Developing of Text Translation and Display Devices via Braille Module

Dong-Hun Yoo[1] and Soo-Whang Baek[2+]
[1, 2]Sangmyung University, Dept. of Human Intelligence and Robot Eng., 31 Sangmyungdae-gil, Dongnam-gu, Cheonan-si, Chungcheongnam-do, 31066, Korea
[1]dbehdgjs98@gmail.com, [2+]swbaek@smu.ac.kr(corresponding author)

Article Info
Page Number: 537 – 547
Publication Issue:
Vol. 71 No. 3 (2022)

Article History
Article Received: *12 January 2022*
Revised: *25 February 2022*
Accepted: *20 April 2022*
Publication: *09 June 2022*

Abstract
Most previous research on braille outputs has focused on the simplification and ease of use of electronic products for the visually impaired. A majority of the existing devices are connected to smart devices, with the aim of obtaining voice outputs or braille information. Instead of such a voice output-based device, this paper proposes a device that can actually print braille to ensure that visually impaired people can read documents. In particular, this study deals with the implementation of a display device and braille module for the purpose of text translation for the visually impaired. To this end, the Raspberry Pi camera module is used to store actual documents, instead of electronic documents. Images captured through this camera module are then stored in the device. Subsequently, characters are extracted from these captured images through the pytesseract library, and the extracted characters are converted to braille. The translated braille information is processed to output the correct information to the braille cell. The braille output device comprises 20 braille cells, based on which a user may receive 20 characters at a time. After the user understands the information from all 20 braille cells, the user receives information related to the next step through the next button. When the user presses the next button, the next word or line of characters is output. The proposed device is easily portable, and the miniaturized device is manufactured via 3D printing. Furthermore, the braille cells are designed to help users recognize the printed braille by hand, using the (Acrylonitrile-butadiene-styrene) ABS resin material instead of rubber.

Keywords: Braille, 3D print, OCR, Raspberry Pi, Portable, Efficient

[+] Corresponding author. Tel.: +82-41-550-5543; fax: +82-41-550-5549.
E-mail address: swbaek@smu.ac.kr.

1. Introduction

Braille is a "character" that is used by the visually impaired. Despite the advantages, only a few braille books have been published in recent years. Despite the advances in electronic devices and e-books, it is clear that such media are not ideal for the visually impaired. The visually impaired often use a braille translator to translate letters into braille. However, this translator has a few disadvantages specifically in terms of time and economy. These devices do not cater to the purpose of learning or education for the visually impaired. With the recent advances in IoT technologies, many devices have been developed (DePountis et al., 2015; Gupta et al., 2016; Rajbongshi et al., 2020; Russomanno et al., 2015). Modern braille translation devices deliver text information to the visually impaired in real time. The braille translation devices are convenient for visually impaired, and they help acquire information by voice output or braille. However, these are difficult to buy and are expensive. In addition, most of the devices are not suitable for translation of general books (Chakraborty and Mallik, 2013; Gayathri et al, 2021; Kurlekar et al., 2020; Eldem and Başçiftçi, 2015). This study was aimed at improving the braille output characteristics for the visually impaired. Optical character recognition (OCR) was mainly used to extract characters from images. OCR acquires printed documents as images and convert them into machine-readable characters. Many OCR technologies are being used in diverse applications such as license plate recognition, delivery, and translator (Agbemenu et al., 2018). Several studies have also employed OCR technology to help the visually impaired (Hegghammer, 2021; Patel et al., 2012; Sahu et al., 2017). Although computers are essential to translate text into braille, they cannot be carried everywhere; hence, many studies have adopted Raspberry Pi, which is a microcomputer, for device design. Raspberry Pi was developed for educational projects that require compact, low-cost computers. Several studies have used Raspberry Pi as a microcomputer for various applications (Ryu et al., 2021). Braille uses six points and is divided into two cases per point. The composition of these six points includes 64 cases, and 63 characters can be expressed, except for blank spaces. Braille is essential for the visually impaired for several reasons, including the task of obtaining information easily. Many existing studies have expressed braille with soft materials (Lee and Lucaszyn, 2005; Wu et al., 2012).

