Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 1, January 2024
https://doi.org/10.47709/cnahpc.v6i1.3386

Submitted : Dec, 29, 2024
Accepted : January 25, 2024
Published : January 26, 2024

Utilizing Convolutional Neural Network for Learning Web-Based Braille Letter Classification System

Ahmad Ridwan[1*], Yoan Purbolingga[2], Hanisah[3]
[1] Department of Informatics, Universitas AMIKOM Yogyakarta, Yogyakarta, Indonesia, 55281
[2] Department of Electrical Engineering, Institut Teknologi Bisnis Riau, Pekanbaru, Indonesia, 28292
[3] Master of Public Health, Universitas Sebelas Maret, Surakarta, Indonesia, 57126
[1]ahmadridwan@amikom.ac.id, [2]johanyurik@gmail.com, [3]hanisah@student.uns.ac.id

ABSTRACT

Braille letters are letters used by blind people to exchange and get information from written media. Because Braille letters are different from the usual alphabet, it requires the ability to read Braille letters where the role of teachers is needed to teach Braille letters to the blind. This paper aims to facilitate prospective teachers and people who want to learn braille letters. The system designed is a website that classifies braille letters using deep learning with the convolutional neural network (CNN) method with the activation functions used, namely ReLU and Softmax. In this research, the input is an image of braille letters with grayscale elements. The output of the data is a regular alphabet letter. Most of this research data consists of training and testing data, which is 2,722 pieces. The accuracy results obtained in the data training process using Max Pooling and epoch 30 for data is 92.15%, epoch 50 is 94.58%, and for training data with epoch 100 is 96.64%. The test results using the system produce an accuracy value of all braille letter image data of 92.30%. Furthermore, for better system development, it is recommended to use hyperparameter tuning to minimize classification uncertainty in braille letter images.

Keywords: Braille Letter; CNN; Image Data; ReLU; Softmax;

INTRODUCTION

Reading is one of the activities carried out to obtain information. For most people, reading is an easy activity to do. Every average person can read if they understand the alphabet (Assefa & Assefa, 2018; Murthy & Hanumanthappa, 2018). However, reading activities are challenging for some people, especially for blind people. People with visual disabilities are people who experience visual impairment. Therefore, blind people cannot use their vision as the primary channel to receive information in written form. Braille is a writing and reading system developed by Louis Braille in 1824. The plan was designed to help blind or visually impaired people read and write. Louis Braille was a blind Frenchman who wanted to create a more efficient way for blind people to communicate and learn (Ramiati, Aulia, Lifwarda, et al., 2020; Singh et al., 2015). As a teenager, he designed the braille system, utilizing ideas from the dotted writing system used in military communications. The braille system uses several combinations of dots arranged in a 2×3 grid placed in 2×3 cells. Each cell can contain up to 6 drops, set in a unique pattern. Combining these dots results in letter characters, numbers, punctuation marks, and mathematical symbols (Molina et al., 2016; Subur et al., 2015). The braille system has been widely used worldwide to exchange information among blind people. However, braille documents could be more beneficial for some people due to the lack of knowledge about braille letters, making it difficult to obtain information from existing braille documents (Hassan et al., 2019; Ovodov, 2021; Ramiati, Aulia, & Lifwarda, 2020). Reading and understanding braille letters takes a long time to learn. It is necessary to train the hands sensitivity to the raised dots and understand and memorize their combination in forming a letter. Braille classification is essential to comprehend braille letters because braille letters are different from the regular alphabet (Herlambang et al., 2021; Hossain et al., 2018; Kausar et al., 2021).

Several non-profit organizations conduct braille training to improve the quality and participation of the visually impaired in education and employment. Therefore, visually impaired people can pursue the highest education possible by providing the necessary specialized services and striving for them to function in society according to their interests and abilities (Bhatia et al., 2022; Mccarthy et al., 2016; Zhi et al., 2016). However, in its implementation, there are obstacles in teaching the introduction of braille letters, where braille teachers need a system capable of increasing competence quickly in understanding braille letters (Ramiati, Aulia, Lifwarda, et al., 2020). Therefore, prospective braille teachers first must be trained to recognize braille letters to make it easier for teaching staff to understand braille letters. The use of braille is not only used by blind people but also by blind teaching staff. The purpose of this research

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 1, January 2024
https://doi.org/10.47709/cnahpc.v6i1.3386

Submitted : Dec, 29, 2024
Accepted : January 25, 2024
Published : January 26, 2024

is to analyze how the Braille letter classification process uses the CNN method which can later be implemented in the Braille letter website application. The result is expected that the designed system can help teachers and people who want to learn braille easily understand braille letters.

