A Hybrid Cost Sensitive Ensemble Approach for Indonesian Local Script Handwritten Recognition

Dian Rakhmawati Lestari
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
dian.lestari002@binus.ac.id

Fadhlan Nur Rachman
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
fadhlan.rachman@binus.ac.id

Henry Lucky
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
henry.lucky@binus.ac.id

Rifqi Alfinnur Charisma
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
rifqi.charisma@binus.ac.id

***Abstract*—This paper discusses the preservation of Indonesia's local scripts, which are increasingly threatened by the dominance of the Latin alphabet in digital communication. To support the recognition of handwritten characters in Nusantara scripts, this study proposes a two-stage hybrid workflow that is sensitive to computational costs and data imbalance. The dataset used includes nine Nusantara scripts, 228-character classes, and 55,713 images with a class imbalance ratio of up to 14.5:1. In the first stage, pre-trained MobileNetV2 and EfficientNet-B0 are used as feature extractors, and 1,280-dimensional features are compressed to 256 dimensions using PCA. The second stage trained XGBoost, LightGBM, and CatBoost with class weights without adding or reducing data. The six models were then combined using WeightedVote based on Macro F1 scores. Experimental results on two T4 GPUs show that WeightedVote achieved a Macro F1 score of 0.9160, which is 3.90% higher than the best individual models, namely CatBoost and MobileNetV2, with a Macro F1 score of 0.8770. The highest performance was achieved with the Lontara script, while the Javanese script showed the lowest performance due to the more complex variations in character shapes.**

Keywords—Nusantara scripts, handwritten character recognition, cost-sensitive ensemble learning, gradient boosting, class imbalance, WeightedVote ensemble.

I. INTRODUCTION

Indonesia is a country with a rich and diverse culture, one aspect of which is reflected in the various regional scripts found across the archipelago, including the Balinese, Batak, Javanese, Jawi, Kawi, Lampung, Lontara, Pallawa, and Sundanese scripts. For thousands of years, these scripts have preserved history, literature, and knowledge while serving as the visual identity of the nation's cultural diversity. Nevertheless, the conserved utilization of these scripts has dynamically moved in favor of Latin alphabet all over education, government and daily utilization, which this legacy is at risk of extinction unless appropriate digital preservation strategies are implemented in the near future [1]. Handwritten Character Recognition (HCR) is considered as a realistic approach to digitize manuscripts and historical documents in local scripts. However, developing a dependable HCR is rather arduous as the writing style varies from person to person and there is a scarcity of labelled data for the majority of these regional scripts [2][3].

Existing HCR studies commonly use one of two approaches. The first approach exploits classical machine learning, coupling handcrafted descriptors such as HOG or Hu Moments with classifiers like SVM, KNN and Random Forest. It is a light-weight method and works well even when the amount of training data is limited [4]. For instance, when the data is limited, gradient boosting models such as XGBoost and LightGBM can be comparable or even better than VGG19 on Javanese, Sundanese and Balinese script datasets [2]. Similarly, KNN and Random Forest with Hu Moment descriptors offer stable results for Lontara script recognition. Second, deep learning approaches based on Convolutional Neural Networks (CNNs) and Vision Transformers are capable of effectively learning feature representations from raw pixels. Individual benchmark results show excellent performance 12-layer DCNN with near perfect accuracy on 120 Javanese script characters [5], a CNN with elastic distortion augmentations on Batak Toba script [6], and transfer learning from a pretrained MobileNetV2 framework achieving near-perfect accuracy of 99.8% on Lampung script [7]. Lightweight architectures such as MobileNetV2 have proven effective in addressing data limitations while maintaining strong feature representations.

However, there is a significant gap in literature. No study has yet conducted an experiment involving all nine Nusantara scripts together in a single standardized experiment, as most studies focus only on one or two scripts [1]. Furthermore, the severe class imbalance in the dataset has not been addressed through cost-sensitive learning strategies in the context of the Nusantara HCR, so the evaluation metrics used tend to be biased toward the majority class [8]. Finally, no previous study has combined feature extraction using pre-trained CNNs with gradient boosting-based ensemble classification for imbalanced multi-script settings. Therefore, the potential for conducting DCNN feature extraction experiments with tree-based ensemble efficiency has not yet been systematically explored [2].

