Indonesian Journal of Data and Science

Volume 6 Issue 2
ISSN 2715-9936
https://doi.org/10.56705/ijodas.v6i2.256

*Research Article*

# Optimization of Nglegena Javanese Script Recognition With Machine Learning Based on Zoning And Normalization of Feature Extraction

**Manuel Tanbica Graciello [1], Anik Nur Handayani [2,*], Aji Prasetya Wibawa [3]**

[1] Universitas Negeri Malang, Malang, Indonesia, manuel.tanbica.2105356@students.um.ac.id
[2] Universitas Negeri Malang, Malang, Indonesia, aniknur.ft@um.ac.id
[3] Universitas Negeri Malang, Malang, Indonesia, aji.prasetya.ft@um.ac.id

Correspondence should be addressed to Anik Nur Handayani; aniknur.ft@um.ac.id

Received 10 January 2025; Accepted 15 June 2025; Published 31 July 2025



**Abstract:**

Machine learning offers promising solutions for the recognition of handwritten Javanese Nglegena script, which is crucial for preserving Indonesia's cultural heritage. This study explores the application of several supervised learning algorithms-K-Nearest Neighbors (KNN), Naïve Bayes, Decision Tree, and Random Forest-for classifying handwritten images of Nglegena Javanese script. Feature extraction is performed using a zoning technique, where each character image is divided into multiple zones (16, 25, 36, and 64) to capture local details. The extracted features are further processed using normalization methods, including Min-Max, Z-Score, and Binary normalization, to ensure uniform data distribution. The dataset, consisting of 600 images representing Javanese Nglegena characters, is split into training and testing sets using various ratios. Experimental results show that the combination of Naïve Bayes classification, 36-zone feature extraction, and Min-Max or Z-Score normalization achieves the highest accuracy of 65%. These findings demonstrate that optimizing zoning and normalization can significantly enhance the accuracy of machine learning models for Javanese script recognition. The research contributes to developing Optical Character Recognition (OCR) technology for Javanese script, supporting the digital preservation of Indonesia's historical and cultural heritage.

**Keywords:** Javanese Script Recognition, Nglegena Script, Optical Character Recognition (OCR), Machine Learning, Data Normalization.

## 1. Introduction

The usage of Latin letters in daily life further marginalizes the Javanese script, which is part of the Indonesian people's cultural legacy. Javanese script actually has a great deal of cultural and historical significance. One of the cultural legacies with significant historical significance that should be preserved is the Nglegena Javanese script [1], [2]. With the passage of time and the replacement of Latin characters throughout the colonial era, the script's use has declined. Even though Javanese script reading and writing abilities are still taught in schools today, particularly in Central and East Java, people's interest in and proficiency with this script is dwindling since it is so challenging to learn and write [3], [4], [5]. Utilizing information technology is one way to improve efforts to preserve Javanese script.

Optical Character Recognition (OCR) technology based on machine learning is being used in an attempt to preserve and advance Javanese script learning [6], [7]. One method for reading, identifying, and converting text in photographs into text format is optical character recognition, or OCR [8], [9]. Translating Javanese script into Latin script with the aid of this technology can aid in script learning and preservation [10]. The zoning method and feature extraction normalization have demonstrated encouraging outcomes in terms of improving Javanese character recognition accuracy. In order to create a more precise and effective model, this study focuses on optimizing the introduction of Nglegena Javanese script utilizing machine learning using a zoning technique and normalizing feature extraction.

In order for machine learning models to effectively capture crucial elements that can be overlooked if examined as a whole [11], the Zoning method splits characters into small zones in order to extract local data like lines and angles [12], [13]. By ensuring uniformity in the scale and orientation of retrieved characteristics, normalization techniques lessen the impact of size and lighting differences that could impair identification accuracy [14]. It is anticipated that zoning and normalizing will improve the model's precision and effectiveness in identifying Nglegena Javanese script, hence aiding in the preservation of culture through the application of contemporary technology. Consequently, this study is crucial for creating new research avenues in the domains of artificial intelligence and image processing.

This study focuses on three critical issues in recognizing Nglegena Javanese script: the relationship between zoning methods and their impact on feature extraction, identifying and applying effective normalization techniques to enhance model accuracy, and evaluating machine learning model performance after optimization through zoning and feature extraction normalization. The study aims to develop and implement zoning approaches for feature extraction specifically designed for Nglegena Javanese script, identify and apply suitable normalization techniques to improve model accuracy, and assess model performance after optimization. This research is expected to provide significant benefits by enhancing the current knowledge base in Optical Character Recognition (OCR) for Javanese script, preserving Indonesia's cultural heritage, and promoting the digitization of ancient writings. Additionally, it will facilitate progress in OCR technology for other complex scripts.

## 2. Method:

This study used data mining tools to take a quantitative approach. Deep analysis is necessary when using complicated and high-dimensional Nglegena Javanese script datasets [15], [16], [17]. In order to uncover patterns and information concealed within the data, machine learning techniques are utilized. The goal of this project is to create precise prediction models and better comprehend the features of Nglegena Javanese script by utilizing machine learning methods. The performance of different machine learning algorithms was assessed in this study using experimental research approaches because of the intricacy and uniqueness of the Nglegena Javanese script. The 600 feature columns in the dataset represent the linguistic and visual traits of the Javanese script. In order to build better Javanese script recognition technology, the goal of this research is to determine the best algorithm for classifying Nglegena Javanese script.

