FastViT-Based Lightweight Batak Script Recognition for Cultural Digital Preservation

[1*)]Egi Dio Bagus Sudewo, [2)]Hainur Rasyid, [3)]Muhammad Kunta Biddinika, [4)]Kariyamin, [5)]Dini Khairunnisa Tanjung
[1,2,5)]Sistem Informasi, Fakultas Ilmu Komputer, Universitas Royal
[3)]Magister Informatika, Fakultas Teknologi Industri, Universitas Ahmad Dahlan
[4)]Teknologi Informasi, Institut Teknologi dan Bisnis Muhammadiyah Wakatobi
[1*)]egidio8gali@gmail.com, [2)]hainurrasyid@royal.ac.id, [3)]muhammad.kunta@mti.uad.ac.id, [4)]kariyaminyamin28@gmail.com, [5)]dinikhairunnisatanjung@gmail.com

INFO ARTIKEL

Riwayat Artikel :
Diterima : 1 Juli 2026
Disetujui : 7 Agustus 2026

Kata Kunci :
Pengenalan Aksara Batak, FastViT-SA12, Deep Learning, Computer Vision, Optical Character Recognition (OCR), Digitalisasi Warisan Budaya

ABSTRAK

Aksara Batak merupakan warisan budaya masyarakat Batak di Sumatera Utara yang perlu dilestarikan melalui digitalisasi. Penelitian ini mengusulkan penggunaan arsitektur FastViT-SA12 untuk pengenalan tulisan tangan aksara Batak menggunakan dataset Batak Char 20 yang terdiri atas 117 kelas karakter. Ketidakseimbangan data diatasi melalui augmentasi menggunakan ImgAug, sehingga setiap kelas memiliki 650 citra. Data dibagi menjadi data latih, validasi, dan pengujian dengan rasio 70:20:10, kemudian seluruh citra diubah ukurannya menjadi 64 × 64 piksel untuk menekan biaya komputasi. Model dilatih selama 30 epoch menggunakan optimizer Adam dengan learning rate 0,001 dan batch size 64. Hasil penelitian menunjukkan akurasi pelatihan 99,97%, akurasi validasi 97,96%, serta akurasi pengujian 89,31%, dengan precision, recall, dan F1-score terbobot masing-masing 91,84%, 89,31%, dan 88,99%. FastViT-SA12 menunjukkan potensi yang baik dalam mengenali 117 kelas aksara Batak, tetapi masih memerlukan optimasi resolusi dan validasi lebih lanjut untuk meningkatkan generalisasi pada karakter berdiakritik kecil, guna mendukung pengembangan sistem OCR yang portabel.

ARTICLE INFO

Article History :
Received : Jul 1, 2026
Accepted : Aug 7, 2026

***Keywords**:*
Batak Script Recognition, FastViT-SA12, Deep Learning, Computer Vision, Optical Character Recognition (OCR), Cultural Heritage Digitization

ABSTRACT

Batak script is a cultural heritage of the Batak people in North Sumatra that requires preservation through digitization. This study proposes the FastViT-SA12 architecture for handwritten Batak script recognition using the Batak Char 20 dataset, which consists of 117 character classes. To address data imbalance, data augmentation was performed using ImgAug, resulting in 650 images per class. The dataset was divided into training, validation, and testing sets with a ratio of 70:20:10, and all images were resized to 64 × 64 pixels to reduce computational overhead. The model was trained for 30 epochs using the Adam optimizer with a learning rate of 0.001 and a batch size of 64. The proposed model achieved a training accuracy of 99.97%, a validation accuracy of 97.96%, and a test accuracy of 89.31%, with weighted precision, recall, and F1-score of 91.84%, 89.31%, and 88.99%, respectively. FastViT-SA12 demonstrates *promising potential for recognizing 117 Batak script classes while maintaining computational efficiency with a compact parameter footprint, making it a viable candidate for mobile-first Optical Character Recognition (OCR) systems in manuscript preservation.*

1. INTRODUCTION

Batak script is an abugida writing system that constitutes an inseparable part of the cultural identity and heritage of the Batak people in North Sumatra, Indonesia. The script comprises five major dialectal variants, namely Toba, Mandailing-Angkola, Karo, Pakpak-Dairi, and Simalungun, which are distributed across different geographical regions. Historically, Batak script has been used to record a wide range of indigenous knowledge, including customary laws, traditional medicine, incantations, and divination practices documented by traditional scholars known as *datu*. These records were inscribed on various media such as bamboo, bone, horn, and *pustaha laklak* (folding books made from tree bark) (Siregar, 2017). The historical, cultural, and anthropological significance of these manuscripts highlights the urgent need for systematic preservation efforts through documentation and digitalization to ensure their accessibility for future generations and the global research community (Sudewo *et al.*, 2025).

