IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 2, April 2026, pp. 1513~1528
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i2.pp1513-1528

MNetNCR: MobileNet model for efficient traditional Nusantara script character recognition

Untari Novia Wisesty[1,2], Aditya Firman Ihsan[1], Mahmud Dwi Sulistiyo[1,2], Donni Richasdy[1], Prasti Eko Yunanto[1,2], Gamma Kosala[1], Arfive Gandhi[1], Febryanti Sthevanie[1,2]
[1]School of Computing, Telkom University, Bandung, Indonesia
[2]Center of Excellence Artificial Intelligence for Learning and Optimization, Telkom University, Bandung, Indonesia

Article Info

Article history:

Received Apr 26, 2025
Revised Jan 5, 2026
Accepted Jan 25, 2026

Keywords:

Character recognition
Cultural preservation
Handwritten script
MobileNetV3
Traditional Nusantara script

ABSTRACT

Preservation of traditional Nusantara scripts is very important because these traditional scripts are part of the cultural heritage that reflects the identity and history of the nation. This research proposed MobileNet for Nusantara character recognition (MNetNCR) model based on MobileNetV3 architecture to recognize traditional Nusantara scripts with lightweight, efficient architecture, and accurate recognition. The novel and comprehensive datasets for traditional Nusantara scripts have been curated in this research, that will later be stored digitally and can be used in further research. This novel dataset includes handwritten Balinese, Batak, Javanese, Lontara, and Sundanese scripts, each with unique visual characteristics. The proposed MNetNCR model is highly effective in recognizing characters, achieving F1-scores of 0.9934 for Balinese, 0.9450 for Batak, 0.9788 for Javanese, 0.9936 for Lontara, and 0.9961 for Sundanese scripts, according to the experimental results. The MNetNCR model built in this research has been proven to be effective and efficient in recognizing traditional scripts accurately. It also supports the preservation and promotion of the nation's cultural and historical heritage.

Corresponding Author:

Untari Novia Wisesty
School of Computing, Telkom University
Telecomunication street no. 1, Bandung, Indonesia
Email: untarinw@telkomuniversity.ac.id

1. INTRODUCTION

The traditional Nusantara script is a collection of traditional writing systems originating from various regions in Indonesia, such as the Javanese, Balinese, Sundanese, and Batak scripts [1]–[6]. Each of these scripts has a long and important history in recording and transmitting local knowledge, culture, and identity. These scripts are used in important documents like inscriptions, ancient manuscripts, and historical records that document the progress of Indonesian society over time. Traditional scripts have lost popularity among young people due to the dominance of the Latin alphabet in education and media. Modernization and globalization have led to a decrease in knowledge about traditional scripts. As a result, fewer people can read and write using these scripts. Without preservation and revitalization, this cultural wealth is at risk of being lost and forgotten over time.

Research on Nusantara script recognition through AI currently faces several critical limitations. Prasetiadi *et al.* [7] develop and apply techniques to detect characters in Nusantara scripts like Balinese, Batak, Bugis, and Javanese. This involves combining YOLOv5 for object detection and U-Net for image segmentation. They found that while the combination of YOLOv5 and U-Net delivered promising results, the accuracy of character segmentation ranged from 75% to 90%, which remains suboptimal. Similarly,

Fidatama *et al.* [8] using local binary patterns (LBP) and k-nearest neighbors (KNN) methods, achieved an accuracy of 86.056%, but it was constrained by specific parameters that may not generalize effectively in all contexts and still need specific preprocessing and feature extraction method. Razali *et al.* [9], [10] investigation into Jawi script was limited by a dataset consisting of only 15 classes, focusing on isolated scripts and failing to encapsulate the complexity of authentic handwriting using Freeman chain code (FCC) for feature extraction and support vector machine (SVM) for classification [9], and ResNet34 and InceptionV3 [10]. Although augmenting the Arabic script data improved accuracy, it also diminished the dataset's diversity and led to misclassification problems with characters that share similar shapes. Suparwito's [11] exploration utilizing YOLOv9 attained a high mean average precision (mAP) of 0.95 but struggled to distinguish between similar characters such as "ba," "la," and "sa," in addition to being confined to specific image formats and weight parameters. In addition, character recognition using YOLOv5 and YOLOv9 need more memory. Furthermore, research on the ancient Sundanese script employing Otsu threshold and convolutional neural networks (CNN) has made significant strides in addressing challenges related to imbalanced datasets and image degradation through sophisticated preprocessing techniques and yields 97.42 accuracy [12].

Dewi *et al.* [13] proposes an ensemble of deep CNNs for recognizing handwritten Balinese characters from palm-leaf manuscripts, aiming to digitize and preserve this cultural heritage. The methodology involves selecting three pre-trained deep learning models, like ResNet, EfficientNet, MobileNet, and swin transformer. The ensemble method significantly outperformed single deep learning models and previous studies, achieving state-of-the-art results with the third ensemble model showing the best performance in precision 0.8644, recall 0.8940, and F1-score 0.8699. While the ensemble improves accuracy, it also increases model complexity, suggesting future work could focus on noise reduction through preprocessing and knowledge distillation to create lighter models for real-time deployment. In summary, several research approaches still rely on specific preprocessing techniques and additional feature extraction methods such as LBP and FCC. These processes increase computational time and require extensive hyperparameter tuning. Furthermore, deep learning methods like YOLO, U-Net, ResNet34, InceptionV3, and ensemble deep learning employ larger architectural designs, demanding greater memory resources and computational power. Also, some previous research only focused on one script, such as Balinese script. Table 1 shows comparative analysis and taxonomy of the previous research.

Therefore, this research proposes MobileNet for Nusantara character recognition (MNetNCR), a MobileNet-based traditional Nusantara script recognition model covering Sundanese, Balinese, Javanese, Lontara, and Batak scripts, in collaboration with local experts to create a handwritten dataset of traditional Nusantara scripts. MobileNetV3 was chosen because it has a lightweight and efficient model achieved through the use of inverted residual (bottleneck) blocks and selective squeeze-and-excitation (SE) modules. This technique reduces parameters and computations in comparison to traditional convolutional networks, making MobileNetV3 suitable for use on devices with limited memory and computation [14]–[16]. MobileNetV3 is really fast, so it can be used in real-time on devices like smartphones, IoT devices, and edge computing platforms with limited hardware. In addition, the MobileNet architecture is highly scalable, allowing for a balance between model size, speed, and accuracy. These parameters allow adapting the model to specific needs, and balance between achieved accuracy and computational efficiency.

This research makes several important contributions that will be instrumental in training deep learning models to identify and understand traditional Nusantara scripts, thereby aiding in the preservation and promotion of Nusantara culture and languages. The proposed research presents the following main contributions. The first contribution is that we curate a novel and comprehensive dataset for Nusantara traditional scripts, covering a variety of handwritten scripts including Sundanese, Balinese, Javanese, Lontara, and Batak [17]. This dataset preserves culturally important scripts and supports research in computational linguistics, digital preservation, and machine learning for recognizing and processing unique writing systems and can be used as a benchmark dataset in recognizing handwritten Nusantara script. Second, we present a lightweight AI framework for low-resource script recognition, combining MobileNetV3 with culturally curated datasets and interpretability tools to advance character-level understanding in heritage digitization. This model eliminates the need for specific preprocessing techniques and additional feature extraction, enabling direct image processing and generating corresponding labels. The third contribution, the proposed MNetNCR model have high accuracy and computational efficiency. The model can recognize and classify traditional scripts efficiently, and it can be used on devices with limited resources. By leveraging the lightweight nature of MobileNet, our model not only achieves superior recognition performance but also ensures accessibility and scalability, making it a practical tool for a variety of applications, including educational tools and cultural preservation initiatives.

The manuscript is organized into four sections. The second section gives a detailed overview of the data specifications and describes the proposed methodology. The third section presents and analyses the

experimental results, discussing the implications of the findings and highlighting the main trends and patterns observed during the research. Finally, the last section summarizes the main conclusions of the study, summarizes the main contributions, discusses potential limitations, and discusses future work to build on the findings presented in this research.