LITERATURE REVIEW

The utilization of technology has made it possible to classify braille letters, namely using image processing with deep learning (Lecun et al., 1998; C. Li & Yan, 2021; T. Li et al., 2014). Deep learning is a part of artificial intelligence and machine learning, which automatically performs representations of data such as images, videos, or text without introducing code rules or human domain knowledge (Made et al., 2019). Some research on the application of artificial intelligence in supporting braille learning, such as in this study (T. Li et al., 2014), uses the Stacked Denoising Auto Encoder (SDAE) extraction feature with an accuracy rate of 86%. This research does not pay too much attention to pre-processing on the image, causing the accuracy results to be not too high. In another study (Smelyakov et al., 2018), using image data only amounted to 33 image data for the training process, and in the testing process, there was a total of 8 image data. The difference between this research and previous research is that it combines image processing techniques with deep learning methods, such as convolutional neural networks, using ReLu and softmax activation functions to calculate the probability of each target over all possible marks to recognize braille text in images.

METHOD

This research method focuses on a prototype approach consisting of case studies and literature, data collection, data processing, system development, and system testing. The system development to classify Braille letters is a web application built using Python programming. The specifications of the software built are digital image input, model training using the CNN method, and the Braille letter classification process from the Braille image inputted through the directory. The design of the classification system can be seen in Figure 1.

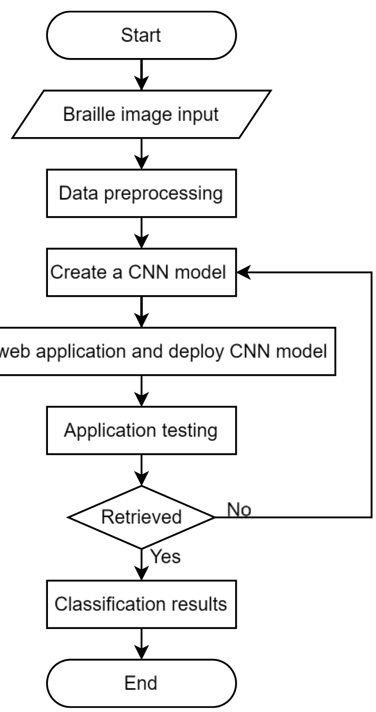

Fig. 1 Flowchart of the system design

Case Studies and Literature

The case study raised in this research is that not all blind people can read Braille letters because Braille letters are different from ordinary letters in the form of six raised dots, so it is necessary to have someone who will teach the Braille letters to blind people, where the teachers must first know the Braille letters one by one. However, there are obstacles where teachers need to learn Braille letters quickly so that it is easy to understand Braille letters. Therefore,

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 1, January 2024
https://doi.org/10.47709/cnahpc.v6i1.3386

Submitted : Dec, 29, 2024
Accepted : January 25, 2024
Published : January 26, 2024

a system is necessary that can improve competence in understanding Braille letters so that teachers can more easily understand Braille letters. Therefore, this research aims to help prospective teachers recognize Braille letters by classifying Braille letters using the CNN method.

Data Collection

The data used in this research is the image of Braille letters. The Braille letter image data used in this research is a Braille letter image with 26 letters of the alphabet. Each letter consists of 100 to 105 Braille letter image data, all in grayscale format. The data will be split in Google Colab and then separated as training data for 2,187 images and validation data for 535 images from 2,722 Braille image data.

Data Processing

Before building a model using the convolutional neural network method, the Braille letter image data is pre-processed, where the Braille letter image data is resized. After that, the image data is prepared to be used as training data and test data. At this stage, each image must have the same quality, format, and size to make the data processing process more accurate. Starting from uploading Braille letter image data obtained and then separating it into validation and test data. Then, the Braille letter image data goes through a convolution process to produce a Braille letter image classification before being implemented in the built web application.

