Advanced Real Time Embedded Book Braille System

Vasile Dan *, Ioan Nașcu, and Silviu C. Folea

Department of Automation, Technical University of Cluj-Napoca, Cluj-Napoca, Romania;
Email: Ioan.Nascu@aut.utcluj.ro (I.N.), Silviu.Folea@aut.utcluj.ro (S.C.F.)
*Correspondence: vasidan.mdt@gmail.com (V.D.)

***Abstract*—Reading is an activity that leads to acquiring information and developing a person's knowledge. Therefore, everyone should have equal access to the same sources of information. Unfortunately, blindness is a disease that restricts the affected people from reading books that are not converted into Braille. This paper describes a novel solution for the real-time conversion of any text into Braille. The system will rely on image processing and a camera to gather the raw text data from any book in physical format. Furthermore, e-books and documents in any digital format, Braille Ready Format (BRF), Portable Embosser Files (PEF), TXT, PDF, or PNG, can be provided for Braille conversion. Image enhancement algorithms, neural networks, and Optical Character Recognition (OCR) algorithms are used to extract accurate content. The process is controlled by a Raspberry Pi 4. A refreshable Braille mechanism, based on an Arduino Due microcontroller, is used to display the dots for each character. The algorithms are implemented to work with the mechanical structure design that was created to reduce the cost and give the user a complete reading experience of any book.**

***Keywords*—embedded braille system, image processing, optical character recognition**

I. INTRODUCTION

In everyday life, people face various problems due to their medical condition. Technology development has led to specialized systems designed to help those in need. Vision is one of the human senses essential for interaction with the world. People who are affected by blindness cannot engage in some activities. There are various proposals of systems for solving this problem.

Mobility of the blind is addressed in [1], where an electrical stimulation display is proposed for blind people to perceive their surroundings. The display consisting of 512 electrodes is applied to the person's forehead. The system uses a camera to acquire images, after which contours are extracted and transformed into tactile sensations. A more advanced solution is proposed by Lumen. This system consists of a pair of glasses, which uses artificial intelligence to understand the environment. The set of five built-in cameras helps to determine the user's position of the user and perceive information. Their transmission to the user is carried out by audio and haptics [2].

Graphiti is an interactive graphical display by Orbit Research that can represent graphical information such as diagrams, sketches, or images. The technology of this device allows the display of topographical maps and other graphic elements, where the variable height of the dots represent shades of gray or different colors. The system display comprises 60×40 pins that can be independently actuated at a variable refresh rate [3].

In order to have access to written information, blind people use Braille. This language is the primary form of reading and writing, which uses distinct patterns of raised dots. Thus, the sense of touch is used for gathering information. A Braille cell consists of 6 dots, arranged in two columns of three dots. Each cell can represent a letter, a word, a combination of letters, a number, or a punctuation mark [4]. The size of the points and the distance between the cells are standardized. Depending on the geographical area, these variables may differ.

This paper is a study that expands the vision of Braille converters by offering a complete solution from design to hardware and software. The system can convert into Braille any written information in physical or digital format. The design of the Braille display mechanism allows a high refresh rate and accuracy of the cells using limited hardware. Certain features were implemented to improve the user experience: an online dictionary, multiple languages selection library, and text-to-audio converter. The focus of this project was to create a system that enables the full potential of the reader at a low cost. Thus, the main components are a camera, Raspberry Pi, Arduino Due microcontroller, and stepper motors.

The paper is organized as follows: the next section presents a technical review of Braille systems. Section III presents the image processing algorithms. Section IV discusses the mechanism and design of the book scanner and the refreshable Braille display. In the final section are presented the conclusions.

II. LITERATURE SURVEY

There have been various proposals for devices that focus on improving the reading experience of a physical book for blind people. Cannute 360 makes it possible to read a book available in any digital format or digital

Manuscript received November 29, 2022; revised January 16, 2023; accepted February 9, 2023; published June 1, 2023.

Braille format: Braille Ready Format (BRF), Portable Embosser Files (PEF) [5]. The refreshable reading area consists of 9 lines of 40 cells each. The cell comprises two plastic pieces, on which three dots can be displayed, forming the six dots corresponding to the Braille alphabet. Several buttons are implemented to navigate through the book content. Raspberry Pi 4 and stepper motors control the refreshable display [6].

Data acquisition is the first step in converting a text into Braille using image processing. Many text recognition methods can be applied: Optical Character Recognition (OCR), adaptive binarization based on wavelet filter, Markov model-based system, or neural networks [7].

Tactile is the first physical text converter. It converts any text in real time and displays the result in Braille. Tactile uses a video camera to capture the text of a book. It slides across the page, and after pressing a button, the text in the camera's field of view is displayed [8]. A similar device is BrailleRing. It is a robust, compact, and portable device that offers a cheaper alternative to systems with an extended display. The Braille dots are made of small cuboids, which rotate 90° clockwise to form the essential 64 combinations [9].

