Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th October 2025. Vol. 5. No. 1; e-ISSN: 2808-4519

Handwritten Batak Toba Script Recognition Based on Deep Learning Using the Convolutional Neural Network (CNN) Algorithm

Wahyu Ardiantito. S[1]*, Hermawan Syahputra[2], Zulfahmi Indra[3], Insan Taufik[4], Susiana[5]

[1,2,3,4,5]*Computer Science, Universitas Negeri Medan*
ardiantitowahyu@gmail.com[1]*, hermawansyahputra@unimed.ac.id[2], zulfahmi.indra@unimed.ac.id[3], insantaufik@unimed.ac.id[4], susianafaisal313@gmail.com[5]

Abstract

The Batak Toba script is one of Indonesia's cultural heritages that has become increasingly rare and less recognized among younger generations. This research aims to develop a handwriting recognition system for Batak Toba characters using the *Convolutional Neural Network* (CNN) method, capable of accurately recognizing characters, transliterating them into Latin script, and translating them into Indonesian. The dataset was self-generated using the Noto Sans Batak font and character combinations, totaling 113 labels, which were processed into 64×64 grayscale images. The CNN model was designed with several convolutional and pooling layers and compiled using the Adam optimizer and categorical cross-entropy loss function. Training results achieved a validation accuracy of **98.36%** and a testing accuracy of 98.12**%**, with respective loss values of 0.0268 and 0.0295. The system was then integrated into a web-based application built as a Progressive Web App (PWA), supporting both online transliteration and translation features. These results demonstrate that the CNN approach is highly effective in recognizing Batak Toba characters. In the future, the system can be further developed into a full sentence-level OCR, integrated into a native Android application, and expanded with datasets from real handwritten samples.

Keywords: *Batak Toba script, handwriting recognition, Convolutional Neural Network, transliteration, deep learning, PWA.*

1. Introduction

Indonesia is a country rich in cultural diversity, including traditional writing systems inherited from various ethnic groups. Regional scripts represent an essential aspect of linguistic and cultural identity, reflecting the history and worldview of the people who use them [1]. UNESCO [2] emphasized that Indonesian culture contributes significantly to the Sustainable Development Goals (SDGs), particularly in education, science, and cultural preservation. However, in the face of rapid technological development and globalization, traditional scripts are becoming increasingly rare in daily use, risking functional extinction.

One of the endangered cultural heritages is the Batak Toba script, historically used by the Batak people of North Sumatra for writing letters, traditional manuscripts, and religious documents [3]. The script, derived from the ancient Indian Brahmi family through the Pallava script, once flourished in the 18th century [4]. Over time, however, its use declined drastically due to the widespread adoption of the Latin alphabet in modern education and communication. As a result, younger generations are becoming increasingly unfamiliar with the shapes and usage of the Batak Toba script [5].

The Batak Toba script is a semi-syllabic writing system consisting of ina ni surat (base characters) and anak ni surat (diacritics) that modify vowel or consonant sounds [6]. This complex structure makes manual recognition challenging, especially for handwritten forms that vary among individuals. Yet, the ability to read and write Batak characters is crucial for cultural preservation, manuscript digitization, and education. Hence, a technological approach is needed to support the automatic identification and transliteration of Batak Toba characters in an efficient and scalable manner.

The advancement of Artificial Intelligence (AI), particularly in Computer Vision, offers a powerful solution for pattern recognition problems. One of the most successful approaches in this domain is Deep Learning, a branch of Machine Learning that employs multilayered neural networks to recognize complex features in data [7]. Unlike conventional algorithms that rely heavily on manual feature extraction, deep learning models automatically extract and learn visual features directly from raw data. This technology has been widely applied in face recognition, object detection, and Handwritten Character Recognition (HCR) tasks.

Among various deep learning models, the Convolutional Neural Network (CNN) has become the most popular architecture for image-based recognition. CNNs are designed to detect hierarchical visual patterns through convolutional and pooling layers that extract essential features from image inputs [8]. This approach has demonstrated outstanding performance in several studies on regional Indonesian scripts. For instance, Mulyanto et al. [9] achieved 80% training accuracy for Lampung script recognition, while Shelvi et al. [11] reached 98.03% accuracy for Sundanese script recognition, confirming CNN's effectiveness for complex character structures.