Table 1. Comparative analysis and taxonomy of previous research

Previous research	Method	Script	Metric performance	Resource requirement	Limitation
[7]	YOLOv5+U-Net	Balinese and Batak	75%-90% (segmentation)	High	Focus on detection and segmentation with sub-optimal accuracy
[8]	LBP and KNN	Bima	86.05% accuracy	Low (traditional machine learning)	Limited to one type of script, relies on manual feature extraction.
[9]	FCC and SVM	Jawi	92.86% accuracy	Low (traditional machine learning)	Limited to one type of script, relies on manual feature extraction.
[10]	ResNet34 and InceptionV3	Jawi	96% accuracy	High	Limited to one type of script.
[11]	YOLOv9	Javanese	0.95 mAP (script detection)	High	Limited to one type of script.
[12]	Otsu threshold and CNN	Sundanese	97.42 accuracy	Low	Limited to one type of script.
[13]	ResNet, EfficientNet, MobileNet, swin transformer	Balinese	F1-score 0.8699	High (ensemble CNN)	Computation and memory needed is really high.
Proposed model	MobileNetV3	Sundanese, Balinese, Javanese, Lontara, and Batak	F1-scores 0.9934 for Balinese, 0.9450 for Batak, 0.9788 for Javanese, 0.9936 for Lontara, and 0.9961 for Sundanese scripts.	Low/Lightweight	

2. METHOD

The proposed methodology is the development of MNetNCR model, a traditional Nusantara script recognition model using the MobileNetV3 architecture, which is designed to produce high performance in terms of accuracy and computational efficiency. This model is optimized to recognize various Nusantara scripts including Javanese, Balinese, Sundanese, Lontara, and Batak scripts, with the aim of supporting cultural preservation and expanding the accessibility of script recognition technology. Algorithm 1 provides a comprehensive description of our proposed MNetNCR model.

Algorithm 1. Proposed MNetNCR model
Input: traditional Nusantara script dataset.
Output: label prediction and performance metrics (precision, recall, and F1-score).
1: Image standardization in dataset to 100×100 pixels.
2: Divide the dataset into train, validation, and testing set.
3: Set trainable layer observation: front layer, back layer, and all layer.
4: For each trainable layer observation do:
5: Do the transfer learning for MobileNetV3-large model.
6: Fine tuning the trainable layer of MobileNetV3-large using train set and Adam optimizer.
7: Predict the label for each image in train set using trained model.
8: Predict the label for each image in validation set using trained model.
9: Calculate train accuracy
y based on train set and validation accuracy based on validation set.
10: End for
11: Choose the best MobileNetV3 model that has been trained with the highest validation accuracy.
12: Save the best MobileNetV3 model params.
13: Predict the label for each image in test set using the best MobileNetV3 model.
14: Calculate precision, recall, and F1-score based on test set.

2.1. Dataset specification and exploration

The Nusantara script dataset built in this research is a dataset containing characters from traditional Indonesian scripts, namely Javanese, Balinese, Sundanese, Lontara, and Batak scripts. The selection of these scripts was based on their distinctive features and varying levels of visual intricacy, providing diverse challenges for deep learning-based pattern recognition systems. These datasets are curated to document and facilitate research and technology development related to traditional script recognition. This dataset contains scripts with different writing styles and variations of individual characters. The styles are from both historical and modern contexts. This dataset aims to support the preservation of Nusantara languages and cultures while also being a basis for the development of applications such as automatic text recognition, language teaching, and digital preservation. The first step to curate the dataset was to set the standard for volunteers to write the specified script on the paper provided. Standardization covers writing layout, character size, and writing methods that must follow the rules. Volunteers are given comprehensive guidance from several experts who understand local culture and the rules of writing Nusantara script, so that the characters written not only meet aesthetic standards, but also historical and cultural accuracy. The template given to volunteers is in the form of a square box for each character of a particular script. The boxes are all the same size to make sure that each character is written neatly and uniformly. This also makes it easier to digitize the text.

After the volunteers have finished writing the Nusantara script on the paper provided, the next step is to digitize the handwritten results. To digitize a script, start by using a high-quality scanner to carefully scan the paper. This will capture all the details of the written characters. The scanned image is carefully cropped into separate character grids based on a pre-determined box. Each separated character image is then further processed to be standardized in terms of its ratio and size. The standardization process adjusts the size of each character to 100×100 pixels, ensuring that all characters in the dataset have the same dimensions. This is very important to ensure that the resulting dataset has uniform quality, which in turn will affect the performance of the developed traditional Nusantara script recognition model. The final process makes sure that the curated dataset is not only usable but also has high scientific value. This enhances the model's ability to accurately recognize scripts. Sample characters from the Batak script dataset are presented in Figure 1.

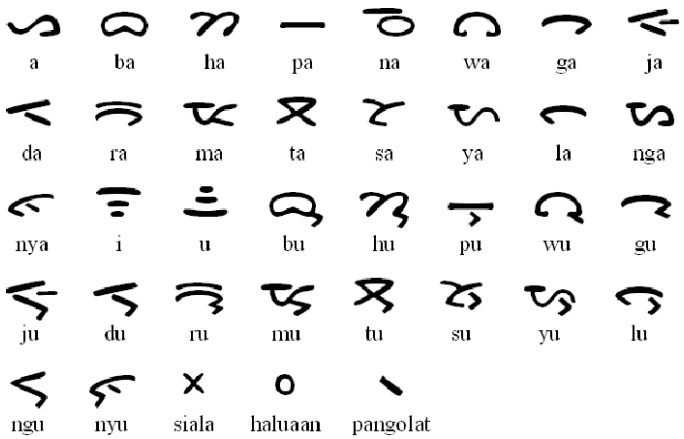


Figure 1. Sample characters of Batak script

Table 2 shows the specifications of the datasets that have been built and will be used in this study. The datasets built consist of Javanese, Balinese, Sundanese, Lontara, and Batak script datasets. The script datasets have different numbers of characters, namely 20 for Javanese, 18 for Balinese, 32 for Sundanese, 23 for Lontara, and 19 for Batak. Each dataset is then divided into training, validation, and testing data with a proportion of 70:20:10. Table 3 shows the amount of data in training, validation, and testing data. Training data is used to teach the model to recognize patterns and make predictions. Validation data is used to assess the model's performance during training and improve it. Once the model is fully trained and optimized, testing data is used to independently test the model and verify its ability to make accurate predictions on new data. Dividing the dataset is crucial to ensure that the model developed performs well not only on the training data, but also on new data. This process of dividing the dataset into different subsets allows for a comprehensive evaluation of the model's performance and generalization capabilities. It helps ensure that the model is not overfitting to the training data and is capable of making accurate predictions on unseen data, which is essential for real-world applications.

Table 2. Dataset specification of traditional Nusantara script

Dataset	Character/class	#Class
Javanese script	"ba", "ca", "da", "dha", "ga", "ha", "ja", "ka", "la", "ma", "na", "nga", "nya", "pa", "ra", "sa", "ta", "tha", "wa", "ya"	20
Balinese script	"ba", "ca", "da", "ga", "ha", "ja", "ka", "la", "ma", "na", "nga", "nya", "pa", "ra", "sa", "ta", "wa", "ya"	18
Sundanese script	"a", "ba", "ca", "da", "e", "eu", "fa", "ga", "ha", "i", "ja", "ka", "kha", "la", "ma", "na", "nga", "nya", "o", "pa", "qa", "ra", "sa", "sya", "ta", "u", "va", "wa", "xa", "ya", "za", "é"	32
Lontara script	"a", "ba", "ca", "da", "ga", "ha", "ja", "ka", "la", "ma", "mpa", "na", "nca", "nga", "ngka", "nra", "nya", "pa", "ra", "sa", "ta", "wa", "ya"	23
Batak script	"a", "ba", "da", "ga", "ha", "i", "ja", "la", "ma", "na", "nga", "nya", "pa", "ra", "sa", "ta", "u", "wa", "ya"	19

Table 3. Amount of data in train, validation, and test set

Nusantara script	Train set	Validation set	Test set	Total data
Balinese script	3,140	894	459	4,493
Batak script	1,330	380	190	1,900
Javanese script	6,988	1,990	1,017	9,995
Lontara script	1,609	459	231	2,299
Sundanese script	14,140	4,031	2,053	20,224

2.2. MobileNetV3 architecture

MobileNetV3 is a type of CNN that was specifically developed to work well on devices with limited resources like smartphones and IoT devices. The third-generation MobileNetV3 combines neural architecture search (NAS), SE blocks, and a new hard-swish activation function to deliver excellent accuracy while using little computational power on mobile devices [14]–[18]. This technique reduces parameters and computational operations needed compared to standard convolution, which combines filtering and pooling in one step. In depthwise separable convolution, each input channel is convolved separately to capture basic spatial features, and then pointwise convolution combines these features across channels to obtain more complex. However, it still maintains good image recognition performance. MobileNetV3 is a great option for mobile and edge computing applications [19]–[21]. It is efficient and works well even with limited computing power and memory. It provides reliable and accurate image and object recognition. MobileNetV3 has several advantages that make it suitable for various practical applications such as face recognition, image classification, object detection, and augmented reality [22]–[27].