To fill this gap, this study proposes an experimental framework called Cost-Sensitive Hybrid Deep Feature Extraction and Ensemble Classification, which separates the feature representation stage from the classification stage. A lightweight pretrained CNN backbone from ImageNet, namely MobileNetV2 or EfficientNet-B0 [9][10], is used as a frozen feature extraction method that generates feature embeddings without the need for gradient-based retraining. These embeddings are then classified by three cost-sensitive gradient boosting ensemble models such as XGBoost,

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

LightGBM, and CatBoost [11][12][13]. Class weights are calculated analytically from the class frequency distribution in the training set using the weight-balanced formula from Scikit-Learn, so that the model incurs a large penalty for misclassifying the minority class without using memory intensive oversampling and undersampling techniques.

This proposed framework brings together classical and deep learning approaches. Unlike traditional machine learning, which depends on manually engineered features, this framework applies CNNs for automatic feature extraction, avoiding the errors and bias of hand-crafted descriptors. This hybrid architecture is also significantly more computation- and data-efficient than a fully data-driven deep learning approach, making it well suited for underrepresented scripts. Cost-sensitive weighting further ensures that all 228 character classes across the nine local scripts are learned equitably. Additionally, the research aligns with the Sustainable Development Goals (SDGs): SDG 4 on Quality Education, by providing a digital resource for cultural heritage [14], and SDG 11 on Sustainable Cities and Communities, by providing an algorithmic framework for the systematic digitization of historical manuscripts [15]. All datasets, source code, and model weights are publicly shared to facilitate continuous research.

II. Related Work

A. Shallow Machine Learning for Script Recognition

Conventional machine learning algorithms have been shown to be reliable for handwritten character recognition systems when the availability of data is limited. Gradient boosting has demonstrated competitive results [2]. XGBoost, LightGBM, and Random Forest are computationally efficient and outperform the VGG19 architecture in classifying three Nusantara scripts. This is backed up by results on the Lontara script where KNN and Random Forest using Hu Moment geometric features return robust results in transliterating [4]. However, these methods are heavily reliant on the quality of raw data. Sutramiani [16] report that KNN segmentation accuracy fell from 92% to 52% when using high-texture Balinese lontar manuscripts as opposed to clean images, while a similar shortcoming occurs with Naive Bayes for Javanese script, which achieved only 65% accuracy without further feature engineering [3]. This high sensitivity to the quality of raw images forms the basis for the selection of Convolutional Neural Networks (CNNs) as automatic feature extractors.

For the classification stage, this study adopted three gradient boosting methods due to their effectiveness in handling high-dimensional data. XGBoost minimizes the risk of overfitting through a regularized additive model with second-order gradient statistics and parallel computation [11]. LightGBM offers significantly faster training times through a leaf-by-leaf tree-growing strategy combined with Gradient-Based One-Side Sampling and Exclusive Feature Bundling techniques [12]. On the other hand, CatBoost prevents target leakage through an ordered boosting mechanism that is crucial for handling large-scale multi-class classification such as the 228 classes of Nusantara scripts [13]. These three algorithms are ideally combined with a CNN architecture because the resulting feature vectors have a histogram distribution that aligns with decision tree-based separation principles [2].

B. Data Augmentation and Handling Class Imbalance

Limited datasets and class imbalance are the most persistent challenges in research on the recognition of Nusantara scripts. An Enhanced Balancing GAN was used to synthesize minority-class samples of ancient Javanese text, and this generative approach successfully improved the F1 score from 95.93% to 96.77% [8]. To handle contrast problems commonly encountered in native handwriting, previous studies applied Top Hat Transform, CLAHE, and DWT image fusion techniques on Kaganga characters [17], which resulted in a 3.73% increase in accuracy. Related research further stresses the importance of domain-specific data [18]. For instance, GPT-4o and Llama 3.2 consistently exhibit near-zero performance on eight local scripts, suggesting that large language and vision models still require domain-specific labels. Along similar lines, incorporating external Arabic data into the dataset boosted ResNet34's accuracy for cursive Jawi recognition to 96% [19]. While these complex data manipulation methods are effective, this study incorporates cost-sensitive weighting to mitigate class imbalance. This approach avoids computationally intensive generative processes and dataset expansion entirely, resulting in a highly memory-efficient solution.