Blitab is an Android tablet that has a touch surface and an intelligent Braille surface on the top. The Braille display has 14 rows of 23 six-dot cells [10]. The device uses a technology that features smart liquids to create tactile pixels (tixels), like bubbles on the screen [11].

Various types of Braille displays determine the cost of the system. The lower the number of cells, the lower the cost. Thus, solutions have been proposed even with a single Braille cell. In Russomanno and Modhrain *et al.*'s research, it was shown that single-cell displays are less efficient if they do not involve sliding contact [12]. In Hossain and Raied *et al.*'s research [13], a display consisting of two sliders with 11 slots using permutations was proposed. Dynamic housing actuation requires low power to perform linear movement compared to piezoelectric displays. In Aryan and Doshi's research, a display mechanism consisting of only two servomotors for a Braille cell is presented [14]. In Karastoyanov and Karastoyanov's research, the authors have discussed microfluidic technology that can produce devices with reduced weight and size [15]. The display would use bubbles of air or liquid to create dots on the screen. Thus, more complex information could be displayed at a low cost. In Fletcher and Downs *et al.*'s research [16], a concept for a magnetic Braille display is proposed. Braille dots will be formed using a ferrofluid through a controlled magnetic field produced by an electromagnet.

In paper Raghunandan and Anuradha's research, the focus is on the translation of a text that can be in different languages [4]. The existing applications have not implemented translations of specific languages or special characters. The proposed approach extracts each character from the text and compares it with the predefined characters in the existing database. Thus, the Pi Tesseract optical recognition engine compares characters and then converts them to Braille. Similar solutions have been proposed by Sharavana and Sivasubramanian *et al.* [17] and Kumari and Akole *et al.* [18]. A step further was made in the paper of Sharavana and Sivasubramanian *et al.* [17], where the Google Text-to-Speech (GTTS) library is used to convert the detected text to audio files.

In Golonka and Kothare *et al.*'s research [19], a mechanism is presented that uses octagonal discs rotated by two stepper motors controlled by a microcontroller. Four linear axes rotate the disks, and another four fix the disks that have been rotated. Due to this sequential method of rotating the discs, the refresh rate of the entire display is a significant disadvantage compared to other systems. An improved mechanism based on octagonal disks is presented in the fourth chapter of this paper.

In the previously exposed works, the architecture of the proposed system is similar: image acquisition using a Raspberry Pi camera, image processing by applying various filters, character recognition, character conversion into Braille, text conversion into an audio file, and display of the Braille characters. The hardware components used are a Raspberry Pi camera or a webcam for data acquisition, a Raspberry Pi board for processing, and solenoids for Braille display. These systems do not provide a complete and reliable solution for improving the user experience. The lack of a complete system design, text display control, the mechanism for displaying Braille dots, and the size of the reading area is a disadvantage.

In this paper, the focus is on the user's reading experience. The system architecture will have some aspects in common with the work done so far. The system structure is designed to allow the easy browsing of a book in physical format. A mechanism will fix the book so that the camera covers both pages of it, and the software solution allows easy navigation of the two pages that will be open. The system has a Perkins-style Braille keyboard through which it will be possible to search for the definition of a specific word. Thus, the user will have all the necessary information for unknown words. The display mechanism has ten lines of 40 cells each. They are made of octagonal plastic disks and are driven by stepper motors.

III. Image Processing

Image processing is one of the essential components of this project. The process can be divided into three large subchapters: pre-processing, text detection, and post-processing.

A. *Pre-processing*

Storing an information source from physical to digital format at a higher quality is done by scanning. In the case of books, the scanned image must be processed to remove unwanted areas outside the region of interest. If two pages are scanned at once, various methods must be applied to delineate the two pages. In Stamatopoulos and Gatos's research, an algorithm was proposed for the detection of pages on scanned images containing two pages [20]. The algorithm splits the two pages and improves the quality of the pages by removing borders. Binarization, noise removal, image smoothing, detection of vertical and horizontal white areas corresponding to pages represent the main steps of this algorithm. Also, in research of Shafait and Beusekom *et al.* [21], an algorithm is presented that involves detecting the content of a page to remove unwanted areas, the edges. Thus, a geometric matching algorithm is applied.

In Fig. 1 are presented the steps for extracting the book pages from the scanned image. The scanning process is accomplished by a Raspberry Pi camera module mounted above the book. The camera has an 8-megapixel sensor with a resolution of 3280 by 2464 and a circular LED module. The book stand will hold the pages at a 120° angle to preserve the book's integrity. Thus, various algorithms are applied to get a high-quality image.