Figure 2 shows the MobileNetV3 architecture used in this research. The architecture used consists of an input image, a 2D convolutional layer, a series of inverted residual bottleneck stage, 2D global average pooling, two dense layers (each layer with rectified linear unit (ReLU) activation function and dropout), and dense layers with SoftMax activation function. The input images consist of previously curated traditional Nusantara script samples. The input image will be processed by the initial 2D convolutional layer using a 3×3 kernel size to quickly reduce the input size and capture basic image features. This layer is followed by batch normalization (BN) and ReLU activation function to maintain gradient stability during training. These stages implement the efficient depthwise separable convolution pattern, specifically composed of an expansion 1×1 convolution (to a large channel count), a depthwise convolution (using varying kernel sizes like 3×3 or 5×5), an optional SE block for channel recalibration, and a final projection 1×1 convolution (to a smaller channel count). These bottleneck stages control feature resolution and computational cost by utilizing varying expansion ratios and strides. A critical design feature for efficiency is the strategic use of activation functions, a mix of ReLU6 and the computationally friendly h-swish (hard swish) activation as in (1) is employed, alongside the selective inclusion of the SE module. The network culminates in a final head, which consists of a 1×1 convolution to expand to a large channel count, followed by global pooling, and a small sequence of fully connected (FC) layers and a SoftMax function for classification. This specific stacking of components ensures both high performance and minimal computational burden.

The core element of MobileNet is depthwise convolution and pointwise convolution. In depthwise convolution, convolution is applied to each input channel independently. If the input image has multiple channels (e.g., red, green, and blue), depthwise convolution applies a separate filter to each channel without combining them. The depthwise convolution process is represented in (2), where the input used is I with dimensions $H \times W \times C$, and the filter K with size $F_H \times F_W \times C$, and $O_{h,w,c}$ is the output value resulting from the convolution at position (h, w) in channel C. Pointwise convolution uses a 1×1 kernel size to combine the depthwise convolution results from each channel into an output with the desired dimension. Pointwise convolution allows the MobileNet architecture to process information between channels linearly, which produces a new feature map by combining information from all channels. The pointwise convolution process is represented in (3), where the input I with dimensions $H \times W \times C$, and the filter K with dimension

$1 \times 1 \times C \times N$ (which N is desired number of filters or output channels), and $O_{h,w,n}$ is the output value resulting from the convolution at position (h, w) for output channel n [28].

$$h - swish(x) = x\frac{ReLU6(x+3)}{6} \quad (1)$$

$$O_{h,w,c} = \sum_{i=0}^{F_H-1} \sum_{j-0}^{F_W-1} I_{h+1,w+j,c} \cdot K_{i,j,c} \quad (2)$$

$$O_{h,w,n} = \sum_{c=0}^{C-1} I_{h,w,c} \cdot K_{1,1,c,n} \quad (3)$$

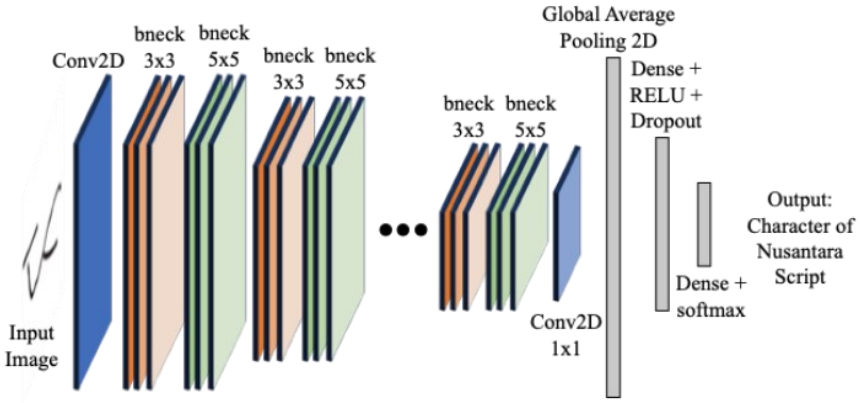


Figure 2. Proposed MNetNCR architecture for traditional Nusantara script

The SE block uses a four-layer structure to learn channel attention, balancing efficiency and representational strength [29]–[31]. The initial layer employs global average pooling to execute the "squeeze" operation, transforming input feature maps with dimensions of [C×H×W] into concise descriptors sized [C×1×1]. This process involves averaging the spatial values of each channel to capture a broader context, effectively condensing the H×W elements into a single scalar for each channel. The second layer introduces a FC transformation with ReLU activation that reduces dimensionality from C channels to C/r channels, where r is the reduction ratio, creating a computational bottleneck that serves dual purposes: minimizing parameter overhead while forcing the network to learn compressed channel relationship representations through non-linear transformations. The third layer performs the excitation operation using an extra FC layer, converting the channel size from C/r back to C channels. This is succeeded by a hard-sigmoid activation function that generates normalized attention weights ranging from 0 to 1. This setup allows the network to prioritize or reduce certain channels based on the global context obtained during the squeeze phase. The fourth layer recalibrates channels by multiplying the learned attention weights with the original input features, enhancing their significance. This scales each channel by its attention coefficient to produce the final output [C×H×W]. This four-layer design allows SE blocks to efficiently adjust feature responses for each channel, making them ideal for mobile architectures like MobileNetV3. It maximizes efficiency while still learning complex channel relationships that enhance model performance.

This research uses the MobileNetV3-Large architecture, which has 14 inverted residual blocks (bottlenecks) and SE modules in inverted residual blocks 4-6 and 11-15. The SE module is placed between depthwise convolution (after the activation function) and projection convolution. The global average pooling layer is applied after the last inverted residual block. This layer calculates the global average of the convolution blocks' results. This layer takes each feature channel and averages its values, creating a more condensed feature vector and reducing the number of parameters. A FC layer (dense layer) with 512 neurons, ReLU activation function, and dropout added. In the last layer, there is the final dense layer with a SoftMax activation function for character classification. The number of neurons used in final dense layer is in accordance with the number of classes/characters of each dataset. For example, in the Lontara script dataset, there are 23 characters, so the number of neurons in the final dense layer is also 23. The detail of proposed MNetNCR model based on MobileNetV3 is shown in Table 4. Then, the Adam optimization algorithm is used in the training process, with a scheduler learning rate between 1e-04 to 1e-06, a batch size of 64, and 50 epochs.