C. Feature Extraction with Deep Learning CNNs

CNNs are commonly utilized in the handwritten character recognition problem because they can learn visual patterns directly from image data, from simple shapes to complex character patterns. In Nusantara scripts, some CNN-based methods have shown excellent performance, such as Deep CNN for Javanese script [5], CNN with elastic distortion augmentation for Batak Toba script [6], and MobileNetV2 for Lampung script [7]. Moreover, transfer learning methods such as HCR-Net were also successful in improving the performance while reducing the number of trainable parameters [20]. In this paper, we also use MobileNetV2 and EfficientNet-B0 as the backbone of the feature extractor because they can provide efficient solutions with a good visual feature representation [9], [10]. Specifically, we removed the built-in classification layers in both backbones, so that the backbones only extract feature vectors via a forward pass without being trained. Then, those features are used as input for the main classification step using the gradient boosting model.

III. METHODOLOGY

A. Experiment Flow

This research conducted a systematic experimental study on handwritten character recognition for local Indonesian scripts. Our approach combines a deep learning feature extraction pipeline and a gradient boosting classification pipeline. The study was divided into three main phases including data collection and preprocessing, modeling, feature extraction, classifier training, evaluation, and results analysis with real time application testing. The general experimental design is shown in Fig. 1.

The first phase, dataset preparation began with the collection of a dataset of handwritten characters from the Indonesian local script, sourced from Mendeley Data (2024) [21]. This raw dataset was processed for standardization to next step, including exploratory data analysis, binarization, pixel resizing, channel replication, normalization, expand splitting into training and testing sets with an 80:20 ratio for the subsequent model training phase.

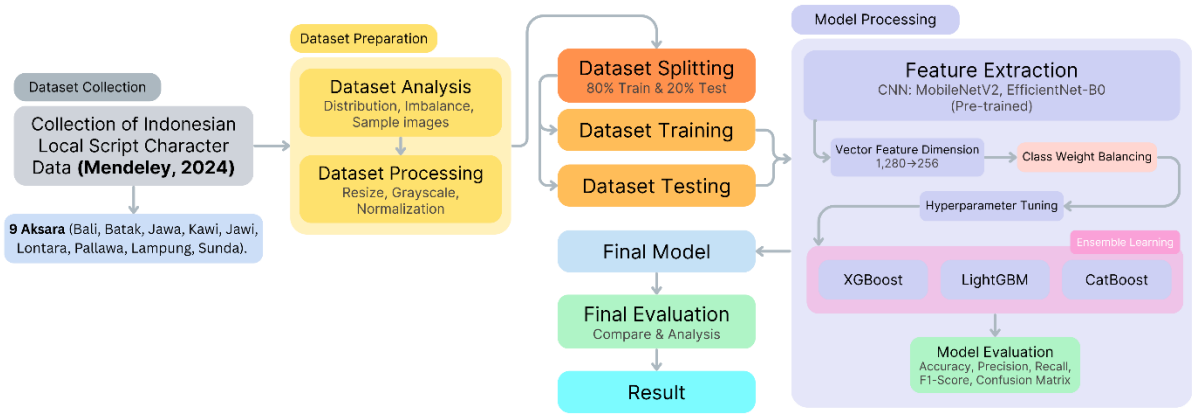

Fig. 1. Flowchart Pipeline

The second is Model Preprocessing where features must first be extracted using a frozen CNN since only the feature extraction layers are used, while the remaining layers are removed because the next step involves calculating class weights and proceeding with gradient boosting classification training and hyperparameter tuning. The final step of this phase is evaluation Metrics like Macro F1 and the Confusion Matrix for each class.

The third and final phase involves the submission of a set of test data that has been set aside and updated. The results of this phase constitute the final findings of this study, which will then be implemented via a web interface based on Streamlit for real-time HCR classification inference.