Despite its importance, the preservation of Batak manuscripts faces several critical challenges. Many surviving documents stored in museums and private collections have experienced deterioration due to aging, humidity, insect infestation, and improper handling. In addition, the limited number of experts capable of reading and interpreting ancient Batak script further complicates preservation initiatives (Hasugian, 2025). Current preservation practices largely rely on manual documentation, where manuscripts are photographed without automated systems capable of recognizing, converting, or transcribing the characters into digital formats. Consequently, the integration of modern technology into manuscript processing and information retrieval remains relatively limited.

Recent advances in computer vision and deep learning have provided promising solutions for automatic script recognition, where Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) have emerged as primary paradigms for Optical Character Recognition (OCR) (Jayachandran, Selvakumar and Pearline, 2025; Xu *et al.*, 2025; Akallouch, Akallouch and Fardousse, 2026). Previous studies have widely explored prominent lightweight and heavy architectures, such as ResNet-18 (Alghyaline, 2024), VGG (Alghyaline, 2024) , EfficientNet (Jayachandran, Selvakumar and Pearline, 2025), and Hybrid-Vit (Ashimgaliyev *et al.*, 2026), to balance classification accuracy and computational overhead across diverse vision tasks. However, the application of FastViT-SA12 which incorporates structural reparameterization remains largely unexplored in comprehensive traditional script recognition involving diacritical variations. (Willian, Rochadiani and Sofian, 2023) developed a Batak Toba script recognition system using CNN, achieving an accuracy of 99.54% on 19 character classes. In parallel, lightweight hybrid architectures like FastViT have gained traction for combining CNN efficiency with Transformer representation capacity across image classification tasks, such as lung ultrasound scoring (Ye *et al.*, 2025), breast cancer classification while (Babita and Nayak, 2024), and diabetic retinopathy detection (Willian, Rochadiani and Sofian, 2023; Long *et al.*, 2025)

However, several critical research gaps remain unaddressed in current literature: existing Batak script recognition studies primarily focus on small subsets of base characters (e.g., 19 classes), ignoring complex character-diacritic combinations (*anak ni surat*) that alter inherent vowels or append final consonants; furthermore, structural reparameterization-based hybrid vision transformers such as FastViT-SA12 remain unexplored in traditional script OCR, and the impact of aggressive image downscaling (64 × 64 pixels) on fine diacritical mark recognition is insufficiently analyzed. To bridge these gaps, the main novelties and contributions of this study lie in applying FastViT-SA12 for the first time to handwritten Batak script recognition, evaluating a full-scale dataset of 117 classes encompassing both base characters (*ina ni surat*) and diacritical variations (*anak ni surat*), and systematically investigating the discriminative trade-offs and limitations of low-resolution inputs on visually similar, diacritic-dependent handwritten characters for mobile-first cultural preservation.

2. METHODS

2.1 Research Stages

The experiment in this study was conducted through a structured and measurable workflow to ensure the validity of the resulting model. The data processing pipeline was designed to optimally transform raw data before being evaluated by the FastViT-SA12 model. The workflow begins with Data Collection from the Batak Char 20 dataset. Next, Data Augmentation is applied using the ImgAug library to balance class distribution. The augmented images then undergo Data Preprocessing, including resizing and normalization. Subsequently, Model Training and Model Validation are performed, followed by diagnostic Performance Evaluation using quantitative metrics and confusion matrix analysis. Figure 1 illustrates the research workflow.

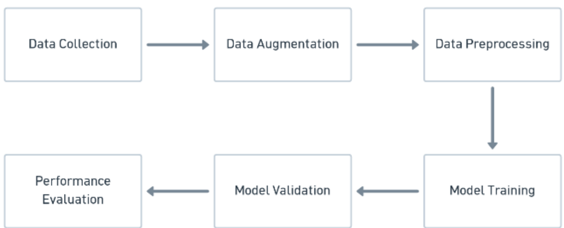

Figure 1. Research Flow

2.2 Dataset

This study utilized a secondary dataset named Batak Char 20, obtained from the Kaggle repository platform https://www.kaggle.com /datasets/rezataqyuddin/batak-char-20. The dataset was specifically designed for developing handwritten Batak script recognition models. The dataset presents a challenging classification task because it consists of 117 unique classes representing not only the basic Batak characters (surat) from various Batak dialects but also combinations of basic characters with diacritical marks (ina ni surat and anak ni surat). These marks modify the inherent vowel /a/ into /i/, /u/, /e/, /o/, or /ə/, or add final consonants such as /-ng/, /-h/, and the vowel suppression marker (pangolat/panongonan). To ensure fair evaluation, the number of samples in each experimental partition was balanced, as presented in Table 1.

