Available online at website: https://jurnal.iaii.or.id/index.php/RESTI

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 9 No. 6 (2025) 1538 - 1550 e-ISSN: 2580-0760

Investigating Shallow Learning Methods for Optical Character Recognition of Indonesia's Nusantara Scripts

Mahmud Dwi Sulistiyo[1], Aji Gautama Putrada[2], Aditya Firman Ihsan[3], Prasti Eko Yunanto[4], Donni Richasdy[5], Hassan Rizky Putra Sailellah[6], Sabrina Adinda Sari[7]
[1,2,3,4,5,6,7]School of Computing, Telkom University, Bandung, Indonesia

[1]mahmuddwis@telkomuniversity.ac.id, [2]ajigps@telkomuniversity.ac.id, [3]adityaihsan@telkomuniversity.ac.id, [4]gppras@telkomuniversity.ac.id, [5]donnir@telkomuniversity.ac.id, [6]hassanrizkyhrs@telkomuniversity.ac.id, [7]sabrinaas@telkomuniversity.ac.id

Abstract

Indonesia has numerous regional scripts—or so-called Nusantara scripts—and recognizing them is important to preserve Indonesia's cultural heritage. The advances of AI and computer vision technologies make it possible for a machine to optically read the handwritten scripts through the Optical Character Recognition (OCR) technique. However, collecting some of the top OCR solutions and comprehensively investigating their performances on the Nusantara scripts is currently lacking. This study investigates and evaluates some shallow learning-based methods on our newly introduced datasets, consisting of more than 38,000-character images across 80 letter classes in total; here, we focus on three regional scripts: Javanese, Sundanese, and Balinese. The methods include Random Forest, SVM, Logistic Regression, and Gaussian Naïve Bayes, as well as boosting techniques such as XGBoost, Light GBM, and CatBoost. A 5-fold cross-validation approach assessed model performance based on accuracy, precision, recall, and F1-score. Based on the experimental results, the methods demonstrated their competitiveness in reaching the best models for scripts; in particular, XGBoost, Light GBM, and Random Forest-Gini were the winners for Javanese, Sundanese, and Balinese scripts, respectively. These findings demonstrate the effectiveness of ensemble learning methods for diverse handwritten scripts. Comparative analysis to prior deep learning studies is also discussed in this paper. In addition, this research also contributes to preserving Indonesian traditional scripts, as well as offers insights for future regional OCR in other countries.

Keywords: balinese; javanese; optical character recognition; shallow learning; sundanese

How to Cite: M. D. Sulistiyo, A. G. Putrada, A. F. Ihsan, P. E. Yunanto, D. Richasdy, and H. R. P. Sailellah, "Investigating Shallow Learning Methods for Optical Character Recognition of Indonesia's Nusantara Scripts", *J. RESTI (Rekayasa Sist. Teknol. Inf.)* , vol. 9, no. 6, pp. 1538 - 1550, Jan. 2026.
Permalink/DOI: https://doi.org/10.29207/resti.v9i6.6648

Received: May 23, 2025
Accepted: December 16, 2025
Available Online: January 11, 2026

This is an open-access article under the CC BY 4.0 License
Published by Ikatan Ahli Informatika Indonesia

1. Introduction

Javanese, Sundanese, and Balinese scripts are cultural heritages of Nusantara with significant historical and cultural value [1]. They play a crucial role in Nusantara's literacy history and continue to be used in cultural and traditional contexts [2]. They function as writing systems and reflect the Balinese, Javanese, and Sundanese communities' identity, culture, and local wisdom [3]. Therefore, efforts to preserve and digitize these scripts are essential to ensure the sustainability of this cultural heritage in the modern era [4]. With the advancement of technology capable of recognizing human handwriting, recognizing these regional scripts becomes vital for developing Indonesia's local text-based technologies, which are unique and not found in other countries. However, OCR for Nusantara scripts remains challenging due to the diversity of handwritten forms that vary greatly between individuals and the similarity of character shapes within the same script system, making accurate recognition a complex task for computers.

To address these challenges, researchers have leveraged computer vision and machine learning techniques to develop OCR systems for Nusantara scripts, including Javanese, Sundanese, and Balinese scripts. Several past studies utilized image processing pipelines and classification techniques to digitize and document these scripts. In Javanese script recognition, metric features, eccentricity, and Local Binary Patterns (LBP) were used on a handwritten dataset with 200 training samples

and 40 testing samples, and classified the characters using K-Nearest Neighbor (KNN), achieving 92.5% accuracy [5]. Various traditional approaches have also been explored, such as circularity and eccentricity feature extraction followed by KNN classification for Javanese script characterization, which reported good recognition performance [6]. In addition, a template-matching approach for transliteration by matching character templates with input images was applied to determine the character class [7]. Ensemble-based shallow learning has also shown potential; for example, a Random Forest could deliver very good accuracy for Javanese script classification [8].

Alongside traditional methods, deep learning approaches have become widely popular and have shown strong performance for various OCR tasks [9], [10], [6]. In the context of Nusantara scripts, the transfer learning approach was used in CNN architectures such as GoogleNet, DenseNet, ResNet, VGG16, and VGG19 for Javanese script classification, where VGG19 achieved the highest accuracy of 99.50% [6]. Moreover, Faster R-CNN was utlized to demonstrate that CNN backbones such as ResNet-50 and Inception ResNet V2 can achieve high mean average precision (mAP) in detecting Balinese characters [9]. A similar Faster R-CNN approach was also applied to detect Javanese script letters, yielding mAP values up to 0.8381 and accuracy up to 96.31% [10]. These results indicate that deep learning models can achieve high recognition performance, but they often require very large datasets and substantial computational resources.

In many practical scenarios, however, large-scale annotated datasets for Javanese, Sundanese, and Balinese scripts are still limited. In contrast, shallow machine learning methods offer a lighter solution and can be applied effectively to smaller datasets. Prior works have discussed comparisons between shallow and deep learning approaches [11], [12], particularly in terms of resource efficiency, raising an important question of whether computational load can be decreased while maintaining competitive recognition ability. Therefore, this study aims to investigate the performance stability of several machine learning methods for recognizing handwritten Indonesia's regional scripts and introduces our Javanese, Sundanese, and Balinese script datasets.

In this study, we focus on shallow learning methods because they provide advantages in speed and computational efficiency, especially when the dataset is limited or the computational infrastructure is inadequate for training deep learning models. This perspective is also relevant for broader adoption, as recognizing Indonesia's regional scripts supports applications such as digitizing ancient manuscripts, language learning applications, and automatic archiving systems. Specifically, we compare several shallow machine learning methods, including Support Vector Machine (SVM), Logistic Regression, Decision Tree, Naïve Bayes, Random Forest, and Gradient Boosting Machine. SVM is a supervised learning method that seeks a maximum-margin separating hyperplane and can handle non-linear patterns using kernel functions, offering relatively low complexity compared to deep learning models [13], [14]. Logistic Regression is a simple probabilistic classifier with low computational cost and fast training and inference, particularly effective when the feature–class relationship is approximately linear [15], [16]. Decision Tree models provide interpretable rule-based decisions and can be trained efficiently, although they may overfit on small datasets [17], [18]. Naïve Bayes is a probabilistic method based on Bayes' theorem with a feature-independence assumption, enabling very fast training and prediction [19], [20]. Random Forest, as an ensemble of multiple decision trees trained on different subsets of data, improves robustness and reduces overfitting compared to a single tree while remaining lighter than deep learning models [21], [22]. Gradient Boosting Machine incrementally builds models to correct previous errors and can yield strong accuracy, although it requires careful parameter tuning to avoid overfitting [23], [24]. In addition, we evaluate an implementation of gradient boosting using LightGBM in our experiments.

Accordingly, this study aims to analyze the effectiveness of various shallow learning methods in recognizing handwritten Javanese, Sundanese, and Balinese scripts. The methods are compared in terms of accuracy and robustness against variations in handwriting styles and limited dataset sizes, with the goal of providing recommendations for practical OCR deployment for Indonesia's regional scripts. We chose to employ shallow learning techniques on purpose because there isn't a lot of data available for Javanese, Sundanese, and Balinese scripts yet, while deep learning approaches such as CRNN, ViT, or modern CNN-based architectures typically work best with huge and diverse datasets. Under these limitations, shallow learning offers a useful, competitive, and accessible OCR solution, particularly when computational resources are constrained, and can also serve as a baseline for future studies that adopt deep learning when larger datasets become available.