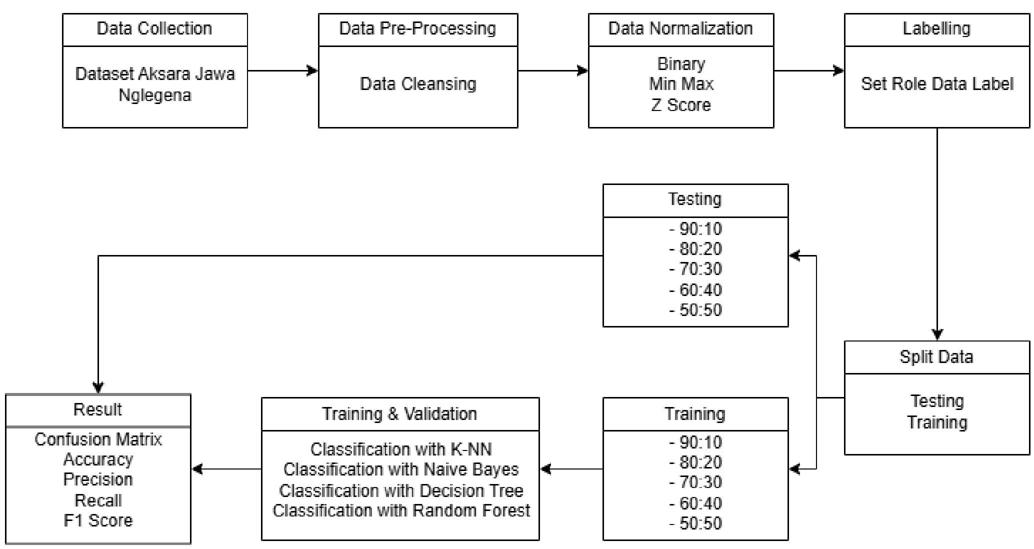


**Figure 1.** Research Design

### Data Collection

Javanese Nglegena script data collection involves scanning or taking photos of documents and handwriting, followed by character segmentation and labelling. Data variety and quality are key considerations for training an accurate OCR model, although challenges such as data limitations and variations in writing styles need to be overcome.

**Figure 2.** Example Image of Handwriting Javanese Nglegena Script

**Data Pre-processing**

Data pre-processing of Javanese Nglegena script images involves zoning as a crucial step for feature extraction, where character images are divided into 16, 25, 36, or 64 zones to capture local details. Before zoning, basic steps such as grayscaling, binarization, and noise removal are performed to clean the images. Each zone is analyzed based on pixel values, generating features like average intensity. The variation in the number of zones allows for the exploration of optimal feature detail levels, with more zones capturing finer details but increasing computational complexity. This zoning variation is tested to determine the best zone configuration for Javanese Nglegena script recognition.

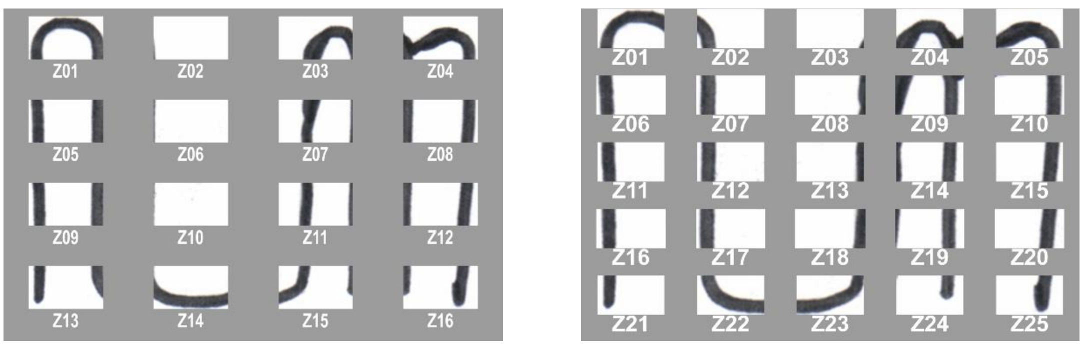


**Figure 3.** Example of 16 and 25 Zones

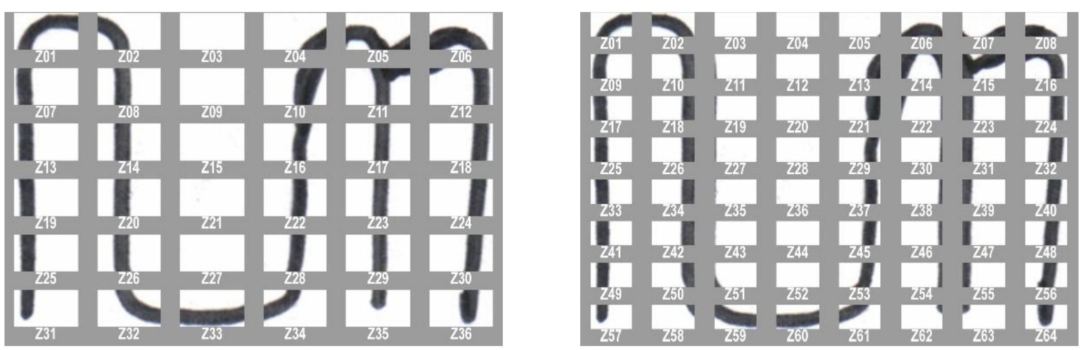


**Figure 4.** Example of 36 and 64 Zones

**Normalization of Data Binary, Z-score, and Min-Max Normalization**

Data normalization is a crucial process in data processing that aims to transform data into a more uniform scale. Binary normalization is typically not applied directly to binary variables (0/1) [18] as they are already in a suitable form for analysis. Binary variables are often used directly in models or encoded for categorical variables with more than two categories. Min-Max Normalization transforms the data range into a specific scale, usually between 0 and 1, using the given formula.

$$\boldsymbol{X'} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

This method is easy to implement and suitable for algorithms that require input within a specific range [19]. Meanwhile, Z-score Normalization (Standardization) transforms data to have a mean of 0 and a standard deviation of 1, using the given formula.

$$\boldsymbol{Z} = \frac{X - \mu}{\sigma} \quad (2)$$

This method is suitable for algorithms that rely on distance calculations and helps reduce the impact of outliers [19]. The choice of normalization method depends on the requirements of the algorithm and the characteristics of the data.