Table 1. Dataset Splitting

Dataset Partition	Samples per Class	Total Samples
Training	455	53,235
Validation	130	15,210
Testing	65	7,605
Total	650	76,050

2.3 Experimental Setup

To ensure reproducibility and accommodate computational efficiency evaluations, all experiments were executed within the Google Colab cloud computing platform accelerated by an NVIDIA Tesla T4 GPU with 15 GB VRAM and standard system memory (Carneiro *et al.*, 2018; Sohail *et al.*, 2024). The software pipeline was implemented on a Python 3.10 runtime using the PyTorch 2.2.0 deep learning framework (`torchvision` 0.17.0, `torchaudio` 2.2.0), PyTorch Image Models (`timm` v1.0.27), NumPy 1.26.4, `ImgAug` 0.4.0, Scikit-Learn (`sklearn.metrics`), Pillow (PIL), `torchsummary`, Matplotlib, and Seaborn under CUDA acceleration (Jha and Samlodia, 2024; Barillaro, 2025).

2.4 Data Augmentation

To address class imbalance and enhance handwriting diversity, data augmentation was performed using the ImgAug library (Ataman, 2025; Jia, Zheng and Gu, 2026; Wang *et al.*, 2026). Augmentation techniques included affine rotations (±10°), subtle translations (±5%), scaling (0.9-1.1), and brightness adjustments. To preserve semantic validity, horizontal flipping was strictly constrained to prevent distorting directional diacritical marks that determine vowel and consonant modifications in Batak script. These transformations simulated real-world handwriting and illumination variations, yielding a balanced dataset of 650 images per class across 117 classes (455 training, 130 validation, and 65 testing samples), as summarized in Table 2

Table 2. Summarized Dataset

Partition Category	Samples per Class
Training	455
Validation	130
Testing	65

2.5 Preprocessing

Prior to model training, all images underwent preprocessing to standardize input specifications and optimize computational throughput (Hasan *et al.*, 2020; Zhang *et al.*, 2021). The augmented images, originally at a resolution of 224 × 224 pixels, were resized to 64 × 64 pixels to reduce memory footprint and accelerate execution. Additionally, pixel intensities ranging from 0 to 255 were normalized to 0.0–1.0 using ($X_{norm} = X/255$) to stabilize gradient backpropagation and enhance optimization convergence (Hurtik, Molek and Hula, 2020; Ilić, 2020).

2.6 FastViT-SA12 Architecture

FastViT-SA12 is a lightweight hybrid vision transformer architecture designed to unify the local feature extraction efficiency of CNNs with the global context modeling capability of transformers across a hierarchical four-stage structure (Gabriel and Zhu, 2023; Stephen *et al.*, 2025). It utilizes RepMixer blocks with structural reparameterization to collapse multi-branch training convolutions into a single execution path during inference, alongside efficient attention modules to capture long-range spatial dependencies (Hatamizadeh *et al.*, 2024; Ran *et al.*, 2025). Comprising approximately 10.670.005 trainable parameters (~41 MB model size), FastViT-SA12 reduces memory access costs and latency while maintaining strong representation capacity, making it well-suited for recognizing 117 handwritten Batak character classes.

2.7 Training Configuration

The FastViT-SA12 model was initialized using ImageNet-pretrained weights from the `timm` library. The original classification head was modified to output 117 target classes, and full fine-tuning was applied across all layers during training. The detailed hyperparameter settings used during training are summarized in Table 3.

Table 3. The training configuration

Training Parameter	Configuration
Model Architecture	FastViT-SA12 (timm library)
Optimizer	Adam
Learning Rate	0.001
Batch Size	64
Epochs	30
Input Size	64 × 64 pixels
Loss Function	Categorical Crossentropy
Trainable Parameters	10,670,005
Fine-tuning Strategy	Full layer fine-tuning

2. Model Evaluation

The classification performance of the model was evaluated using several quantitative metrics to assess its effectiveness in recognizing 117 Batak script classes.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (4)$$

In addition, a confusion matrix was generated to visualize the distribution of classification errors across all 117 classes, providing deeper diagnostic insights into the visual similarities and misclassification patterns among Batak script characters.