This study realizes braille output through braille cells. By using hard materials such as ABS resin in 3D printing, we intend to improve the ability to recognize accurate braille information for the visually impaired. Hard materials can be easily felt, as compared to soft materials; thus, they can provide more accurate information to users. Furthermore, portability was improved by using Raspberry Pi as the reading device. The device outputs 20 characters at a time, and the user can move on to the next word through a next button. The use of the device may help replicate the feeling of reading a braille book. Moreover, a camera module of Raspberry Pi was used to photograph documents. The text was then extracted through OCR from the image of the document. The extracted text is translated into braille information on the Raspberry Pi board. The translated braille information is output to the braille output device. In this work, an experiment was conducted using a device composed of a main body and braille cells. The text output accuracy of OCR was confirmed by varying the conditions for photographing the document. The text extracted from the images was converted into braille information, which was then converted and output through the braille output device. In this manner, a device for photographing printed documents and outputting braille was realized, and the validity of the design was confirmed.

2. Device and braille translation

2.1 Braille cell

The braille cells used in this study were designed based on the “low-cost braille electronic display” of the “MOLBED” project. This device takes the form of an actuator and recognizes electric signals while moving linearly. Braille requires six points to express one character. Thus, one cell consists of six braille pins, and each braille pin consists of two parts. The first part is a magnetholder, which includes a magnet in the actuator. The upper part of the magnetholder is recognized by the user as a braille dot and is composed of resin, which can provide the user with a solid feeling. This implies that accurate information can be delivered to the user. For the magnetholder, two cylindrical magnets with a circumference and height of 2 mm are used. The second part is the body wherein the magnetholder is inserted, and the coil is wound in the actuator. When a current flows through this coil, the magnetholder moves according to the magnetic field. Furthermore, the magnetholder moves linearly under the effect of the magnetic field. These two parts are combined into a single pin; in this manner, six braille pins are assembled with specially designed PCB plates to operate as the braille cells. The PCB plate features a structure wherein the braille pins are arranged along two rows. It is also configured to include a header pin, in order to facilitate control over the movement of the braille pins using Raspberry Pi. The header pin is connected to the breadboard. These braille cells, consisting of the pins, PCBs, and magnetholders, are illustrated in Fig. 1.

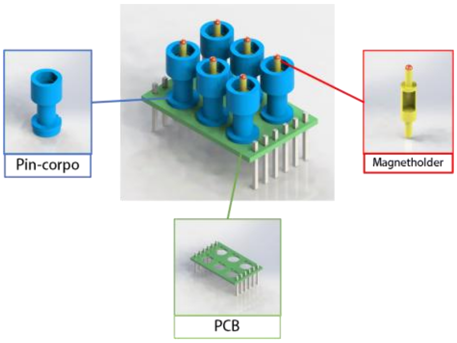

Fig. 1: Configuration of Braille cell

2.2 Design of device

The braille device consists of a main body, braille cells, and buttons, as shown in Fig. 2. The main body was manufactured using 3D printing, and ABS resin was used for the purpose of miniaturization and weight reduction. The main body features a width of 200 mm, length of 70 mm, and height of 40 mm; it is implemented such that its weight is less than or equal to 200 g. The braille cell comprises braille pins and the PCBs, with a total of 20 braille cells arranged along two rows. It also consists of two buttons that can be used to move to the previous/next page. Thus, users can receive a total of 20 braille information datapoints at a time. The device is designed such that it expresses braille through a total of 120 holes, thereby allowing for the acquisition of braille information. After completing a page, the remaining pages of the document can be easily read using the page turning buttons located on either side of the braille device. Thus, the user can recognize the line changes in the document. The overall device is configured such that it replicates the feeling of reading a braille book, achieved through the braille cells and buttons of the device.