Table 4. Detail of proposed MNetNCR model

Layer name	Operators
Input layer	
2D convolution	Conv2D, BN, ReLU
Inverted residual block 1-3:	Expanded Conv2D, BN, activation function, zero padding 2D Depthwise Conv2D, BN, activation function Projection Conv2D, BN
Inverted residual block (with SE module) 4-6:	Expanded Conv2D, BN, activation function, zero padding 2D Depthwise Conv2D, BN, activation function SE block (global average pooling 2D, SE Conv2D, ReLU, hard sigmoid, multiply) Projection Conv2D, BN
Inverted residual block 7-10:	Expanded Conv2D, BN, activation function, zero padding 2D Depthwise Conv2D, BN, activation function Projection Conv2D, BN
Inverted residual block (with SE module) 11-15:	Expanded Conv2D, BN, activation function, zero padding 2D Depthwise Conv2D, BN, activation function SE block (global average pooling 2D, SE Conv2D, ReLU, hard sigmoid, multiply) Projection Conv2D, BN
Global average pooling 2D	Global average pooling 2D
Dense layer	Dense layer (512), ReLU, dropout
Dense layer for classification	Dense layer (num_class), SoftMax function

3. RESULTS AND DISCUSSION

In this section, we present the experiments results conducted to test the performance of the MobileNet model in recognizing traditional Nusantara scripts, including Javanese, Balinese, Sundanese, Lontara, and Batak scripts. Each script has unique characteristics in terms of shape and writing pattern, which adds complexity to the recognition task. Therefore, this experiment is designed to evaluate how effectively the proposed model can recognize and classify these script characters with high accuracy. The experimental framework consists of three phases: i) trainable layer observation within the MobileNet architecture to establish optimal layers for each Nusantara script, ii) performance analysis on five Nusantara script datasets, and iii) comprehensive comparison between the proposed model and other deep learning architectures. The results of this experiment will be analyzed in depth and measured using accuracy, precision, recall, and macro F1-score metrics for each script dataset used.

3.1. Trainable layer observation within the MobileNetV3 architecture

The first experiment is designed to evaluate the effectiveness of training on different layers of the MobileNetV3 architecture in recognizing all characters in five types of traditional Nusantara scripts, namely Balinese, Batak, Javanese, Lontara, and Sundanese scripts. These scripts were chosen because each one has unique characteristics and visual complexity, which presents different challenges for pattern recognition by deep learning models. Three training scenarios are used in this experiment: i) training the front layers to capture basic features, ii) training the back layers to focus on complex features, and iii) training all layers to process information from early to late stages, potentially giving the model the ability to understand features of different levels of complexity. Each type of script will be used to train the model in these three scenarios, with the aim of identifying which training approach provides the highest validation accuracy, namely the ability of the model to generalize the learned patterns to data that has not been seen before in the training process, which is very important in real-world applications. In addition to conducting experiments on the trainable MobileNetV3 layer, data augmentation experiments also conducted on the training data to improve the model's robustness and generalization. The data augmentation applied includes rotation (20), width shift (0.2), height shift (0.2), shear (0.20), zoom (0.2), and horizontal flip (true).

The experimental results in Table 5 demonstrate that the model's training approaches for different layers and augmentation data can yield diverse outcomes depending on the script type used. This analysis shows that, in general, the fine-tuning process on all layers accompanied by augmentation consistently produces the highest performance on most complex datasets such as Balinese, Batak, and Lontara. For example, on the Balinese script without augmentation with fine-tuning front layers, the accuracy is around 0.9709 and F1-score is 0.9708. However, when augmentation is applied and all layers are trained, the accuracy increases to 0.9866 and F1-score to 0.9888, indicating that deeper feature representation coupled with data generalization through augmentation strengthens the model's ability to capture letter variations. Similarly, for the Batak and Lontara scripts, the all layers and augmentation configuration achieved the highest evaluation scores, with an accuracy of 0.9211 and an F1-score of 0.9363 for the Batak script, and an accuracy of 0.9630 and an F1-score of 0.9719 for the Lontara script, confirming that the combination of greater network depth and increased synthetic data helped the model balance precision and recall. However, for the Javanese and Sundanese scripts, where sufficient data was already available, the use of data augmentation did not have a significant impact. For example, for the Javanese script, fine-tuning the front

layers without augmentation demonstrated high accuracy and F1-scores of 0.9839 and 0.9834, respectively. Meanwhile, for the Sundanese script, the highest validation F1-score of 0.9912 was achieved when fine-tuning the back layers with augmentation. However, these results were not significantly different from the other configurations, indicating that the Sundanese feature distribution is quite stable. Based on several experiments, applying data augmentation to training data has a significant impact on small data sets because the process can increase the variety of data patterns. This demonstrates the importance of tailoring the training strategy to the specific characteristics of each script type.

Table 5. Trainable layer and augmentation data observation on MobileNetV3 architecture for traditional Nusantara script recognition

Dataset	Augmentation	Trainable layer	Validation accuracy	Validation F1-score
Balinese script	No	Front layers	0.9709	0.9708
		Back layers	0.8758	0.8770
		All layers	0.9620	0.9620
	Yes	Front layers	0.9732	0.9731
		Back layers	0.8031	0.8013
		All layers	0.9866	**0.9888**
Batak script	No	Front layers	0.8000	0.8061
		Back layers	0.5737	0.5756
		All layers	0.7132	0.7048
	Yes	Front layers	0.7895	0.7821
		Back layers	0.7105	0.7078
		All layers	0.9211	**0.9363**
Javanese script	No	Front layers	0.9839	**0.9834**
		Back layers	0.9307	0.9195
		All layers	0.9769	0.9706
	Yes	Front layers	0.9548	0.9535
		Back layers	0.8724	0.8521
		All layers	0.9749	0.9747
Lontara script	No	Front layers	0.8736	0.8755
		Back layers	0.7168	0.6705
		All layers	0.4336	0.4230
	Yes	Front layers	0.8039	0.7859
		Back layers	0.7473	0.7411
		All layers	0.9630	**0.9719**
Sundanese script	No	Front layers	0.9744	0.9749
		Back layers	0.9896	0.9897
		All layers	0.9883	0.9887
	Yes	Front layers	0.9829	0.9830
		Back layers	0.9878	**0.9912**
		All layers	0.9871	0.9875

3.2. Performance analysis on Nusantara script datasets

To assess true generalization, the optimal model for each script is evaluated using a dedicated testing set comprising samples entirely distinct from those used during training and validation. Performance is measured through accuracy, precision, recall, and the macro F1-score. These metrics provide a comprehensive overview: precision evaluates prediction correctness, recall assesses the identification of positive instances, and the F1-score offers a balanced harmonic mean of both. This rigorous evaluation identifies the model's practical strengths and weaknesses, ensuring its reliability and consistency in real-world Nusantara script recognition scenarios. The experimental results shown in Table 6 illustrate the performance of the MobileNet model in recognizing five types of traditional Indonesian scripts (Balinese, Batak, Javanese, Lontara, and Sundanese) with and without data augmentation. Overall, the results show that augmentation significantly improves performance on most datasets, particularly on the Batak and Lontara scripts, which have limited data and high levels of visual variation between characters. On these two datasets, the F1-score increased by more than 11-13%, indicating that augmentation effectively improved the model's generalization ability. This improvement means the model is able to recognize more varied character patterns and is robust to distortion, rotation, and lighting changes. Conversely, on datasets like the Javanese and Sundanese scripts, which already perform well even without augmentation, the effect of augmentation decreased slightly or was not statistically significant. This confirms that augmentation does not always provide universal benefits but must be tailored to the specific dataset.

The experimental results demonstrate that data augmentation significantly enhances model performance for most Nusantara scripts, though its impact varies based on script complexity. For the Balinese script, metrics improved consistently, with accuracy rising from 0.9673 to 0.9935. This suggests that augmentation effectively accommodates complex ornamentation by broadening data representation

without causing overfitting, resulting in a superior balance between precision and recall. The most significant improvement occurred in the Batak script, where the F1-score surged from 0.8090 to 0.9450 and accuracy increased by 13%. This highlights that augmentation is most effective for datasets with high intra-class variance or limited samples, as it enables the model to better distinguish between visually similar characters. Meanwhile, on the Lontara script, augmentation yielded very positive results. Accuracy increased from 0.8615 to 0.9784, with an F1-score increase of over 11%. This demonstrates a similar effect to the Batak dataset, were augmentation significantly improved model generalization. Furthermore, the difference between precision and recall is very small (less than 0.003), indicating a good predictive balance between avoiding false positives and capturing all positive cases. Unlike the previous scripts, Javanese script actually experienced a slight performance decline. The accuracy value decreased from 0.9803 to 0.9636, and the F1-score dropped by about 2%. This small decrease indicates that overly aggressive augmentation can excessively change the visual shape of letters, thus worsening the model's ability to recognize patterns that are already well-learned. This means that augmentation needs to be applied selectively, for example by limiting extreme rotations or distortions that can change the distinctive structure of Javanese script. For Sundanese script, both without and with augmentation, performance remained high, with an F1-score of around 0.993. The small decrease (around 0.1-0.2%) after augmentation can be considered statistically insignificant. This suggests that the Sundanese dataset likely already has good data quality and variety, so augmentation no longer provides significant benefits. The model is already at the point of optimal convergence, and the addition of synthetic variation introduces little noise. Detailed classification errors and successes are further illustrated via the confusion matrix in Figure 3.