System Development

The design of the Braille letter classification system is done by inputting Braille letter image data obtained from Kaggle. Then, data pre-processing is carried out, where there is a process of cropping and resizing the data obtained. After getting the appropriate data, the next step is to build a model using the CNN method. After receiving a CNN model that has good accuracy, continue by creating a web application and entering the CNN model that has been made into the web application. In the application testing stage, they use new Braille letter image data to determine whether the Braille letter classification results are appropriate. However, if they are reasonable, the system is complete. When the classification results are unsuitable, rebuild the CNN model with different epochs or optimizers.

Braille letter image data, totaling 2,722 image data, will be trained on Google Colab using CNN, then the result of the training is a model that will be stored in the directory as shown in Figure 2. Then, the CNN model will implement into the Braille letter classification web application built to classify Braille letters by users.

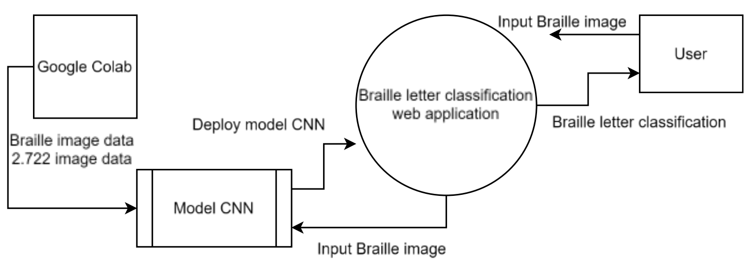

Fig. 2 Braille system flow

System Testing

After developing the application, it tested the application that has been built by trying the training data and data set to determine whether the application is following the design that has been made before.

RESULT

In this section, explain this research modeling and implementation stages, including the data modeling process with CNN, classification results using CNN, test results analysis, and interface using a website built with the Django framework and Python programming language. Here, users can classify Braille letter images, and each user can also see which contains the built application. Then, the user can also see the review and results where the classification results of the image that has been inputted.

Modelling Process CNN

Identifying Braille letters with the CNN method is carried out in several stages. Before implementation, the data is

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 1, January 2024
https://doi.org/10.47709/cnahpc.v6i1.3386

Submitted : Dec, 29, 2024
Accepted : January 25, 2024
Published : January 26, 2024

first separated into 26 folders based on the letters of the alphabet. After that, training is carried out on the data using the CNN method. Furthermore, data preparation begins with separating all data into 26 folders according to the letters of the alphabet and giving names that match the notes. The modelling dataset preparation process can be seen in Figure 3. Furthermore, the data is collected and trained on Google collabs with the CNN method so that later, it will get a model to reference the testing data. The first step is to import the required library. Then, take the Braille letter image data, where previously the Braille letter image data was uploaded to Google Drive to facilitate the retrieval of training data in Google collabs. In addition, Braille letter data is taken from Google Drive, and data separation is carried out to determine training data and validation data, where 2,187 image data are obtained as training data and 535 image data as validation data from a total of 2,722 Braille letter image data.

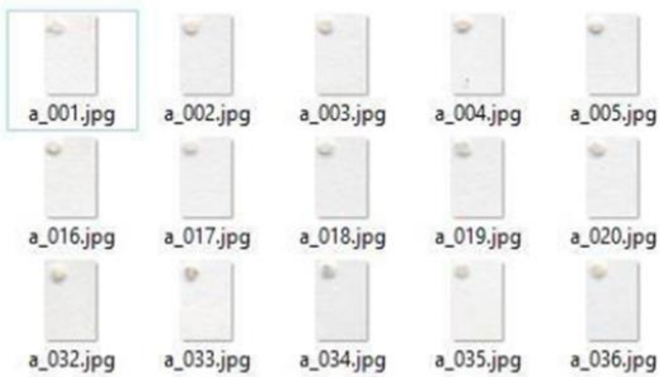

Fig. 3 Example of training data for alphabet letter A

Afterward, the data was trained using the CNN method. The data is processed with three convolution layers using the ReLU activation function in the feature extraction process and the classification process using the Softmax function. Then, the loss value and accuracy of the model generated from the data are calculated. The epoch value for testing data is 100, which is the repetition of the training process in one skip session 100 times to get the slightest possible error and produce the performance of the model made. Epoch 100 is used after experimenting with Epoch 30, Epoch 50, and Epoch 100.