3. RESULT AND DISCUSSION

3.1 Training Results

The training process of the FastViT-SA12 model exhibited a rapid and stable convergence rate throughout the 30 training epochs. The model progressively learned the visual patterns of handwritten Batak characters and demonstrated consistent improvements in classification performance. A summary of the training and validation accuracy and loss values is presented in Table 4.

Table 4. Training Result

Epoch	Train Accuracy	Validation Accuracy	Train Loss	Validation Loss
1	0.8124	0.6998	0.709	1.103
5	0.9115	0.8340	0.256	0.512
10	0.9482	0.8912	0.143	0.305
15	0.9710	0.9245	0.089	0.198
20	0.9854	0.9480	0.052	0.124
25	0.9938	0.9621	0.031	0.091
30	0.9997	0.9796	0.021	0.064

At the beginning of the training process (Epoch 1), the model achieved a training accuracy of 81.24% with a loss value of 0.709, while the validation accuracy started at 69.98% with a validation loss of 1.103. As training progressed, the model continuously improved its internal feature representations, resulting in significant performance gains. By the final epoch, the model achieved an almost perfect training accuracy of 99.97% with a very low training loss of 0.021. Simultaneously, the highest validation accuracy reached 97.96% with a validation loss of 0.064, indicating effective learning and strong convergence behavior, can be seen in Figure 2.

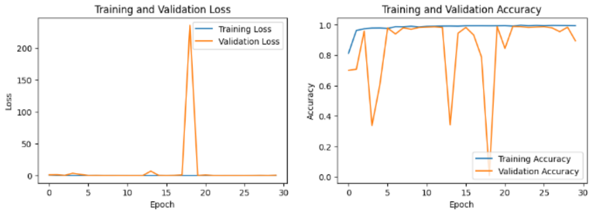

Figure 2. Training chart

3.2 Learning Curve Analysis

The learning curve analysis, which illustrates the relationship between training epochs and both accuracy and loss values, revealed a healthy convergence pattern. The training and validation accuracy curves increased consistently during the first ten epochs before gradually stabilizing near an asymptotic value above 95%. Correspondingly, both training and validation loss curves decreased sharply and monotonically toward zero.

Although minor fluctuations were observed in the validation loss during the final epochs, the quantitative performance gap between training accuracy (99.97%) and validation accuracy (97.96%) was precisely 2.01% (99.97% - 97.96% = 2.01%). This narrow gap of approximately 2% suggests stable learning with minimal validation variance, indicating that data augmentation effectively mitigated severe overfitting during the training phase.

3.3 Testing Results

The final evaluation was conducted using an independent testing dataset containing 7,605 images. The overall performance of the trained FastViT-SA12 model is summarized in Table 5.

Table 5. Testing Result

Evaluation Metric	Performance
Test Accuracy	89.31%
Weighted Precision	91.84%
Weighted Recall	89.31%
Weighted F1-Score	88.99%

Evaluation on the unseen test dataset (7,605 images) yielded an overall test accuracy of 89.31%. To provide an unweighted and equitable diagnostic across all 117 classes, both Macro-averaged and Weighted-averaged metrics were evaluated, as presented in Table 5. This performance gap confirms the adverse effect of reducing image resolution to 64 × 64 pixels. At such a low resolution, many fine-grained visual details that distinguish handwritten Batak characters become blurred, limiting the model's ability to generalize effectively to unseen test samples.

3.4 Confusion Matrix and Classification Report Analysis

A detailed analysis of the classification report provides valuable insights into the model's ability to distinguish among diverse Batak script characters.

Analysis of the Best-Performing Classes

Several character classes were classified almost perfectly, achieving F1-scores close to or equal to 1.00. The ten best-performing classes are presented in Table 6.

Table 6. 10 Best-Performing Classes

Class	Precision	Recall	F1-Score
b	1.000000	1.000000	1.000000
ba	1.000000	1.000000	1.000000
n	1.000000	1.000000	1.000000
e	1.000000	0.984615	0.992248
no	1.000000	0.984615	0.992248
ti	1.000000	0.984615	0.992248
nyi	1.000000	0.984615	0.992248
ni	1.000000	0.984615	0.992248
wu	0.969231	1.000000	0.984615
bo	0.969231	1.000000	0.984615

Characters such as ba and b exhibit relatively simple geometric structures dominated by straight vertical and horizontal strokes with minimal curvature. These distinctive visual features remain recognizable even after image downscaling, enabling FastViT-SA12 to establish highly discriminative decision boundaries and achieve near-perfect classification performance, can be seen in Figure 3.