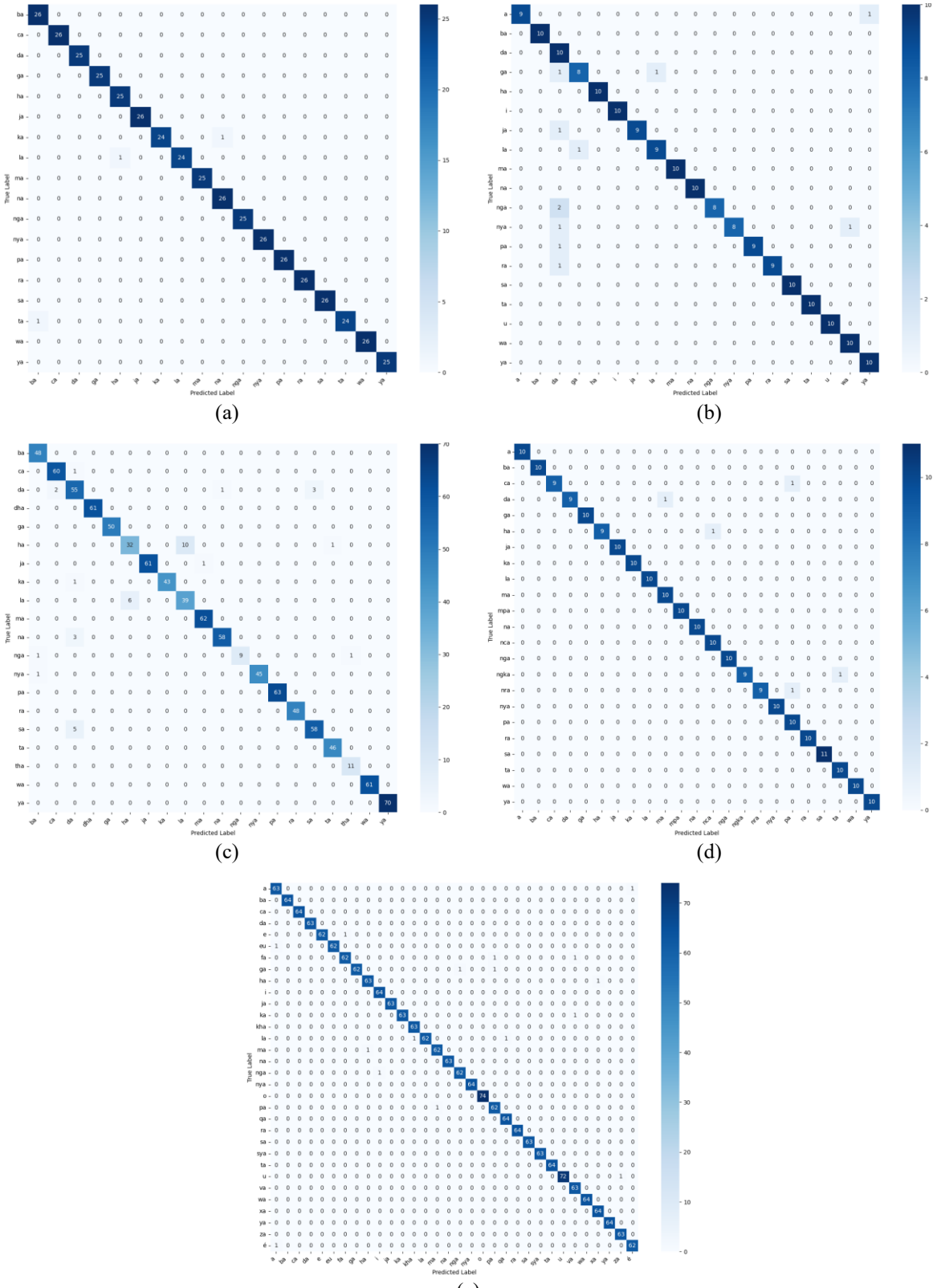
Table 6. Best MobileNet model testing performance for traditional Nusantara script recognition

Dataset	Augmentation	Accuracy	Precision	Recall	F1-score
Balinese script	No	0.9673	0.9687	0.9673	0.9672
	Yes	0.9935	0.9937	0.9933	0.9934
Batak script	No	0.8105	0.8790	0.8105	0.8090
	Yes	0.9421	0.9576	0.9421	0.9450
Javanese script	No	0.9803	0.9798	0.9785	0.9788
	Yes	0.9636	0.9613	0.9563	0.9579
Lontara Script	No	0.8615	0.9346	0.8613	0.8636
	Yes	0.9784	0.9809	0.9783	0.9784
Sundanese script	No	0.9937	0.9937	0.9936	0.9936
	Yes	0.9922	0.9922	0.9922	0.9922

A detailed analysis of the model's performance is presented through confusion matrix in Figure 3, illustrating the classification results across various traditional Nusantara scripts, where Figure 3(a) shows the Balinese script, Figure 3(b) shows the Batak script, Figure 3(c) shows the Javanese script, Figure 3(d) shows the Lontara script, and Figure 3(e) shows the Sundanese script. A thorough analysis of the confusion matrix of the five Nusantara scripts reveals that while the MNetNCR model achieved high overall accuracy, most of the misclassification errors were concentrated on issues of high inter-class similarity. This pattern of weakness was particularly evident in the Javanese script, where significant major errors were observed (31.3% of 'ha' samples were misclassified as 'la') and substantial confusion between visually similar characters ('la' vs. 'ha', 'sa' vs. 'da'). Susceptibility to stroke similarity was also prevalent in the Balinese and Sundanese scripts. Meanwhile, the Batak script exhibited a distributed error pattern where the character 'da' acted as an error attractor, absorbing 7 errors from 6 different characters, indicating that the biased towards the most dominant stroke pattern when faced with visual uncertainty due to the lack of standardization of letter shapes and sizes. The geometric Lontara script certainly faced its challenges, where the model showed minor confusion between similar single characters ('ca' vs. 'pa' and 'ngka' vs. 'ta'), highlighting the challenge in distinguishing basic elements. These error limitations collectively demonstrate that MobileNetV3's computational efficiency comes with a trade-off, namely it fails to capture minor discriminatory features that separate characters with extreme visual similarity.

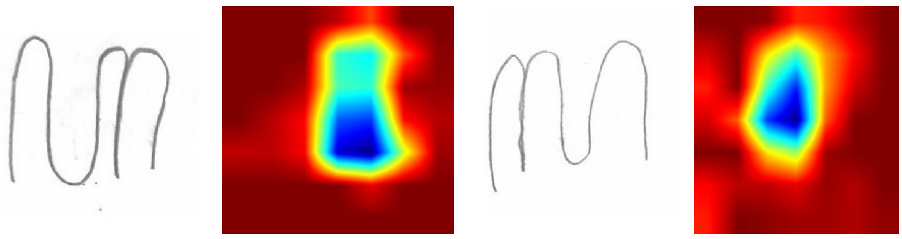
Figure 4 presents gradient-weighted class activation mapping (Grad-CAM) results for 'ha' (Figure 4(a)) and 'la' (Figure 4(b)) in Javanese script, indicating the model's attention during classification. This visualization shows which areas of the image contribute most to the model's decisions. The blue-green areas for the character 'ha' show that the model focuses on the main curves of its vertical structure and double wave. The surrounding red areas indicate low-contribution zones, demonstrating that the model successfully ignores the background and irrelevant edges. For the character 'la', the model focuses on the top middle, shown by the blue to bright green area, which sets it apart from 'ha'. The model successfully identifies key differences between characters, especially the subtle middle curve structures that often cause errors in typical optical character recognition (OCR) systems. Grad-CAM results show that the proposed model has an effective attention mechanism, successfully identifying key areas of character shapes without interference

from noise or the background. Improving model interpretability supports the idea that accuracy gains come from a real grasp of Javanese script structure instead of just overfitting to patterns.