This study makes several key contributions. First, it introduces a novel dataset consisting of handwritten ethnical scripts from three Indonesian languages, namely Javanese, Sundanese, and Balinese, which can serve as a valuable resource for future research. Second, the experimental results demonstrate that a shallow learning approach using LightGBM outperforms deep learning methods for OCR on both Javanese and Balinese handwritten scripts. Finally, for Sundanese handwritten script recognition, the Random Forest–based model achieves superior performance compared to deep learning and KNN approaches.

The remainder of this paper has the following systematics: Section 2 explains the research methodology, dataset collecting, preprocessing

techniques, shallow learning methods, and the evaluation methods. Section 3 presents the results and analyzes the model's performance in recognizing handwritten Javanese, Sundanese, and Balinese scripts. Finally, Section 4 summarizes the study's findings and offers recommendations for future research directions.

2. Research Methods

2.1 Overview of the Proposed System

Figure 1 overviews the OCR system in this study. In terms of accuracy and resilience to variations in handwriting styles and small dataset sizes, the study will evaluate the performance of techniques like SVM, Decision Trees, Naïve Bayes, and Logistic Regression with certain ensemble learning-based techniques. The performance stability of several machine learning techniques in the three types of regional scripts used in Indonesia is examined in this study. It is anticipated that the analysis's findings would offer suggestions for the best machine learning techniques for Javanese, Sundanese, or Balinese script recognition in practical settings.

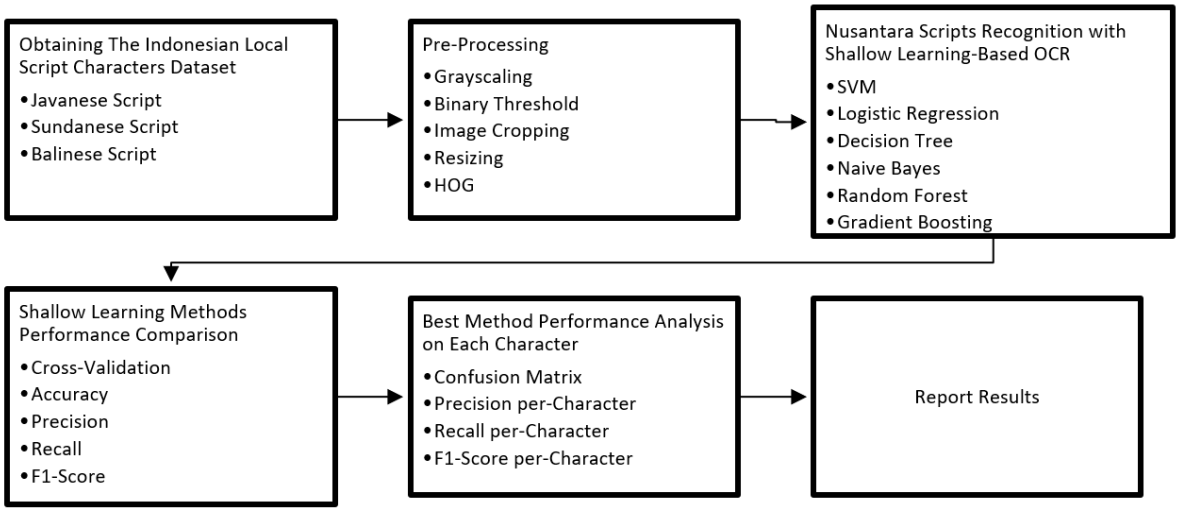


Figure 1. The method workflow for the Nusantara scripts recognition with shallow learning-based OCR

2.2 The Nusantara Scripts: Datasets and Preprocessing

This section describes the self-collected datasets, including statistics and distribution related to each type of Nusantara script. This paper introduces Javanese, Sundanese, and Balinese script datasets. Furthermore, the dataset is accessible through Mendeley Data, with the repository name "Indonesian Local Script Characters." It can be accessed at: https://data.mendeley.com/datasets/vfj32bpjsf/1.

The Aksarawa (Aksara Jawa) dataset contains images of the Javanese script. This script is used to write the Javanese language and was developed around the Central and East Java regions on Java Island. The dataset includes 20 original letter classes and eight additional variation classes, totaling 11,061 images—specifically, 9,995 images in their original form and 1,066 images representing variations. The distribution of images varies between 103 and 695 per class, and the images come in mixed dimensions, all formatted as PNG files. The data distribution of Javanese original characters is presented in Figure 2. Meanwhile, some samples of character variations are shown in Figure 3.

The Aksarunda (Aksara Sunda) dataset consists of 20,224 handwritten Sundanese script images divided into two main subsets: 16,179 for training and 4,045 for validation. This division ensures the model can be trained with sufficient data while being evaluated accurately on data separated from the training process. The data distribution of Aksarunda is shown in Figure 4.

The Aksarali (Aksara Bali) dataset contains images of handwritten Balinese script. This dataset has 6,981 images, divided into 5,585 images for training and 1,396 images for validation; class distribution is shown in Figure 5. The dataset division was carried out to ensure that the model can optimally learn patterns from the training data while being thoroughly evaluated using the validation data.

The Aksarunda (Aksara Sunda) dataset consists of 20,224 handwritten Sundanese script images divided into two main subsets: 16,179 for training and 4,045 for validation. This division ensures the model can be trained with sufficient data while being evaluated accurately on data separated from the training process. The data distribution of Aksarunda is shown in Figure 4.

The Aksarali (Aksara Bali) dataset contains images of handwritten Balinese script. This dataset has 6,981 images, divided into 5,585 images for training and 1,396 images for validation; class distribution is shown in Figure 5. The dataset division was carried out to ensure that the model can optimally learn patterns from the training data while being thoroughly evaluated using the validation data.