Further research by Alvin and Wasito [13] implemented CNN for recognizing handwritten Javanese characters, achieving a final accuracy of 84%. Despite these promising results, challenges remain due to variations in handwriting styles and inconsistent image quality. These findings highlight the importance of developing robust models and representative datasets to support the recognition of other local scripts, including Batak Toba, which remains underexplored in current deep learning studies.

Based on previous studies, a clear research gap exists—there is still no automated system capable of accurately recognizing and translating handwritten Batak Toba characters. Existing digitization efforts, such as font creation and documentation, remain largely static and lack interactive or educational functionality [14]. Therefore, this study proposes the development of a CNN-based handwriting recognition system that can identify Batak Toba characters and integrate automatic transliteration and translation features.

2. Literature Reviews

Research on handwritten character recognition has evolved significantly with the advancement of *Deep Learning* and *Computer Vision*. Early approaches relied on traditional *Machine Learning* algorithms such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Backpropagation Neural Networks (BPNN), which required manual feature extraction and were limited in handling complex variations in handwriting. With the introduction of *Convolutional Neural Network (CNN)* architectures, recognition systems gained the ability to automatically extract hierarchical visual features, resulting in substantial improvements in accuracy and generalization [16], [8].

Several studies have explored CNN-based models for recognizing traditional Indonesian scripts. Mulyanto *et al.* [9] implemented CNN for classifying Lampung characters, using scanned handwritten datasets to develop an Optical Character Recognition (OCR) model. The system achieved 80% training accuracy and 57% testing accuracy, highlighting CNN's potential for feature extraction but also the need for larger datasets and improved architecture design. Similarly, Rikendry and Maharil [10] compared VGG-16 and ResNet-50 architectures for Lampung handwritten recognition, where VGG-16 achieved a higher accuracy (91%) and faster computation time compared to ResNet-50 (65%), suggesting that deeper networks are not always superior in small-scale datasets.

Another related study by Shelvi *et al.* [11] focused on recognizing Sundanese script characters using CNN. Their research achieved 98.03% accuracy after extensive training with 500 epochs and multiple learning rates, outperforming previous methods such as edge detection and BPNN, which only achieved around 90%. The results demonstrated CNN's strength in automatically learning visual features from script images without manual preprocessing. Likewise, Handoko and Indahyanti [12] successfully applied CNN to Bima script recognition, achieving 97.34% test accuracy using a dataset of 2,640 handwritten images across 22 classes, further validating CNN's robustness for multi-class character recognition.

Research on Javanese script recognition also supports CNN's effectiveness for handwritten characters. Alvin and Wasito [13] developed a CNN-based model for digital handwritten Javanese character recognition, obtaining an accuracy of 84%. However, they noted several challenges, such as data inconsistency, handwriting variation, and hardware limitations that affected classification accuracy. These findings underline the importance of high-quality data collection, preprocessing, and augmentation techniques to enhance model reliability and robustness in real-world applications.

Despite extensive research on local scripts such as Lampung, Sundanese, Javanese, and Bima, studies on Batak Toba script recognition remain very limited. Previous works have mainly focused on documentation, font creation, or transliteration efforts rather than *machine learning*-based recognition [14]. Therefore, this study addresses a clear research gap by proposing a CNN-based model for handwritten Batak Toba character recognition. Unlike prior works that only performed classification, this system integrates three core components—character recognition, Latin transliteration, and Indonesian translation—within a web-based application framework. By combining *Deep Learning* and cultural computing, this research aims to contribute both to the technological development of image-based character recognition and to the preservation of Indonesia's endangered writing systems.

3. Research Methods

3.1. Research Design

This study adopts an experimental research design focused on developing and evaluating a *Deep Learning*-based handwriting recognition model for the Batak Toba script. The research framework consists of several stages: dataset preparation, data preprocessing, model design and training, model evaluation, and web-based deployment. The overall workflow aims to build an automated system capable of recognizing Batak Toba characters and transliterating them into Latin script as well as translating them into the Indonesian language. The experiment was conducted at the Department of Computer Science, Faculty of Mathematics and Natural Sciences, Universitas Negeri Medan, from December 2024 to July 2025. All experiments were carried out using Python programming language and the *TensorFlow* library for model development and evaluation.