CNN Classification Results

The test results are carried out after the model is implemented into the system. Before being implemented, the model is first selected with the best accuracy obtained from the training results. Three trainings were conducted using different epochs in each activity to get the best results. The accuracy result at epoch 30 is 92.15% with a loss of 0.8%, and the time required during the modeling process is 238 seconds. The accuracy and loss graphs for epoch 30 are shown in Figures 4 (a) and (b).

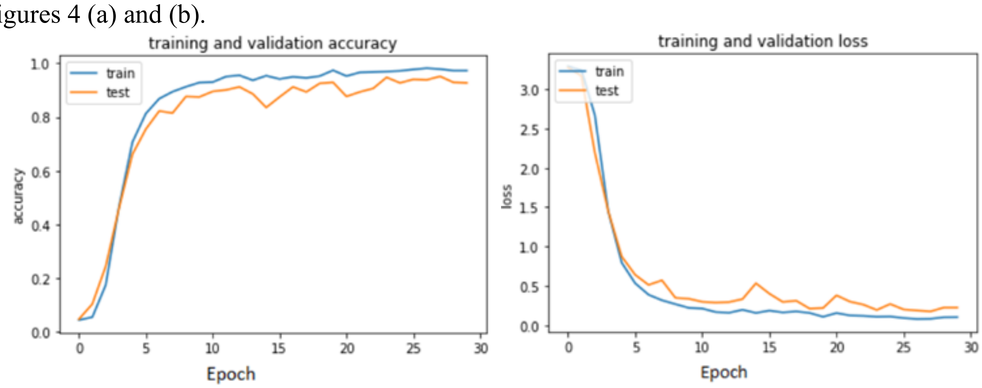

Fig. 4 CNN model training result graph epoch 30. (a) Accuracy graph. (b) Loss graph.

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 1, January 2024
https://doi.org/10.47709/cnahpc.v6i1.3386

Submitted : Dec, 29, 2024
Accepted : January 25, 2024
Published : January 26, 2024

In the experiment with epoch 50, the accuracy result obtained was 94.58% with a loss of 0.5%. The investigation results with epoch 50 took 362 seconds during the modelling process. The accuracy and loss graphs for the epoch 50 test are shown in Figures 5 (a) and (b).

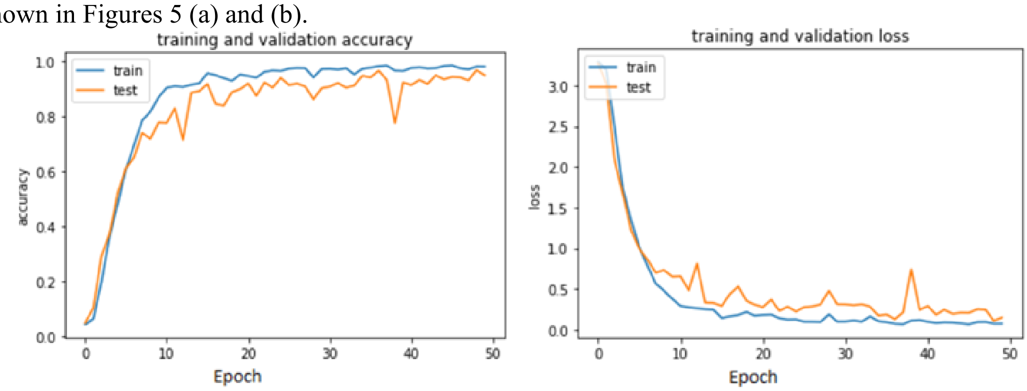

Fig. 5 CNN model training result graph epoch 50. (a) Accuracy graph. (b) Loss graph.

The accuracy result at epoch 100 is 96.07% with a loss of 0.4%, and the time required during the modelling process is 702 seconds. The accuracy and loss graphs on the test with epoch 50 are shown in Figure 6 (a) and (b).