**Labelling**

Data Labeling is a crucial process in developing machine learning (ML) models that involves adding labels to raw data to provide meaningful context and categorization [20]. These labels enable ML models to understand the data and make accurate predictions. Data labelling is essential because it allows ML models to learn from the data and make informed decisions based on recognized patterns. Labelling methods include Manual Labeling, which is accurate but time-consuming; Automated Labeling, which is fast but requires quality control; and Hybrid Labeling, which combines manual and automated methods to achieve a balance between accuracy and efficiency. The labelling process involves data collection, pre-processing, data annotation, and quality testing to ensure that the assigned labels are accurate and consistent.

**Split Data**

Data Splitting is crucial in machine learning (ML) development to prevent overfitting. One common method is Random Train-Test Split, where the dataset is randomly divided into a Training Set and Test Set [21] with various ratios such as 90:10, 80:20, etc. The Training Set is used to train the model, while the Test Set is used to evaluate its performance on unseen data. This method is simple and effective, but may not be ideal for imbalanced or small datasets.

**Training and Validation**

Training and validation are two essential phases in the development of machine learning (ML). The training process entails utilizing the Training Set, enabling the model to discern patterns within the data. Validation employs the Validation Set to assess the model's performance throughout training, aiding in parameter optimization and mitigating overfitting. The Validation Set functions as an intermediary between the Training Set and Test Set, guaranteeing the model's efficacy on novel data.

**Classification**

- K-Nearest Neighbors (KNN)

  K-Nearest Neighbors (KNN) works by finding the nearest neighbors of a new data point based on distance, such as Euclidean distance [22], [23]Click or tap here to enter text.. KNN is classified based on the majority vote of the nearest neighbors [24]. The advantage of KNN is that it is simple and easy to implement, but it can be slow for large datasets.

- Naïve Bayes

  Naïve Bayes uses Bayes' theorem with the assumption of conditional independence between features given the class. Naïve Bayes calculates the probability of a class based on features and selects the class with the highest probability [25], [26]. This algorithm is fast and effective for large datasets, but the independence assumption may not be accurate in some cases.

- Decision Tree

  Decision Tree uses a tree structure to divide data based on the most relevant features. Each internal node represents an attribute, and leaf nodes represent classes or prediction values [27]. Decision Trees are easy to understand and interpret, but can suffer from overfitting if the tree is too complex [23]. Concepts of entropy and information gain are used to select the most influential attributes in dividing the data.

- Random Forest

  Random Forest is an ensemble of multiple Decision Trees built randomly to improve accuracy and reduce overfitting [28], [29]. For classification, Random Forest uses the majority vote of all trees, while for regression, the average value is used [30]. Random Forest is more stable and accurate than a single Decision Tree, but requires more computational resources [31].

**Confusion Matrix**

Confusion Matrix is an instrument employed to assess the efficacy of a classification model by illustrating the concordance between the model's predictions and the actual outcomes. It comprises True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). The Confusion Matrix is utilized to compute metrics including Accuracy, Precision, Recall, and F1-Score, offering insights into the model's strengths and limitations in categorization.

## 3. Results and Discussion

After conducting model evaluation using cross-validation, we measured the performance of the models using different test data. We quantified accuracy, precision, recall, and F1-score using a Confusion Matrix to assess how effectively the models can identify the Javanese Nglegena script. In the context of recognizing the Javanese Nglegena script, this assessment provides a deeper understanding of the effectiveness of the algorithms used.

**KNN Model Results**

**Table 1.** K-NN Model Result 36 Zones

| | | Split | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|---|
| **36 Zones** | Binary | 90:10 | 36.67% | 44.08% | 36.67% | 40.04% |
| | | 80:20 | 35.83% | 34.19% | 35.83% | 34.99% |
| | | 70:30 | 35.56% | 37.56% | 35.56% | 36.53% |
| | | 60:40 | 40.83% | 42.25% | 40.83% | 41.53% |
| | | 50:50 | 34.67% | 37.24% | 34.67% | 35.91% |
| | **Min-Max** | 90:10 | 51.67% | 59.25% | 51.67% | 55.20% |
| | | 80:20 | 52.5% | 54.91% | 52.5% | 53.68% |
| | | 70:30 | 47.78% | 49.12% | 47.78% | 48.44% |
| | | **60:40** | **59.58%** | **61.16%** | **59.58%** | **60.36%** |
| | | 50:50 | 50% | 53.5% | 50% | 51.69% |
| | Z-score | 90:10 | 40% | 45.33% | 40% | 42.5% |
| | | 80:20 | 52.5% | 56.21% | 52.5% | 54.29% |
| | | 70:30 | 48.89% | 50.44% | 48.89% | 49.65% |
| | | 60:40 | 55% | 56.65% | 55% | 55.81% |
| | | 50:50 | 47.67% | 51.22% | 47.67% | 49.38% |