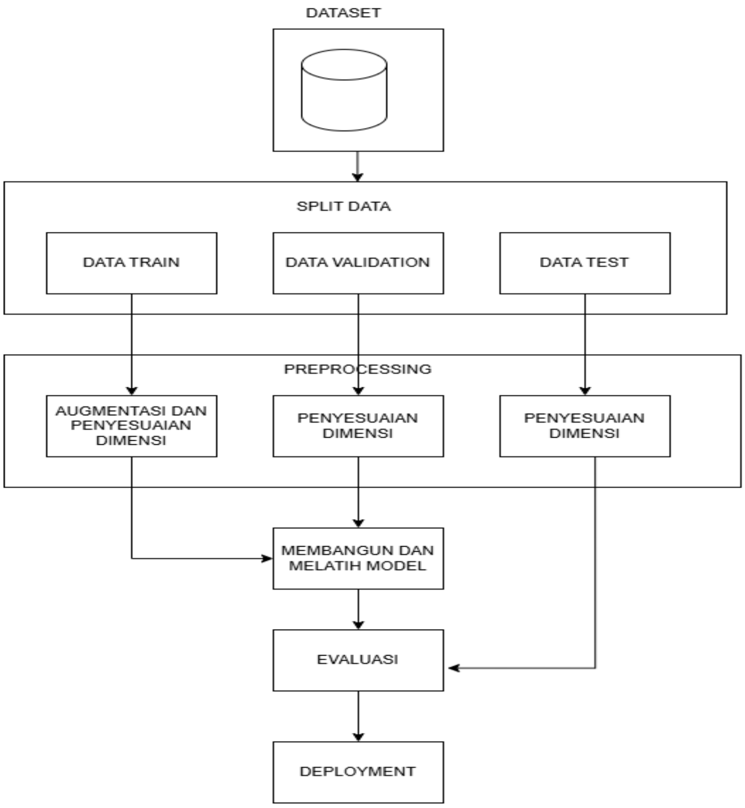

Fig. 1: Research Design

Figure 1 illustrates the overall workflow of the research process for developing the Batak Toba handwritten character recognition system. The process begins with dataset creation, which is then divided into three subsets: training, validation, and testing data. Each subset undergoes preprocessing, including dimension adjustment and normalization, while the training data additionally goes through data augmentation to enhance model generalization. The preprocessed data is subsequently used to build and train the Convolutional Neural Network (CNN) model. After training, the model is evaluated using the validation and testing data to measure its performance in terms of accuracy and reliability. Once satisfactory results are obtained, the trained model is deployed into a web-based application for real-time handwritten character recognition, transliteration, and translation.

3.2. Dataset Preparation

Since no public dataset exists for Batak Toba handwritten characters, a synthetic dataset was constructed specifically for this research. The dataset was generated using the *Noto Sans Batak* font provided by Google Fonts, which supports all primary Batak Toba characters. Each character image was rendered in grayscale format with a resolution of 64×64 pixels, resulting in 113 unique labels representing combinations of *ina ni surat* (base characters) and *anak ni surat* (diacritics), as shown in Table 1. To simulate handwritten variations, data augmentation was applied, including random rotation, horizontal flipping, and elastic distortion, following techniques proposed by Fadillah *et al.* [15]. The final dataset was divided into 70% training, 20% validation, and 10% testing subsets to ensure balanced class distribution and reliable model evaluation.

Table 1: Character Combination

Component	Count
ᯀ and variants	3
ᯂ and variants	6
ᯄ and variants	6
ᯔ, ᯉ, ᯒ, ᯖ, ᯘ, ᯇ, ᯞ, ᯎ, ᯐ, ᯑ, ᯝ, ᯅ, ᯋ, ᯛ, ᯠ × 6	90
ᯡ and variants	6
ᯤ, ᯥ	2
Total	**113**

Table 1 presents the list of Batak Toba script character combinations used in this study as the dataset for model training and evaluation. The dataset consists of a total of 113 character classes, representing various base characters (*ina ni surat*) and their diacritic combinations (*anak ni surat*). Each base character, such as ᯀ, ᯂ, ᯄ, and others, includes multiple variants that differ according to the attached diacritical marks, which modify the vowel or consonant sound. The largest group consists of fifteen base characters—ᯔ, ᯉ, ᯒ, ᯖ, ᯘ, ᯇ, ᯞ, ᯎ, ᯐ, ᯑ, ᯝ, ᯅ, ᯋ, ᯛ, and ᯠ—each with six variants, totaling 90 combinations. Other groups, such as ᯀ, ᯂ, ᯄ, and ᯡ, include between three and six variants, while ᯤ and ᯥ are represented as individual vowel symbols. Altogether, these 113 combinations comprehensively represent the primary structure of Batak Toba script characters, providing a complete dataset for training the Convolutional Neural Network (CNN) to accurately recognize and classify handwritten forms.