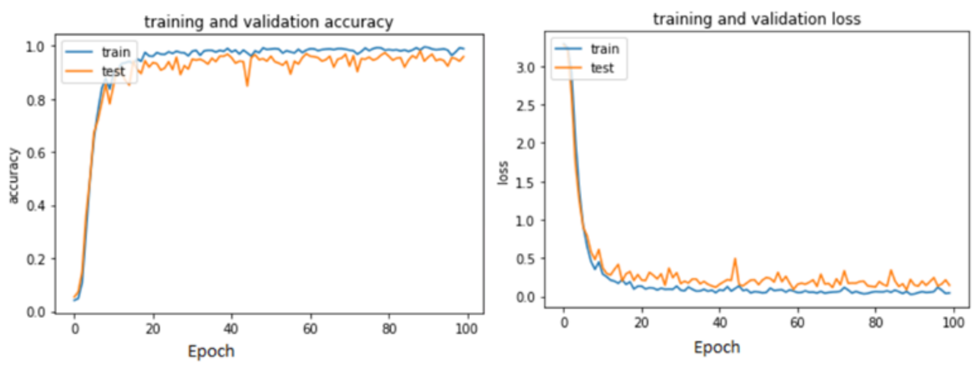

Fig. 6 CNN model training result graph epoch 100. (a) Accuracy graph. (b) Loss graph.

Table 1 shows the test results show that the model's accuracy with an epoch 30 experiment has an accuracy rate of 92.15% with a loss rate of 11.06% and requires a train time of 238 seconds or about 3.96 minutes. The epoch 50 experiment has an accuracy rate of 94.58% with a loss rate of 9.09% and requires a train time of 362 seconds or about 6.03 minutes. Then, the epoch 100 experiment has an accuracy rate of 96.07% and a loss rate of 4.52% with a train time of 702 seconds or about 11.7 minutes. Based on the experiments above, epoch 100 has the highest accuracy rate compared to experiments at epoch 30 and epoch 50, so the model with epoch 100 will be implemented into the system that has been created.

Table 1 Training result

Epoch	*Accuracy*	*Loss*	*Time (Sec)*
30	92.15 %	11.06 %	238
50	94.58 %	9.09 %	362
100	96.07 %	4.52 %	702

User Interface Implementation

This web-based Braille letter classification application uses the deep learning CNN method with Python programming language and Django framework. Figure 7 shows the display on the home page and the Classification input page on the web.

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 1, January 2024
https://doi.org/10.47709/cnahpc.v6i1.3386

Submitted : Dec, 29, 2024
Accepted : January 25, 2024
Published : January 26, 2024

Fig. 7 Front page and classification view

DISCUSSIONS

The experiment process uses test data 520 times where each letter of the alphabet totaling 26 letters has 20 Braille image data to be tested. The testing process is done by testing all test data individually. In Table 2, it can be seen that of the 20 Braille image data of letter A tested, there is one wrong test data. Where the error occurred in the 17th trial, the results showed the letter C. At the same time, the other problems showed correct results where the letter A was appropriately read. Figure 8 shows the accurate experiment, A1.jpg, while in Figure 9 the failed experiment is A17.jpg.

Table 2 Braille letter a classification test results

Input	*Output*	*Classification Status*
A1.jpg	Letter A	Correct
A2.jpg	Letter A	Correct
A3.jpg	Letter A	Correct
A4.jpg	Letter A	Correct
A5.jpg	Letter A	Correct
A6.jpg	Letter A	Correct
A7.jpg	Letter A	Correct
A8.jpg	Letter A	Correct
A9.jpg	Letter A	Correct
A10.jpg	Letter A	Correct
A11.jpg	Letter A	Correct
A12.jpg	Letter A	Correct
A13.jpg	Letter A	Correct
A14.jpg	Letter A	Correct
A15.jpg	Letter A	Correct
A16.jpg	Letter A	Correct
A17.jpg	Letter A	Incorrect
A18.jpg	Letter A	Correct
A19.jpg	Letter A	Correct
A20.jpg	Letter A	Correct

