Indonesian Journal of Data and Science

Volume 6 Issue 2
ISSN 2715-9936
https://doi.org/10.56705/ijodas.v6i2.281

Research Article

Classification of L ontara Script Using K-NN Algorithm, Decision Tree, and Random Forest Based on Hu Moments and Canny Segmentation

Berlian Septiani [1,*]; Tasrif Hasanuddin [2]; Wistiani Astuti [3]
[1] Universitas Muslim Indonesia, Makassar, 90231, Indonesia, 13020210143@umi.ac.id
[2] Universitas Muslim Indonesia, Makassar, 90231, Indonesia, tasrif.hasanuddin@umi.ac.id
[3] Universitas Muslim Indonesia, Makassar, 90231, Indonesia, wistiani.astuti@umi.ac.id

Correspondence should be addressed to Thomas Edyson; tarigan@utdi.ac.id

Received 13 January 2025; Accepted 21 May 2025; Published 31 July 2025

Abstract:
Lontara script is a traditional writing system of the Bugis-Makassar people in South Sulawesi, used to write the Bugis, Makassar, and Mandar languages. This system is based on an abugida, in which each letter represents a consonant with an inherent vowel. It was once used to record history, customary law, and literature, but its use has declined due to the influence of the Latin alphabet. Today, the Lontara script is preserved through education and digitization as part of the cultural heritage of the Indonesian archipelago. In this article, the researchers attempt to use a dataset of handwritten Lontara Bugis-Makassar characters. The process begins with the collection of character datasets, which are then processed through Canny segmentation and Hu Moment feature extraction to obtain a representation of the shape that is invariant to rotation and scale. The processed data was divided into training and testing data, then classified using the K-NN, Decision Tree, and Random Forest algorithms. The results showed that the KNN algorithm with 6 neighbors achieved the highest accuracy, precision, and recall of 98%. The Decision Tree algorithm achieved an accuracy of 96.67%, precision of 96.22%, recall of 95.33%, and an F1-score of 95.98%. Meanwhile, Random Forest showed an accuracy of 96.67%, precision of 96.34%, recall of 96%, and an F1-score of 95.98%.

Keywords: Aksara Lontara; K-Nearest Neighbors (KNN); Decision Tree; Random Forest; Hu Moments; Canny Segmentation.

Dataset link: https://www.kaggle.com/code/berlianseptiani/aksara-lontara

1. Introduction

Indonesia's cultural heritage is extremely diverse, one example of which is the regional languages that have their own scripts or writing systems across the archipelago [1]. The Lontara script plays a significant role in the lives of communities, used to record history, customary law, literature, and various other important documents. However, with the introduction of the Latin alphabet due to colonialism and modernization, the use of the Lontara script has significantly declined. In recent years, various efforts have been made to preserve the Lontara script, both through education in schools and through digitization. One of the challenges in digitizing the Lontara script is the development of an automatic character recognition system that can accurately recognize the handwritten form of this script with a high degree of accuracy.

Object classification in images is generally one of the problems in computer vision, namely how a computer can mimic the human ability to understand image information and recognize objects like humans do, such as recognizing handwriting or recognizing certain patterns in an image. For humans, this is a very simple and easy task, but in reality

it is a difficult task for computers, because computers only see pixel values and pixel data, which are difficult to process [2].

Handwriting recognition is the ability of a computer to receive and interpret handwritten image input that can be understood from sources such as paper documents, photos, touch screens, and other devices. Template matching is one method that can be used in the final stage of pattern recognition, namely the classification stage [3]. The K-Nearest Neighbors (K-NN) classifier helps determine the class to which the k nearest neighbors belong for classifying an object [4]. Handwriting recognition of Lontara script is a new development, especially in South Sulawesi, while pattern recognition methods such as K-Nearest Neighbors (K-NN) have the advantage of being able to recognize handwritten numeric patterns but have the disadvantage of consuming more memory [5]. Handwriting recognition of the Lontara script is a new concept, especially in the South Sulawesi region, while pattern recognition methods such as K-Nearest Neighbors (K-NN) have the advantage of being able to recognize handwritten number patterns, but they have the disadvantage of consuming more memory [6].