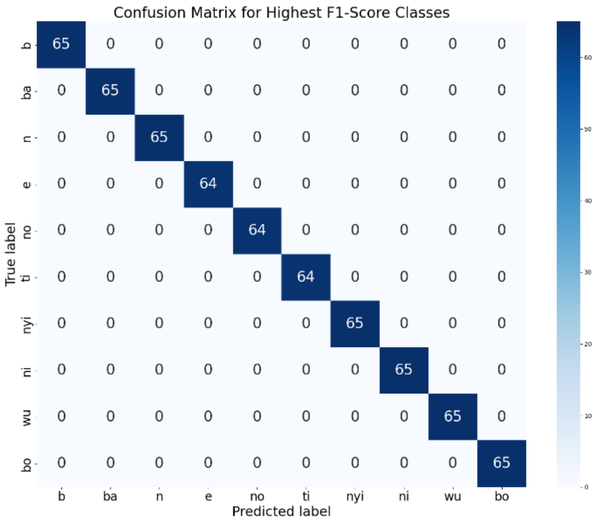

Figure 3. Best-Performing Classes Confusion Matrix

Analysis of the Worst-Performing Classes

In contrast, several classes exhibited significantly lower performance, indicating severe misclassification issues. The ten worst-performing classes are shown in Table 7.

Table 7. Worst-Performing Classes

Class	Precision	Recall	F1-Score
ri	0.484848	0.492308	0.488550
gi	1.000000	0.492308	0.659794
ru	0.480000	0.969231	0.640816
rang	0.650000	0.615385	0.632297
le	0.500000	0.492308	0.496124
ng	0.900000	0.430769	0.583333
ge	0.688889	0.476923	0.563636
g	1.000000	0.307692	0.470588
gang	0.863636	0.292308	0.436782
gu	0.818182	0.138462	0.236842

Morphological analysis suggests that most classification errors were caused by extreme visual similarities between characters and the loss of thin diacritical marks after image downscaling, shown in Figure 4.

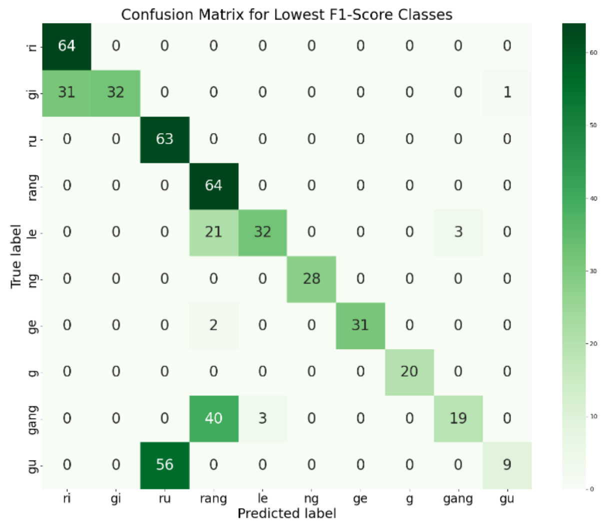

Figure 4. Worst-Performing Classes Confusion Matrix

The gu class exhibited the lowest recall value (13.85%). This character is formed by combining the base character ga with the vowel diacritic representing /u/ located beneath the main symbol. After resizing to 64 × 64 pixels, the spatial separation between the base character and its diacritic became significantly reduced or disappeared entirely due to interpolation effects. Consequently, many gu samples were incorrectly classified as the base character g or ga.

Similarly, the ri and ru classes experienced substantial confusion. The ri class achieved an F1-score of only 0.4885, while ru showed a low precision value of 0.4800 despite achieving a recall of 96.92%. These results indicate that the model frequently confused the upper-positioned /i/ diacritic with the lower-positioned /u/ diacritic. At low image resolutions, the model struggled to accurately distinguish the vertical placement of these markers, causing both classes to overlap in the feature space.