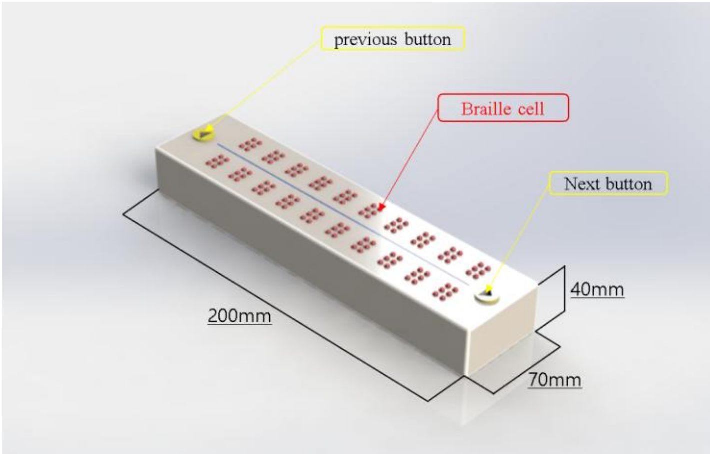

Fig. 2: Design of device

2.3 Translation text into braille

For the purpose of reading a book, it is necessary to extract characters from images of the book. As a result, the OCR library was used. The translation hardware devices for braille translation were designed using OCR and Raspberry Pi, a microcomputer for embedded systems. The Raspberry Pi camera module is used for information interworking between Raspberry Pi and OCR. Through the camera module, text-based documents such as books are photographed, and the images are stored. The characters are extracted using the Tesseract OCR library, among the OCR libraries. The Tesseract OCR library is characterized by high accuracy in extracting English alphabets. Therefore, programming was conducted using the Tesseract OCR library. Fig. 3 shows the results of extracting English alphabet text using the pytesseract OCR library. The extraction accuracy of the English alphabet text displayed in the shell window was approximately 85%. The typos or missing characters in the shell window are the parts that are covered by shadows when taking pictures. Therefore, it is important to photograph a clean image for accurate text extraction.

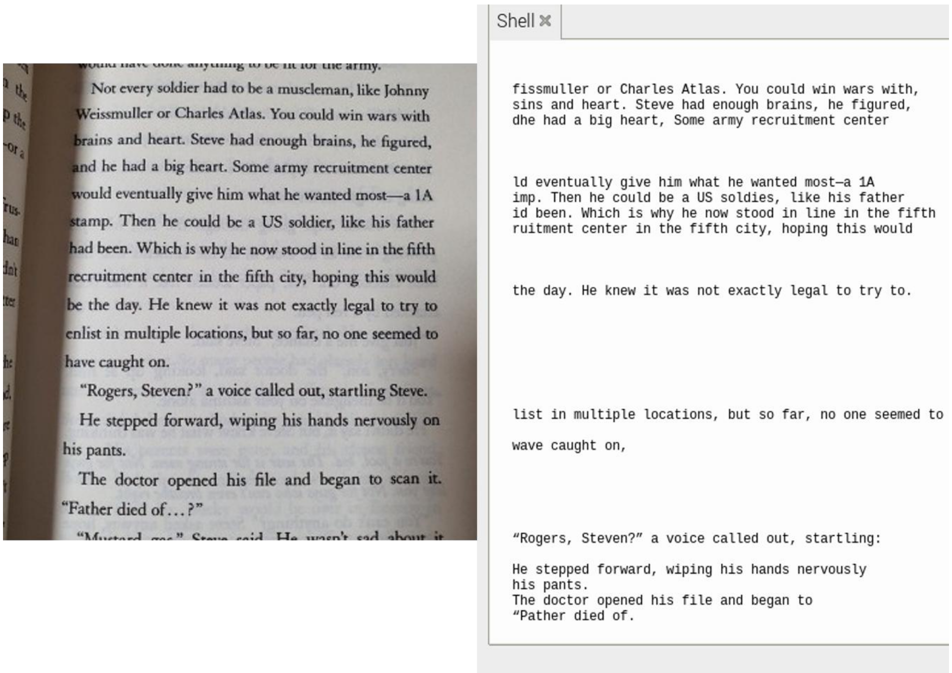

Fig. 3: Result of English alphabet text extraction using pytesseract OCR

The extracted character is converted into braille information through a "code-table" that is created in advance. The created code-table matches the English alphabet, with a six-digit number consisting of 0 and 1 in the form of ON and OFF. Table 1 shows the conversion of the English alphabets into braille information. Finally, the device receives the converted braille information and outputs it as braille.