In this study, the K-NN, Decision Tree, and Random Forest methods were used. The K-NN method works by calculating the distance between the test data and all the training data, then determining the class based on the majority of the K nearest neighbors[7]. K-NN is known to be simple and effective for small datasets with clear visual patterns. Meanwhile, Decision Tree builds a decision tree model by dividing the data based on features that provide the best information, and can produce a model that is easy to understand. However, this method tends to overfit on the training data if not limited. To address this, Random Forest is used, which is an ensemble algorithm that builds multiple decision trees randomly and combines their results through majority voting [8]. This approach makes Random Forest more stable and accurate, and reduces overfitting commonly found in a single decision tree. All three methods are used to evaluate the performance of Lontara character classification based on accuracy, precision, recall, and F1-score metrics.

This study uses three classification methods, namely K-NN, Decision Tree, and Random Forest to recognize Lontara characters based on images. The process begins with segmentation using Canny Edge Detection, followed by feature extraction using Hu Moment. The test results show that the K-method is 98% accurate. Decision Tree and Random Forest also yielded high results with an accuracy of 96.67% and a similar F1-score of 95.98% [9]. When compared to Altwaijy Al-Turaiki's (2020) research, which used CNN for Arabic characters, the highest accuracy only reached 88% on the children's dataset (Hijaiyah) and 97% on the adult dataset (AHCD). Meanwhile, research by A'ayunnisaa et al. (2022) used Gaussian Naïve Bayes and only achieved an accuracy of 12%. This shows that GNB is less suitable for complex data such as characters, especially if the features are not normally distributed. As for the research by Saputri et al. (2021) using Naïve Bayes only achieved a maximum testing accuracy of 13.04%. This indicates that the Naïve Bayes method is less effective in handling the complexity of Lontara script forms. In contrast, the method used in this study is more adaptive and accurate in recognizing the visual patterns of traditional scripts [10].

2. Method

This study describes the process of character classification using a machine learning approach, beginning with the collection of a dataset consisting of character images as the main object. After the data is collected, preprocessing is performed through two main stages: Canny Segmentation to extract image edges so that character shapes are easier to recognize, and Hu Moment Segmentation to represent character shapes numerically and invariance to rotation, scale, and translation [11]. The extracted data is then divided into two parts: training data and testing data. The training data is used to train the classification model, while the testing data is used to evaluate the model's performance. Three machine learning algorithms are applied in the classification stage: K-NN, Decision Tree, and Random Forest, each of which is used to recognize characters based on the extracted features, as shown in **Figure 1**.

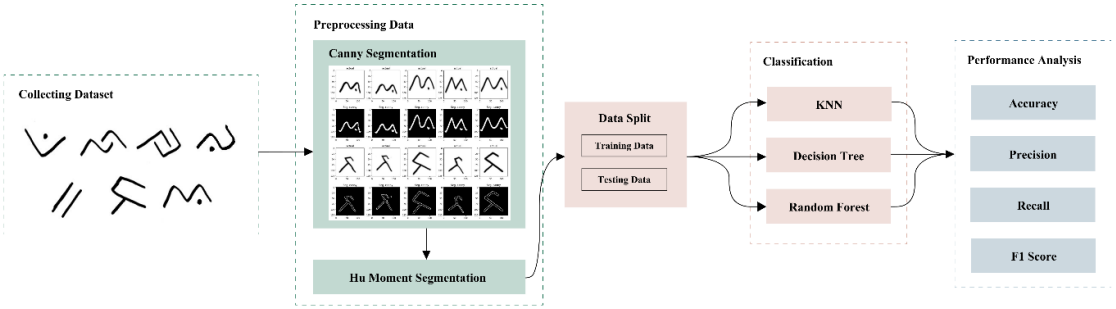

Figure 1. Research Process Sequence

The classification results were evaluated using accuracy, precision, recall, and F1-score metrics to assess the model's performance in recognizing characters as a whole. Thus, this study included stages ranging from dataset collection, data preprocessing, data division, classification, to model performance evaluation.