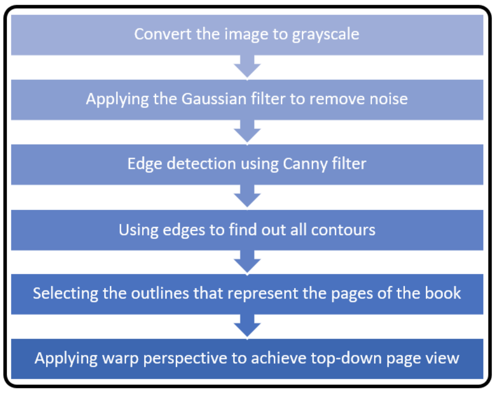

Figure 1. Book pages extraction algorithm.

The first four steps of the algorithm are completed by using the edges to find out all the contours within the image, as can be seen in Fig. 2b). The book's pages are delimited by a thin strip connecting the two glass plates, which are pressing the pages from above. Thus, segmentation errors will be minimized. The algorithm for extracting the pages is based on selecting the first two largest contours, approximated with a rectangular shape, as shown in Fig. 2d).

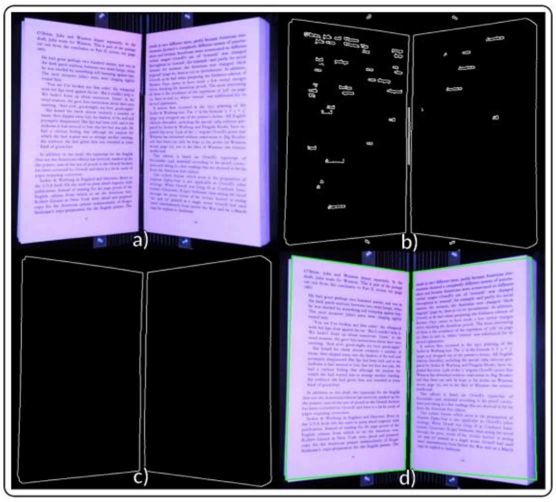

Figure 2. a) Original image; b) Contours of the image; c) Contours of the pages; d) Approximate rectangular contours of the pages.

In Fig. 3, we can see the last step's result, which consists of applying the warped perspective to obtain the top-down view of the pages. This will increase the accuracy of text detection. The algorithms for image enhancement will also be applied: binarization, noise removal, and smoothing.

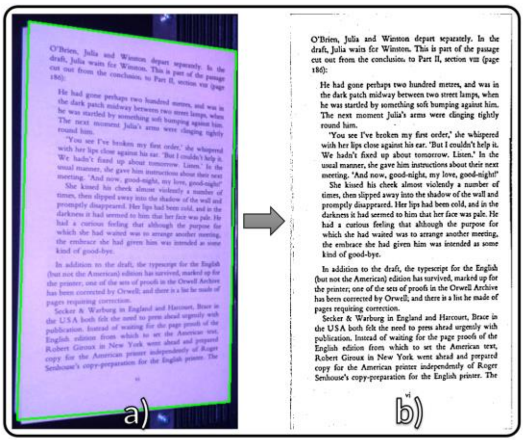

Figure 3. a) Initial page; b) Page after applying warp perspective and enhancement algorithms.

B. *Text Detection*

OCR is widely used in numerous applications. One field of use is the automatic recognition of license plates. Shashidhar and Manjunath *et al.* [22], Onim and Nyeem *et al.* [23], Cai and Sun *et al.* [24] have proposed solutions based on various neural networks and the OpenCV library. Extracting certain information from images with structured content is another way of using OCR. Thus, solutions for extracting valuable data from receipts, tax invoices, curriculum vitae, sales contracts, identity cards, or passports were proposed by Antonio and Putra *et al.* [25], Zhang and Xu *et al.* [26], Jiju and Tuscano *et al.* [27]. Nonetheless, not all the information of structured content is typewritten. Nath and Aggarwal *et al.* have focused on building a framework to detect handwritten text from registration forms [28].

Another field of use for OCR is document digitization. Zhu and Sokhandan *et al.* have presented an approach to digitizing newspapers due to its complex layout [29]. They have used advanced segmentation and detection methods, such as Mask R-CNN from Detectron2. Also, post-processing algorithms were used to improve text recognition metrics.

Depending on the data input quality, the result of the conversion can have errors. Post-processing algorithms are implemented for detecting and correcting errors. Phamtoan and Nguyen *et al.* [30] have proposed a new approach by using an adapted version of the Hill Climbing algorithm for OCR error correction.

Figure 4. Optical character recognition process.

The detected text, which can be stored in a database, as shown in Fig. 4, can be converted to Braille or audio. Sharavana and Sivasubramanian *et al.* [17], Kumari and Akole *et al.* [18], Dhandapani and Arumbu *et al.* [31], Mathumitha and Vinodhini *et al.* [32] have given solutions based on Tesseract OCR framework.

There are different frameworks for OCR with distinctive features. Hegghammer have presented a performance evaluation report comparing three frameworks [33]: Tesseract, Amazon Textract, and Google Document AI. The tests included various scanned pages where artificial noise was applied. The differences between these frameworks are installation time, number of available languages, architecture, and cost. Tesseract OCR has 116 languages, Long Short-Term Memory (LSTM) architecture, local installation, and free cost. Thus, it is the primary choice for this project.