To overcome the classification bottleneck observed in diacritic-sensitive classes such as `gu`, `ri`, and `gang`, future architectural improvements should incorporate technical strategies such as adopting higher spatial resolutions (128 x 128 or 224 x 224 pixels) to preserve subtle stroke details, integrating spatial attention modules to amplify feature maps around small diacritics, applying Focal Loss to focus training gradients on hard-to-classify samples, and implementing hierarchical multi-task learning to predict base characters (*ina ni surat*) and diacritical modifiers separately.

Discussion of FastViT-SA12 Performance

The overall experimental results demonstrate the effectiveness and efficiency of the FastViT-SA12 architecture for recognizing Batak script characters across 117 distinct classes. The model successfully learned complex decision boundaries and extracted discriminative visual features from handwritten manuscripts through its rich hierarchical feature representations and pretrained weights. Data augmentation using the ImgAug library played a crucial role in stabilizing the training process by increasing the diversity of spatial and illumination conditions, which helped maintain the training-validation accuracy gap below 2%.

However, image resizing from 224 × 224 pixels to 64 × 64 pixels emerged as the primary factor limiting classification performance on the testing dataset. While the reduced resolution significantly accelerated computation and reduced memory consumption, it also removed fine-grained visual details that are essential for distinguishing Batak diacritical markers. This limitation contributed directly to the high misclassification rates observed in sensitive classes such as gu, ri, and gang.

From an architectural perspective, FastViT-SA12 demonstrated remarkable efficiency with a total of 10,670,005 trainable parameters. Through structural reparameterization techniques that merge multiple training branches into a single inference pathway, the model achieves low memory consumption and reduced inference latency without sacrificing its global feature representation capability. These characteristics make FastViT-SA12 highly suitable for deployment in mobile-first manuscript digitization systems. The model can potentially be integrated into smartphone-based applications capable of performing real-time scanning, segmentation, and transcription of Batak manuscripts directly on-device without requiring cloud-based processing, thereby supporting large-scale, cost-effective, and accessible digital cultural preservation initiatives.

Table 8. Summary of Previous Studies

Research Title	Authors and Year	Method and Dataset	Results
Design of Batak Toba Script Recognition System Using Convolutional Neural Network Algorithm	Steven Willian et al. (2024)	CNN with RMSprop optimizer on a handwritten Batak Toba script dataset consisting of 19 character classes.	Achieved an accuracy of 99.54% for handwritten Batak Toba script recognition.
Lightweight Deep Learning Model for Lung Ultrasound Image Scoring	Heliang Ye et al. (2025)	FastViT combined with Channel Shuffle on a Lung Ultrasound (LUS) image dataset.	Achieved 96.70% test accuracy and 95.14% external validation accuracy with only 3.2 million parameters.
LiCT-Net: Lightweight Convolutional Transformer Network for Multiclass Breast Cancer Classification	Babita and Deepak Ranjan Nayak (2024)	LiCT-Net based on FastViT using a multiclass breast cancer histopathology image dataset.	Achieved accuracies ranging from 94.21% to 96.16% across different image magnifications.
A Diabetic Retinopathy Classification Method Based on Image-Text Contrastive Learning	Fei Long et al. (2025)	FastViT-based image-text contrastive learning on the APTOS diabetic retinopathy dataset.	Achieved 92.38% accuracy for diabetic retinopathy detection.
Batak Script Recognition Based on FastViT for Traditional Manuscript Preservation in Digital Form	This Research (2026)	FastViT-SA12 on the Batak Char 20 dataset consisting of 117 Batak script classes with ImgAug-based data augmentation.	Achieved 89.31% accuracy, 91.84% precision, 89.31% recall, and 88.99% F1-score for Batak script recognition.

While previous research achieved a high accuracy of 99.54% on a limited dataset of 19 basic Batak Toba characters, the proposed FastViT-SA12 framework addresses a substantially more complex classification task encompassing 117 classes with intricate diacritical variations. FastViT-SA12 delivers competitive recognition performance across this expanded multi-class dataset while maintaining a compact parameter footprint of 10.67 million parameters (~41 MB model size), striking a favorable trade-off between representation capacity and computational efficiency suitable for resource-constrained environments.