Dataset

The dataset used in this study consists of 11,178 Lontara character images taken from the Kaggle repository, divided into training data and test data. The training data consists of 7,452 images and the test data consists of 3,726 images, each measuring 128×128 pixels. Each image represents a single Lontara character belonging to one of 23 classes, such as 'a', 'ba', 'ca', 'da', 'ga', 'ha', 'ja', 'ka', and so on up to 'ya'. These classes represent letters in the Lontara script that have distinctive visual characteristics. This dataset is used to train and test classification models in recognizing characters based on the visual features of each character. Data preprocessing is a crucial stage in the classification process because it involves a series of steps to prepare raw data or data collected from various sources for analysis or modelling [12]. **Figure 2** shows the data distribution after visualization using a pie chart.

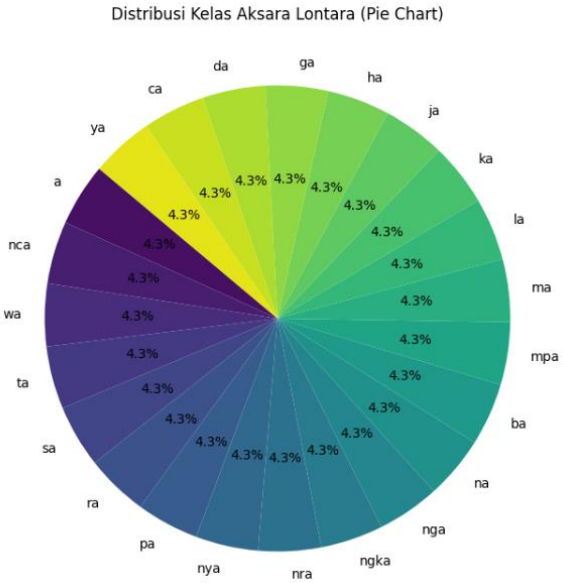

Figure 2. Visualization of the Class Aksara Lontara

K-Nearest Neighbor

KNN is a method for classifying objects based on learning data from the object's nearest neighbors. The proximity or distance of neighbors is usually calculated based on Euclidean distance [13]. As a system capable of searching data, a classification system is required [14]. Supervised learning algorithms are used in KNN, which classify new query results based on the majority category of its nearest neighbors [15]. This space is divided into

several parts based on the classification of the training samples. A point in this space is labeled with a specific class if that class is most frequently found among the nearest neighbors of that point. The proximity or distance of neighbors is typically calculated using Euclidean distance [16]. The general formula used in the KNN method is as follows.

$$d(x,y) = i = 1\sum n(xi - yi)2 \quad (1)$$

Random Forest

Random Forest is one of the ensemble learning algorithms that works by combining a number of decision trees using the Bootstrap Aggregating (bagging) approach [17]. This method builds many trees randomly from subsets of data and features, then combines the prediction results of each tree through a voting mechanism for classification or averaging for regression. This approach aims to improve model accuracy and reduce the risk of overfitting that often occurs in single decision trees. Mathematically, the final prediction for classification is formulated as.

$$y\hat{} = mode\{h1(x), h2(x), \dots, hB(x)\} \quad (2)$$

Decision Tree

understand and interpret. This model works by constructing a tree structure in which each node tests specific features, and each branch represents the results of those tests until reaching the leaf node that determines the class label [18] In this study, the Decision Tree algorithm was implemented using the Scikit-learn library and evaluated using the 5-fold cross-validation technique. The evaluation was conducted using four main metrics: accuracy, precision, recall, and F1-score. The formulas for calculating each metric are as follows:

$$\text{Accuracy } = (TP + TN)/(TP + TN + FP + FN) \quad (3)$$

$$\text{Precision } = TP/(TP + FP) \quad (4)$$

$$\text{Recal } = TP/(TP + FN) \quad (5)$$

$$\text{F1} - \text{Score } = (2\ x\ Precision\ x\ Recall)/(Precision + Recall) \quad (6)$$