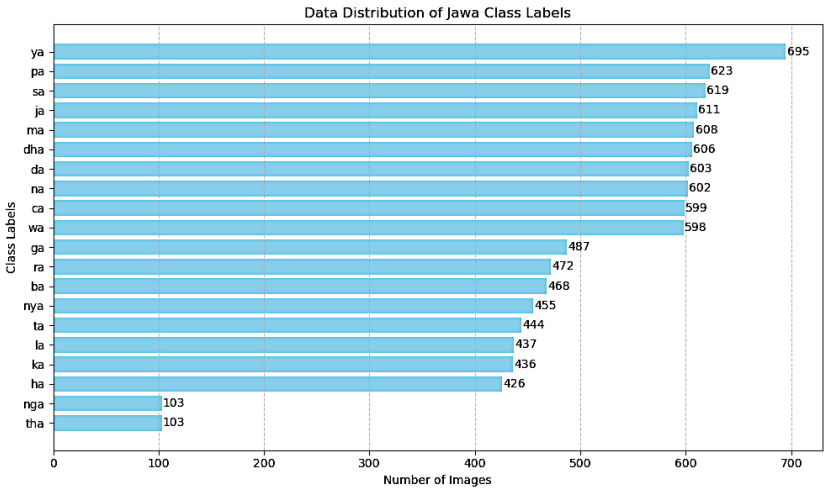


Figure 2. Class Distribution of the Character Images in Aksarawa Dataset

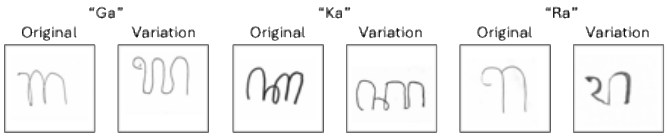


Figure 3. Samples of Original and Variation Characters from the Data Collection

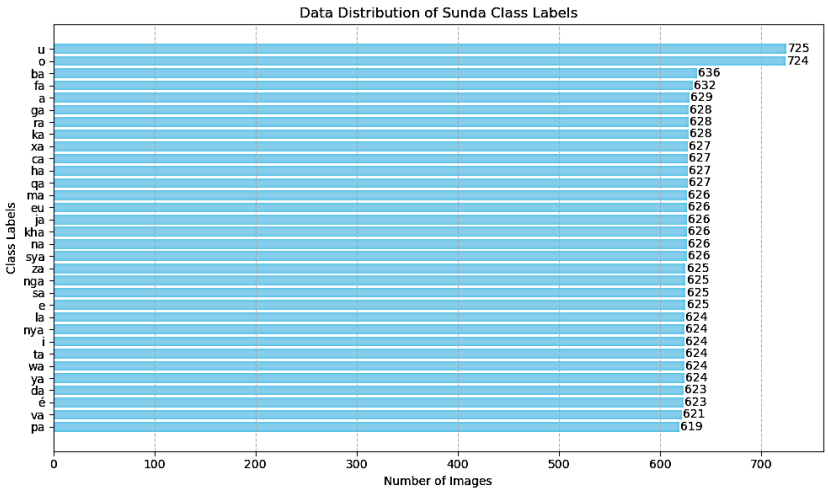


Figure 4. Class Distribution of the Character Images in Aksarunda Dataset

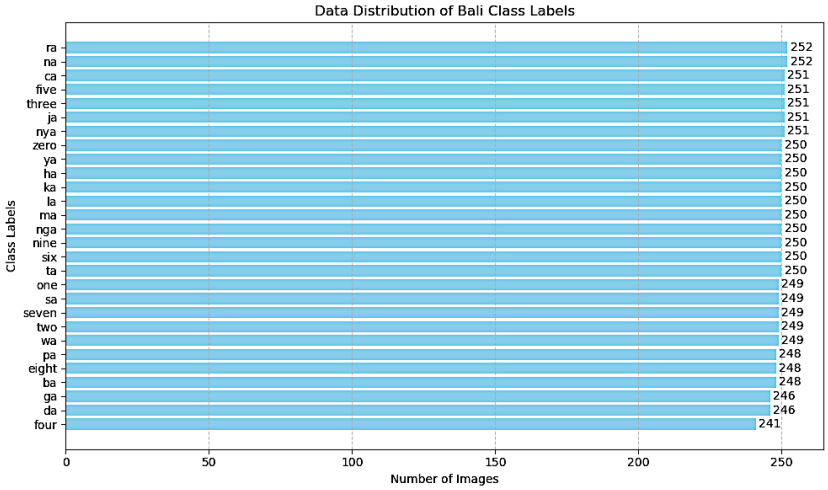


Figure 5. Class Distribution of the Character Images in Aksarali Dataset

2.3 Data Preprocessing

Before entering the classification or model training process, the input script images are preprocessed with some steps. Figure 6 shows the preprocessing stages with samples of some Aksarawa characters. First, greyscaling, is a process where at this stage, the input image is converted to grayscale to simplify processing and reduce the data dimensions from three channels (RGB) to one channel.

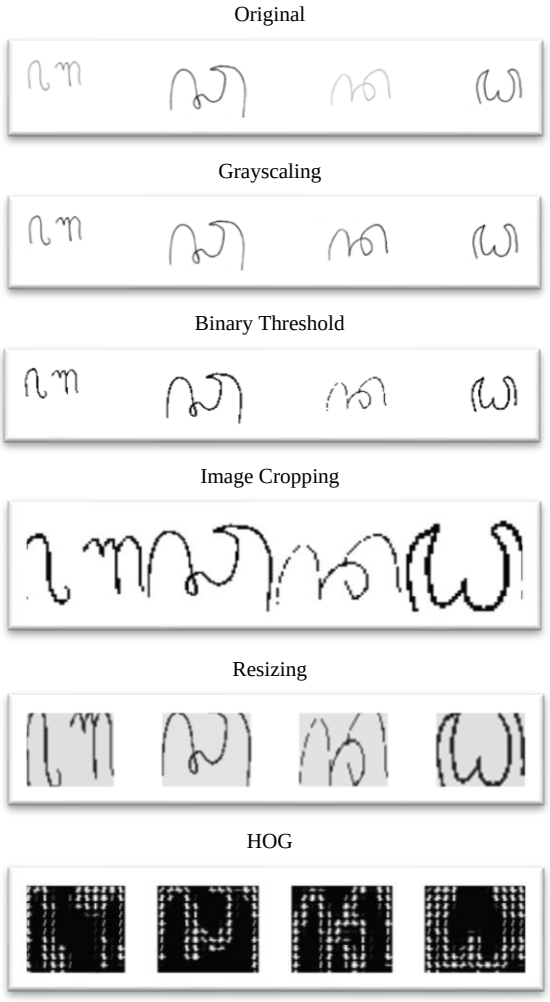


Figure 6. Image Visualization of each Preprocessing Stage

Binary thresholding is to convert the image into only two colors: black and white (0 and 1). Pixels below 50 will be converted to black, while those above 50 will be converted to white. This step aims to eliminate noise in the image and enhance the clarity of character contours. Image cropping is performed to precisely locate the main area consisting of a character. The cropping area is determined by detecting the presence of black pixels. After cropping, the image will have varying sizes and aspect ratios. Resizing is where the image is resized and padded to a 96×96-pixel size to ensure all images have a uniform size. This size is chosen as the optimal balance between image size and clarity, facilitating character recognition.

Then, by histogram of oriented gradient (HOG), the structural features of the characters are extracted using the HOG technique. HOG measures the gradient orientation in the image to capture important edge and texture information, aiding in detecting characters more sharply and accurately. Lastly, pixel intensity is captured and combined with information from the HOG process. This combination provides a richer character representation, integrating textural details and pixel intensity to improve recognition accuracy. Figure 6 shows the visualization of the image at every stage of the preprocessing methods.

2.4 Implementation Detail

Almost every machine learning method includes some hyperparameters to be set by developers. Two splitting criteria were used for Random Forest: Gini and Entropy, while SVM employed Polynomial and RBF kernels. The boosting methods applied included XGBoost, Light GBM, and CatBoost. Details of the implementation for the discussed methods are described in Table 1.

Table 1. Parameter Settings for the Model Training

Model	Parameter	Value
Random Forest	n_estimators	300
	max_depth	None
	min_samples_split	2
	min_samples_leaf	1
	criterion	Gini, Entropy
SVM-RBF	C	1
	kernel	RBF
	gamma	Scale
	class_weight	None
	probability	False
SVM-Poly	C	1
	kernel	Poly
	gamma	Scale
	degree	3
	class_weight	None
	probability	False
XGBoost	n_estimators	300
	learning_rate	0.3
	max_depth	6
	min_child_weight	1
	subsample	1
	gamma	0
Logistic Regression	max_iter	200
	penalty	L2
	C	1
	solver	lbfgs
	multi_class	Multinomial
Gaussian Naïve Bayes	prior	None
Light GBM Classifier	n_estimators	300
	learning_rate	0.1
	num_leaves	31
	boosting_type	gbdt
	max_depth	-1
	subsample	1
	regulatization_L1	0
	regularization_L2	0
CatBoost Classifier	n_estimator	300
	learning_rate	0.03
	depth	6
	bagging_temperature	1
	grow_policy	Symmetric Tree

The modeling process employs 5-fold cross-validation (set to 5), where the training set is divided into five equal parts. Each fold is used once as a validation set, while the remaining four folds form the training set. This process is repeated five times, and the model's performance is averaged to select the best model based on accuracy scores. This 5-fold cross-validation is applied to each implemented method. The best models from each method are then compared using precision, recall, and F1-score metrics. Both macro-averaged and weighted-average metrics are calculated to evaluate the models' performance comprehensively. The comparison and analysis of these metrics help determine the most effective method for the given dataset.

3. Results and Discussion

3.1 Cross Validation Results

The models implemented in this study include Random Forest (Gini), Random Forest (Entropy), SVM (Polynomial), SVM (RBF), Logistic Regression, Gaussian Naïve Bayes, XGBoost, Light GBM, and CatBoost. For each method, 5-fold cross-validation was employed, where the best model from the five trials was selected based on accuracy metrics. The best results for all methods were compared, and the top-performing model was chosen based on average accuracy.

Figure 7 summarizes the 5-fold cross-validation results for all methods in the shallow learning model for all, including Javanese script OCR.

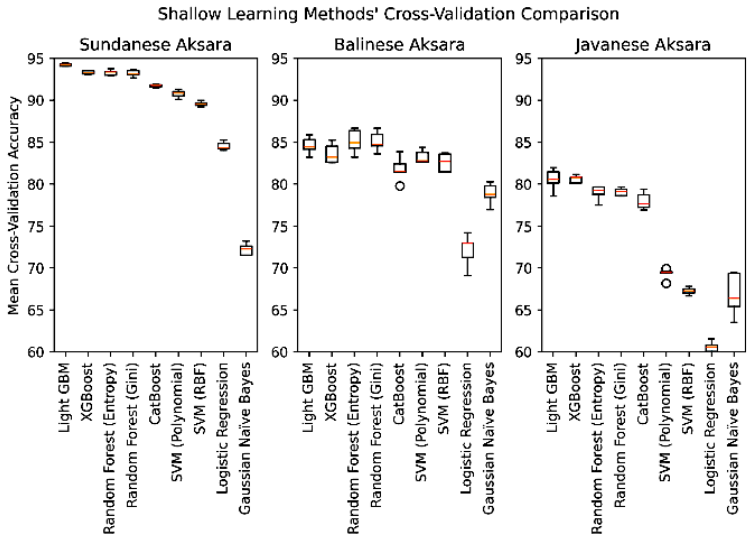


Figure 7. The Boxplot of Cross-Validation Results from Every Shallow Learning Method

Accuracy in the testing data was used to evaluate each trained model. Considering the average accuracy from the 5-fold cross-validation results for each method, XGBoost generally performed the best with an average score of 80.61%. However, when looking at individual results, LightGBM achieved the highest accuracy in one of the folds with a score of 81.99%, slightly outperforming the best XGBoost model, which achieved 81.14% in one of the folds.