Pytesseract is the library that must be imported within the python script and provides an extensive set of parameters that can be configured:

- Extended language support;
- Page Segmentation Mode (PSM): how the image is segmented into paragraphs, lines of text, and words. There are 14 options available;
- OCR engine mode (OEM): affects detection performance and accuracy. There are four options available.

The detection process creates a dictionary that contains the number of words, the words' bounding box coordinates, the detection's accuracy, or the number of the detected line of text. In Fig. 5, the detected words are highlighted in green bounding boxes.

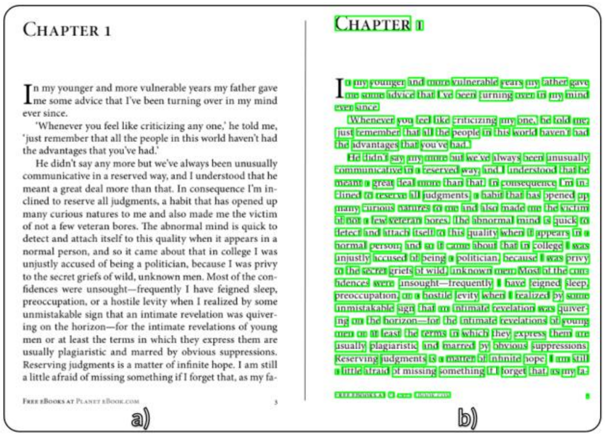

Figure 5. Pytesseract words bounding boxes: a) Original image; b) Highlighted words.

OEM has four available options:

a) Legacy mode: is a 3-step process: word detection, line detection, and character classification;

b) LSTM neural networks;

c) Legacy mode and LSTM neural networks;

d) Default mode: depending on what is available.

By comparing the detection results of the three modes, the accuracy is 91% by using the Legacy mode. For LSTM neural networks, the accuracy in modes b) and c) is 97–98%. Thus, mode b) was selected for OEM. Errors that may occur can be divided into three categories: undetected characters, semantic errors, and punctuation marks.

C. Post-processing

The detected text, saved into a file, can be converted into Braille or audio. PyBraille is a python library that can be used for this purpose [34]. However, the library only contains all letters of the English alphabet, numbers, and some punctuation marks. Thus, a new library was created with the following features:

- Indicators for numbers, uppercase, symbol, italics, and bold;
- All punctuation and grouping;
- Signs of operations and comparison;
- Signs for various currencies and measurements;
- All special symbols;
- Characters from the Romanian language alphabet.

The new library includes support for the English and Romanian alphabet characters, and there is also the possibility to add characters for other languages. The characters are saved in Unicode format. Each character has assigned to it a vector of 6 binary values that constitutes a location in the 6-point Braille encoding. Thus, the first three values correspond to points 1, 2, and 3, and the last three correspond to points 4, 5, and 6.

Text-to-Speech (TTS) libraries are used to synthesize intelligible and natural speech. The development of neural networks and artificial intelligence has significantly improved audio quality [35]. Two python libraries can be used to convert text to audio: Text-to-speech x-platform (pyttsx3) and Google Text-to-speech (gtts).

Pyttsx3 features are speaking rate, volume, and voice change control. It can be used without an internet connection. However, using gtts, the text can be translated into various languages, and the speech is more natural. Thus, the gtts library is used for this project. The Raspberry Pi 4 has a jack output, which can be used to play sound.

During a reading, there is a possibility that the text contains certain words whose meaning is not known to the reader. In these situations, explanatory dictionaries are used, which can be in physical or electronic format. Thus, a feature for searching the definitions of a word has been implemented. The word's definition will also be partially displayed on the Braille display or audible, depending on the method selected. The online dictionary is accessed using the Beautiful Soup library, specific to the Python language. Hence, definitions can be extracted from HTML and XML files of specific web pages [36].

IV. BRAILLE SYSTEM DESIGN

User experience and cost are fundamental factors in designing and implementing the chosen solution. The two main mechanisms are a book scanner and a refreshable Braille display.

A. Book Scanner Mechanism

There are several book scanners: manual and automated. CZUR ET Smart Book Scanner [37] uses various technologies for tilt correction, cropping areas of interest, flattening, and anti-reflection. BFS-Auto [38] is a system that scans books automatically. Obtaining the 3D shape of the book, tracking each page, and processing, are the main methods of the system for a higher-quality result [39].

The systems mentioned above use software to flatten the book pages. An alternative to this, a structural solution, can be used: a “V” shaped stand. ScanRobot 2.0 MDS [40], Linear Book Scanner [41], and Archivist Quill Book Scanner [42] are using this type of stand that decreases the image processing algorithms and book damage.