The test results show that the Decision Tree model in Lontara character classification produces an average accuracy of 96.67%, precision of 96.22%, recall of 95.33%, and F1-score of 95.99%, which indicates excellent performance in recognizing visual patterns of characters.

Feature Extraction

Feature extraction converts pixel data into representations of shape, motion, color, and texture. An important step in building these pattern classifications is to obtain information about the characteristics of each class so that it can be used for the next step [19]. Feature extraction in computer vision and machine learning refers to a set of key data that is measured and constructed based on anticipated features so that it is not excessive and informative. Image recognition involves the feature extraction process [20]. The recognition rate is directly influenced by the reliability of the feature vector. M.L.K. Hu proposed the Hu invariant moment theory in 1962. The Hu moment feature extraction method is used to generate seven features that can identify objects. The objects extracted can include location, area, direction, and others. Objective moments with translation invariance, rotation, scale, and scaling are based on the theory of region moment invariance, which can describe the shape of a spatial region. Extracting the shape aspects of an image, namely [21].

$$h_1 = \eta_{\{20\}} + \eta_{\{02\}} \quad (7)$$

$$h_2 = \left(\eta_{\{20\}} - \eta_{\{02\}}\right)^2 + 4\eta_{\{11\}}^2 \quad (8)$$

$$h_3 = \left(\eta_{\{30\}} - 3\eta_{\{12\}}\right)^2 + \left(3\eta_{\{21\}} - \eta_{\{03\}}\right)^2 \quad (9)$$

$$h_4 = \left(\eta_{\{30\}} + \eta_{\{12\}}\right)^2 + \left(\eta_{\{21\}} + \eta_{\{03\}}\right)^2 \quad (10)$$

$$h\_5 \& = (\eta\_\{30\} - 3\eta\_\{12\})(\eta\_\{30\} + \eta\_\{12\}) \vdash[(\eta_{\{30\}} + \eta_{\{12\}})^2 - 3(\eta_{\{21\}} + \eta_{\{03\}}) \ ] + (3\eta_{\{21\}} - \eta_{\{03\}})(\eta_{\{21\}} + \eta_{\{03\}})\left[3(\eta_{\{30\}} + \eta_{\{12\}})^2 - (\eta_{\{21\}} + \eta_{\{03\}})^2\right] \quad (11)$$

$$h_6 = (\eta_{\{20\}} - \eta_{\{02\}})\left[(\eta_{\{30\}} + \eta_{\{12\}})^2 - (\eta_{\{21\}} + \eta_{\{03\}})^2\right] + 4\eta_{\{11\}(\eta_{\{30\}} + \eta_{\{12\}})(\eta_{\{21\}} + \eta_{\{03\}})} \quad (12)$$

$$h_7 = (3\eta_{\{21\}} - \eta_{\{03\}})(\eta_{\{30\}} + \eta_{\{12\}})\left[(\eta_{\{30\}} + \eta_{\{12\}})^2 - 3(\eta_{\{21\}} + \eta_{\{03\}})^2\right] - (\eta_{\{30\}} - 3\eta_{\{12\}})(\eta_{\{21\}} + \eta_{\{03\}})\left[3(\eta_{\{30\}} + \eta_{\{12\}})^2 - (\eta_{\{21\}} + \eta_{\{03\}})^2\right] \quad (13)$$