| | true HA | true NA | true CA | true RA | true KA | true DA | true TA | true SA | true WA | true LA | true PA | true DHA | true JA | true YA | true NYA | true MA | true GA | true BA | true THA | true NGA | class pr |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| pred. HA | 6 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 3 | 0 | 0 | 42.86% |
| pred. NA | 1 | 7 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 58.33% |
| pred. CA | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 57.14% |
| pred. RA | 0 | 0 | 0 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 92.31% |
| pred. KA | 0 | 2 | 0 | 0 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 72.73% |
| pred. DA | 1 | 2 | 0 | 0 | 0 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 75.00% |
| pred. TA | 3 | 0 | 0 | 0 | 0 | 0 | 8 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 57.14% |
| pred. SA | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 7 | 1 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 50.00% |
| pred. WA | 0 | 1 | 2 | 0 | 1 | 0 | 0 | 1 | 4 | 0 | 1 | 4 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 25.00% |
| pred. LA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. PA | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 5 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 1 | 1 | 38.46% |
| pred. DHA | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 1 | 3 | 0 | 4 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 36.36% |
| pred. JA | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 75.00% |
| pred. YA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 12 | 1 | 0 | 0 | 1 | 0 | 0 | 85.71% |
| pred. NYA | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 0 | 0 | 0 | 0 | 0 | 80.00% |
| pred. MA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 6 | 0 | 0 | 1 | 0 | 54.55% |
| pred. GA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 1 | 90.91% |
| pred. BA | 0 | 0 | 1 | 0 | 0 | 0 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 6 | 0 | 4 | 40.00% |
| pred. THA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 3 | 66.67% |
| pred. NGA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 1 | 25.00% |
| class rec... | 50.00% | 58.33% | 33.33% | 100.00% | 66.67% | 75.00% | 66.67% | 58.33% | 33.33% | 83.33% | 41.67% | 66.67% | 50.00% | 100.00% | 66.67% | 50.00% | 83.33% | 50.00% | 50.00% | 8.33% | |

**Figure 5.** Confusion Matrix kNN with 36 Zones, Min-Max Normalization, 60:40 Split Data

In this case, the 60:40 data split ratio provided the best results compared to other ratios. With this split, the model achieved an accuracy of 59.58%, precision of 61.16%, recall of 59.58%, and an F1-score of 60.36%. This data split ratio proved optimal for training and testing the model in recognizing the Javanese Nglegena script.

Min-Max Normalization produced the best results in improving the accuracy of recognizing the Javanese Nglegena script compared to other normalization methods. Min-Max Normalization is superior because it can transform the data range into a consistent scale (usually between 0 and 1), making it easier for the K-NN model to calculate distances more accurately. Meanwhile, Binary Normalization was not effective as it did not provide significant benefits in recognizing script that requires continuous variation. Z-score Normalization also did not yield the best results in this case, possibly because it did not optimize the data distribution for the K-NN algorithm, which relies on distance calculations.

The K-Nearest Neighbors (K-NN) algorithm with K=3 yielded the best results compared to other K values. This indicates that using three nearest neighbors is the optimal strategy for making predictions in recognizing the Javanese Nglegena script. The use of 36 zones was found to be most effective in improving script recognition accuracy. These zones help divide the data into specific areas that facilitate the recognition process, thereby increasing overall accuracy. Combining these three factors Min-Max Normalization, K-NN with K=3, and 36 zones represent the optimal strategy for recognizing the Javanese Nglegena script.

### Naïve Bayes Model Results

**Table 2.** Naïve Bayes Model Result 36 Zones

| | | Split | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|---|
| **36 Zones** | Binary | 90:10 | 40% | 48.25% | 40% | 43.74% |
| | | 80:20 | 36.67% | 36.74% | 36.67% | 36.70% |
| | | 70:30 | 40% | 42.23% | 40% | 41.08% |
| | | 60:40 | 40.83% | 45.41% | 40.83% | 43% |
| | | 50:50 | 38% | 44.89% | 38% | 41.16% |
| | **Min-Max** | **90:10** | **65%** | **63.75%** | **65%** | **64.37%** |
| | | 80:20 | 50% | 52.65% | 50% | 51.29% |
| | | 70:30 | 50.56% | 52.98% | 50.56% | 51.71% |
| | | 60:40 | 49.17% | 56.41% | 49.17% | 52.54% |

| | Split | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| | 50:50 | 50.33% | 55.41% | 50.33% | 52.75% |
| Z-score | **90:10** | **65%** | **63.75%** | **65%** | **64.37%** |
| | 80:20 | 50% | 52.65% | 50% | 51.29% |
| | 70:30 | 50.56% | 52.98% | 50.56% | 51.71% |
| | 60:40 | 49.17% | 56.41% | 49.17% | 52.54% |
| | 50:50 | 50.33% | 55.41% | 50.33% | 52.75% |

| | true HA | true NA | true CA | true RA | true KA | true DA | true TA | true SA | true WA | true LA | true PA | true DHA | true JA | true YA | true NYA | true MA | true GA | true BA | true THA | true NGA | class pr |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| pred. HA | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 66.67% |
| pred. NA | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00% |
| pred. CA | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 60.00% |
| pred. RA | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. KA | 0 | 1 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 66.67% |
| pred. DA | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. TA | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. SA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. WA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 50.00% |
| pred. LA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. PA | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00% |
| pred. DHA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 50.00% |
| pred. JA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. YA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. NYA | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 75.00% |
| pred. MA | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 0 | 40.00% |
| pred. GA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 100.00% |
| pred. BA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 1 | 50.00% |
| pred. THA | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 2 | 16.67% |
| pred. NGA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00% |
| class rec... | 66.67% | 0.00% | 100.00% | 100.00% | 66.67% | 100.00% | 33.33% | 33.33% | 66.67% | 100.00% | 0.00% | 66.67% | 100.00% | 100.00% | 100.00% | 66.67% | 100.00% | 66.67% | 33.33% | 0.00% | |