The book scanner mechanism consists of two stands in a “V” shape at a 120° angle. One fixed, at the bottom of the structure, made of plastic and aluminum plates. The second one is mobile, made of two glass plates and a plastic frame. The glass stand will press the book and flatten the pages.

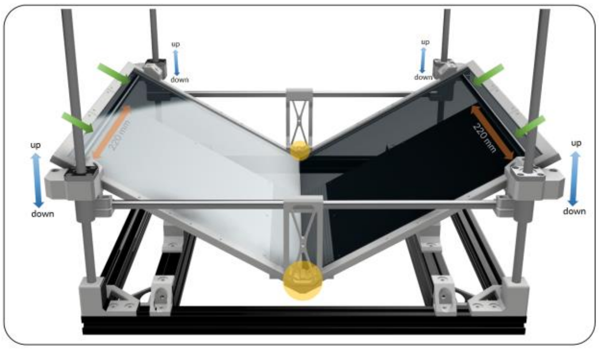

Figure 6. Book scanner mechanism.

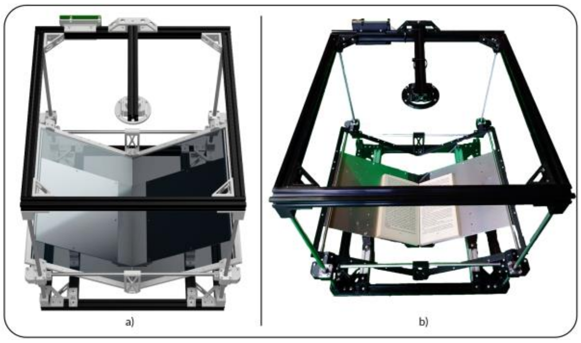

Figure 7. Final scanner structure: a) Design; b) Reality.

The frame of the two glass plates is supported in 6 points: at the extremities, by the four linear bearings, and in the middle part, as shown in Fig. 6 by the yellow dots. Further, the stabilization of the frame was achieved by adding four horizontal axes, which also have a secondary role: supporting the frame in the default position on the two aluminum plates of the “V” shape stand. The green arrows indicate the two axes used for support. The horizontal axis at the front is also used to manipulate the mechanism vertically. The movement of the glass plates along the four linear vertical axes of 8 mm is made manually.

Fig. 7 shows the final structure of the scanning mechanism. Besides the aluminum plates, glass plates, linear bearings, 8 mm axes, and screws, all the additional pieces were designed in Fusion 360, and 3D printed.

B. Refreshable Braille Display Mechanism

The type of display influences the cost of the final product. Thus, optimizing the mechanism was considered to have a minimal cost for this project. The mechanism will consist of stepper motors, octagonal discs, and a microcontroller.

The Marburg Medium Braille Font standard was used in this project, similar to ECMA Euro Braille, International Building Standard, and American Standard Sign [43]. The specifications of the Braille cell and display are:

a) Point diameter: 1.55 mm;

b) Point height: 0.55 mm;

c) The distance between the centers of adjacent vertical and horizontal points: 2.5 mm;

d) The horizontal distance between the centers of points of adjacent cells: 7 mm;

e) The distance between vertical lines and the nearest points’ centers: 27 mm. The standard value is 10 mm. However, as it was also with Cannute 360 [5], a compromise had to be made. The distance had to be increased due to the shape of the octagonal disks.

The octagonal disks have a thickness of 3.3 mm, a side length of 10 mm, and a distance between the parallel sides of 24.141 mm. Each side of the disk has a combination of three points, as shown in Fig. 8. Two octagonal disks make up a Braille cell.

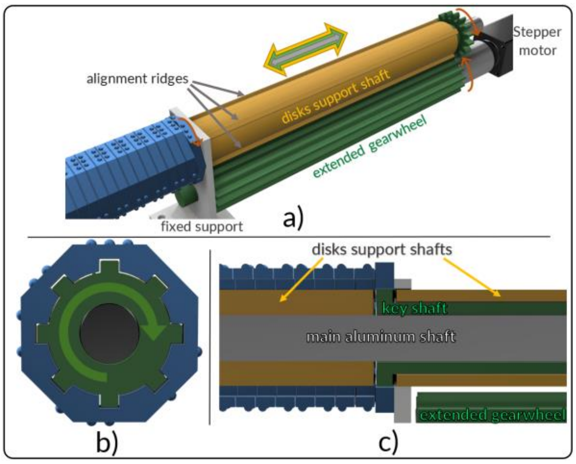

Figure 8. Octagonal disk rotating mechanism: a) Top view; b) Front view; c) Cross section.

Fig. 8 shows the mechanism for rotating the octagonal disks. Each disk will be rotated sequentially by a shaft, colored green in the image, that has at the end a key. After the shaft has rotated the octagonal disk, the perforated pattern of the disks will allow the key to shifting horizontally regardless of the Braille combination. The yellow and grey rounded middle shafts will only move horizontally and will not rotate. They are the primary support for the key shaft and the octagonal disks. Further, the disks support shafts have four alignment ridges for improved accuracy.