B. *Dataset*

The dataset used is the Indonesia Local Script Characters Handwriting Dataset, published on Mendeley Data and publicly [21]. The dataset contains images of handwritten characters from nine scripts of the Indonesian local script Balinese, Batak, Javanese, Jawi, Kawi, Lampung, Lontara, Pallawa, and Sundanese. It contains 55,713 handwritten images and exhibits distinct characteristics in terms of variation, resolution, and class distribution, necessitating further analysis.

TABLE I. DATASET COMPOSITION

Script	Region	Classes	Images	Percentage
Bali	Bali	18	4493	8.06%
Batak	NorthSumatera	19	1900	3.41%
Jawa	Central Java	20	9995	17.94%
Jawi	Malay Region	32	1600	2.87%
Kawi	Historical Java	31	6246	11.21%
Lampung	Lampung	20	4996	8.97%
Lontara	Sulawesi	23	2299	4.13%
Pallawa	South Asia	33	3960	7.11%
Sunda	West Java	32	20224	36.30%
Total :		228	55713	100%

Given the significant data imbalance between majority classes, such as the Sunda class, and minority classes, such as the Jawi class, single-class accuracy cannot be used as an evaluation metric because it risks introducing bias toward the majority class, as shown in Table I.

All images in the dataset are in image format. The resolution and color mode of the images are different for each character. All the images were converted into uniform 224×224 pixels grayscale images for training in preprocessing stage IIIA to bring uniform input dimension for the model. The dataset was then split into final training and test sets using Stratified split with a ratio of 80% training set and 20% test set.

C. *Feature Extraction*

Unlike conventional machine learning methods that rely on manually engineered features such as HOG or Hu Moments, this work employs a CNN architecture for automatic feature extraction. This approach is specifically chosen to avoid human error and subjectivity during the visual feature selection step. This phase commences with image preprocessing, Binarization and Thresholding of the character stroke with respect to the background. Since pre-trained CNN architectures like MobileNetV2 and EfficientNet-B0 expect color (RGB) images, the grayscale image is reconstructed by duplicating the image for 3 RGB channels and by resizing it to 224×224 pixels [9] [10].

Then process the modified images through a pretrained ImageNet model with its final classification layer removed. During this stage, the neural network functions strictly as a frozen feature extractor. A single forward pass converts visual elements such as lines, angles, and complex curves into a compact one-dimensional feature vector, with no retraining or backpropagation involved.

D. *Model Experiment Setting*

The one-dimensional feature vectors obtained from the deep learning stage were used as input to train three gradient boosting-based ensemble models. These three models were selected because they can handle high-dimensional features and perform well on multi-class classification tasks. To address class imbalance in the dataset, this study applies a cost sensitive learning strategy through class weighting. Additionally, hyperparameter optimization is performed with a search space tailored to each model, as shown in Table II. The number of estimators is set within the range of 100 to 350, while the learning rate is within the range of 0.05 to 0.2. In

XGBoost and CatBoost, model complexity is controlled via a maximum tree depth of 4 to 8, while LightGBM uses a leaf-wise growth approach with 15 to 63 leaves to capture complex feature patterns more efficiently.

TABLE II. MODEL HYPERPARAMETER SETTINGS

Hyperparameter	XGBoost	LightGBM	CatBoost
n_estimators (search space)	100-350	30-120	100-350
n_estimators (final training)	500	100	500
max_depth	4-8	leaf-wise	4-8
learning_rate	0.05-0.25	0.05-0.25	0.03-0.25
subsample	0.7-1.0	-	-
colsample_bytree	0.6-1.0	0.6-1.0	-
num_leaves	-	15-63	
min_child_samples	-	0.001-1.0	-
l2_leaf_reg	-	-	0.01-5.0
random_states	42	42	42
eval_metric	mlogloss	multi_logloss	MultiClass

In addition to optimizing key hyperparameters, regularization was also applied to reduce the risk of overfitting the training data. Settings such as subsampling and model penalties were used in accordance with the built-in mechanisms of XGBoost, LightGBM, and CatBoost. Since this study employed multiclass classification, the objective function was adjusted accordingly. To ensure experimental consistency, all models were run with the same random state, and the best configuration was selected based on the highest Macro F1 score on the validation dataset.