**Figure 6.** Confusion Matrix Naïve Bayes with 36 Zones, Min-Max Normalization, 90:10 Split Data

| | true HA | true NA | true CA | true RA | true KA | true DA | true TA | true SA | true WA | true LA | true PA | true DHA | true JA | true YA | true NYA | true MA | true GA | true BA | true THA | true NGA | class pr |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| pred. HA | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 66.67% |
| pred. NA | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00% |
| pred. CA | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 60.00% |
| pred. RA | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. KA | 0 | 1 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 66.67% |
| pred. DA | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. TA | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. SA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. WA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 50.00% |
| pred. LA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. PA | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00% |
| pred. DHA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 50.00% |
| pred. JA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. YA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. NYA | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 75.00% |
| pred. MA | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 0 | 40.00% |
| pred. GA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 100.00% |
| pred. BA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 1 | 50.00% |
| pred. THA | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 2 | 16.67% |
| pred. NGA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00% |
| class rec... | 66.67% | 0.00% | 100.00% | 100.00% | 66.67% | 100.00% | 33.33% | 33.33% | 66.67% | 100.00% | 0.00% | 66.67% | 100.00% | 100.00% | 100.00% | 66.67% | 100.00% | 66.67% | 33.33% | 0.00% | |

**Figure 7.** Confusion Matrix Naïve Bayes with 36 Zones, Z-score Normalization, 90:10 Split Data

In this case, the 90:10 data split ratio provided the best results. With this split, the Naïve Bayes model achieved an accuracy of 65%, precision of 63.75%, recall of 65%, and an F1-score of 64.37%. This data split ratio proved optimal for training and testing the model to recognize the Javanese Nglegena script.

Both Min-Max Normalization and Z-score Normalization produced the best results and were identical in improving the accuracy of recognizing the Javanese Nglegena script. Both normalization methods yielded the same accuracy, precision, recall, and F1 score, indicating that they are equally effective in preparing data for the Naïve

Bayes model. This is because Min-Max Normalization and Z-score Normalization can transform the data scale into a more uniform one, making it easier for the Naïve Bayes model to calculate probabilities and make more accurate predictions. Meanwhile, Binary Normalization was not effectively used in this case as it did not provide significant benefits in recognizing script that requires continuous variation.

Using 36 zones proved to be the most effective in improving script recognition accuracy compared to other zones. These zones help divide the data into specific areas that facilitate the recognition process, thereby increasing overall accuracy. Combining a 90:10 split ratio, either Min-Max Normalization or Z-score Normalization, and 36 zones resulted in optimal performance for recognizing the Javanese Nglegena script using the Naïve Bayes model.

**Decision Tree Model Results**

**Table 3.** Decision Tree Model Result 64 Zones

| | | Split | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|---|
| **64 Zones** | **Binary** | 90:10 | 35% | 29,58% | 35% | 32,06% |
| | | **80:20** | **37,5%** | **42,38%** | **37,5%** | **39,79%** |
| | | 70:30 | 35% | 37,94% | 35% | 36,41% |
| | | 60:40 | 34,58% | 38,7% | 34,58% | 36,52% |
| | | 50:50 | 32,67% | 34,63% | 32,67% | 33,62% |
| | Min-Max | 90:10 | 21,67% | 17,82% | 21,67% | 19,56% |
| | | 80:20 | 20% | 12,93% | 20% | 15,71% |
| | | 70:30 | 18,89% | 15,82% | 18,89% | 17,22% |
| | | 60:40 | 20% | 14,33% | 20% | 16,7% |
| | | 50:50 | 19% | 19,94% | 19% | 19,46% |
| | Z-score | 90:10 | 21,67% | 17,82% | 21,67% | 19,56% |
| | | 80:20 | 20% | 12,93% | 20% | 15,71% |
| | | 70:30 | 18,89% | 15,82% | 18,89% | 17,22% |
| | | 60:40 | 20% | 14,33% | 20% | 16,7% |
| | | 50:50 | 19% | 19,94% | 19% | 19,46% |

| | true HA | true NA | true CA | true RA | true KA | true DA | true TA | true SA | true WA | true LA | true PA | true DHA | true JA | true YA | true NYA | true MA | true GA | true BA | true THA | true NGA | class pr |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| pred. HA | 2 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 40.00% |
| pred. NA | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. CA | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0.00% |
| pred. RA | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 66.67% |
| pred. KA | 1 | 1 | 0 | 0 | 3 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 33.33% |
| pred. DA | 0 | 1 | 0 | 0 | 0 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 60.00% |
| pred. TA | 0 | 0 | 0 | 0 | 3 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 33.33% |
| pred. SA | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 3 | 1 | 1 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 9.09% |
| pred. WA | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 33.33% |
| pred. LA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 66.67% |
| pred. PA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. DHA | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 3 | 0 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 30.00% |
| pred. JA | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 50.00% |
| pred. YA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 40.00% |
| pred. NYA | 2 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 28.57% |
| pred. MA | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 20.00% |
| pred. GA | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 4 | 0 | 0 | 0 | 57.14% |
| pred. BA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 0 | 1 | 50.00% |
| pred. THA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0.00% |
| pred. NGA | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 1 | 5 | 5 | 29.41% |
| class rec... | 33.33% | 33.33% | 0.00% | 66.67% | 50.00% | 50.00% | 50.00% | 16.67% | 16.67% | 66.67% | 16.67% | 50.00% | 33.33% | 33.33% | 33.33% | 16.67% | 66.67% | 33.33% | 0.00% | 83.33% | |

**Figure 8.** Confusion Matrix Decision Tree with 64 Zones, Binary Normalization, 80:20 Split Data

The 80:20 data split ratio yielded the most favourable outcomes in this instance. With this division, the Decision Tree model attained an accuracy of 37.5%, precision of 42.38%, recall of 37.5%, and an F1-score of 39.79%. The specified data split ratio was appropriate for training and testing the model to identify the Javanese Nglegena script.