Fig. 2: Generate Synthetic Dataset and Elastic Distortion

Figure 2 illustrates the process of generating a synthetic dataset for Batak Toba script recognition. The image on the left represents a clean character sample generated using the *Noto Sans Batak* font, while the image on the right shows the same character after applying elastic distortion. This transformation introduces subtle variations in the character's shape, curvature, and stroke thickness, mimicking the natural inconsistencies found in human handwriting. By applying elastic distortion, the dataset gains greater variability, which helps the Convolutional Neural Network (CNN) model generalize better to real handwritten inputs. This approach ensures that the model does not merely memorize fixed font patterns but learns to recognize the essential structural features of each Batak Toba character despite variations in style or writing form.

3.3. Data Preprocessing

Before training, all image samples underwent a series of preprocessing steps to enhance data quality and model performance. Each image was resized to 64×64 pixels, converted to grayscale, and normalized to a pixel value range between 0 and 1. Data augmentation was implemented on the training subset to increase diversity and reduce overfitting. Transformations such as random rotation (±10°), zoom (±10%), and translation (±10%) were applied using the *ImageDataGenerator* module from the Keras library. Meanwhile, the validation and testing subsets were only normalized without augmentation to maintain objectivity during evaluation. All class labels were encoded using one-hot encoding to match the *categorical cross-entropy* loss function used in the CNN model.

3.4. CNN Model Architecture and Training

The proposed model was implemented using the Sequential API from *TensorFlow Keras*. The architecture consists of three convolutional layers with 32, 64, and 128 filters respectively, each using a 3×3 kernel size and *ReLU* activation function. Each convolutional block is followed by a MaxPooling2D layer to reduce spatial dimensions and computational complexity. The output from the last convolutional block is flattened and passed to a fully connected layer with 128 neurons using *ReLU* activation, followed by a Dropout layer (0.3) to prevent overfitting. The final output layer uses the Softmax activation function to perform multi-class classification across 113 Batak Toba Character Classes. The model was compiled using the Adam optimizer with a learning rate of 0.001 and trained using the categorical cross-entropy loss function [17]. Training was conducted over 10 epochs with a batch size of 32, and performance metrics such as *accuracy*, *precision*, *recall*, and *F1-score* were recorded for both training and validation datasets. Figure 3 visualize the CNN architecture that used in this research

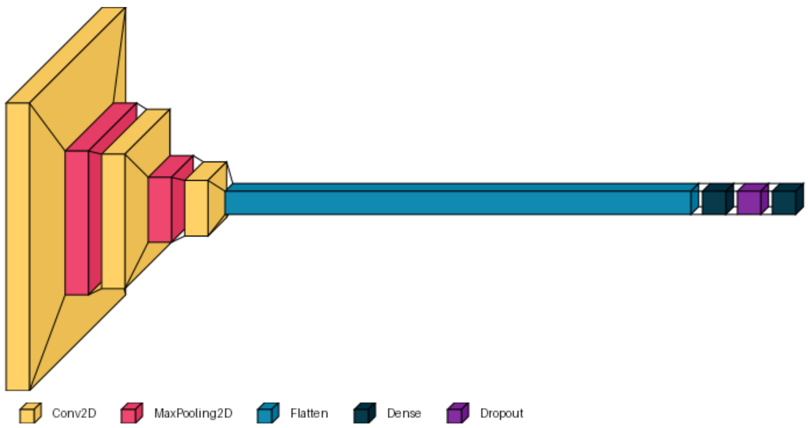

Fig. 3: CNN Architecture

3.5. System Implementation

After achieving satisfactory performance, the trained CNN model was exported in the *.h5* format and integrated into a web-based application using the *Flask* framework for backend processing. The web system was developed as a Progressive Web App (PWA) to ensure accessibility across different devices and partial offline functionality. The frontend interface, designed with HTML, Tailwind CSS, and JavaScript, allows users to draw or upload handwritten Batak Toba characters. The application then processes the image, predicts its corresponding Latin transliteration, and provides an automatic Indonesian translation through an integrated Google Translate API. This implementation demonstrates the practical usability of the system for both educational and cultural preservation purposes, bridging traditional heritage with modern digital technology.