These results indicate that while XGBoost is consistently strong across all folds, Light GBM has the potential to achieve higher peak performance in certain instances. Further analysis of precision, recall, and F1-score metrics will provide a more comprehensive evaluation of each model's effectiveness. Then, these results indicate that while advanced gradient boosting methods like XGBoost and Light GBM are highly effective for complex tasks, simpler models like Logistic Regression may need help with the same level of complexity. This finding highlights the importance of choosing the right model for the dataset's specific characteristics.

3.2 Method's Performances Comparison

In this section, we compare the performance of each method using more comprehensive metrics: accuracy, precision, recall, and F1-score, calculated as both macro-average and weighted average. Macro-average treats all classes equally by averaging the metric across all classes, while weighted-average accounts for each class's support (number of true instances), giving more weight to classes with more instances.

As described earlier, we compare the models that achieve the highest accuracy, and the results are summarized in Figure 8. Two models stand out as the strongest contenders: XGBoost and LightGBM. LightGBM slightly surpasses XGBoost in accuracy (and becomes the best model in one fold), while XGBoost delivers more consistent performance across the remaining metrics, including both macro-average and weighted-average scores. XGBoost achieves the highest values in five metrics, indicating stronger robustness and stability across different evaluation criteria.

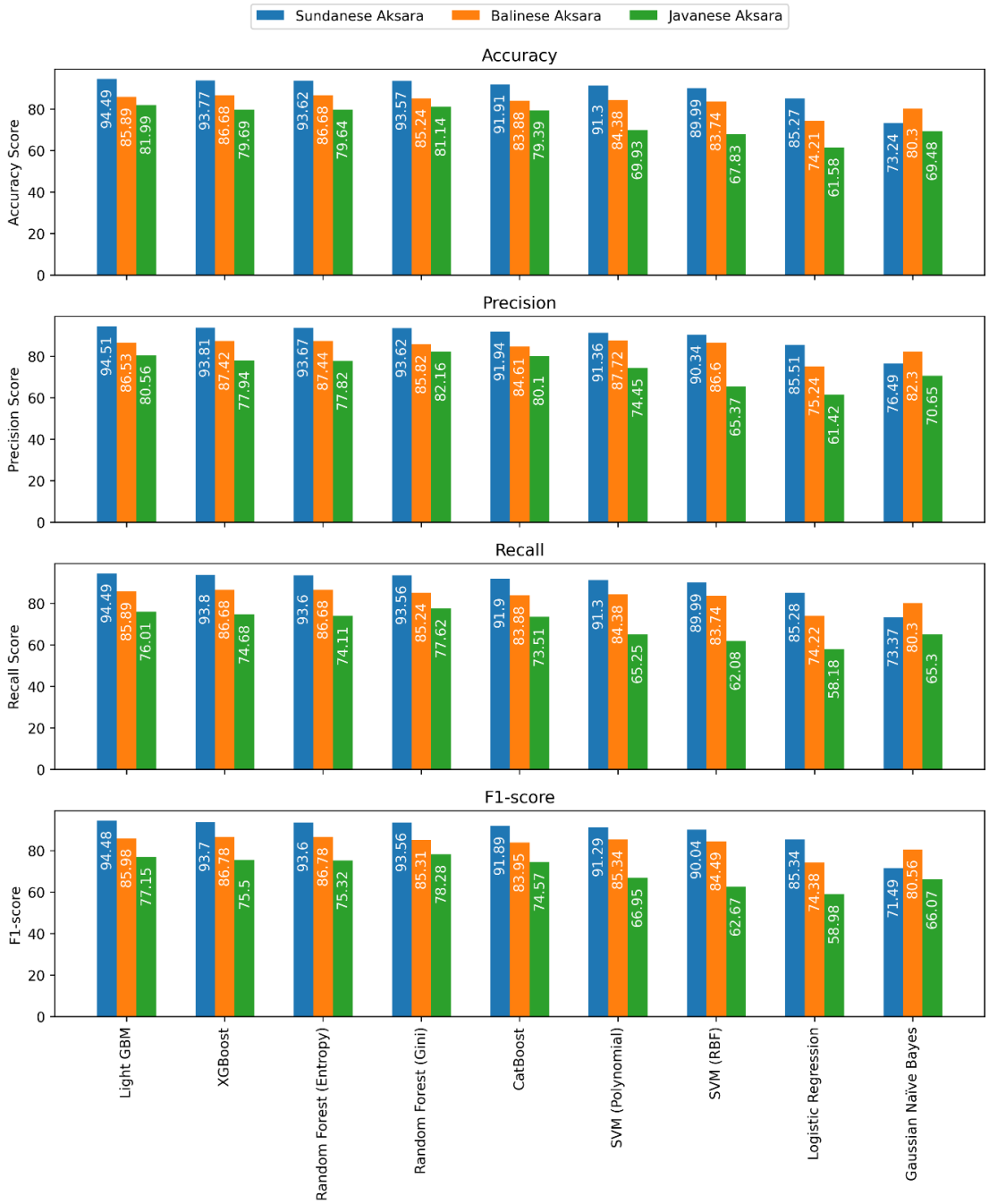


Figure 8. Performance Comparison of Three Nusantara Script OCR with Several Shallow Learning Methods

This advantage is likely supported by XGBoost's regularization and tree pruning, which help prevent overfitting and improve generalization, as well as its efficient handling of missing values and parallelized training. Meanwhile, LightGBM benefits from GOSS and EFB to accelerate training and from its leaf-wise growth strategy that can capture complex patterns. In contrast, Logistic Regression consistently performs the worst, likely because its linear assumption cannot model the non-linear variability of handwritten Javanese scripts, highlighting that gradient boosting methods are more suitable for this complex dataset than simpler linear models.

LightGBM, while achieving the highest accuracy in one of the folds, benefits from its Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) techniques. These methods reduce the number of data points and features considered during training, accelerating the process without compromising accuracy. LightGBM's leaf-wise tree growth allows for deeper trees, capturing more complex patterns in the data. On the other hand, Logistic Regression consistently ranks the lowest across all metrics. This result is likely due to its linear nature, which limits its ability to capture the non-linear relationships inherent in the handwritten Javanese script data. Logistic Regression's sensitivity to outliers and its assumption

of a linear relationship between input features and the log odds of the outcome further contribute to its lower performance. This result highlights the importance of selecting models that can effectively handle the complexity and variability of the dataset. While advanced gradient boosting methods like XGBoost and Light GBM demonstrate superior performance, simpler models like Logistic Regression may need help with complex datasets, underscoring the need for careful selection based on dataset characteristics.

3.3 Per-Class Classification Report

We created a confusion matrix to display the alignment between the target and predicted classes. This matrix helps visualize the performance of the classification model by showing where the model correctly predicted the classes and where it made errors. Results for Aksarawa, Aksarunda, and Aksarali from the corresponding best models are shown in Figures 9, 10, and 11, respectively.