The key shaft is rotated by an extended gearwheel connected to a Nema 8 stepper motor. The gear ratio is 1:1, which means that to rotate a disk from one side to the next one, the shaft of the motor will turn 45°.

There are 80 octagonal disks for a row, which forms 40 Braille cells. A row has two key shafts on both sides. Hence, two octagonal disks will be rotated at once. The key shafts can reach only the middle of the row. Thus, the left motor will rotate disks 1–40, and the right motor will rotate disks 41–80. For example, when the left motor is rotating disk 40, the right motor will rotate disk 80. The support shaft is moved to the left until the disk pair reaches 1 and 41. The implemented algorithm will rotate only the necessary disks and can start from any side.

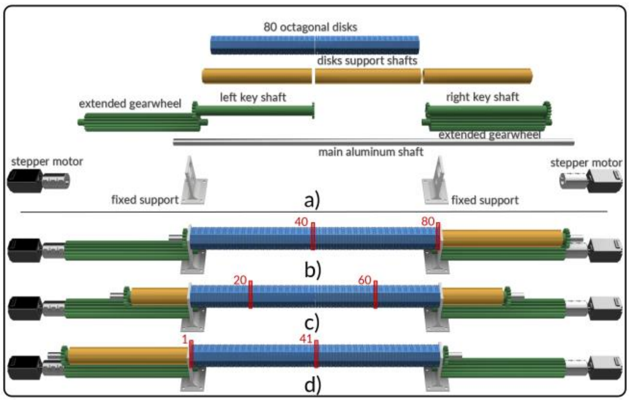

Figure 9. Row mechanism: a) Expanded components; b) Disk pair [40,80]; c) Disk pair [20,60]; d) Disk pair [1,41].

The middle disks support shaft is fixed to the main aluminum shaft, while the external ones are not. They allow the rotation of the key shafts. Nonetheless, the fixed supports will not allow the rotation of the disks support shafts, only the translation, as shown in Fig. 9.

Increasing the number of rows means increasing the number of stepper motors. Each row will have only two motors for rotating the key shafts. The two Nema 11 motors are used to move all the support shafts simultaneously. The shafts that can move only horizontally are interconnected, as shown in Fig. 10.

The rod that moves the shafts is connected to the Nema 11 motors timing belts on each side for smooth movement. Regardless of the number of rows, the refresh rate time will be the same. The Nema 8 stepper motors will be turned at once by using multithreading. Thus, there is a substantial improvement in the refresh rate.

The Braille display has 10 rows. Thus, the length of a character line is 280 mm, and the length from the first line to the last line is 247 mm.

The Arduino Due microcontroller will receive the Braille characters from the Raspberry Pi via serial communication. The Arduino algorithm will control the motors using stepper motor drivers connected to a power supply. The hardware is enclosed under the scanner and the display mechanism.

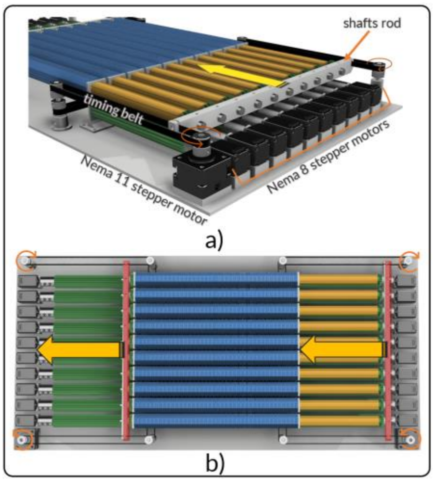

Figure 10. Multiple rows mechanism: a) Side view; b) Top view.

Fig. 11 shows the enclosure of this refreshable Braille display that contains 10 buttons, after the Perkins-style Braille keyboard model, for controlling the displayed text. The buttons can be used to move to the next or the previous section of the page, select a digital book, select a specific page of that book, enable the audio, change the volume or search for an unknown word in the dictionary. Each row is notated from 1 to 10 in Braille so the user can easily find it. There are also three buttons for starting the device, blocking the Braille keyboard, and changing the book mode: digital format or physical format.

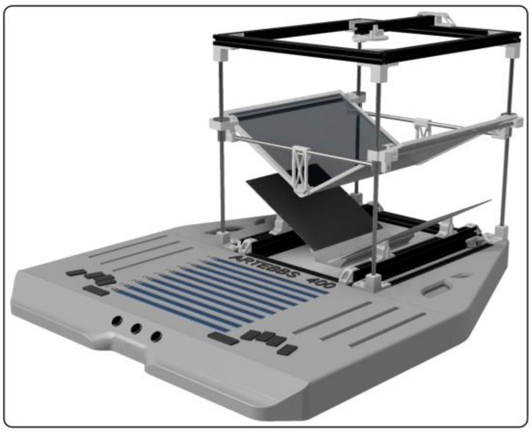

Figure 11. Advanced real time embedded book braille system — design concept.