4. CONCLUSION AND SUGGESTION

4.1. Conclusion

Based on the experimental results, this study successfully evaluated the FastViT-SA12 architecture for handwritten Batak script recognition across a comprehensive set of 117 character classes. The model achieved a training accuracy of 99.97% and a validation accuracy of 97.96% across 30 epochs, demonstrating rapid and stable convergence. On an independent test dataset, the proposed model achieved a test accuracy of 89.31%, with weighted precision, recall, and F1-score of 91.84%, 89.31%, and 88.99%, alongside macro-averaged precision, recall, and F1-score of 88.42%, 89.31%, and 88.10%, respectively. Diagnostic error analysis revealed that misclassifications primarily occurred in visually similar characters and fine diacritical markers (*anak ni surat*) that lost spatial separation after resizing to 64 x 64 pixels. However, this study is limited by the loss of high-frequency stroke details under aggressive spatial compression and the lack of direct hardware latency benchmarking on physical mobile platforms. Despite these limitations, FastViT-SA12 demonstrates strong overall recognition performance while maintaining a compact parameter footprint of 10.67 million parameters, proving its potential as a lightweight backbone for portable manuscript preservation systems.

4.2. Suggestion

For future research, several improvements can be considered to further enhance the performance and applicability of Batak script recognition systems. First, higher input image resolutions, such as 128 × 128 or 224 × 224 pixels, should be explored to preserve fine-grained visual details, particularly small diacritical marks that play a crucial role in distinguishing similar characters. Second, comparative studies involving other lightweight deep learning architectures, including MobileNetV3, GhostNet, and EfficientNet, are recommended to provide a more comprehensive evaluation of the trade-off between classification accuracy and computational efficiency. Finally, the trained FastViT-SA12 model can be deployed on mobile platforms by converting it into formats such as ONNX or TensorFlow Lite, enabling the development of an offline mobile-based Optical Character Recognition (OCR) application capable of recognizing handwritten Batak script directly in the field and supporting broader digital cultural preservation initiatives.

5. REFERENCE

Akallouch, O., Akallouch, M. and Fardousse, K. (2026) "TifinNet: CNN–Transformer Hybrid Architecture for Tifinagh Handwritten Recognition," *IEEE Access*, 14, pp. 34830–34844. Available at: https://doi.org/10.1109/ACCESS.2026.3669937.

Alghyaline, S. (2024) "Optimised CNN Architectures for Handwritten Arabic Character Recognition," *Computers, Materials and Continua*, 79(3), pp. 4905–4924. Available at: https://doi.org/https://doi.org/10.32604/cmc.2024.052016.

Ashimgaliyev, M. *et al.* (2026) "Development of Deep Learning Methods for Visual Document Classification Using Hybrid Vision Transformer–EfficientNet Architecture," *IEEE Access*, 14, pp. 28041–28053. Available at: https://doi.org/10.1109/ACCESS.2025.3646481.

Ataman, F. (2025) "Data Augmentation Techniques in Deep Image Processing," in *Advances in Medical Diagnosis, Treatment, and Care (AMDTC) Book Series*, pp. 231–262. Available at: https://doi.org/10.4018/979-8-3693-9816-6.ch010.

Babita and Nayak, D.R. (2024) "LiCT-Net: Lightweight Convolutional Transformer Network for Multiclass Breast Cancer Classification," in *TENCON 2024 - 2024 IEEE Region 10 Conference (TENCON)*, pp. 1359–1363. Available at: https://doi.org/10.1109/TENCON61640.2024.10902690.

Barillaro, L. (2025) "Deep Learning Platforms: PyTorch," in *Encyclopedia of Bioinformatics and Computational Biology (Second Edition)*. Oxford: Elsevier, pp. 167–170. Available at: https://doi.org/https://doi.org/10.1016/B978-0-323-95502-7.00093-2.

Carneiro, T. *et al.* (2018) "Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications," *IEEE Access*, 6, pp. 61677–61685. Available at: https://doi.org/10.1109/ACCESS.2018.2874767.

Gabriel, J. and Zhu, J. (2023) "FastViT : A Fast Hybrid Vision Transformer using Structural Reparameterization," *Computer Vision and Pattern Recognition* [Preprint]. Available at: https://doi.org/https://doi.org/10.48550/arXiv.2303.14189.

Hasan, M.M. *et al.* (2020) "Preprocessing of Continuous Bengali Speech for Feature Extraction," *2020 11th International Conference on Computing, Communication* *and Networking Technologies, ICCCNT 2020*, pp. 1–4. Available at: https://doi.org/10.1109/ICCCNT49239.2020.9225469.

Hasugian, J.H. (2025) "PUTUSNYA PEWARISAN PENGETAHUAN PUSTAHA LAKLAK BATAK: ANALISIS HISTORIS DAN SOSIO-KULTURAL," *JURNAL NAGUR*, 3(2), pp. 52–62.