4. Result and Discussion

4.1. System Implementation

The model training process was conducted using 6000 synthetic Batak Toba character images with a resolution of 64×64 pixels, divided into 70% training data, 20% validation data, and 10% testing data. The training used the Adam optimizer and categorical cross-entropy loss function for 10 epochs with a batch size of 32. During training, the model demonstrated rapid convergence, where both training and validation accuracy consistently increased after the third epoch and stabilized after the eighth epoch. The validation accuracy reached 98.36%, while the testing accuracy achieved 98.12%, indicating strong generalization and minimal overfitting.

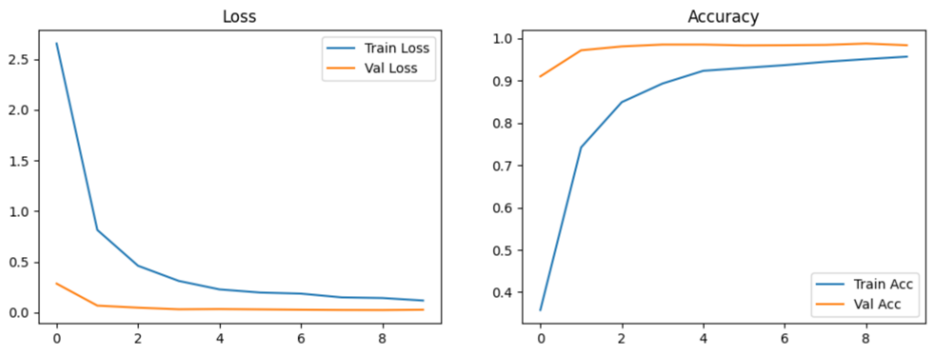

Fig. 4: System Implementation

4.2. Evaluation Metrics

To thoroughly assess the model's performance, multiple evaluation metrics were calculated, including accuracy, precision, recall, and F1-score. These metrics provide a more comprehensive view of classification effectiveness, particularly for multi-class datasets such as the 113 Batak Toba character labels used in this study. The model achieved an overall accuracy of 98.12%, average precision of 0.981, average recall of 0.979, and an F1-score of 0.980. These high scores demonstrate that the CNN model was not only able to correctly classify most input characters but also maintain consistent performance across all classes. Here's the result that obtain from training purposed model in Table 2 below.

Table 2: Training Purposed

Metric Type	Precision	Recall	F1-score
Macro Average	0.9735	0.9823	0.9764
Weighted Average	0.9758	0.9839	0.9785

The precision metric reflects the model's ability to avoid false positives, meaning that when a character is predicted as a particular class, it is highly likely to be correct. The recall metric indicates that the model successfully identified nearly all instances of each class, showing minimal missed detections. The F1-score, which combines both precision and recall, confirms that the model maintains a strong balance between accuracy and consistency across character categories.

However, a small number of misclassifications occurred between visually similar characters, especially among those that share similar base forms but differ in diacritical marks, such as [illegible] and [illegible] or [illegible] and [illegible]. These errors are expected, as diacritics in Batak Toba script often appear as minor variations in small regions of the character image. Nevertheless, these misclassifications accounted for less than 2% of all predictions, indicating excellent robustness and discriminative power of the CNN model.

4.3. Effects of Data Augmentation and Elastic Distortion

The incorporation of data augmentation and elastic distortion proved essential in improving model generalization and realism in handwritten character recognition. Initially, when the model was trained only on uniform, font-generated characters, the accuracy plateaued around 91–92%, with clear signs of overfitting after a few epochs. The limited visual diversity in the dataset restricted the CNN's ability to handle handwritten variations.

By introducing elastic distortion, random rotation, and geometric transformations, the training data more closely resembled real-world handwriting characteristics. Elastic distortion, in particular, mimics the natural irregularities of human writing — such as uneven strokes, variable curvature, and stroke width differences — which helped the model learn invariant structural patterns rather than memorizing fixed shapes. As a result, the model achieved a remarkable 7–8% improvement in overall accuracy and demonstrated greater stability across multiple runs.