V. Conclusion

The main objective of the presented project was to create a system with the help of which blind people will be able to read any book in physical or electronic format using the Braille language.

The paper showed a complete solution of a refreshable Braille system that can be implemented. From the algorithms used for image processing, the algorithms for converting a text into Braille or audio, the scanner, and Braille display design, to the features for improving the user experience, such as the online dictionary, multiple languages selection, and page navigation.

By creating this system, we want the experience of blind people to be complete. This system can be used in a personal environment or a library. Thus, it will be possible to reduce the costs of buying special or normal books, which can be borrowed from the library or read there.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Vasile Dan conducted the research, designed the diagrams, and wrote the paper. Ioan Nașcu and Silviu Folea supervised the work development. All authors provided critical feedback and approved the final version.

REFERENCES

[1] H. Kajimoto, Y. Kanno, and S. Tachi. (2006). Forehead electro-tactile display for vision substitution. *Computer Science*. 2983156. [Online]. Available: https://www.semanticscholar.org/paper/Forehead-Electro-tactile-Display-for-Vision-Kajimoto/4d61efe8fd515c55fb6f8c3f5c5bf8d48cffc2b4

[2] LUMEN. Empowering the blind. [Online]. Available: https://www.dotlumen.com/

[3] Graphiti® — A Breakthrough in Non-Visual Access to All Forms of Graphical Information. Orbit Research. [Online]. Available: https://www.orbitresearch.com/product/graphiti/

[4] A. Raghunandan and M. Anuradha, "The methods used in text to braille conversion and vice versa," *International Journal of Innovative Research in Computer and Communication Engineering*, 2017.

[5] The future of Braille has more than one line. Bristol Braille Technology CIC. [Online]. Available: http://www.bristolbraille.co.uk/index.htm

[6] Canute 360 Technical Specification. (2021). [Online]. Available: https://github.com/Bristol-Braille/canute-ui/wiki/Canute-360-Technical-Specification

[7] P. Manwatkar and K. Singh, "A technical review on text recognition from images," in *Proc. International Conference on Intelligent Systems and Control*, 2015.

[8] M. Wei-Haas. (2017). This device translates text to braille in real time. [Online]. Available: https://www.smithsonianmag.com/innovation/device-translates-text-braille-real-time-180963171/

[9] W. L. Zagler, M. Treml, D. Busse, M. Busboom, and I. Deák, "BrailleRing: The shortest long braille-display in the world — A review of the state-of-the-art and a new approach," in *Proc. International Conference on Computers Helping People with Special Needs*, 2018, pp. 313–321.

[10] Blitab: Android tablet with 14 row braille display. [Online]. Available: https://www.perkinselearning.org/technology/posts/blitab-android-tablet-14-row-braille-display

[11] BLITAB — Feelings get visible. [Online]. Available: http://www.blitab.com/#project

[12] A. Russomanno, S. O'Modhrain, R. B. Gillespie, and M. W. M. Rodger, "Refreshing refreshable braille displays," *IEEE Transactions on Haptics*, vol. 8, no. 3, pp. 287–297, 2015.

[13] S. Hossain, A. A. Raied, A. Rahman, Z. R. Abdullah, D. Adhikary, A. R. Khan, A. Bhattacharjee, C. Shahnaz, and S. A. Fattah, "Text to braille scanner with ultra low cost refreshable braille display," in *Proc. 2018 IEEE Global Humanitarian Technology Conference (GHTC)*, pp. 1–6, 2018.

[14] R. E. Aryan and S. Doshi, "Refreshable braille module using cam actuated mechanism," *IOP Conference Series: Materials Science and Engineering*, vol. 1123, 012028, 2021.

[15] D. Karastoyanov and N. Stoimenov, "Access of visually impaired people to cultural and historical heritage using braille visualization," *International Journal of Systems Applications, Engineering & Development*, vol. 13, pp. 10–14, 2019.

[16] A. Fletcher, S. Downs, W. Frost, D. Clarke, K. O'Grady, and A. Hirohata, "Magnetic braille using ferrofluids," *Journal of Physics D: Applied Physics*, vol. 54, 2021.

[17] K. Sharavana, E. Sivasubramanian, N. Arpitha, M. M. Gowda, G. M. Naik, and Vijayalakshmi, "Real-time text to braille and audio convertor," *International Research Journal of Engineering and Technology*, vol. 7, no. 8, pp. 2869–2878, 2020.

[18] S. Kumari, A. Akole, P. Angnani, Y. Bhamare, and Z. Naikwadi, "Enhanced braille display use of OCR and solenoid to improve text to braille conversion," in *Proc. 2020 International Conference for Emerging Technology (INCET)*, pp. 1–5, 2020.