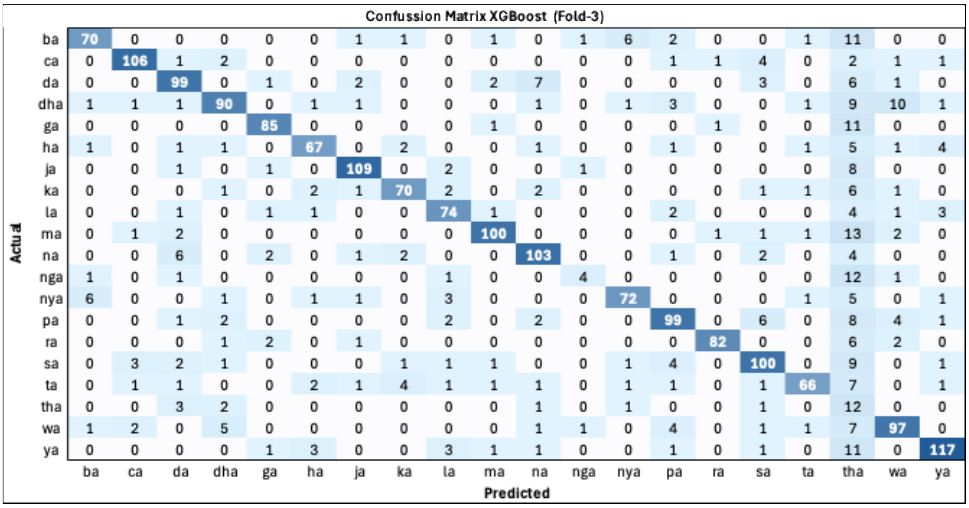


Figure 9. Confusion Matrix for Aksarawa from the Best Model Achieved, i.e., XGBoost

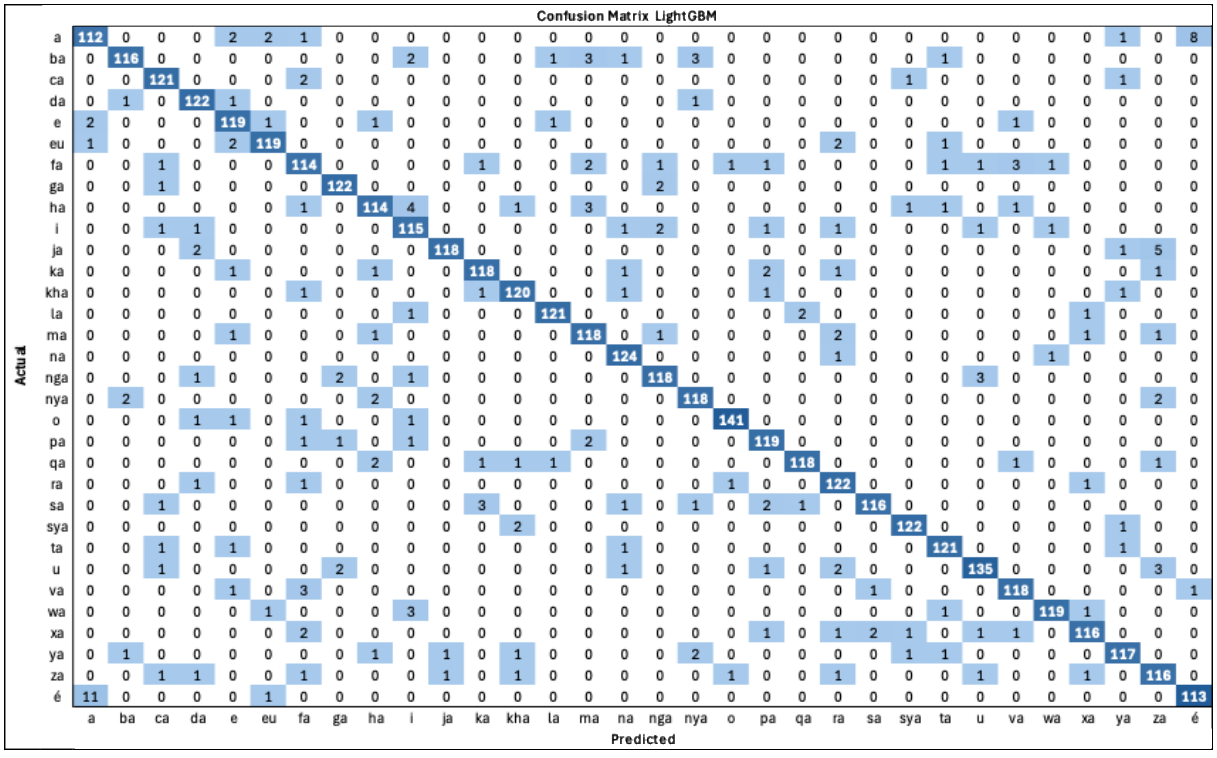


Figure 10. Confusion Matrix for Aksarunda from the best model achieved, i.e., Light GBM

The confusion matrix in Figure 9 reveals the XGBoost model's performance in recognizing Javanese script characters. It demonstrates strong accuracy for characters like "ca" and "ja" (106/118 and 109/122 correct predictions, respectively), but moderate performance for "ba" (70/94). On the other hand, it is clearly shown that the model significantly struggles with "nga" (18/60) and "tha" (18/30); many characters from other classes are misclassified as "tha". Despite the model working quite well on majority classes, the

results highlight the need for improved feature extraction, balanced datasets, and maybe additional training data to enhance accuracy for these challenging characters.

Next, Figure 10 shows results from the best Light GBM model for Aksarunda. In general, unlike for the Aksarawa cases, this model achieves quite similar results across all characters, although higher performances can be noticed for some classes. In particular, the model demonstrates excellent accuracy for characters like "ca" (121/123 correct predictions), "da" (122/124), and "ga" (122/125), indicating reliable identification for these characters. Moderate performance is observed for characters such as "ha" (114/124), "é" (113/125), and "xa" (116/124), where precision is high but recall is slightly lower due to occasional misclassifications. The model struggles a little bit with distinguishing characters "é" and "a"; it is seen on the matrix where misclassified cases are the highest. This might be due to their visually similar characters. The results emphasize the need for improved feature extraction or additional training.

Figure 11 reveals the model's performance in recognizing various characters and numerals in the Aksarali dataset. The model excels with characters like "ja" (48/49 correct predictions), "da" (44/48), and "ra" (47/50); this demonstrates high precision and recall in these cases. However, the model struggles with characters "ha" (36/50) and "nine" (37/50), which are among the lowest accuracies. Based on the confusion matrix in Figure 11, we can also observe the most frequent mispredictions occurring for classes "ha" (14 cases are misclassified from this class to other classes), "ma" (13 cases), and "nine" (13 cases). Additionally, the class that is most often misclassified as "zero", with 27 cases from other classes being incorrectly predicted as "zero".

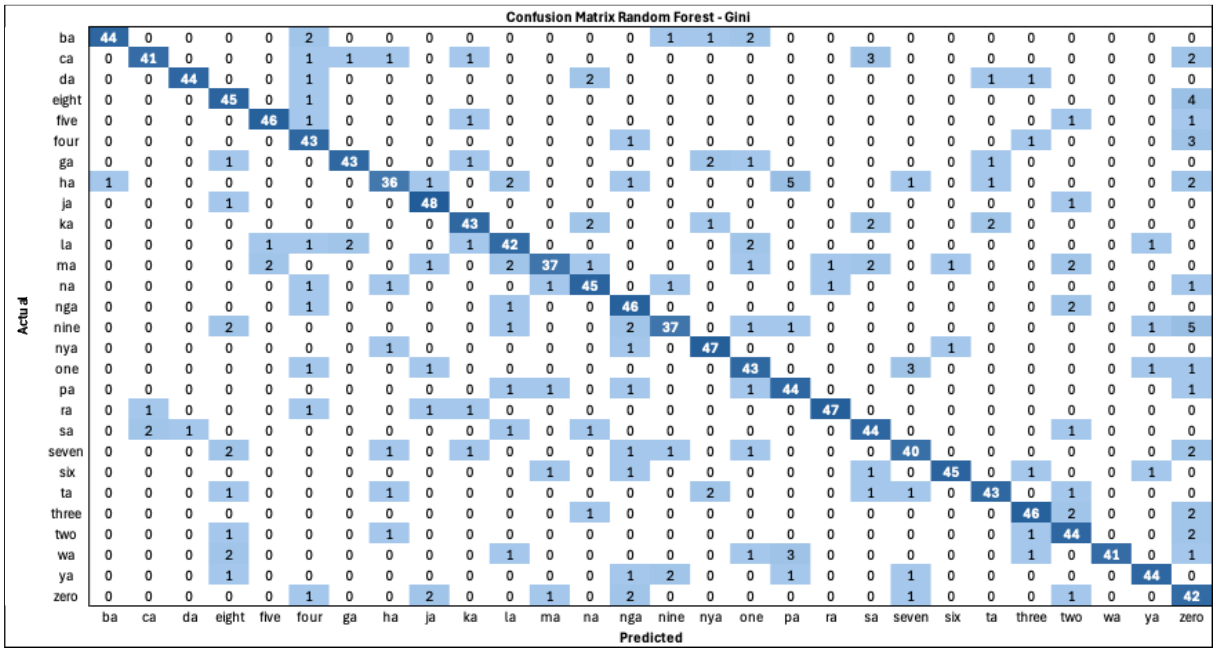


Figure 11. Confusion Matrix for Aksarali from the best model achieved, i.e., Random Forest with Gini

Additionally, we provide detailed quantitative results in precision, recall, and F1-score for each class as produced by all chosen models in Table 2. For each column, we mark the highest scores in **bold** and the lowest ones in *underlined italics*.