This finding aligns with previous studies [15], [16], which reported that applying distortion-based augmentation significantly enhances the robustness of CNN models for handwriting recognition tasks. Thus, data augmentation is a key contributor to the model's success in generalizing to unseen handwritten inputs.

4.4. Web Application Implementation

After achieving optimal model performance, the trained CNN model was integrated into a Progressive Web App (PWA) for practical use. The deployment aimed to demonstrate the system's real-world functionality as an interactive handwriting recognition tool for the Batak Toba script. The backend of the application was developed using the Flask framework in Python, while the frontend was built with HTML, Tailwind CSS, and JavaScript, providing a user-friendly interface accessible from both desktop and mobile devices. Figure 4 display the interface of purposed web that this research makes.

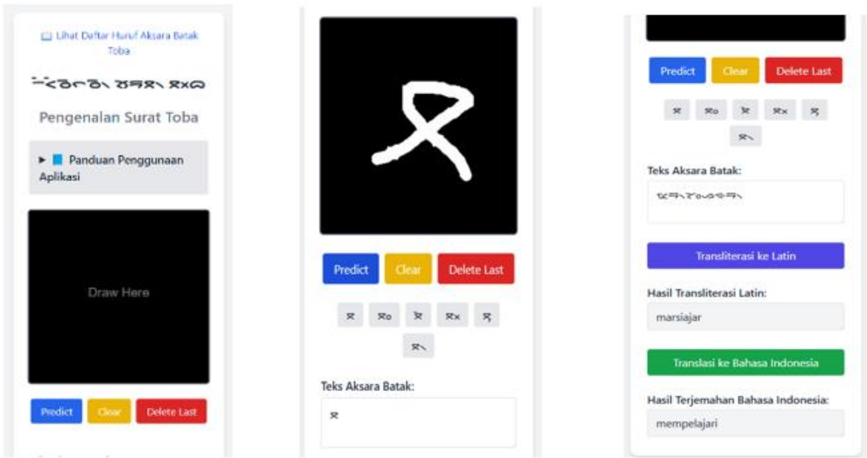

Fig. 5: Web Application Implementation

The web system allows users to draw or upload handwritten Batak Toba characters, which are then processed by the CNN model in real time. Once recognition is complete, the application performs automatic transliteration into Latin script and translation into Indonesian, utilizing an integrated translation API. Testing of the deployed system demonstrated that the average prediction time for a single image was under one second, even on standard computing hardware, indicating that the system is suitable for real-time applications. From a usability perspective, the PWA design ensures that the system can function with minimal internet dependency and supports offline caching, allowing users to access it in remote areas with limited connectivity. The deployment of this model in a web-based environment highlights its potential as a digital cultural learning tool, bridging heritage preservation with accessible modern technology.

4.5. Discussion

The results of this study confirm that the Convolutional Neural Network (CNN) is an effective and reliable approach for recognizing Batak Toba handwritten characters. The model's high accuracy and low loss values demonstrate its strong capacity for learning discriminative features, even within a complex multi-class environment involving 113 distinct characters. The ability to recognize subtle diacritic variations validates the network's capability to extract fine-grained spatial features from the input data.

Compared to related research on regional scripts, the proposed model achieves comparable or superior performance. For instance, Shelvi *et al.* [11] obtained 98.03% accuracy for Sundanese script recognition, while Alvin and Wasito [13] achieved 84% accuracy for Javanese handwritten characters. The Batak Toba CNN model, with 98.12% testing accuracy and fewer data samples, demonstrates that well-constructed synthetic datasets can substitute for large-scale handwritten datasets in low-resource contexts. Moreover, its PWA-based implementation extends its utility beyond academic use—serving as an interactive learning platform for cultural preservation.

This research also contributes to the broader field of computational cultural heritage, where AI technologies are applied to safeguard endangered writing systems. By enabling instant recognition, transliteration, and translation, the developed system not only functions as a handwriting recognizer but also as a digital revitalization tool for the Batak Toba language and script. Future improvements could involve integrating real handwritten samples from native users, expanding the dataset, and deploying the model in mobile-native applications for wider accessibility and offline learning experiences.