(a) (b) (c) (d) (e)

Figure 3. Testing confusion matrix for traditional Nusantara script using MobileNet, (a) Balinese Script, (b) Batak Script, (c) Javanese Script, (d) Lontara Script, and (e) Sundanese Script



(a) (b)

Figure 4. Grad-CAM visualization for (a) 'ha' character and (b) 'la' character in Javanese script

3.3. Comparative analysis across model

Finally, the best MobileNetV3 model for each script is compared with ResNet50, InceptionV3, EfficientNetV2, and tiny vision transformer (TinyViT) models to evaluate performance advantages and computational efficiency. This comparison involves analysis based on key evaluation metrics such as accuracy and macro F1-score, which allow identifying the strengths and weaknesses of each model in recognizing traditional Nusantara scripts. By comparing MobileNetV3 with other more complex models, such as ResNet50 which is known for its deep residual architecture [32], [33], InceptionV3 which uses inception modules to capture various feature scales [34], [35], EfficientNetV2 which using fused mobile inverted residual bottleneck convolution (Fused-MBConv) to speed up the training process [36]–[38], TinyViT which is a smaller version of ViT [39], [40], this research can provide deeper insights into how each model handles the task of recognizing character patterns with various levels of complexity. Furthermore, to ensure that the performance differences between models, 10-fold cross-validation used on each comparison and calculated the p-value to analyze the statistical significance of the differences in the obtained metric results.

Table 7 presents the performance of different deep learning models on five Nusantara scripts, namely Balinese, Batak, Javanese, Lontara, and Sundanese. Each model is evaluated based on mean accuracy, mean F1-score, standard deviation of the F1-score, and p-value indicating the statistical significance of the comparison against the proposed model. If p-values are less than 0.05, this suggests that the differences observed are statistically significant.

Table 7. Performance comparison across model for traditional Nusantara script recognition

Dataset	Model	Mean accuracy	Mean F1-score	Stdev F1-score	p-value (compared to proposed model)
Balinese script	Proposed model	0.9913	0.9913	0.0039	-
	EfficientnetV2	0.9734	0.9734	0.0041	2.40E-06
	InceptionV3	0.9845	0.9846	0.0067	1.76E-02
	ResNet50	0.9813	0.9813	0.0045	1.24E-03
	TinyViT	0.9861	0.9861	0.0031	9.33E-04
Batak script	Proposed model	0.9247	0.9211	0.0376	-
	EfficientnetV2	0.7247	0.7072	0.0405	7.59E-08
	InceptionV3	0.9547	0.9546	0.0203	3.84E-02
	ResNet50	0.9079	0.9047	0.0309	3.71E-01
	TinyViT	0.9763	0.9759	0.0088	2.02E-03
Javanese script	Proposed model	0.9722	0.9654	0.0070	-
	EfficientnetV2	0.9728	0.9686	0.0037	0.2730
	InceptionV3	0.9811	0.9795	0.0037	0.0007
	ResNet50	0.9790	0.9757	0.0046	0.0114
	TinyViT	0.9678	0.9595	0.0070	0.0885
Lontara script	Proposed model	0.9823	0.9815	0.0224	-
	EfficientnetV2	0.7307	0.7224	0.0885	1.53E-05
	InceptionV3	0.9732	0.9722	0.0253	3.38E-01
	ResNet50	0.8762	0.8711	0.0458	1.13E-04
	TinyViT	0.9922	0.9921	0.0065	1.52E-01
Sundanese script	Proposed model	0.9850	0.9850	0.0045	-
	EfficientnetV2	0.9799	0.9799	0.0061	0.0893
	InceptionV3	0.9898	0.9897	0.0015	0.0152
	ResNet50	0.9851	0.9851	0.0034	0.9571
	TinyViT	0.9939	0.9938	0.0013	0.0003

The proposed model achieved the highest mean accuracy and F1-score of 0.9913 for Balinese script, with a slight deviation of 0.0039. The p-values for other models such as EfficientNetV2 (2.40E-06),

InceptionV3 (1.76E-02), ResNet50 (1.24E-03), and TinyViT (9.33E-04) were all under 0.05, meaning that the performance difference with the proposed model was statistically significant. The proposed model is both more accurate and offers more stable results than other models. Despite the comparable performance of InceptionV3 and TinyViT, the proposed model outperforms in consistency, achieving the smallest standard deviation. In Batak script, the performance gap between models is even larger.

The proposed model had a mean accuracy of 0.9247 and an F1-score of 0.9211, outperforming EfficientNetV2, which recorded 0.7247 and 0.7072. The p-value for EfficientNetV2 is 7.59E-08, indicating a highly significant difference. Other models such as InceptionV3 and TinyViT also performed well with F1-scores of 0.9546 and 0.9759, respectively. The proposed model's ability to outperform larger models, including ResNet50, shows that it balances complexity and generalization well.

The experimental results for the Javanese script indicate a highly competitive landscape. The proposed model achieved a mean accuracy of 0.9722 and an F1-score of 0.9654, performing almost identically to EfficientNetV2 ($p = 0.273$). While heavier architectures like InceptionV3 and ResNet50 surpassed the proposed model's F1-score by approximately 0.01, the proposed model significantly outperformed TinyViT ($p = 0.007$).

This demonstrates that for balanced datasets like Javanese, the proposed model provides an efficient alternative with comparable performance to much larger models. In the Lontara script, the proposed model excelled with an F1-score of 0.9815, dramatically outperforming EfficientNetV2. Although TinyViT achieved a high F1-score (0.9921), the p-value of 0.152 suggests that the proposed model maintains statistical parity while offering slightly better average stability. In the Sundanese script, all models showed exceptional performance, with the proposed model recording an F1-score of 0.9850, showing no significant statistical difference compared to ResNet50.

In terms of architectural complexity, the proposed model (MobileNetV3) maintains the smallest footprint with only 3.5 million parameters and a memory usage of 40.7 MB (Table 8). This is considerably more efficient than EfficientNetV2 (7.2M params, 76.1 MB), InceptionV3 (23.9M, 262.6 MB), and ResNet50 (25.6M, 282.5 MB). Notably, the proposed model delivers its high performance with less than one-sixth the memory footprint of ResNet50. While TinyViT reported lower memory usage (21.2 MB), this difference is likely due to its PyTorch implementation compared to the TensorFlow environment used for other models, which utilizes a best-fit with coalescing (BFC) allocator and static-graph optimization.

Table 8. Number of parameters and memory usage accoss models

Model	Number of params	Memory usage
Proposed model	3.5M	40.7 MB
EfficientnetV2	7.2M	76.1 MB
InceptionV3	23.9M	262.6 MB
ResNet50	25.6M	282.5 MB
TinyViT	5.53M	21.2 MB

Ultimately, the scalability of MobileNetV3's depthwise separable convolution design allows for a significant reduction in redundant computations without sacrificing representational power. With its statistically significant performance advantages ($p < 0.05$ across most datasets), the proposed model offers a superior accuracy-to-memory trade-off. Making it ideal for deployment in resource-constrained or embedded OCR systems requiring reliable recognition of diverse Nusantara script.

4. CONCLUSION

The preservation of Nusantara script is crucial for maintaining the nation's heritage and identity, as it reflects history and civilization. To support this digital preservation, we propose a script recognition model based on the MobileNetV3 architecture, called the MNetNCR model. This research uses a specially curated dataset, encompassing characters from five major scripts, including Balinese, Batak, Javanese, Lontara, and Sundanese, which allows for an accurate evaluation of the model's ability to classify various traditional visual patterns. Based on the experimental results, the MNetNCR model demonstrates excellent performance in recognition tasks, consistently achieving high results in the precision, recall, and F1-score metrics. The model achieves outstanding performance for all scripts, the proposed model obtained best testing F1-score 0.9934 on the Balinese script, 0.9450 on Batak script, 0.9788 on Javanese script, 0.9784 on Lontara script, and 0.9936 on Sundanese script. The proposed model outperforms EfficientnetV2, InceptionV3, ResNet50, and TinyViT on Balinese script with a higher F1-score, greater stability, and lightest number of parameters. Overall, the proposed model proved to be efficient and accurate, making it an ideal

choice for fast and reliable character recognition applications, especially in environments with limited computing resources. Future research will be expanded to include the addition of other Nusantara script datasets, exploration of ensemble deep learning approaches, multi-task learning (character classification and script family prediction), development of a system for sentence-level recognition, and adaptation of the methodology for ancient manuscript recognition to improve the digital accessibility of valuable cultural and historical documents.