Hatamizadeh, A. *et al.* (2024) "FASTERVIT: FAST VISION TRANSFORMERS WITH HIERARCHICAL ATTENTION," *ICLR*, pp. 1–24.

Hurtik, P., Molek, V. and Hula, J. (2020) "Data Preprocessing Technique for Neural Networks Based on Image Represented by a Fuzzy Function," *IEEE Transactions on Fuzzy Systems*, 28(7), pp. 1195–1204. Available at: https://doi.org/10.1109/TFUZZ.2019.2911494.

Ilić, A.S. (2020) "Preprocessing Image Data for Deep Learning," in *Sinteza*, pp. 312–317. Available at: https://doi.org/10.15308/SINTEZA-2020-312-317.

Jayachandran, S., Selvakumar, Y. and Pearline, S.A. (2025) "Multi-Attention Based Convolutional Neural Network for Tamil Handwritten Character Recognition," *IEEE Access*, 13, pp. 184360–184375. Available at: https://doi.org/10.1109/ACCESS.2025.3625588.

Jha, R.G. and Samlodia, A. (2024) "GPU-acceleration of tensor renormalization with PyTorch using CUDA," *Computer Physics Communications*, 294, p. 108941. Available at: https://doi.org/https://doi.org/10.1016/j.cpc.2023.108941.

Jia, Y., Zheng, S. and Gu, L. (2026) "Enhancing adversarial transferability via hybrid data and model augmentation," *Neurocomputing*, 697, p. 134144. Available at: https://doi.org/https://doi.org/10.1016/j.neucom.2026.134144.

Long, F. *et al.* (2025) "A diabetic retinopathy classification method based on image-text contrastive learning," *Pattern Recognition Letters*, 193, pp. 142–148. Available at: https://doi.org/https://doi.org/10.1016/j.patrec.2025.04.017.

Ran, Z. *et al.* (2025) "FastViT: Real-Time Linear Attention Accelerator for Dense Predictions of Vision Transformer (ViT)," in *2025 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5. Available at: https://doi.org/10.1109/ISCAS56072.2025.11043624.

Siregar, A.M. (2017) "Pelestarian Aksara Batak Toba sebagai Warisan Budaya Melalui Media Edukasi Interaktif," *Jurnal Bahasa dan Budaya*, 3(2), pp. 88–95.

Sohail, M. *et al.* (2024) "Deep Learning Based Multi Pose Human Face Matching System," *IEEE Access*, 12, pp. 26046–26061. Available at: https://doi.org/10.1109/ACCESS.2024.3366451.

Stephen, R.K. *et al.* (2025) "Fast Vision Transformer Framework for Proactive Diabetic Retinopathy Diagnosis in Fundus Images," in *International Conference on Intelligent Systems and Digital Transformation (ICISD 2025)*. Atlantis Press, pp. 774–786. Available at: https://doi.org/10.2991/978-94-6463-866-0_63.

Sudewo, E.D.B. *et al.* (2025) "Evaluating the Impact of Optimizer Hyperparameters on ResNet in Hanacaraka Character Recognition," *Preservation, Digital Technology & Culture*, pp. 1–11. Available at: https://doi.org/10.1515/pdtc-2024-0061.

Wang, Z. *et al.* (2026) "A Comprehensive Survey on Data Augmentation," *IEEE Transactions on Knowledge and Data Engineering*, 38(1), pp. 47–66. Available at: https://doi.org/10.1109/TKDE.2025.3622600.

Willian, S., Rochadiani, T.H. and Sofian, T. (2023) "Design of Batak Toba Script Recognition System Using Convolutional Neural Network Algorithm," 8(3), pp. 1609–1618.

Xu, D. *et al.* (2025) "DEVICE: Depth and Visual Concepts Aware Transformer for OCR-based image captioning," *Pattern*

Recognition, 164, p. 111522. Available at: https://doi.org/https://doi.org/10.1016/j.patcog.2025.111522.

Ye, H. *et al.* (2025) "Lightweight Deep Learning Model for Lung Ultrasound Image Scoring," in *2025 IEEE International Ultrasonics Symposium (IUS)*, pp. 1–3. Available at: https://doi.org/10.1109/IUS62464.2025.11201808.

Zhang, H. *et al.* (2021) "A Data Preprocessing Method for Automatic Modulation Classification Based on CNN," *IEEE Communications Letters*, 25(4), pp. 1206–1210. Available at: https://doi.org/10.1109/LCOMM.2020.3044755.