E. Evaluation Metrics

Due to class imbalance, we need to evaluate the model's performance using four complementary metrics: precision, recall, and F1 score per class, with the macro F1 score as the primary comparison metric. Although accuracy metrics are intuitive, they can be misleading in class-imbalanced conditions, as high accuracy can be achieved simply by overpredicting the majority of class. Before calculating Macro-F1, we must first determine the F1 scores, as Macro-F1 is the overall average of the F1 scores across all labels, calculated as the unweighted average of all K classes, as shown in the following equation 1.

$$MacroF1 = \frac{1}{K}\sum_{k=1}^{K} F1Score_k \quad (1)$$

This employed the Macro-F1 metric, in which all the classes are equally weighted regardless of the number of samples. This is conducive to imbalanced detection datasets where minority classes such as Jawi and Batak are not systematically ignored [22]. Finally, we present a dimensionally restricted confusion matrix for the specific region described in Section IV-E to visualize our findings.

IV. EXPERIMENT RESULTS

This section describes the experimental results of a cost-sensitive hybrid pipeline implemented on two T4 GPUs. The analysis consists of five subsections, each with their corresponding results and discussions.

A. Feature Extraction and Dimensionality Reduction

Both CNN backbones produce feature vectors of 1,280 dimensions per image. With a total of 55,713 images and an 80:20 stratified split, a training feature matrix of size $44{,}570 \times 1{,}280$ and a test feature matrix of size $11{,}143 \times 1{,}280$ were obtained. Before training the classifier, the dimensionality was reduced using PCA to 256 components. MobileNetV2 retained 95.9% of the variance with 256 components, while EfficientNet-B0 retained 94.1%. This compression significantly reduces memory load and training time, from 1,280 dimensions to 256 dimensions, without sacrificing meaningful discriminative information. All classifiers are trained on 256-dimensional PCA feature representations.

B. Model Comparison

Table III summarizes the evaluation results for the Macro Average F1-Score, Macro Precision, and Macro Recall for all single model combinations and ensemble voting as well as the ensemble model tested on the validation set. All configurations use cost-sensitive class weights calculated analytically using the balanced formula from Scikit-Learn.

TABLE III. COMPARISON OF MODEL PERFORMANCE

Method Combination	Macro Precision	Macro Recall	Macro F1
MobileNetV2 + CatBoost	0.8778	0.8805	0.8770
MobileNetV2 + XGBoost	0.8610	0.8578	0.8571
MobileNetV2 + LightGBM	0.7996	0.8010	0.7976
EfficientNet-B0 + CatBoost	0.8681	0.8698	0.8668
EfficientNet-B0 + XGBoost	0.8650	0.8561	0.8574
EfficientNet-B0 + LightGBM	0.8001	0.7984	0.7957
Soft Vote (6 Models)	0.9181	0.9161	0.9149
Weighted Vote (6 Models)	**0.9192**	**0.9170**	**0.9160**

The Weighted Vote Ensemble approach, which combines all 6 individual model method combinations, proved to deliver the best results with a Macro F1 score of 0.9160. The synergy of various algorithmic architectures, different boosting methods as well as features from MobileNetV2 and EfficientNet-B0 successfully compensated for the predictive weaknesses of each individual model. Meanwhile, for pure individual models, the best model was CatBoost + MobileNetV2 with a Macro F1 score of 0.8770. This finding is quite counterintuitive because EfficientNet-B0 has a larger architecture and generally produces richer representations. However, after PCA reduction to 256 dimensions, the capacity advantage of EfficientNet-B0 retaining 94.1% of the variance becomes smaller compared to MobileNetV2, which retains 95.9% of the variance. The representation quality of the two backbones becomes comparable. Under these conditions, the regularization in CatBoost proves highly effective in separating the 228 classes in the 256-dimensional feature space.

C. Weight Balancing and Cost-Sensitive Learning

Class weights were calculated analytically using Scikit-Learn. From the 228 classes, the following weight ranges were obtained a minimum of 0.337 for majority class Sundanese, an average of 1.891, and a maximum of 4.887 for the most critical minority class Jawi script. The ratio between the maximum and minimum weights is 14.5 times that of the majority class, consistent with the actual imbalance ratio of the dataset [21]. These weights are integrated via the class weight parameter in three models that we use.