Table 1: Text into braille code-table

Alphabet	Braille code	Alphabet	Braille code
A	100000	**N**	110110
B	101000	**O**	100110
C	110000	**P**	111010
D	110100	**Q**	111110
E	100100	**R**	101110
F	111000	**S**	011010
G	111100	**T**	011110
H	101100	**U**	100011
I	011100	**V**	101011
J	100010	**W**	011101
K	100010	**X**	110011
L	101010	**Y**	110111
M	110010	**Z**	100111

When the next button is pressed, 20 pieces of the subsequent text information are output sequentially. This device has the ability to continuously present the following information, according to the user's manipulation via a continuous next button.

2.4 Algorithm

The components of the hardware of the braille device are shown in Fig. 4. Using the camera module of the rRaspberry pPi, the captured image is stored on the rRaspberry pPi board. Various operations for braille translation are carried out within the raspberry pithis board. OCR is then applied to the stored image. The extracted text information is converted into braille information and output as braille cells. This Iinformation is then transmitted given to 20 braille cells to express multiple words

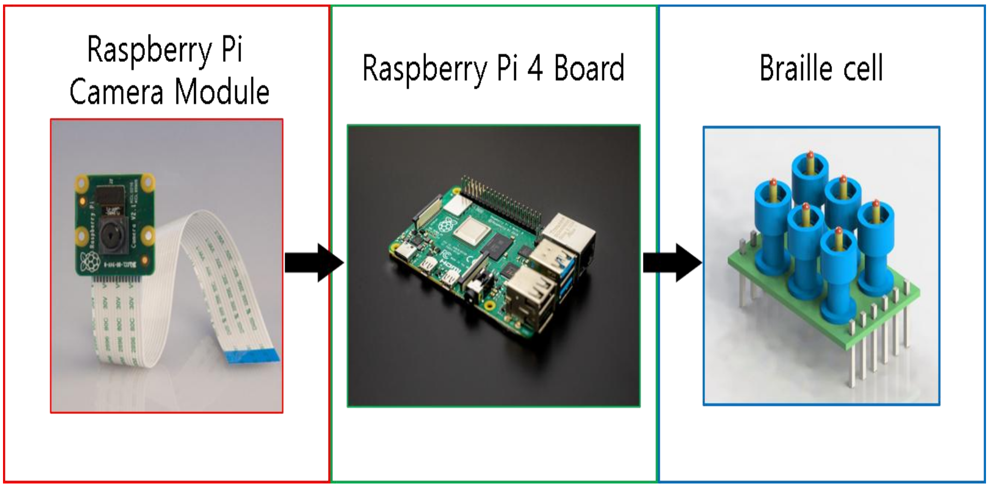

Fig. 4: Hardware components

Figure 5 shows the algorithm used for the device. The camera module captures an image to read the text of the document. If text is not recognized in the photographed image, the photographing process is restarted. Once character recognition is completed, the characters are translated into braille information. The translated braille information is then transferred to the device. The device finally outputs the received 20 braille information datapoints, and the user subsequently receives this braille information through their fingertips. Once users have completed reading the received information, they can press the next button. In this case, instead of the previously output text information, the text corresponding to the next word and line is recognized and output.