5. Conclusion

This research successfully developed a Batak Toba handwritten character recognition system using the Convolutional Neural Network (CNN) algorithm, capable of recognizing characters, transliterating them into Latin script, and translating them into Indonesian. The dataset consisted of 113-character classes generated synthetically from the *Noto Sans Batak* font, enhanced through data augmentation and elastic distortion to resemble natural handwriting. The final model achieved 98.36% validation accuracy and 98.12% testing accuracy, confirming CNN's high effectiveness in recognizing complex diacritic-based characters and its strong generalization capability despite the limited dataset size. The integration of the trained model into a Progressive Web App (PWA) demonstrated the system's practical usability and cultural significance. Users can upload or draw handwritten Batak Toba characters and instantly obtain transliteration and translation results in real time

References

[1] R. Lorentius, R. Setiawan, and D. Hutabarat, "Preserving local scripts through digital recognition systems," *J. Cult. Inform.*, vol. 5, no. 2, pp. 45–53, 2020.
[2] UNESCO, "Culture for sustainable development: The role of heritage in achieving SDGs," *UNESCO Rep. Cult. Dev.*, Paris, 2017.
[3] J. Butar-Butar, T. Sihombing, and M. Silalahi, "Revitalization of Batak Toba script through digital media," *Indones. J. Lang. Cult.*, vol. 4, no. 1, pp. 22–30, 2022.
[4] U. Kozok, *Buku Tanda-Tanda: The Batak Script and Its History*. Jakarta: KITLV Press, 2009.
[5] P. Turnip, F. Sinaga, and R. Pardede, "Digital archiving of Batak script manuscripts for cultural preservation," *J. Teknol. Komput. Indones.*, vol. 12, no. 3, pp. 110–118, 2023.
[6] S. Pasaribu, "Aksara Batak dan fungsinya dalam tradisi tulisan Batak Toba," *J. Kebahasaan Kesastraan Indones.*, vol. 7, no. 2, pp. 65–73, 2015.
[7] R. Pratiwi, U. Hasanah, and A. Rachman, "A review of deep learning algorithms for image recognition," *Int. J. Artif. Intell. Res.*, vol. 5, no. 1, pp. 56–64, 2021.
[8] M. Omori and K. Shima, "Convolutional neural network architecture for visual pattern recognition," *Procedia Comput. Sci.*, vol. 176, pp. 2544–2551, 2020.
[9] R. Mulyanto, M. Fathoni, and D. Putra, "Lampung script recognition using convolutional neural networks," *J. Ilm. Teknol. Inf.*, vol. 19, no. 2, pp. 87–95, 2021.
[10] R. Rikendry and M. Maharil, "Comparison of VGG-16 and ResNet-50 architectures for Lampung handwritten character recognition," *J. Comput. Vis. Image Process.*, vol. 8, no. 1, pp. 33–41, 2022.
[11] D. Shelvi, R. Wijaya, and A. Rahman, "Handwritten Sundanese script recognition using convolutional neural network," *Indones. J. Artif. Intell. Data Sci.*, vol. 2, no. 4, pp. 121–128, 2021.
[12] S. Handoko and D. Indahyanti, "CNN-based recognition of Bima script characters," *J. Inform. Komput. Cerdas*, vol. 3, no. 1, pp. 14–22, 2024.
[13] A. Alvin and I. Wasito, "Handwritten Javanese character recognition using deep convolutional neural network," in *Proc. 2023 Int. Conf. Inf. Technol. Innov.*, pp. 88–94, 2023.
[14] A. Cahya, R. Nugraha, and D. Sari, "Digitalization of local scripts: Efforts to preserve Indonesian writing systems," *J. Kebud. Nusantara*, vol. 11, no. 2, pp. 97–105, 2021.
[15] N. Fadillah, H. Kurniawan, and R. Setiadi, "Enhancing CNN performance using elastic distortion for handwritten character recognition," *Int. J. Mach. Vis. Signal Process.*, vol. 9, no. 3, pp. 201–210, 2021.
[16] F. Anhar and Y. Rahma, "Improving deep CNN models for multi-class handwritten recognition through data augmentation," *J. Comput. Sci. Appl.*, vol. 15, no. 2, pp. 142–151, 2023.
[17] M. Fawwaz, S. Putri, and T. Hidayat, "Optimization of CNN with Adam and cross-entropy for character image classification," *J. Teknol. Syst. Komput.*, vol. 8, no. 4, pp. 155–162, 2020.