FUNDING INFORMATION

The financial support provided by Indonesia's DRTPM, DITJEN DIKTIRISTEK, KEMDIKBUDRISTEK through grant 043/SP2H/RT-MONO/LL4/2024 and 062/LIT07/PPM-LIT/2024.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Untari Novia Wisesty	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓		✓
Aditya Firman Ihsan	✓		✓	✓		✓	✓	✓		✓				
Mahmud Dwi Sulistiyo		✓		✓					✓			✓		✓
Donni Richasdy		✓		✓	✓	✓	✓	✓	✓					
Prasti Eko Yunanto					✓		✓	✓		✓			✓	✓
Gamma Kosala			✓			✓		✓		✓	✓			
Arfive Gandhi			✓			✓	✓			✓	✓		✓	
Febryanti Sthevanie						✓	✓			✓	✓		✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are openly available in Mendeley Data at https://data.mendeley.com/datasets/vfj32bpjsf/1, reference [17].

REFERENCES

[1] A. Apriyanto, N. Nurjanah, and Ruhaliah, "Structure of the Sundanese language in the Pegon script," in *Proceedings of the Fifth International Conference on Language, Literature, Culture, and Education (ICOLLITE 2021)*, 2021, doi: 10.2991/assehr.k.211119.006.

[2] D. F. Lubis and T. A. Bowo, "Batak Toba script, preserving its authenticity in globalization stream," in *2nd International Conference on Social, Politics, and Humanities (ICoSoPH) 2021*, Jan. 2022, pp. 28–34, doi: 10.11594/nstp.2021.1704.

[3] G. Indrawan, Sariyasa, and I. K. Paramarta, "A new method of Latin-To-Balinese script transliteration based on Bali Simbar font," in *2019 Fourth International Conference on Informatics and Computing (ICIC)*, Oct. 2019, pp. 1–6, doi: 10.1109/ICIC47613.2019.8985675.

[4] M. F. Adilazuarda, M. I. Wijanarko, L. Susanto, K. Nur'aini, D. T. Wijaya, and A. F. Aji, "NusaAksara: a multimodal and multilingual benchmark for preserving Indonesian indigenous scripts," *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, Jul. 2025, doi: 10.18653/v1/2025.acl-long.1377.

[5] F. Tambunan, E. Ginting, E. V. Haryanto, and M. Fauzi, "Pattern recognition of Batak script using Habbian method," in *2020 8th International Conference on Cyber and IT Service Management (CITSM)*, Oct. 2020, pp. 1–4, doi: 10.1109/CITSM50537.2020.9268839.

[6] A. R. Widiarti, A. Harjoko, Marsono, and S. Hartati, "The model and implementation of Javanese script image transliteration," in *2017 International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIT)*, Sep. 2017, pp. 51–57, doi: 10.1109/ICSIIT.2017.17.

[7] A. Prasetiadi, J. Saputra, I. Kresna, and I. Ramadhanti, "YOLOv5 and U-Net-based character detection for Nusantara script," *Jurnal Online Informatika*, vol. 8, no. 2, pp. 232–241, Dec. 2023, doi: 10.15575/join.v8i2.1180.

[8] M. I. Fidatama, F. Bimantoro, G. S. Nugraha, B. Irmawati, and R. Dwiyansaputra, "Recognition of Bima script handwriting patterns using the local binary pattern feature extraction method and k-nearest neighbour classification method," in *International Conference on Biomedical Engineering (ICoBE 2021)*, 2023, doi: 10.1063/5.0111770.

[9] S. Razali, F. Arnia, R. Muharrar, K. Muchtar, and A. Bintang, "Improved classification of handwritten Jawi script based on main part of script body," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 1, pp. 94–104, Feb. 2023, doi: 10.29207/resti.v7i1.4600.

[10] S. Razali, K. Muchtar, M. H. Rinaldi, Y. Nurdin, and A. Rahman, "Augmentation of additional Arabic dataset for Jawi writing and classification using deep learning," *Jurnal Rekayasa Elektrika*, vol. 20, no. 1, Mar. 2024, doi: 10.17529/jre.v20i1.33722.

[11] H. Suparwito, "Image detection analysis for Javanese character using YOLOv9 models," *International Journal of Applied Sciences and Smart Technologies*, vol. 6, no. 1, pp. 197–208, Jun. 2024, doi: 10.24071/ijasst.v6i1.8779.

[12] A. A. Hidayat, K. Purwandari, T. W. Cenggoro, and B. Pardamean, "A convolutional neural network-based ancient Sundanese character classifier with data augmentation," *Procedia Computer Science*, vol. 179, pp. 195–201, 2021, doi: 10.1016/j.procs.2020.12.025.

[13] D. A. S. Dewi, D. M. S. Arsa, G. A. A. Putri, and N. L. P. L. S. Setiawati, "Ensembling deep convolutional neural neworks for Balinese handwritten character recognition," *ASEAN Engineering Journal*, vol. 13, no. 3, pp. 133–139, Aug. 2023, doi: 10.11113/aej.v13.19582.

[14] A. Howard *et al.*, "Searching for MobileNetV3," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019, pp. 1314–1324. doi: 10.1109/ICCV.2019.00140.

[15] S. Qian, C. Ning, and Y. Hu, "MobileNetV3 for image classification," in *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, Mar. 2021, pp. 490–497, doi: 10.1109/ICBAIE52039.2021.9389905.

[16] L. Zhao and L. Wang, "A new lightweight network based on MobileNetV3," *KSII Transactions on Internet and Information Systems*, vol. 16, no. 1, Jan. 2022, doi: 10.3837/tiis.2022.01.001.

[17] A. F. Ihsan, "Indonesian local script characters," *Mendeley Data*. Accessed: Apr. 24, 2025, [Online], Available: https://data.mendeley.com/datasets/vfj32bpjsf/1

[18] B. Koonce, "MobileNetV3," in *Convolutional Neural Networks with Swift for Tensorflow*, Berkeley, United States: Apress, 2021, pp. 125–144, doi: 10.1007/978-1-4842-6168-2_11.

[19] S. Shen, R. Xiao, M. Li, and S. Yuan, "Application of lightweight method based on YOLOv5s+mobileV3 on edge computing platform," in *Proceedings of the 4th International Conference on Artificial Intelligence and Computer Engineering*, Nov. 2023, pp. 787–792, doi: 10.1145/3652628.3652759.

[20] B. Wu, Z. Miu, and X. Huang, "Edge computing-enhanced k-means and MobileNetV3 for short-term photovoltaic forecasting," in *2024 5th International Conference on Computers and Artificial Intelligence Technology (CAIT)*, Dec. 2024, pp. 611–615, doi: 10.1109/CAIT64506.2024.10962993.

[21] D. Saha, M. P. Mangukia, and A. Manickavasagan, "Real-time deployment of MobileNetV3 model in edge computing devices using RGB color images for varietal classification of chickpea," *Applied Sciences*, vol. 13, no. 13, Jul. 2023, doi: 10.3390/app13137804.

[22] S. Bisht, A. Singhal, and C. Kaushik, "Face recognition using deep neural network with MobileNetV3-Large," in *Advances and Applications of Artificial Intelligence & Machine Learning (ICAAAIML 2022)*, 2023, pp. 115–123, doi: 10.1007/978-981-99-5974-7_11.

[23] S. B. R. Prasad and B. S. Chandana, "MobileNetV3: a deep learning technique for human face expressions identification," *International Journal of Information Technology*, vol. 15, no. 6, pp. 3229–3243, Aug. 2023, doi: 10.1007/s41870-023-01380-x.

[24] X. Tian, L. Shi, Y. Luo, and X. Zhang, "Garbage classification algorithm based on improved MobileNetV3," *IEEE Access*, vol. 12, pp. 44799–44807, 2024, doi: 10.1109/ACCESS.2024.3381533.