Based on Table 2, XGBoost model demonstrates strong performance for several Aksarawa characters, particularly "ra", "ca", "ja", "ga", and "ya", which achieve high precision, recall, and F1-scores. These results indicate the model reliably identifies these characters with minimal errors. In general, we can take F1-score as consideration, and thus "ra" with a 91.62% is the easiest class to be predicted. However, the model struggles with characters like "tha" and "nga,", which have very low F1-scores (13.64% and 29.63%, respectively), due to poor precision and recall. Additionally, classes such as "ta", "ba", and "dha" show moderate precision but low recall, suggesting the model misses many true instances of these characters. The class "tha" stands out with extremely low precision (7.69%), confirming frequent misclassifications of other characters as "tha".

For Aksarunda dataset, the Light GBM model basically work quite well; Most classes achieve an F1-score above 90%. Among highest F2-score classes are "sya" (97.21%), "la" (97.19%), "o" (97.58%), "ga" (96.83%), and "da" (96.06%), demonstrating near-perfect recognition accuracy. On the other hand, the lowest-but still considered high-performing classes are "a" (88.89%) based on F1-scores. In this case, class "a" indicates a need for improvement through data augmentation or feature refinement. Overall, the model is highly effective for recognizing Sundanese characters compared to other two scripts.

Lastly, for Aksarali dataset, the Random Forest model shows quite strong performance. Excellent results in F1-score are achieved for classes “wa” (90.11%), “ra” (94.00%, the best one), “six” (92.78%), “da” (93.62%), “ja” (92.31%). However, some classes get relatively low F1-score on classes such as “zero” (70.59%) and “ha” (78.26%), which also have lower precision or recall. Specifically, “zero” has the worst performance due to low precision (60.87%), while “ha” has low recall (72%). Classes like “three” and “five" (92.93%) demonstrate balanced precision and recall.

Table 2. Summary of Detailed Performances on Every Class in Each Nusantara Script Produced by Corresponding Best Models (Shown in Percentage)

Aksarawa (XGBoost Model)				Aksarunda (LightGBM Model)				Aksarali (Random Forest Model)			
Class	Precision	Recall	F1-Score	Class	Precision	Recall	F1-Score	Class	Precision	Recall	F1-Score
ba	87.50	74.47	80.46	a	*88.89*	*88.89*	*88.89*	ba	97.78	88.00	92.63
ca	92.98	89.08	90.99	ba	96.67	91.34	93.93	ca	93.18	82.00	87.23
da	82.50	81.82	82.16	ca	94.53	96.80	95.65	da	97.78	89.80	93.62
dha	84.91	74.38	79.30	da	94.57	97.60	96.06	eight	80.36	90.00	84.91
ga	91.40	86.73	89.01	e	92.25	95.20	93.70	five	93.88	92.00	92.93
ha	87.01	78.82	82.72	eu	95.97	95.20	95.58	four	78.18	89.58	83.50
ja	92.37	**89.34**	90.83	fa	89.06	89.76	89.41	ga	93.48	87.76	90.53
ka	87.50	80.46	83.83	ga	96.06	97.60	96.83	ha	85.71	*72.00*	78.26
la	83.15	84.09	83.62	ha	93.44	90.48	91.94	ja	88.89	**96.00**	92.31
ma	92.59	82.64	87.34	i	89.84	92.74	91.27	ka	87.76	86.00	86.87
na	85.83	85.12	85.48	ja	**98.33**	93.65	95.93	la	82.35	84.00	83.17
nga	57.14	*20.00*	29.63	ka	95.16	94.40	94.78	ma	90.24	74.00	81.32
nya	87.80	79.12	83.24	kha	95.24	96.00	95.62	na	86.54	88.24	87.38
pa	83.19	79.20	81.15	la	97.58	96.80	97.19	nga	80.70	92.00	85.98
ra	**96.47**	87.23	**91.62**	ma	92.19	94.40	93.28	nine	88.10	74.00	80.43
sa	82.64	80.65	81.63	na	94.66	**98.41**	96.50	nya	88.68	94.00	91.26
ta	90.41	74.16	81.48	nga	95.16	94.40	94.78	one	81.13	86.00	83.50
tha	*7.69*	60.00	*13.64*	nya	94.40	95.16	94.78	pa	81.48	89.80	85.44
wa	80.17	80.83	80.50	**o**	97.92	97.24	**97.58**	**ra**	95.92	92.16	**94.00**
ya	90.00	84.17	86.99	pa	92.97	95.97	94.44	sa	83.02	88.00	85.44
				qa	97.52	94.40	95.93	seven	85.11	81.63	83.33
				ra	91.73	96.83	94.21	six	95.74	90.00	92.78
				sa	97.48	92.80	95.08	ta	89.58	86.00	87.76
				sya	96.83	97.60	97.21	three	90.20	90.20	90.20
				ta	95.28	96.80	96.03	two	80.00	89.80	84.62
				u	95.07	93.10	94.08	wa	**100.00**	82.00	90.11
				va	94.40	95.16	94.78	ya	91.67	88.00	89.80
				wa	97.54	95.20	96.36	*zero*	*60.87*	84.00	*70.59*
				xa	95.87	92.80	94.31				
				ya	95.12	93.60	94.35				
				za	89.92	92.80	91.34				
				e	92.62	90.40	91.50				

3.4 Discussion

XGBoost and Light GBM excel due to their advanced gradient-boosting techniques. XGBoost’s regularization (L1 and L2) helps prevent overfitting, and its ability to handle missing values internally simplifies preprocessing [25]. XGBoost’s tree pruning technique also keeps only the most relevant splits, enhancing model generalization [26]. Light GBM, on the other hand, uses Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) to reduce the number of data points and features considered during training, which significantly speeds up the process without sacrificing accuracy [27]. Light GBM’s leaf-wise tree growth allows for deeper trees and better accuracy on complex datasets [28].

Logistic Regression performed the weakest, with an average accuracy of 60.62%. This result is likely due to its simplicity and linear nature, which makes it less capable of capturing complex, non-linear relationships in the data. Logistic Regression assumes a linear relationship between the input features and the log odds of the outcome, which may not hold for the intricate patterns present in handwritten Javanese script. Logistic Regression is also sensitive to outliers and may require larger datasets to perform better [17].

Several studies have developed Indonesian ethnic script datasets. Susanto et al. [29] developed a dataset for Javanese script handwriting containing 3,360 data items. Pratama et al. [9] developed OCR on Balinese script handwriting with a special dataset. Their Balinese script handwriting dataset contains 253 data items. Paulus et al. [30] used OCR to detect Sundanese script handwriting from their dataset. The dataset contains 7,361 data items. In this study, we developed a dataset containing three handwriting scripts: Javanese, Balinese, and Sundanese. Each dataset has 11,061, 20,224, and 6,981 data items, respectively. Our research contribution is a novel dataset consisting of three handwriting scripts: Javanese, Balinese, and Sundanese, and has a total of 38,266 records. Table 3 summarizes the comparison of our study with state-of-the-art studies in terms of the script used.

Table 3. Record Size Comparisons in Indonesian Local Script Handwriting Datasets

No	Reference	Nusantara Script Handwriting in Dataset			Record Size
		Javanese	Sundanese	Balinese	
1	Susanto *et al.* [29]	✓	✗	✗	3.360
2	Pratama *et al.* [9]	✗	✓	✗	253
3	Paulus *et al.* [30]	✗	✗	✓	7.361
4	This Research	✓	✓	✓	38.266

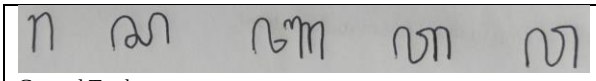
We hypothesize that shallow learning outperforms deep learning on smaller datasets. Several previous studies have used deep learning for OCR on several Nusantara scripts. Kesaulya *et al.* [31] used CNN for Javanese script and obtained an accuracy of 62%. Rifaldy *et al.* [32] used the LeNet-5 model, a variation of CNN and a deep learning technique. The accuracy of the model was 80%. Putra *et al.* [33] used CNN-based OCR on Balinese handwriting and obtained a performance of 81%. Indrawan *et al.* [34] used Tesseract OCR based on long short-term memory (LSTM), another branch of deep learning, and obtained an accuracy of 67%. Maliki *et al.* [35] created OCR for Sundanese script with a CNN extreme learning machine (CNN-ELM) and found an accuracy of 88%. Gerhana *et al.* [36] did not use deep learning but KNN for OCR on the Sundanese script, and their accuracy was 83%. The contribution of our research is two-fold: First, a shallow learning method, Light GBM, has a better performance than deep learning methods for OCR on both Javanese and Balinese script handwriting. Second, a shallow learning method, Random Forest, performs better than deep learning and KNN methods on Sundanese script handwriting. Table 4 summarizes the performance comparison between our proposed shallow and deep learning methods in state-of-the-art research.