Binary normalization yielded superior results in enhancing the accuracy of recognizing the Javanese Nglegena script relative to alternative normalization techniques. Binary normalization proved more effective due to the potential presence of pronounced binary characteristics in the data, thereby preserving critical information. Simultaneously, Min-Max Normalization and Z-score Normalization failed to produce optimal results, likely due to their ineffectiveness in enhancing the data distribution for the decision tree architecture employed by the Decision Tree model.

Implementing 64 zones has shown superior efficacy in enhancing script recognition accuracy relative to alternative configurations. These zones partition the data into distinct areas, enhancing the recognition process and improving overall accuracy. The optimal performance for recognizing the Javanese Nglegena script using the Decision Tree model was achieved by combining an 80:20 split ratio, Binary Normalization, and 64 zones.

**Random Forest Model Results**

**Table 4.** Random Forest Model Result 64 Zones

| | | Split | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|---|
| **64 Zones** | **Binary** | **90:10** | **58,33%** | **65,79%** | **58,33%** | **61,84%** |
| | | 80:20 | 55,83% | 59,62% | 55,83% | 57,66% |
| | | 70:30 | 54,44% | 55,59% | 54,44% | 55,01% |
| | | 60:40 | 54,58% | 54,52% | 54,58% | 54,55% |
| | | 50:50 | 54% | 54,49% | 54% | 54,24% |
| | Min-Max | 90:10 | 26,67% | 20,88% | 26,67% | 23,42% |
| | | 80:20 | 28,33% | 32,85% | 28,33% | 30,42% |
| | | 70:30 | 40% | 45,82% | 40% | 42,71% |
| | | 60:40 | 36,25% | 42,16% | 36,25% | 38,98% |
| | | 50:50 | 37,33% | 35,58% | 37,33% | 36,43% |
| | Z-score | 90:10 | 26,67% | 21,15% | 26,67% | 23,59% |
| | | 80:20 | 28,33% | 32,9% | 28,33% | 30,44% |
| | | 70:30 | 39,44% | 44,31% | 39,44% | 41,73% |
| | | 60:40 | 36,25% | 42,3% | 36,25% | 39,04% |
| | | 50:50 | 37,67% | 37,51% | 37,67% | 37,59% |

| | true HA | true NA | true CA | true RA | true KA | true DA | true TA | true SA | true WA | true LA | true PA | true DHA | true JA | true YA | true NYA | true MA | true GA | true BA | true THA | true NGA | class pr |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| pred. HA | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 33.33% |
| pred. NA | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. CA | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 66.67% |
| pred. RA | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. KA | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. DA | 1 | 1 | 0 | 0 | 1 | 3 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 37.50% |
| pred. TA | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 50.00% |
| pred. SA | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 20.00% |
| pred. WA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00% |
| pred. LA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. PA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 50.00% |
| pred. DHA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 50.00% |
| pred. JA | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 50.00% |
| pred. YA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. NYA | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 1 | 1 | 33.33% |
| pred. MA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 100.00% |
| pred. GA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 75.00% |
| pred. BA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 100.00% |
| pred. THA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 100.00% |
| pred. NGA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 50.00% |
| class rec... | 33.33% | 66.67% | 66.67% | 100.00% | 66.67% | 100.00% | 33.33% | 33.33% | 0.00% | 33.33% | 33.33% | 66.67% | 66.67% | 100.00% | 66.67% | 66.67% | 100.00% | 33.33% | 33.33% | 66.67% | |

**Figure 9.** Confusion Matrix Random Forest with 64 Zones, Binary Normalization, 90:10 Split Data

This instance, the optimal outcomes were achieved with a 90:10 data split ratio. The Random Forest model attained an accuracy of 58.33%, precision of 65.79%, recall of 58.33%, and an F1-score of 61.84% with this division. The specified data split ratio demonstrated optimal efficacy for training and testing the model in identifying the Javanese Nglegena script.

Binary normalization yielded superior results in enhancing the accuracy of Javanese Nglegena script recognition relative to alternative normalization techniques. Binary normalization proved more effective in this instance due to the potential presence of pronounced binary characteristics in the data, hence preserving critical information. Simultaneously, Min-Max Normalization and Z-score Normalization failed to produce optimal results, likely due to their inadequacy in enhancing the data distribution for the ensemble architecture employed by the Random Forest model.

The implementation of 64 zones has shown superior efficacy in enhancing script recognition accuracy relative to alternative configurations. These zones delineate the data into distinct areas that enhance recognition, improving overall accuracy. The integration of a 90:10 split ratio, Binary Normalization, and 64 zones yielded the best performance in recognizing the Javanese Nglegena script with the Random Forest model.

This study demonstrates strengths in evaluating various classification models, such as K-NN, Naïve Bayes, Decision Tree, and Random Forest, for recognizing Nglegena Javanese script. A comprehensive assessment using metrics like accuracy, precision, recall, and F1-score provides a deep understanding of model performance. The study also explores different configurations, including data normalization and zoning, to enhance recognition accuracy. However, the study also has some limitations. The focus on specific algorithms might limit comparisons with more advanced machine learning approaches, such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), which could be more effective in recognizing complex script patterns. Additionally, the study does not explicitly address the issue of imbalanced data, which is common in script datasets and can affect model performance.