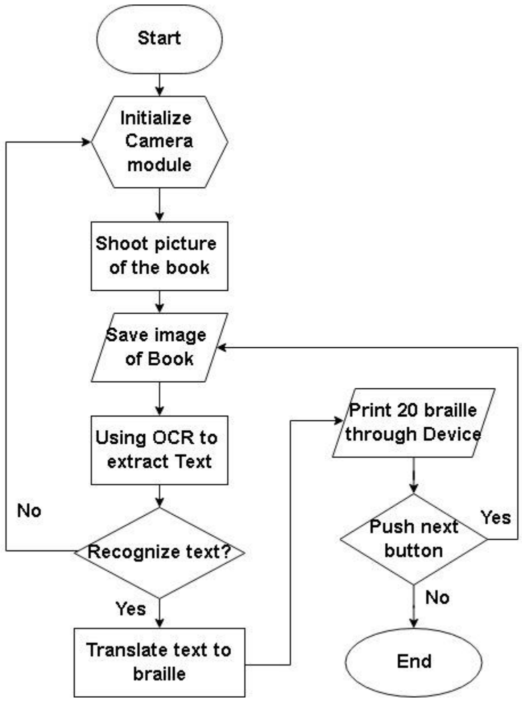

Fig. 5: Operation algorithm

3. Prototype production and experimental

A prototype of the braille device was manufactured in this study, and the validity and feasibility of this proposed device was verified through experiments. Figure 6 shows the process of reading text using Raspberry Pi camera module. First, the text of the book is photographed using the Raspberry Pi camera module. The device is designed such that it can film through the top side of a book; the advantage being that the user can continue to read while filming. This method helps acquire the information in real time quickly. Through a combination of the camera module and the Raspberry Pi board, the braille device can constantly photograph image information, and then analyze and extract text. In order to increase the recognition rate of pytesseract OCR and to test the accuracy of text extraction of braille devices, four experimental conditions that correspond to Clean, Both Sides, Curved, and With Shadow were selected as follows. Figure 7 shows the results of the pytesseract OCR under four conditions. The first image is the result of the “Clean” condition. The images were taken using a white base in the vertical direction of the print. The second image is the result of the “both sides” conditions. The left and right pages of the book were captured in a single image. In the third condition, i.e., “Curved”, the image was captured by bending the book at a certain angle. The fourth image is captured under the “With Shadow” condition. The image was captured in a dark place.