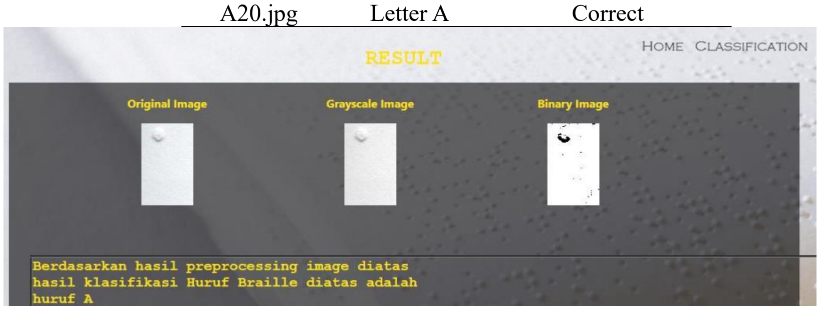

Fig. 8 Correct classification result of data A.1

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 1, January 2024
https://doi.org/10.47709/cnahpc.v6i1.3386

Submitted : Dec, 29, 2024
Accepted : January 25, 2024
Published : January 26, 2024

Fig. 9 Incorrect classification result of data A.17

Based on the test results of the classification of Braille letters using the deep learning CNN method, the accuracy of the image data of the letters K, N, O, P, S, and U Braille is 100%. While the accuracy of the image data of the letters A, D, E, G, L, and T is worth 95%. Then, the image data of the letters B, I, and X with an accuracy value of 90%. Then, the image data of the letter W with an accuracy value of 85%. Then, the letters J, R, and Z with an accuracy value of 90%. Then, for the letters C and H with an accuracy value of 85%. Then, the letters V and F have an accuracy of 90% and 95%, respectively. And finally, the letters M, Q, and Y have an accuracy of 85%, 80%, and 75%, respectively. Errors in classifying data in testing are due to Braille letter image data that are almost similar. So, the accuracy results are as follows for all the Braille letter images that have been tested (Gonçalves et al., 2020):

$$Accuracy = \frac{All\ True}{Total\ True} \times 100\% \tag{1}$$

$$Accuracy = \frac{480}{520} \times 100\% = 92{,}30\% \tag{2}$$

Based on these calculations, the accuracy value of Braille letter classification using the CNN method is 92.30%.

CONCLUSION

Based on research using image mining conducted on Braille letter classification, the CNN method can classify based on Braille letter patterns. The Braille letter category produces an accuracy rate on training data using Max Pooling and epoch 100 for training data of 96.64%. However, the test value of Braille letter classification accuracy using the CNN method is still low, only 92.30%. Therefore, for better system development, it is recommended to use hyperparameter tuning to minimize classification uncertainty in Braille letter images. Furthermore, the following system is expected to use the convolutional neural network method to classify Braille sentences and Braille numbers.

REFERENCES

Assefa, T. H., & Assefa, T. H. (2018). Design of Neural Network System to Communicate a Blind Person with a Computer Using a Braille. *Innovative Systems Design and Engineering*, *9*(1), 19–29. https://doi.org/10.7176/ISDE/9-1-04

Bhatia, S., Devi, A., Alsuwailem, R. I., & Mashat, A. (2022). Convolutional Neural Network Based Real Time Arabic Speech Recognition to Arabic Braille for Hearing and Visually Impaired. *Frontiers in Public Health*, *10*. https://doi.org/10.3389/fpubh.2022.898355

Gonçalves, D., Santos, G. G., Campos, M. B., Amory, A. M., & Manssour, I. H. (2020). Braille character detection using deep neural networks for an educational robot for visually impaired people. *2020: Anais Do XVI Workshop de Visão Computacional*, 123–128. https://github.com/lsa-pucrs/donnie-assistive-robot-sw

Hassan, K. M. N., Biswas, S. K., Anwar, M. S., Siam, M. S. I., & Shahnaz, C. (2019). A Dual-Purpose Refreshable Braille Display Based on Real Time Object Detection and Optical Character Recognition. *2019 IEEE International Conference on Signal Processing, Information, Communication & Systems (SPICSCON)*. https://doi.org/10.1109/SPICSCON48833.2019.9065110

Herlambang, M. F., Hermana, A. N., & Putra, K. R. (2021). Pengenalan Karakter Huruf Braille dengan Metode Convolutional Neural Network. *Systemic: Information System and Informatics Journal*, *6*(2), 20–26. https://doi.org/10.29080/systemic.v6i2.969

Hossain, S., Raied, A. A., Rahman, A., Abdullah, Z. R., Adhikary, D., Khan, A. R., Bhattacharjee, A., Shahnaz, C., &

* Corresponding author

Submitted : Dec, 29, 2024
Accepted : January 25, 2024
Published : January 26, 2024