For future research, it is recommended that the scope of algorithms used be expanded and that domain-specific features or temporal information be integrated to improve the model's ability to capture complex nuances in Nglegena Javanese script. Strategies to address imbalanced data and enhance model interpretability should also be developed to improve practical applications in script recognition. Interdisciplinary collaborations with linguistics or cognitive science experts can provide valuable insights into refining models to better align with the complexities of script recognition. This study also has the potential to significantly contribute to the development of Optical Character Recognition (OCR) technology for Javanese script, which can help preserve Indonesia's cultural heritage and improve accessibility and educational opportunities for younger generations. Furthermore, it can serve as a foundation for developing technology-driven applications that can recognize and transliterate Javanese characters into Latin script, thereby promoting the digitization of ancient writings and the preservation of historical records. Ultimately, it will catalyze further progress in OCR technology for other intricate scripts, facilitating the creation of more adaptable and generally applicable OCR systems.

## 4. Conclusion

This study evaluated the performance of several machine learning algorithms, namely K-NN, Naïve Bayes, Decision Tree, and Random Forest, in recognizing the Nglegena Javanese script. The results showed that Naïve Bayes with Min-Max and Z-Score normalization achieved the highest accuracy of 65%, followed by K-NN with an accuracy of 59.58%. Decision Tree and Random Forest performed lower, with accuracies of 37.5% and 58.33%, respectively. Different zones also affected model performance, with zone 36 yielding the best results for K-NN and Naïve Bayes, and zone 64 for Decision Tree and Random Forest. This study answered questions about which algorithm is most effective in recognizing Nglegena Javanese script and how normalization and zoning affect model performance. The results indicated that Naïve Bayes with Min-Max or Z-Score normalization is the best choice, and zone 36 provides optimal results. This study contributes to developing Optical Character Recognition (OCR) technology for Javanese script by demonstrating that appropriate zoning and normalization approaches can improve recognition accuracy. Additionally, it helps preserve Indonesia's cultural heritage by promoting the digitization of ancient writings.

For future research, it is recommended that the scope of algorithms used, such as Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN), be expanded, and domain-specific features should be integrated to enhance model performance. Furthermore, strategies to address imbalanced data should be developed to improve practical applications in script recognition.

**Acknowledgments**

The author would like to express gratitude to all parties who have supported this research. Thank you to the academic advisors who provided guidance and direction throughout the research process. Thank you also to friends and family who have provided moral support and assistance in data collection. We hope this study can make a meaningful contribution to the field of Nglegena Javanese script recognition

**References:**

[1] A. Budiman, A. Fadlil, and R. Umar, "Improving The Results of Learning Nglegena Javanese Handwriting Using Backpropagation Artificial Neural Network," *Edunesia: Jurnal Ilmiah Pendidikan*, vol. 4, no. 1, pp. 259–269, Jan. 2023, doi: 10.51276/edu.v4i1.339.

[2] F. T. Anggraeny, Y. V. Via, and R. Mumpuni, "Image preprocessing analysis in handwritten Javanese character recognition," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 2, pp. 860–867, Apr. 2023, doi: 10.11591/eei.v12i2.4172.

[3] R. Sutjiadi, T. J. Pattiasina, and P. Santoso, "The Implementation of Deep Learning Technique in Mobile Application as a Preservation and Learning Media of Javanese Letter," 2024, pp. 161–169. doi: 10.1007/978-3-031-48453-7_16.

[4] M. M. Al Haromainy, A. N. Sihananto, D. A. Prasetya, A. P. Sari, M. Alfyando, and R. Purnomo, "Classification Of Javanese Script Using Convolutional Neural Network With Data Augmentation," in *2022 IEEE 8th Information Technology International Seminar (ITIS)*, IEEE, Oct. 2022, pp. 288–291. doi: 10.1109/ITIS57155.2022.10010262.

[5] 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.

[6] W. Torres, M. G. J. van den Brand, and A. Serebrenik, "Xamã : Optical character recognition for multi-domain model management," *Innov Syst Softw Eng*, vol. 20, no. 3, pp. 225–249, Sep. 2024, doi: 10.1007/s11334-022-00453-7.

[7] Irham Ferdiansyah Katili, Mochamad Arief Soeleman, and Ricardus Anggi Pramunendar, "Character Recognition of Handwriting of Javanese Character Image using Information Gain Based on the Comparison of Classification Method," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 1, pp. 193–200, Feb. 2023, doi: 10.29207/resti.v7i1.4488.

[8] R. Najam and S. Faizullah, "Analysis of Recent Deep Learning Techniques for Arabic Handwritten-Text OCR and Post-OCR Correction," *Applied Sciences*, vol. 13, no. 13, p. 7568, Jun. 2023, doi: 10.3390/app13137568.

[9] M. Li *et al.*, "TrOCR: Transformer-Based Optical Character Recognition with Pre-trained Models," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 11, pp. 13094–13102, Jun. 2023, doi: 10.1609/aaai.v37i11.26538.

[10] G. N. Adli Kesaulya, A. Fariza, and T. Karlita, "Javanese Script Text Image Recognition Using Convolutional Neural Networks," in *2022 International Electronics Symposium (IES)*, IEEE, Aug. 2022, pp. 534–539. doi: 10.1109/IES55876.2022.9888527.

[11] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," *SN Comput Sci*, vol. 2, no. 3, p. 160, May 2021, doi: 10.1007/s42979-021-00592-x.