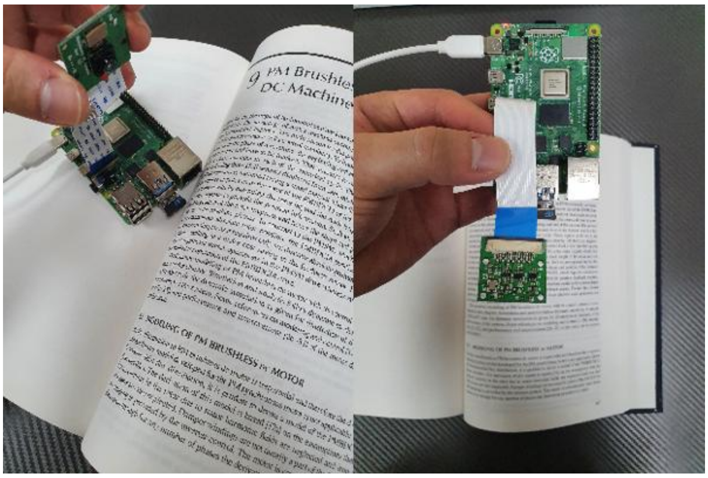

Fig. 6: Using raspberry pi camera module

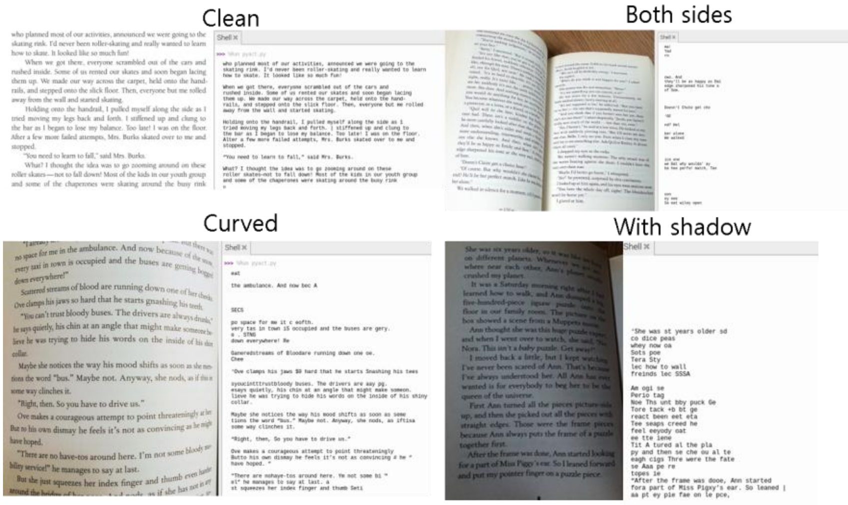

Fig. 7: OCR results in 4 conditions

For all four experimental conditions, the pytesseract OCR showed different accuracies. Table 2 shows the pytesseract OCR accuracy according to the four experimental conditions. In general, the character extraction significantly reduces the accuracy of character extraction when photographed under conditions other than the clean conditions. In particular, under the condition of both sides of the book, the recognized text was not clear. In addition, only some text is recognized under the Curved and With Shadow conditions. However, when the end of the document was photographed under severe dark conditions, the device did not recognize many characters.

Table 2: Pytesseract OCR accuracy in 4 conditions

Situation	Accuracy
Clean	99%
Both sides	3%
Curved	34%
With shadow	31%

The text acquired through pytesseract OCR is converted into braille information. The braille device is driven with the character information that is obtained under the Clean condition. The acquired character is "whoplanned mostofour." It is converted into braille information through code-table. Then, the converted braille information is output through the device. Figure 8 shows the results of the device that output braille information.

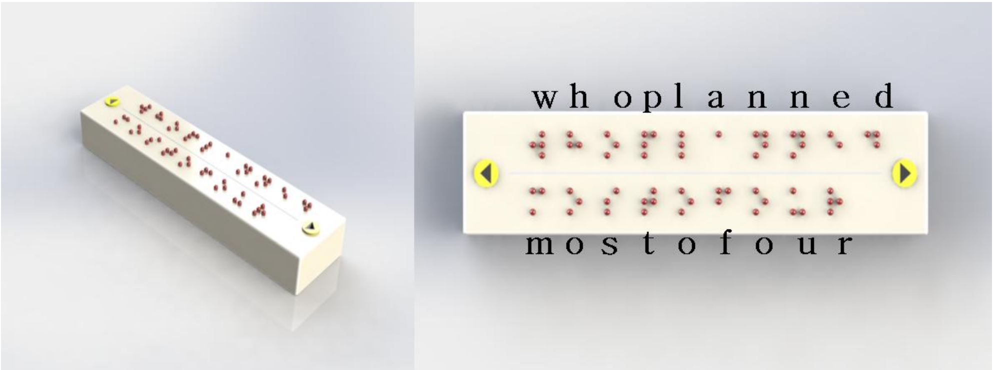

Fig. 8: Device outputs Braille

4. Conclusion

In this paper, a braille output device that incorporated a camera module was proposed based on Raspberry Pi. For 3D printing, the ABS resin material was used, and a Raspberry Pi was incorporated, with regard to price competitiveness and portability, as compared to existing studies. In particular, accurate information delivery to users is possible by using a hard material such as ABS resin in braille cells to ensure fast braille output. The printed output was photographed through the Tesseract OCR library to check whether text information could be received, and the process of converting it into braille information was confirmed. In some cases, the text to sound (TTS) may be more useful for some users. However, in libraries or places where voice output is restricted, tactile senses need to be used instead of hearing. The use of earphones restricts the information through surrounding sounds. Thus, this study stands out from other technologies that are being studied based on TTS. When an image of the document was captured and translated, it showed a good recognition rate of 99% in clean conditions. When photographing the entire open two-page book, it is necessary to recognize the left and right pages separately. In the future, we plan to develop and optimize a braille device that can conveniently deliver more accurate information to the visually impaired by combining the Internet of Things and embedded systems. In particular, it is expected that improved text extraction results can be obtained by adding preprocessing such as image shadow removal and distortion flattening.

5. Acknowledgements

This study was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1F1A1061567).

