.

References

[1] Asfaw, N., Belay, B. & Alemu, K. A deep learning approach for line-level Amharic Braille image recognition. Sci Rep 14, 24172 (2024). https://doi.org/10.1038/s41598-024-73895-7

[2] A. Sai Laxmi, R. Sneha, P. Supraja and R. Prasath, "Braille Communication System for Visually Impaired with Haptic Feedback System," 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2022, pp. 697-702, doi: 10.1109/ICOSEC54921.2022.9952150.

[3] A. AlSalman, A. El-Zaart, S. Al-Salman and A. Gumaei, "A novel approach for Braille images segmentation," 2012 International Conference on Multimedia Computing and Systems, Tangiers, Morocco, 2012, pp. 190-195, doi: 10.1109/ICMCS.2012.6320146.

[4] Baumgärtner, C., Schwarz, T., Stiefelhagen, R. (2020). "Image-Based Recognition of Braille Using Neural Networks on Mobile Devices". In: Miesenberger, K., Manduchi, R., Covarrubias Rodriguez, M., Peňáz, P. (eds) Computers Helping People with Special Needs. ICCHP 2020. Lecture Notes in Computer Science(), vol 12376. Springer, Cham. https://doi.org/10.1007/978-3-030-58796-3_41

[5] Braille Computer Vision Project. Roboflow, 2024, [online] Available: https://universe.roboflow.com/mingyu1123/braille-ifwv9.

[6] C N Ravi Kumar (2011). "An Insight into Optical Braille Character Recognition since its Conceptualisation". International Journal of Computer Applications, 33, 1-4.

[7] 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, Bangalore, India, 2014, pp. 292-296, doi: 10.1109/ICSIP.2014.52.

[8] D. BARCZYK and D. JASINSKA, "Experimental studies of the quality of embossed characters of the Braille alphabet", BULLETIN OF THE POLISH ACADEMY OF SCIENCES TECHNICAL SCIENCES, pp. 607-614, 2016.

[9] D. Joshi et al., "An Interactive and Economical Embedded Braille Machine for the Visually Challenged People," 2023 9th International Conference on Signal Processing and Communication (ICSC), NOIDA, India, 2023, pp. 632-637, doi: 10.1109/ICSC60394.2023.10441493.

[10] Harris LN, Gladfelter A, Santuzzi AM, Lech IB, Rodriguez R, Lopez LE, Soto D, Li A (2023) Braille literacy as a human right: a challenge to the "inefficiency" argument against braille instruction. Int J Psychol 58(1):52–58

[11] H. Chen et al., "A Novel Refreshable Braille Display Based on the Layered Electromagnetic Driving Mechanism of Braille Dots", IEEE Transactions on Haptics, vol. 16, no. 1, pp. 96-105, Jan.-March 2023.

[12] Jie Li, Xiaoguang Yan and Dayong Zhang, "Optical Braille recognition with Haar wavelet features and Support-Vector Machine," 2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering, Changchun, 2010, pp. 64-67, doi: 10.1109/CMCE.2010.5610062.

[13] J. Chen and N. Takagi, "A homogeneous region based methodology for text extraction from natural scene images," 2015 International Conference on Informatics, Electronics & Vision (ICIEV), Fukuoka, Japan, 2015, pp. 1-6, doi: 10.1109/ICIEV.2015.7334037.

[14] Korde, A., Gaikar, O., Nikam, S., Rukhande, S. (2020). Wireless Emanation of Braille to Text/Voice and Vice Versa. In: Pant, M., Kumar Sharma, T., Arya, R., Sahana, B., Zolfagharinia, H. (eds) Soft Computing: Theories and Applications. Advances in Intelligent Systems and Computing, vol 1154. Springer, Singapore. https://doi.org/10.1007/978-981-15-4032-5_37

[15] Manjari K, Verma M, Singal G (2020) A survey on assistive technology for visually impaired. Internet of Things 11:100188

[16] Murthy, V.V., Hanumanthappa, M., Vijayanand, S. (2021). Braille Cell Segmentation and Removal of Unwanted Dots Using Canny Edge Detector. In: Chiplunkar, N.N., Fukao, T. (eds) Advances in Artificial Intelligence and Data Engineering. AIDE 2019. Advances in Intelligent Systems and Computing, vol 1133. Springer, Singapore. https://doi.org/10.1007/978-981-15-3514-7_7

[17] M. Hanumanthappa and V. V. Murthy, "Optical Braille recognition and its correspondence in the conversion of Braille script to text — A literature review," 2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), Bengaluru, India, 2016, pp. 297-301, doi: 10.1109/CSITSS.2016.7779374.

[18] M. A. Rahman, K. Dhar and M. A. Ullah, "Hardware design and implementation of Bengali Braille embosser," 2017 4th International Conference on Advances in Electrical Engineering (ICAEE), Dhaka, Bangladesh, 2017, pp. 613-617, doi: 10.1109/ICAEE.2017.8255430.

[19] Nguyen, K. & Nakagawa, N. M. Deep convolutional recurrent network for segmentation-free offline handwritten Japanese text recognition. In Proc. International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November (2017).

[20] N. P. D. P. Utami, F. Hamami and R. Y. Fa'rifah, "Development of Braille Text Recognition System into Alphabet Using YOLOv8," 2024 International Conference on ICT for Smart Society (ICISS), Bandung, Indonesia, 2024, pp. 1-5, doi: 10.1109/ICISS62896.2024.10751344.

[21] Park, BS., Im, SM., Lee, H. et al. Visual and tactile perception techniques for braille recognition. Micro and Nano Syst Lett 11, 23 (2023). https://doi.org/10.1186/s40486-023-00191-w