[12] J. Huang, I. U. Haq, C. Dai, S. Khan, S. Nazir, and M. Imtiaz, "Isolated Handwritten Pashto Character Recognition Using a K-NN Classification Tool based on Zoning and HOG Feature Extraction Techniques," *Complexity*, vol. 2021, no. 1, Jan. 2021, doi: 10.1155/2021/5558373.

[13] I. K. A. G. Wiguna and I. M. D. P. Asana, "Implementasi Zoning Dan Fitur Arah Sebagai Ekstraksi Fitur Pada Pengenalan Tulisan Tangan Aksara Bali," *Jurnal RESISTOR (Rekayasa Sistem Komputer)*, vol. 4, no. 1, pp. 85–92, Apr. 2021, doi: 10.31598/jurnalresistor.v4i1.751.

[14] M. Mazziotta and Pareto Adriano, "Everything you always wanted to know about normalization (but were afraid to ask)," *Riv Ital Econ Demogr Stat*, vol. 75, no. 1, pp. 41–52, 2021.

[15] A. N. Handayani, H. W. Herwanto, K. L. Chandrika, and K. Arai, "Recognition of Handwritten Javanese Script using Backpropagation with Zoning Feature Extraction," *Knowledge Engineering and Data Science*, vol. 4, no. 2, p. 117, Dec. 2021, doi: 10.17977/um018v4i22021p117-127.

[16] H. W. Herwanto, A. N. Handayani, K. L. Chandrika, and A. P. Wibawa, "Zoning Feature Extraction for Handwritten Javanese Character Recognition," in *2019 International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, IEEE, Oct. 2019, pp. 264–268. doi: 10.1109/ICEEIE47180.2019.8981462.

[17] H. W. Herwanto, A. N. Handayani, A. P. Wibawa, K. L. Chandrika, and K. Arai, "Comparison of Min-Max, Z-Score and Decimal Scaling Normalization for Zoning Feature Extraction on Javanese Character Recognition," in *2021 7th International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, IEEE, Oct. 2021, pp. 1–3. doi: 10.1109/ICEEIE52663.2021.9616665.

[18] I. Izonin, R. Tkachenko, N. Shakhovska, B. Ilchyshyn, and K. K. Singh, "A Two-Step Data Normalization Approach for Improving Classification Accuracy in the Medical Diagnosis Domain," *Mathematics*, vol. 10, no. 11, p. 1942, Jun. 2022, doi: 10.3390/math10111942.

[19] H. Henderi, "Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer," *IJIIS: International Journal of Informatics and Information Systems*, vol. 4, no. 1, pp. 13–20, Mar. 2021, doi: 10.47738/ijiis.v4i1.73.

[20] A. Onan, "SRL-ACO: A text augmentation framework based on semantic role labeling and ant colony optimization," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 7, p. 101611, Jul. 2023, doi: 10.1016/j.jksuci.2023.101611.

[21] Q. H. Nguyen *et al.*, "Influence of Data Splitting on Performance of Machine Learning Models in Prediction of Shear Strength of Soil," *Math Probl Eng*, vol. 2021, pp. 1–15, Feb. 2021, doi: 10.1155/2021/4832864.

[22] A. Susanto, I. U. W. Mulyono, C. A. Sari, E. H. Rachmawanto, and R. R. Ali, "Javanese Character Recognition Based on K-Nearest Neighbor and Linear Binary Pattern Features," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, Sep. 2022, doi: 10.22219/kinetik.v7i3.1491.

[23] M. Bansal, A. Goyal, and A. Choudhary, "A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning," *Decision Analytics Journal*, vol. 3, p. 100071, Jun. 2022, doi: 10.1016/j.dajour.2022.100071.

[24] M. Suyal and P. Goyal, "A Review on Analysis of K-Nearest Neighbor Classification Machine Learning Algorithms based on Supervised Learning," *International Journal of Engineering Trends and Technology*, vol. 70, no. 7, pp. 43–48, Jul. 2022, doi: 10.14445/22315381/IJETT-V70I7P205.

[25] Zena Lusi, Ayu Eka Saputri, and Tri Basuki Kurniawan, "Identifikasi Komentar Spam Pada Sosial Media," *Neptunus: Jurnal Ilmu Komputer Dan Teknologi Informasi*, vol. 2, no. 2, pp. 71–76, May 2024, doi: 10.61132/neptunus.v2i2.100.

[26] K. Wabang, Oky Dwi Nurhayati, and Farikhin, “Application of The Naïve Bayes Classifier Algorithm to Classify Community Complaints,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 5, pp. 872–876, Nov. 2022, doi: 10.29207/resti.v6i5.4498.

[27] B. Charbuty and A. Abdulazeez, “Classification Based on Decision Tree Algorithm for Machine Learning,” *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20–28, Mar. 2021, doi: 10.38094/jastt20165.

[28] M.M Jibril *et al.*, “An overview of streamflow prediction using random forest algorithm,” *GSC Advanced Research and Reviews*, vol. 13, no. 1, pp. 050–057, Oct. 2022, doi: 10.30574/gscarr.2022.13.1.0112.

[29] P. Mekha and N. Teeyasuksaet, “Image Classification of Rice Leaf Diseases Using Random Forest Algorithm,” in *2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering*, IEEE, Mar. 2021, pp. 165–169. doi: 10.1109/ECTIDAMTNCON51128.2021.9425696.

[30] 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, Jun. 2021, doi: 10.11591/eei.v10i3.3036.

[31] D. Murphy, “Using Random Forest Machine Learning Methods to Identify Spatiotemporal Patterns of Cheatgrass Invasion through Landsat Land Cover Classification in the Great Basin from 1984 - 2011,” Apr. 2019, doi: 10.13140/RG.2.2.36786.86723.