6. References

1. Agbemenu, AS. Yankey, J. & Addo, EO. (2018). An automatic number plate recognition system using openCV and tesseract OCR engine. *International Journal of Computer Applications*, 180(43), 1-5. https://doi.org/10.5120/ijca2018917150
2. Chakraborty, P. & Mallik, A. (2013). An open source tesseract based tool for extracting text from images with application in braille translation for the visually impaired. *International Journal of Computer Applications*, 68(16), 26-32. https://doi.org/10.5120/11664-7254
3. DePountis, VM. Pogrund, RL. Griffin-Shirley, N. & Lan, WY. (2015). Technologies used in the study of advanced mathematics by students who are visually impaired in classrooms: teachers' perspectives. *Journal of Visual Impairment & Blindness*, 109(4), 265-278. https://doi.org/10.1177/0145482X1510900403
4. Eldem, A. & Başçiftçi, F. (2015). Electronic and Computer-Assisted Refreshable Braille Display Developed for Visually Impaired Individuals. *International Journal of Medical, Medicine and Health Sciences*, 8.0(1), 1-4. https://doi.org/10.5281/zenodo.1337930
5. Gayathri, S. Sujathakumari, BA. Murthy, NS. Balabhadra, M. Joshi, NS. & Sandesh, C. (2021). Digitized braille cell: a novel solution for visually impaired. *International Journal of Information Technology*, 13(3), 1201-1208. https://doi.org/10.1007/s41870-020-00506-9
6. Gupta, R. Singh, P K. & Bhanot, S. (2016). Design and implementation of Arduino based refreshable braille display controller. *Indian Journal of Science and Technology*, 9(33), 1-6. https://doi.org/10.17485/ijst/2016/v9i33/99593
7. Hegghammer, T. (2021). OCR with Tesseract, Amazon Textract, and Google Document AI: a benchmarking experiment. *Journal of Computational Social Science*, 1-22. https://doi.org/10.1007/s42001-021-00149-1
8. Ryoo, HG. Kim, DH. & Ryoo, KK. (2021). Apartment notice voice guidance service using TTS and raspberry pi. *International Next-generation Convergence technology Association*, 5(6), 986-993. https://doi.org/10.33097/JNCTA.2021.05.06.986
9. Kurlekar, S. Deshpande, OA. Kamble, AV. Omanna, AA. & Patil, DB. (2020). Reading device for blind people using python OCR and GTTS. *International journal of Science and Engineering Applications*, 9(4), 49-52. https://doi.org/10.7753/ijsea0904.1003
10. Lee, JS. & Lucyszyn, S. (2005). A micromachined refreshable Braille cell. *Journal of Microelectromechanical Systems*, 14(4), 673-682. https://doi.org/10.1109/JMEMS.2005.845415
11. Patel, C. Patel, A. & Patel, D. (2012). Optical character recognition by open source OCR tool tesseract: a case study. *International Journal of Computer Applications*, 55(10), 50-56. https://doi.org/10.5120/8794-2784
12. Rajbongshi, A. Islam, I. Rahman, M. Majumder, A. Islam, E. & Biswas, AA. (2020). Bangla optical character recognition and text-to-speech conversion using raspberry pi. *International Journal of Advanced Computer Science and Applications*, 11(6), 274-278. https://doi.org/10.14569/ijacsa.2020.0110636
13. Russomanno, A. O'Modhrain, S. Gillespie, RB. & Rodger, MWM. (2015). Refreshing refreshable braille displays. *IEEE transactions on haptics*, 8(3), 287-297. https://doi.org/10.1109/TOH.2015.2423492

14. Sahu, N. & Sonkusare, M. (2017). A study on optical character recognition techniques. *International Journal of Computational Science, Information Technology and Control Engineering*, 4(1), 1-15. https://doi.org/10.5121/ijcsitce.2017.4101
15. Wu, X. Kim, S.-H. Zhu, H. Ji, C.-H. & Allen, MG. (2012). A refreshable braille cell based on pneumatic microbubble actuators. *Journal of Microelectromechanical Systems*, 21(4), 908-916. https://doi.org/10.1109/JMEMS.2012.2190043