Table 4. Performance Comparison between Shallow Learning (Proposed) and Deep Learning Methods form State-of-the-art Research

No	Script Handwriting	Reference	Method	Machine Learning Type	Accuracy
1	Javanese	Kesaulya *et al.* [31]	CNN	Deep Learning	62%
		Rifaldy *et al.* [32]	LeNet-5	Deep Learning	80%
		Proposed Method	Light BGM	Shallow Learning	82%
2	Balinese	Putra *et al.* [33]	CNN	Deep Learning	81%
		Indrawan *et al.* [34]	LSTM	Deep Learning	67%
		Proposed Method	Light BGM	Shallow Learning	95%
3	Sundanese	Maliki *et al.* [35]	CNN-ELM	Deep Learning	88%
		Gerhana *et al.* [36]	KNN	Deep Learning	83%
		Proposed Method	Random Forest	Shallow Learning	87%

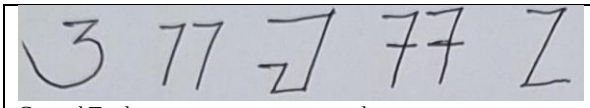
3.3 Cross-Script Models

We added an experiment in the form of testing simultaneous character recognition for several regional scripts, namely Javanese, Sundanese, and Balinese. In this experiment, images containing characters from different scripts are tested using the same shallow learning algorithms, including Random Forest (Gini & Entropy), SVM (Poly & RBF), Logistic Regression, Gaussian Naive Bayes, XGBoost, LightGBM, and CatBoost. Figure 12 displays the results of the cross-script model test.



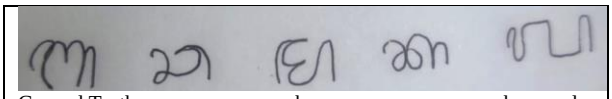
Ground Truth	ra	sa	nya	ta	wa
Random Forest-Gini	ra	na	ya	ha	ra
Random Forest-Entropy	ra	na	na	ga	ra
SVM-Poly	na	na	na	na	na
SVM-RBF	ra	na	na	ha	wa
Logistic Regression	dha	na	na	ha	ta
Gaussian Naïve Bayes	ra	na	ka	na	wa
XGBoost	ra	na	ka	ha	pa
LightGBM	ra	sa	nya	ha	wa
CatBoost	ra	sa	nya	ha	ra

(a) Sample of Aksarawa



Ground Truth	a	ka	ma	sa	na
Random Forest-Gini	e	ra	ra	sa	na
Random Forest-Entropy	e	ta	ra	sa	na
SVM-Poly	i	i	i	i	i
SVM-RBF	e	e	e	sa	e
Logistic Regression	ta	eu	ta	eu	ca
Gaussian Naïve Bayes	e	ta	ra	sa	na
XGBoost	u	ra	ta	sa	na
LightGBM	pa	ka	ta	sa	na
CatBoost	e	ka	va	sa	na

(b) Sample of Aksarunda



Ground Truth	ba	ca	ma	ka	la
Random Forest-Gini	ba	ca	ma	ka	la
Random Forest-Entropy	ba	ca	ma	ka	la
SVM-Poly	ba	ca	five	five	la
SVM-RBF	ba	ca	ma	ka	la
Logistic Regression	ba	ka	na	ka	nya
Gaussian Naïve Bayes	ja	ja	wa	wa	ga
XGBoost	ba	ca	ma	ka	la
LightGBM	ba	ca	ma	ka	la
CatBoost	ba	ca	ma	ka	La

(c) Sample of Aksarali

Figure 12. Example of Cross-Script Models for Akrawa (a), Aksarunda (b), and Aksarali (c), where the red writing means the character is incorrectly predicted. Best viewed in color mode

The test results show that most of the shallow learning models are still able to recognize characters with good accuracy despite having to process multi-character inputs simultaneously. Whether Aksarawa, Aksarunda or Aksarali, LightGBM is able to classify almost all characters correctly. CatBoost only makes one to two minor errors. Random Forest and XGBoost misrecognized three to four characters. Whereas Gaussian Naive Bayes and Logistic Regression models

tend to produce more deviant predictions. These findings indicate that the ensemble boosting and SVM methods still consistently provide relatively stable performance even with the variations in script form.

4. Conclusions

This research evaluated the effectiveness of various shallow learning methods for recognizing handwritten Javanese script characters using the Aksarawa dataset. The methods implemented included Random Forest, SVM, Logistic Regression, Gaussian Naïve Bayes, and several boosting techniques such as XGBoost, Light GBM, and CatBoost. Through a comprehensive analysis using 5-fold cross-validation, ensemble learning-based methods—namely XGBoost, Light GBM, and Random Forest—emerged as the best-performing models for the Aksarawa, Aksarunda, and Aksarali datasets, respectively. Confusion matrices for each best performer were demonstrated, and detailed performance metrics, including precision, recall, and F1-score, were analyzed.

The results revealed that the XGBoost model excelled in recognizing Aksarawa characters like “ra” (91.62% F1-score) and “ja” (90.83%), though it extremely struggled with visually similar characters such as “tha” (13.64%) and “nga” (29.63%). For the Aksarunda dataset, the Light GBM model achieved near-perfect accuracy for classes like “sya” (97.21%) and “o” (97.58%), while “a” (88.89%) showed room for improvement. Similarly, the Random Forest model performed well for Aksarali characters such as “ra” (94.00%) and “six” (92.78%), but underperformed for “zero” (70.59%) and “ha” (78.26%). In addition, by benchmarking our models to prior studies, we noticed that shallow learning outperforms other comparative methods on smaller datasets. CNN and LSTM have achieved accuracies ranging from 62% to 88% for Nusantara scripts, while our shallow learning methods in this study significantly showed superior performances for Javanese, Sundanese, and Balinese scripts.

For future work, efforts should focus on improving feature extraction techniques, augmenting training data for underperforming classes, and exploring hybrid models that combine shallow and deep learning approaches to enhance recognition accuracy. Additionally, expanding the dataset to include more diverse handwriting samples and optimizing hyperparameters could further refine model performance. This study provides valuable insights into the application of shallow learning methods for OCR tasks, contributing to the preservation and digitalization of Indonesia’s Nusantara scripts.

Acknowledgements

We would like to thank the Directorate of Research and Community Service (PPM) of Telkom University for funding this research through the university’s Leading Applied Research Scheme Number: 544/PNLT3/PPM/2023. We also thank Aditya Firman Ihsan who has shared the Indonesian Local Script Characters dataset in Mendeley Data.

References

[1] I. Krismayani, J. Jumino, and J. Wasisto, “Selected Webliographies on Indonesian Cultural Heritages,” in E3S Web of Conferences, EDP Sciences, 2021, p. 05008. Accessed: Dec. 14, 2024. [Online]. Available: https://www.e3s-conferences.org/articles/e3sconf/abs/2021/93/e3sconf_icenis2021_05008/e3sconf_icenis2021_05008.html

[2] A. Zainal, M. Raden, R. Rustopo, and T. Haryono, “Transforming Sundanese Script From Palm Leaf to Digital Typography,” vol. 509, no. Icollite, pp. 645–652, 2020.

[3] E. S. N. Sumarlina, R. S. M. Permana, and U. A. Darsa, “The Role of Sundanese Letters as the One of Identity and Language Preserver,” 2020, Accessed: Dec. 14, 2024. [Online]. Available: https://books.google.com/books?hl=en&lr=&id=xB0oEAAAQBAJ&oi=fnd&pg=PA273&dq=The+Role+of+Sundanese+Letters+as+the+One+of+Identity+and+Language+Preserver&ots=QP5FnH8lO5&sig=nviMD_rTkH-T9L72DScnIGE2_Xs

[4] Y. A. Ajani, B. D. Oladokun, S. A. Olarongbe, M. N. Amaechi, N. Rabiu, and M. T. Bashorun, “Revitalizing Indigenous Knowledge Systems via Digital Media Technologies for Sustainability of Indigenous Languages,” Preservation, Digital Technology & Culture, vol. 53, no. 1, pp. 35–44, Apr. 2024, doi: 10.1515/pdtc-2023-0051.

[5] I. U. W. M. Susanto, C. A. Sari, E. H. Rachmawanto, and D. R. I. M. Setiadi, “Javanese Script Recognition based on Metric, Eccentricity and Local Binary Pattern,” 2021 International Seminar on Application for Technology of Information and Communication (iSemantic), Sep. 2021, doi: 10.1109/isemantic52711.2021.9573232.