The effect of cost-sensitive weighting is most evident in classes with very few test samples. The Lontara script, despite having only about 20 images per class in the test set, achieved a Macro F1 score of 0.9554. Similarly, the Batak script, with about 20 images per class, was able to achieve a Macro F1 score of 0.8583. In contrast, the Jawi script, with about 10 images per class, showed very large variations in F1 scores because the test set was too small for stable estimation. Without class weights, the model tends to ignore minority classes, which in this dataset can lead to misleading performance evaluations because they are unpredictable.

D. Character Evaluation & Error Analysis

Table IV is an overview of the average Macro F1 per script from the best performing model. In the evaluation results, we can see big changes in the metrics across character classes, which are deeply related to the level of morphological complexity and the uneven distribution of training data samples.

TABLE IV. MACRO F1 PER SCRIPT EVALUATION

Script	Classes	F1 Min	F1 Max	MacroF1
Lontara	23	0.9500	1.0000	0.9892
Pallawa	33	0.9362	1.0000	0.9862
Kawi	31	0.8525	1.0000	0.9849
Lampung	20	0.9388	1.0000	0.9760
Batak	19	0.8421	1.0000	0.9307
Sunda	32	0.7711	0.9804	0.9138
Jawi	32	0.5926	1.0000	0.8417
Bali	18	0.6869	0.9400	0.8225
Jawa	20	0.4516	0.9286	0.7464

The Lontara, Pallawa and Kawi scripts showed the highest F1 scores, a result supported by their discernible and often very consistent visual character forms. The Pallawa script, however, has very well class-balanced distribution, thereby reducing risk of evaluation bias. The Lontara script, on the other hand, showed multiple classes with perfect F1 scores, reflecting its highly distinct morphological classes.

The Javanese script is ranked the lowest with an average macro F1 score of 0.7464, although it is one of the groups with the most data. The classes with the lowest performance are Jawa_nga with an F1 score of 0.4516 with only 20 supporting data points, Jawa_da with an F1 score of 0.5888, and Jawa_ha with an F1 score of 0.5276. They are all classes of Javanese characters with visually similar features consisting of vertical strokes and curves at the bottom. This intra-character confusion in the Javanese script is inherent and was also reported in a previous 12-layer DCNN-based study [5]. Then there is the Balinese script, which exhibits a similar pattern, with the Bali_ba class having an F1 score of 0.6869 and the Bali_nga class having an F1 score of 0.7273, making them the most difficult to distinguish due to their morphological similarity to Kawi script characters, with which they are historically related.

E. Confusion Matrix Analysis

The 228 × 228 confusion matrix from the best individual model reveals morphologically structured patterns of classification errors. The most frequent inter-class errors are centered on two main patterns. First, intra-script confusion, namely errors between characters within the same script that share a similar basic form, for example, among Kawi and Pallawa characters that share a visual heritage from the Brahmi script family. Second, inter-script confusion occurs between the Javanese and Kawi scripts, which are historically and morphologically closely related, so that some of their basic characters have nearly identical stroke structures [1]. This pattern shows that additional accuracy can be obtained with script-by-script class balancing strategies or by fine-tuning the backbone model on Nusantara script. The confusion-matrix of region script can be seen in Fig. 2.