Fattah, S. A. (2018). Text to Braille Scanner with Ultra Low-Cost Refreshable Braille Display. *2018 IEEE Global Humanitarian Technology Conference (GHTC)*. https://doi.org/10.1109/GHTC.2018.8601552

Kausar, T., Manzoor, S., Kausar, A., Lu, Y., Wasif, M., & Adnan Ashraf, M. (2021). Deep Learning Strategy for Braille Character Recognition. *IEEE Access*, *9*, 169357–169371. https://doi.org/10.1109/ACCESS.2021.3138240

Lecun, Y., Bottou, E., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, *86*(11). https://doi.org/10.1109/5.726791

Li, C., & Yan, W. (2021). Braille Recognition Using Deep Learning. *Proceedings of the 4th International Conference on Control and Computer Vision*, 30–35. https://doi.org/10.1145/3484274.3484280

Li, T., Zeng, X., & Xu, S. (2014). A Deep Learning Method for Braille Recognition. *2014 Sixth International Conference on Computational Intelligence and Communication Networks (CICN)*. https://doi.org/10.1109/CICN.2014.229

Made, I., Agastya, A., Oktarina, D., Handayani, D., & Mantoro, T. (2019). A Systematic Literature Review of Deep Learning Algorithms for Personality Trait Recognition. *2019 5th International Conference on Computing Engineering and Design (ICCED)*. https://doi.org/10.1109/ICCED46541.2019.9161107

Mccarthy, T., Rosenblum, L. P., Johnson, B. G., Dittel, J., & Kearns, D. M. (2016). An Artificial Intelligence Tutor: A Supplementary Tool for Teaching and Practicing Braille. *Journal of Visual Impairment & Blindness*, *110*(5), 309–322. https://doi.org/10.1177/0145482X1611000503

Molina, S., Pérez, B., & Gómez, J. (2016). Literary Braille language translator to Spanish text. *2016 IEEE International Conference on Automatica (ICA-ACCA)*. https://doi.org/10.1109/ICA-ACCA.2016.7778514

Murthy, V. V., & Hanumanthappa, M. (2018). Improving Optical Braille Recognition in Pre-processing stage. *2018 International Conference on Soft-Computing and Network Security (ICSNS)*. https://doi.org/10.1109/ICCIMA

Ovodov, I. G. (2021). Semantic-based Annotation Enhancement Algorithm for Semi-supervised Machine Learning Efficiency Improvement Applied to Optical Braille Recognition. *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, 2190–2194. https://doi.org/10.1109/ElConRus51938.2021.9396534

Ramiati, R., Aulia, S., & Lifwarda, L. (2020). Aplikasi Identifikasi Huruf Braille Menggunakan Computer Vision Berbasis Raspberry Pi. *JURNAL NASIONAL TEKNIK ELEKTRO*, *9*(1), 12. https://doi.org/10.25077/jnte.v9n1.707.2020

Ramiati, R., Aulia, S., Lifwarda, L., & Ningrum, N. S. (2020). Recognition of Image Pattern to Identification of Braille Characters to be Audio Signals for Blind Communication Tools. *IOP Conference Series: Materials Science and Engineering*, *846*(1). https://doi.org/10.1088/1757-899X/846/1/012008

Singh, G., Kumar, B., & Jassi, J. S. (2015). Odia Braille: Text transcription via image processing. *2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE)*. https://doi.org/10.1109/ABLAZE.2015.7154983

Smelyakov, K., Chupryna, A., Yeremenko, D., Sakhon, A., & Polezhai, V. (2018). Braille Character Recognition Based on Neural Networks. *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*. https://doi.org/10.1109/DSMP.2018.8478615

Subur, J., Sardjono, T. A., & Mardiyanto, R. (2015). Braille Character Recognition Using Find Contour Method. *5th 2015 International Conference on Electrical Engineering and Informatics (ICEEI)*. https://doi.org/10.1109/ICEEI.2015.7352588

Zhi, T., Duan, L.-Y., Wang, Y., & Huang, T. (2016). Two-stage pooling of deep convolutional features for image retrieval. *2016 IEEE International Conference on Image Processing (ICIP)*. https://doi.org/10.1109/ICIP.2016.7532802

* Corresponding author