[6] C. A. S. Susanto, E. H. Rachmawanto, I. U. W. Mulyono, and N. M. Yaacob, “A Comparative Study of Javanese Script Classification with GoogleNet, DenseNet, ResNet, VGG16 and VGG19,” Scientific Journal of Informatics, vol. 11, no. 1, pp. 31–40, Jan. 2024, doi: 10.15294/sji.v11i1.47305.

[7] R. Widiarti and F. T. Adji, “Enhancing the Transliteration of Words written in Javanese Script through Augmented Reality,” etasr.com, Oct. 2024, doi: 10.48084/etasr.8312.

[8] M. A. Rasyidi, T. Bariyah, Y. I. Riskajaya, and A. D. Septyani, “Classification of handwritten Javanese script using random forest algorithm,” Bulletin of Electrical Engineering and Informatics, vol. 10, no. 3, pp. 1308–1315, May 2021, doi: 10.11591/eei.v10i3.3036.

[9] A. Pratama, M. D. Sulistiyo, and A. F. Ihsan, “Balinese Script Handwriting Recognition Using Faster R-CNN,” Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), vol. 7, no. 6, pp. 1268–1275, Nov. 2023, doi: 10.29207/resti.v7i6.5176.

[10] M. H. Faishal, M. D. Sulistiyo, and A. F. Ihsan, “Javanese Script Letter Detection Using Faster R-CNN,” Indonesian Journal of Artificial Intelligence and Data Mining, vol. 6, no. 2, p. 243, Aug. 2023, doi: 10.24014/ijaidm.v6i2.24641.

[11] Ö. F. Ertuğrul, J. M. Guerrero, and M. Yilmaz, “Shallow Learning vs. Deep Learning: A Practical Guide for Machine Learning Solutions,” Springer, 2024.

[12] Y. Meir, O. Tevet, Y. Tzach, S. Hodassman, R. D. Gross, and I. Kanter, “Efficient shallow learning as an alternative to deep learning,” *Sci. Rep.*, vol. 13, no. 1, pp. 1–7, 2023, doi: 10.1038/s41598-023-32559-8.

[13] V.-H. Nhu and others, “Comparison of Support Vector Machine, Bayesian Logistic Regression, and Alternating Decision Tree Algorithms for Shallow Landslide Susceptibility Mapping along a Mountainous Road in the West of Iran,” Applied Sciences, vol. 10, no. 15, p. 5047, Jul. 2020, doi: 10.3390/app10155047.

[14] G. Tzougas and K. Kutzkov, “Enhancing Logistic Regression Using Neural Networks for Classification in Actuarial Learning,” Algorithms, vol. 16, no. 2, p. 99, Feb. 2023, doi: 10.3390/a16020099.

[15] Goel and D. Gorse, "Deep vs. Shallow Learning: A Benchmark Study in Low Magnitude Earthquake Detection." [Online]. Available: https://arxiv.org/abs/2205.00525
[16] P. Schober and T. R. Vetter, "Logistic Regression in Medical Research," *Anesth. Analg.*, vol. 132, no. 2, pp. 365–366, 2021, doi: 10.1213/ANE.0000000000005247.
[17] J. J. Levy and A. J. O'Malley, "Don't dismiss logistic regression: the case for sensible extraction of interactions in the era of machine learning," BMC Medical Research Methodology, vol. 20, no. 1, Jun. 2020, doi: 10.1186/s12874-020-01046-3.
[18] V.-H. Dang and N.-D. Hoang, "L.-M.-D," Nguyen, D. T. Bui, and P. Samui, "A Novel GIS-Based Random Forest Machine Algorithm for the Spatial Prediction of Shallow Landslide Susceptibility, "Forests, vol. 11, no. 1, p. 118, Jan. 2020, doi: 10.3390/f11010118.
[19] M. Ismail, N. Hassan, and S. S. Bafjaish, "Comparative Analysis of Naive Bayesian Techniques in Health-Related for Classification Task," *J. Soft Comput. Data Min.*, vol. 1, no. 2, pp. 1–10, 2020, doi: 10.30880/jscdm.2020.01.02.001.
[20] Wickramasinghe and H. Kalutarage, "Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation," Soft Computing, vol. 25, no. 3, pp. 2277–2293, Sep. 2020, doi: 10.1007/s00500-020-05297-6.
[21] R. Iranzad and X. Liu, "A review of random forest-based feature selection methods for data science education and applications," International Journal of Data Science and Analytics, Feb. 2024, doi: 10.1007/s41060-024-00509-w.
[22] H. A. Salman, A. Kalakech, and A. Steiti, "Random Forest Algorithm Overview," *Babylonian J. Mach. Learn.*, vol. 2024, pp. 69–79, 2024, doi: 10.58496/bjml/2024/007.
[23] M. S. Hosen and R. Amin, "Significant of Gradient Boosting Algorithm in Data Management System," *Eng. Int.*, vol. 9, no. 2, pp. 85–100, 2021, doi: 10.18034/ei.v9i2.559.
[24] F. G. Boldini, D. Kuhn, L. Friedrich, and S. A. Sieber, "Practical guidelines for the use of gradient boosting for molecular property prediction," Journal of Cheminformatics, vol. 15, no. 1, Aug. 2023, doi: 10.1186/s13321-023-00743-7.
[25] Mljourney, "XGBoost vs LightGBM: Detailed Comparison - ML Journey." [Online]. Available: https://mljourney.com/xgboost-vs-lightgbm-detailed-comparison/
[26] "XGBoost, "vs LightGBM: Gradient Boosting in the Spotlight." [Online]. Available: https://dataheadhunters.com/academy/xgboost-vs-lightgbm-gradient-boosting-in-the-spotlight/
[27] Mlq, "Ai, 'XGBoost and LightGBM | Machine Learning Course.'" [Online]. Available: https://mlq.ai/academy/machine-learning/5/xgboost-lightgbm/
[28] A. C. Bentéjac and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," Artificial Intelligence Review, vol. 54, no. 3, pp. 1937–1967, Aug. 2020, doi: 10.1007/s10462-020-09896-5.
[29] A. Susanto, I. U. W. Mulyono, C. A. Sari, E. H. Rachmawanto, M. De Rosal Ignatius Moses Setiadi, and K. Sarker, "Handwritten Javanese script recognition method based 12-layers deep convolutional neural network and data augmentation," Int J Artif Intell ISSN, vol. 2252, no. 8938, p. 1449, 2023.
[30] E. Paulus, M. Suryani, S. Hadi, and F. Natsir, "An initial study to solve imbalance sundanese handwritten dataset in character recognition," in 2018 Third International Conference on Informatics and Computing (ICIC), IEEE, 2018, pp. 1–6. Accessed: Dec. 15, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8780496/
[31] Text Image Recognition Using Convolutional Neural Networks," in 2022 International Electronics Symposium (IES), IEEE, 2022, pp. 534–539. Accessed: Dec. 15, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9888527/
[32] M. R. Rifaldy, "Implementation Of Architecture LeNet-5 For Javanese Scripts Handwriting Recognition," in FoITIC: The 3rd Faculty of Industrial Technology International Congress, 2021. Accessed: Dec. 15, 2024. [Online]. Available: https://econference.itenas.ac.id/index.php/foitic2021/foitic2021/paper/view/17
[33] I. G. N. A. C. Putra et al., "Implementasi Metode Convolutional Neural Network Pada Pengenalan Aksara Bali Berbasis Game Edukasi," SINTECH (Science and Information Technology) Journal, vol. 6, no. 1, pp. 1–15, 2023.
[34] G. Indrawan, A. Asroni, L. J. E. Dewi, I. G. A. Gunadi, and I. K. Paramarta, "Balinese Script Recognition Using Tesseract Mobile Framework," Lontar Komputer: Jurnal Ilmiah Teknologi Informasi, vol. 13, no. 3, p. 160, 2022.
[35] I. Maliki and A. Febriansyah, "Implementation of Convolutional Neural Network–Extreme Learning Machine for Handwriting Recognition of Sundanese Script," in 2023 9th International Conference on Signal Processing and Intelligent Systems (ICSPIS), IEEE, 2023, pp. 1–4. Accessed: Dec. 15, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10402761/
[36] Y. A. Gerhana, A. R. Atmadja, and M. F. Padilah, "Comparison of Template Matching Algorithm and Feature Extraction Algorithm in Sundanese Script Transliteration Application using Optical Character Recognition," Jurnal Online Informatika, vol. 5, no. 1, pp. 73–80, 2020.