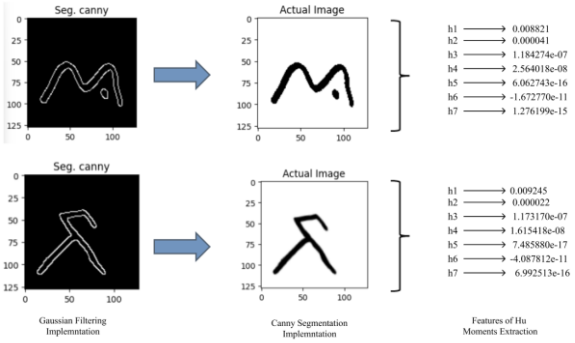

Figure 3. Feature Extraction Process

In this study, the Lontara script image dataset was processed at the feature extraction stage using Hu Moments after undergoing image segmentation using Gaussian Filtering and Canny segmentation. This segmentation process aimed to extract the edge shapes of each Lontara script character. After that, feature extraction was performed using Hu Moments, which produced seven numerical values representing the shape characteristics of each character [22].

3. Result and Discussion

Results

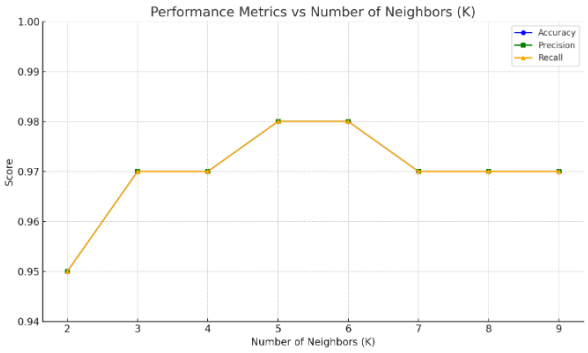

Figure 4. Scatter Plot in K-NN

The K-NN algorithm was evaluated on the iris dataset using five-fold cross-validation with a total of 150 samples. In each iteration, 120 data points were used for training and 30 data points for testing. For eight configurations of K values (K = 2 to K = 9), the total data used in training reached 4,800 samples and the testing data reached 1,200 samples. The evaluation results showed that K = 5 and K = 6 provided the best performance with accuracy, precision, and recall of 0.98, while K = 2 provided the lowest performance of 0.95. The average model performance was around 0.97, indicating stable classification performance for K values greater than 3.

Figure 5. Scatter Plot Random Forest

The Random Forest algorithm was evaluated on a dataset consisting of a total of 11,178 samples, divided into 7,452 training data and 3,726 test data, representing 25 letter classes in 128x128 pixel RGB image format. The evaluation process used 5-fold cross-validation to measure the overall performance of the model [23]. The evaluation results showed that the Random Forest algorithm performed very well with an average accuracy of 0.96, precision of 0.963, recall of 0.96, and F1-Score of 0.966. These values indicate that the model has a high level of consistency and generalization ability toward the test data. The F1-Score, which is higher than the other metrics, indicates an optimal balance between precision and recall.

Figure 6. Scatter Plot Decision Tree

The Decision Tree algorithm was evaluated on a dataset consisting of 11,178 samples, divided into 7,452 training data and 3,726 testing data, with 25 letter classes represented in 128x128 pixel RGB color images. The evaluation process using five-fold cross-validation showed that the Decision Tree model achieved an average accuracy of 0.9666, precision of 0.9623, recall of 0.9533, and an F1-Score of 0.9599. These results indicate that the Decision Tree model performs very well and is relatively consistent in handling multi-class classification tasks [24].

Discussion

Table 1. Dataset Performance Decision Tree

$\sum installment - installments$	**Decision Tree**
Balanced Accuracy	0,96
Accuracy	0.96
Precision Weighted	0.96
Recal Weighted	0.95
F1-Score Weighted	0.95

The model in **Table 1** shows excellent performance with Balanced Accuracy and Accuracy values of 0.96, indicating the model's ability to perform balanced and accurate classification across all data classes. The Precision Weighted value of 0.96 shows that the model has a high level of accuracy in predicting each class proportionally.