[25] Y. Zhao, H. Huang, Z. Li, H. Yiwang, and M. Lu, "Intelligent garbage classification system based on improve MobileNetV3-Large," *Connection Science*, vol. 34, no. 1, pp. 1299–1321, Dec. 2022, doi: 10.1080/09540091.2022.2067127.

[26] Y. Yang and J. Han, "Real-time object detector based MobileNetV3 for UAV applications," *Multimedia Tools and Applications*, vol. 82, no. 12, pp. 18709–18725, May 2023, doi: 10.1007/s11042-022-14196-x.

[27] P. Li, F. Chen, L. Pan, T. Hoang, Y. Zhu, and L. Yang, "LightLiveAuth: a lightweight continuous authentication model for virtual reality," *IoT*, vol. 6, no. 3, Sep. 2025, doi: 10.3390/iot6030050.

[28] A. G. Howard *et al.*, "MobileNets: efficient convolutional neural networks for mobile vision applications," 2017, arXiv: 1704.04861.

[29] S. Kantu, H. S. Kaja, V. Kukkala, S. A. Aly, and K. Sayed, "Integrating MobileNetV3 and SqueezeNet for multi-class brain tumor classification," *Journal of Imaging Informatics in Medicine*, Jul. 2025, doi: 10.1007/s10278-025-01589-1.

[30] S. Zhu *et al.*, "Squeeze-and-excitation-attention-based mobile vision transformer for grading recognition of bladder prolapse in pelvic MRI images," *Medical Physics*, vol. 51, no. 8, pp. 5236–5249, Aug. 2024, doi: 10.1002/mp.17171.

[31] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 7132–7141. doi: 10.1109/CVPR.2018.00745.

[32] N. Behar and M. Shrivastava, "ResNet50-based effective model for breast cancer classification using histopathology images," *Computer Modeling in Engineering & Sciences*, vol. 130, no. 2, pp. 823–839, 2022, doi: 10.32604/cmes.2022.017030.

[33] M. Elpeltagy and H. Sallam, "Automatic prediction of COVID-19 from chest images using modified ResNet50," *Multimedia Tools and Applications*, vol. 80, no. 17, pp. 26451–26463, Jul. 2021, doi: 10.1007/s11042-021-10783-6.

[34] N. Dong, L. Zhao, C. H. Wu, and J. F. Chang, "Inception v3 based cervical cell classification combined with artificially extracted features," *Applied Soft Computing*, vol. 93, Aug. 2020, doi: 10.1016/j.asoc.2020.106311.

[35] G. Jignesh Chowdary, N. S. Punn, S. K. Sonbhadra, and S. Agarwal, "Face mask detection using transfer learning of InceptionV3," in *Big Data Analytics (BDA 2020)*, 2020, pp. 81–90, doi: 10.1007/978-3-030-66665-1_6.

[36] O. A. Abioye, A. E. Evwiekpaefe, and A. J. Awujoola, "Performance evaluation of EfficientNetV2 models on the classification of histopathological benign breast cancer images," *Science Journal of University of Zakho*, vol. 12, no. 2, pp. 208–214, May 2024, doi: 10.25271/sjuoz.2024.12.2.1261.

[37] M. Tan and Q. V. Le, "EfficientNetV2: smaller models and faster training," in *Proceedings of the 38th International Conference on Machine Learning*, 2021, pp. 10096–10106.

[38] R. S. S. Devi, V. R. V. Kumar, and P. Sivakumar, "EfficientNetV2 model for plant disease classification and pest recognition," *Computer Systems Science and Engineering*, vol. 45, no. 2, pp. 2249–2263, 2023, doi: 10.32604/csse.2023.032231.

[39] K. Wu *et al.*, "TinyViT: fast pretraining distillation for small vision transformers," in *Computer Vision – ECCV 2022*, 2022, pp. 68–85, doi: 10.1007/978-3-031-19803-8_5.

[40] X. Mao *et al.*, "Towards robust vision transformer," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 12032–12041, doi: 10.1109/CVPR52688.2022.01173.

BIOGRAPHIES OF AUTHORS

Untari Novia Wisesty received a Doctor of Electrical Engineering and Informatics degree from Bandung Institute of Technology, Indonesia in 2023. She also received bachelor's and master's degree on Informatics Engineering from Telkom Institute of Technology (now Telkom University), Bandung, Indonesia in 2010 and 2012. Since December 2024, she is an associate professor of Artifcial Intelligence in Telkom University. Her research interest includes machine learning, deep learning, artificial intelligence, bioinformatics, and biomedical engineering. She can be contacted at email: untarinw@telkomuniversity.ac.id.

Aditya Firman Ihsan obtained degrees of bachelor, master, and doctor in Department of Mathematics at Institut Teknologi Bandung (ITB) consecutively in 2016, 2017, and 2022. He was actively involved in many research projects in the research consortium optimization of pipeline network (OPPINET) under mathematical modelling and simulation CoLaboratory in ITB from 2017 until present. He also teaches mathematics in the School of Computing in Telkom University from 2020. His main research interests are mathematical modelling, dynamical systems, and deep learning. He can be contacted at email: adityaihsan@telkomuniversity.ac.id.

Mahmud Dwi Sulistiyo received his Doctor of Informatics degree from Nagoya University, Japan, in 2021. He also earned his bachelor's and master's degrees in Informatics Engineering from Telkom Institute of Technology, Indonesia, in 2010 and 2012, respectively. He is currently a lecturer at the School of Computing, Telkom University, Indonesia, where he also serves as the artificial intelligence laboratory supervisor. His research interests include artificial intelligence, machine learning, and computer vision. He has received multiple research grants from Telkom University, as well as external funding from the Ministry of Research, Technology, and Higher Education, and the Indonesia Endowment Fund for Education (LPDP) under the Ministry of Finance. He is a member of IEEE and ACM. He can be contacted at email: mahmuddwis@telkomuniversity.ac.id.

Donni Richasdy is a lecturer in the Informatics Study Program, Faculty of Informatics, Telkom University, Bandung, Indonesia. He holds a Master of Engineering degree and serves as a lecturer in the field of Computer Science. In the field of research, he has a primary interest in conversational AI, natural language processing, data engineering, and software engineering. In addition to teaching and research, he is also active in other academic activities. He has been a mentor for a student team in a competition. He currently lives in Bandung, Indonesia. He can be contacted at email: donnir@telkomuniversity.ac.id.

Prasti Eko Yunanto earned his bachelor's and master's degrees in Informatics Engineering from Telkom Institute of Technology, Indonesia, in 2012 and 2015, respectively. He is currently a lecturer at the School of Computing, Telkom University, Indonesia. His research interests include biometrics security, machine learning, and computer vision. He can be contacted at email: gppras@telkomuniversity.ac.id.

Gamma Kosala earned a Bachelor of Science degree in Instrumentation Electronics from Universitas Gadjah Mada. He was awarded the master's to doctoral education scholarship for outstanding undergraduates, which enabled him to pursue his master's and doctoral studies in Computer Science at Universitas Gadjah Mada. He completed his doctorate with a dissertation on intelligent traffic monitoring systems (ITMS). He is currently a lecturer in the School of Computing Faculty at Telkom University. His research interests include artificial intelligence, machine learning, and computer vision. He can be contacted at email: gammakosala@telkomuniversity.ac.id.

Arfive Gandhi is a software engineering specialist at Telkom University, Bandung. He obtained his Ph.D. in Computer Science from Universitas Indonesia. He is affiliated with the Special Interest Group of Software Engineering and Algorithm and the Center of Excellence for Smart City at Telkom University. He has extensive experience in developing smart city master plans, including projects in Kaimana, Temanggung, and Purworejo Regencies. His research interests include software project management, smart city, information security, e-government, and user experience. He can be contacted at email: arfivegandhi@telkomuniversity.ac.id.

Febryanti Sthevanie holds a Master of Informatics degree from Telkom University, Bandung, Indonesia in 2013. She received her B.Sc. (Informatics) also from Telkom University, in 2010. She is currently an assistant professor at School of Computing in Telkom University. Her research includes computer vision, pattern recognition, and machine learning. She has published over 30 papers in international journals and conferences. She can be contacted at email: sthevanie@telkomuniversity.ac.id.