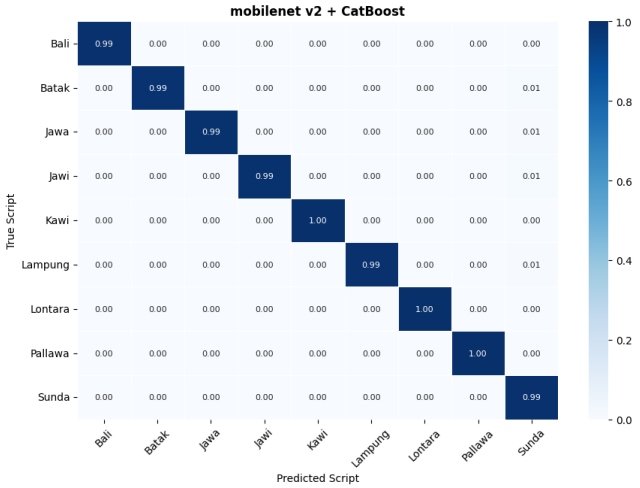

Fig. 2. Confusion Matrix

F. Selection of the Best Model

The evaluation found that the strongest overall approach was the WeightedVote ensemble of six models with a Macro F1 score of 0.9160. But in terms of single pipeline efficiency, the combination of CatBoost and MobileNetV2 gave a very efficient alternative with an F1 score of 0.8770. The optimal hyperparameters from Optuna are n_estimators is 200, max_depth is 4, learning_rate is 0.0750, subsample is 0.810, and colsample_bytree is 0.837. The depth of the tree only max depth is 4, so the 256-dimensional PCA features have enough information, and deeper trees are not required. From this configuration, we will deploy the final model for the inference phase using a web interface powered by Streamlit.

V. CONCLUSION

This study present and evaluate a two-stage hybrid workflow for recognition of handwritten characters in nine Indonesian scripts with 55,713 images in 228 classes and an imbalanced class ratio of up to 14.5:1. We extracted 1,280-dimensional features using pre-trained MobileNetV2 and EfficientNet-B0 backbones, reduced their dimensionality to

256 using PCA, and classified them with class-weighted XGBoost, LightGBM and CatBoost. Then we combined the six resulting models by using a Weighted Vote to do the final prediction. This method allows for a systematic benchmark of the recognition of these nine Nusantara scripts in one experiment using uniform metrics. Our class-weighting strategy achieved high performance on minority scripts with no dataset manipulation, with a Macro F1 score of nearly 98% for Lontara and 93% for Batak. Consequently, the macro F1 score of the weighted vote method reached about 0.9160, which is 3.9% higher than the best single model, showing the capability of combining different backbones and classification algorithms to enhance the stability and accuracy of the system.

From this success, a limitation of this experiment can be identified the Javanese script poses a major performance barrier with a Macro F1 of 0.7464. This is caused by the similarity in patterns and inherent structures among characters such as jawa_nga, jawa_ha, and jawa_da not due to data scarcity, as Javanese itself has 7,997 training samples. Furthermore, although the Jawi script achieved a Macro F1 score of 0.8417, it exhibited high inter-class variation due to the small class-specific sample size only 50 test samples per class which limited the stability of metrics per class. Finally, the Optuna hyperparameter search was limited to 20 trials per model on a subsample of only 30%, leaving room for further optimization

For future work, script specific morphological augmentation strategies targeting Javanese and Balinese characters, such as elastic distortion on stroke components, are recommended to directly address within class visual ambiguity. Partial fine-tuning of the CNN backbone's final convolutional layers on Nusantara domain data is also a promising direction to improve feature discriminability for morphologically similar characters, specifically within Javanese script.

AUTHOR CONTRIBUTION

Dian Rakhmawati Lestari: Conceptualization, Investigation, Writing - original draft, Writing - review & editing, Formal analysis, Visualization. Fadhlan Nur Rachman: Conceptualization, Data Curation, Methodology, Software, Formal analysis, Visualization, Writing - original draft. Henry Lucky: Supervision, Validation, Writing - review & editing. Rifqi Alfinnur Charisma: Supervision, Project administration, Validation, Writing - review & editing.

REFERENCES

[1] B. Krisna et al., "Deep Learning Approaches for Nusantara Scripts Optical Character Recognition," IJCCS, 2023. [Online]. Available: https://journal.ugm.ac.id/ijccs/article/view/86302/36989

[2] M. D. Sulistiyo, "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. 2025. [Online]. Available: https://jurnal.iaii.or.id/index.php/RESTI/article/view/6648/1200

[3] F. Firmansyah et al., "Optimization of Nglegena Javanese Script Recognition With Machine Learning Based on Zoning," IJODAS, 2025. [Online]. Available: https://jurnal.yoctobrain.org/index.php/ijodas/article/view/256