[19] S. Golonka, R. Kothare, and R. Lam. Braillebook. [Online]. Available: http://www.ecs.umass.edu/ece/sdp/sdp16/team15/sdp16.my-free.website/index.html

[20] N. Stamatopoulos, B. Gatos, and T. Georgiou, "Page frame detection for double page document images," *ACM International Conference Proceeding Series*, pp. 401–408, 2010.

[21] F. Shafait, J. Beusekom, D. Keysers, and T. Breuel, "Document cleanup using page frame detection," *IJDAR*, vol. 11, pp. 81–96, 2008.

[22] R. Shashidhar, A. S. Manjunath, R. Santhosh Kumar, M. Roopa, and S. B. Puneeth, "Vehicle number plate detection and recognition using YOLO-V3 and OCR method," in *Proc. 2021 IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC)*, 2021, pp. 1–5. doi: 10.1109/ICMNWC52512.2021.9688407

[23] M. S. Onim, H. Nyeem, K. Roy, M. Hasan, A. Ishmam, M. Akif, and T. Ovi, "BLPnet: A new DNN model and Bengali OCR engine for automatic license plate recognition," *Array*, vol. 15, 100244, 2022.

[24] J. Cai, E. Sun, and Z. Chen, "OCR service platform based on OpenCV," *Journal of Physics: Conference Series*, vol. 1883, 012043, 2021.

[25] J. Antonio, A. R. Putra, H. Abdurrohman, M. S. Tsalasa Putra, and A. Chandra, "A survey on scanned receipts OCR and information extraction," Preprint, 2022, doi: 10.13140/RG.2.2.24735.84643

[26] P. Zhang, Y. Xu, Z. Cheng, S. Pu, J. Lu, L. Qiao, Y. Niu, and F. Wu, "TRIE: End-to-end text reading and information extraction for document understanding," arXiv pre-print, 2020, doi: 10.48550/arXiv.2005.13118

[27] A. Jiju, S. Tuscano, and C. Badgujar, "OCR text extraction," *International Journal of Engineering and Management Research*, pp. 83–86, 2021.

[28] G. Nath, R. Aggarwal, V. Pant, K. Golla, A. Sekseria, and M. Kumar, "Isolated OCR for handwritten forms: An application in the education domain," in *Proc. 8th International Conference of Business Analytics and Intelligence*, 2022.

[29] W. Zhu, N. Sokhandan, G. Yang, S. Martin, and S. Sathyanarayana, "DocBed: A multi-stage OCR solution for documents with complex layouts," arXiv pre-print, 2022, doi: 10.48550/arXiv.2202.01414

[30] D. Phamtoan, D. Nguyen, A. Le, M. Phan, and P. Kromer, "Candidate word generation for OCR errors using optimization algorithm," in *Proc. AIP Conference*, vol. 2406, 2021.

[31] K. Dhandapani, V. P. Arumbu, K. Surendhirababu, K. Selvakumar, P. Divya, P. Suhasini, and R. Palanisamy, "Sophisticated and modernized library running system with OCR algorithm using IoT," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, issue 3, pp. 1680–1691, 2021.

[32] I. Mathumitha, B. Vinodhini, S. Jayaprakash, D. R. Malini, and K. Sangeetha, "OCR for medical data processing," *International Research Journal on Advanced Science Hub*, vol. 3, pp. 78–81, 2021.

[33] T. Hegghammer, "OCR with tesseract, Amazon textract, and Google document AI: A benchmarking experiment," *Journal of Computational Social Science*, vol. 5, pp. 861–882, 2021.

[34] Pybraille 1.0.0. (2020). A library to convert text to 6-dot braille pattern (Grade 1). [Online]. Available: https://pypi.org/project/pybraille/

[35] X. Tan, T. Qin, F. Soong, and T.-Y. Liu, "A survey on neural speech synthesis," arXiv pre-print, 2021, doi: 10.48550/arXiv.2106.15561

[36] Beautiful soup documentation. [Online]. Available: https://www.crummy.com/software/BeautifulSoup/bs4/doc/
[37] CZUR-et16plus. [Online]. Available: https://www.czur.com/product/et16plus
[38] BFS-Auto: High Speed & High Definition Book Scanner. [Online]. Available: http://ishikawa-vision.org/vision/BFS-Auto
[39] S. Noguchi, M. Yamada, Y. Watanabe, and M. Ishikawa, "Real-time 3D page tracking and book status recognition for high-speed book digitization based on adaptive capturing," in *Proc. IEEE Winter Conference on Applications of Computer Vision*, pp. 137–144, 2014.
[40] ScanRobot® 2.0 MDS. [Online]. Available: https://expmstore.com/products/scanrobot%C2%AE-2-0-mds
[41] Mango product design. [Online]. Available: https://studiomango.nl/portfolio/revyve/
[42] Archivist Quill Guide. [Online]. Available: http://tenrec.builders/quill/guide
[43] Pharmaceutical Braille. [Online]. Available: https://www.pharmabraille.com/pharmaceutical-braille/marburg-medium-font-standard/