[4] A. Mardiani et al., "Classification of Lontara Script Using K-NN Algorithm, Decision Tree, and Random Forest Based on Hu Moments and Canny Segmentation," IJODAS, 2025. [Online]. Available: https://jurnal.yoctobrain.org/index.php/ijodas/article/view/281

[5] W. Kusuma et al., "Handwritten Javanese Script Recognition Based on 12-Layers DCNN," IJAI, 2023. [Online]. Available: https://ijai.iaescore.com/index.php/IJAI/article/view/22593/13718

[6] Rajagukguk et al., "Handwritten Batak Toba Script Recognition Based on Deep Learning Using CNN," J. AIEA, 2024. [Online]. Available: https://ioinformatic.org/index.php/JAIEA/article/view/1795

[7] T. Saputra et al., "Development of Lampung Script Characters Recognition Model using TensorFlow," IJECS, 2023. [Online]. Available: https://ejournal.radenintan.ac.id/index.php/IJECS/article/view/19878/pdf

[8] R. Cahyani et al., "Evaluation of Synthetic Data Effectiveness Using GAN in Improving Javanese Script Recognition on Ancient Manuscripts," JUTI ITS, 2025. [Online]. Available: https://juti.if.its.ac.id/index.php/juti/article/view/1256

[9] M. Sandler et al., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in Proc. CVPR, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8578572

[10] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in Proc. ICML, Long Beach, CA, USA, 2019. [Online]. Available: https://proceedings.mlr.press/v97/tan19a/tan19a.pdf

[11] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD), San Francisco, CA, USA, 2016, pp. 785-794. [Online]. Available: https://dl.acm.org/doi/10.1145/2939672.2939785

[12] G. Ke et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html

[13] L. Prokhorenkova et al., "CatBoost: Unbiased Boosting with Categorical Features," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf

[14] United Nations, "Goal 4: Quality Education," Sustainable Development Goals, 2015. [Online]. Available: https://sdgs.un.org/goals/goal4

[15] United Nations, "Goal 11: Sustainable Cities and Communities," Sustainable Development Goals, 2015. [Online]. Available: https://sdgs.un.org/goals/goal11

[16] N. P. Sutramiani, I. W. A. S. Darma, and D. M. S. Arsa, "Handwritten Balinese Script Recognition on Palm Leaf Manuscript using Projection Profile and K-Nearest Neighbor," Jurnal Ilmiah Merpati, vol. 10, no. 3, pp. 133-144, Dec. 2022, doi: 10.24843/JIM.2022.v10.i03.p02.

[17] E. D. Putra, Ermatita, and Abdiansah, "Handwritten Kaganga Script Classification Using Deep Learning and Image Fusion," Bulletin of Electrical Engineering and Informatics, vol. 14, no. 2, pp. 1290-1297, Apr. 2025, doi: 10.11591/eei.v14i2.8747.

[18] A. Santoso et al., "NusaAksara: A Multimodal and Multilingual Benchmark for Preserving Indonesian Indigenous Scripts," in Proc. ACL, 2025. [Online]. Available: https://aclanthology.org/2025.acl-long.1377.pdf

[19] M. Zulkifli et al., "Augmentation of Additional Arabic Dataset for Jawi Writing and Classification Using Deep Learning," J. Rekayasa Elektrika, 2024. [Online]. Available: https://jurnal.usk.ac.id/JRE/article/view/33722

[20] V. K. Chauhan, S. Singh, and A. Sharma, "HCR-Net: a deep learning based script independent handwritten character recognition network," *Multimedia Tools and Applications*, vol. 83, pp. 78433–78467, 2024, doi: 10.1007/s11042-024-18655-5.

[21] F. Ihsan, "Indonesian Local Script Characters Handwriting Dataset," Mendeley Data, V1, 2024. [Online]. Available: https://data.mendeley.com/datasets/vfj32bpjsf/1. doi: 10.17632/vfj32bpjsf.1.

[22] M. C. Hinojosa-Lee, J. Braet, and J. Springael, "Performance Metrics for Multilabel Emotion Classification: Comparing Micro, Macro, and Weighted F1-Scores," Applied Sciences, vol. 14, no. 21, p. 9863, 2024, doi: 10.3390/app14219863