Additionally, the Weighted Recall value of 0.95 indicates that the model is able to recognize most samples from each class well. The Weighted F1-Score of 0.95 shows an optimal balance between precision and recall [25].

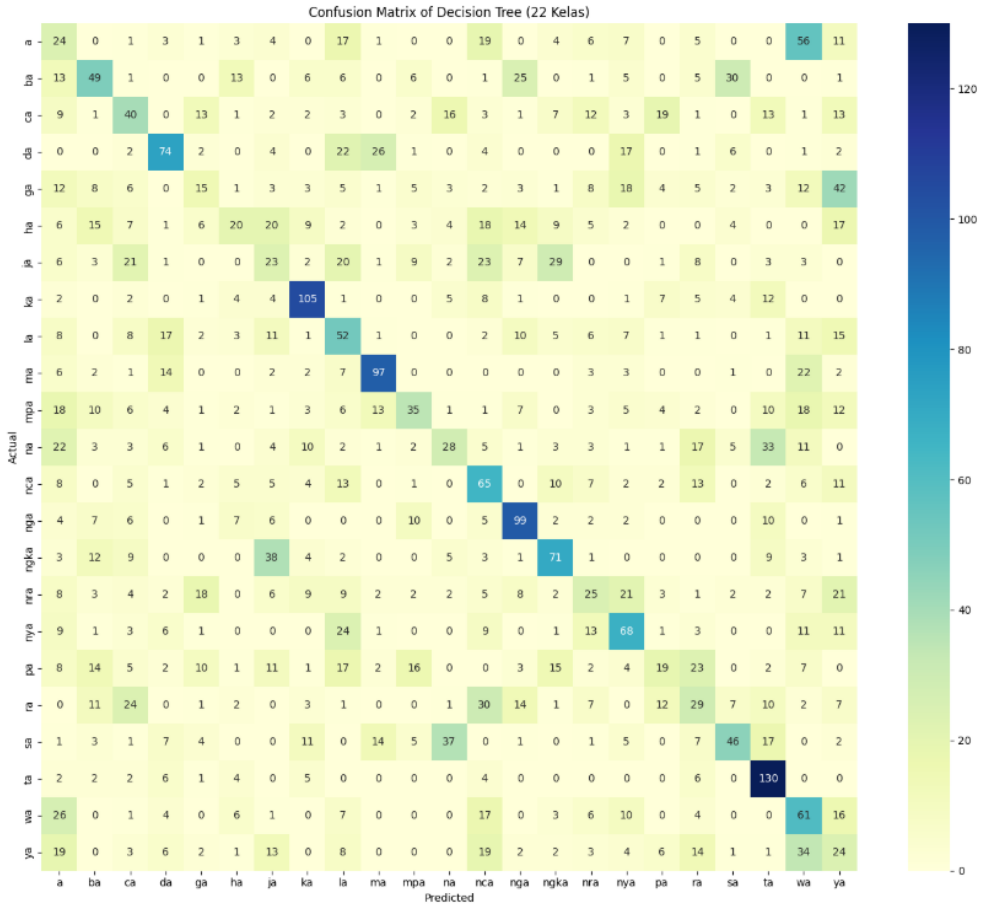

Figure 4. Confusion Matrix in Decision Tree Method

Table 2. Dataset Performance Random Forest

$\sum installment - installments$	**Random Forest**
Balanced Accuracy	0,95
Accuracy	0.96
Precision Weighted	0.96
Recal Weighted	0.96
F1-Score Weighted	0.96

The model in **Table 2** shows excellent performance with Balanced Accuracy and Accuracy values of 0.95 each, which reflects the model's ability to perform consistent and accurate classification, even in an unbalanced class distribution. Precision Weighted and Recall Weighted are 0.96 each, which indicates that the model has a high level of accuracy in classifying data and is able to recognize the majority of instances from each class well. The F1-Score Weighted value of 0.96 further reinforces that the model has an optimal balance between precision and recall [26].

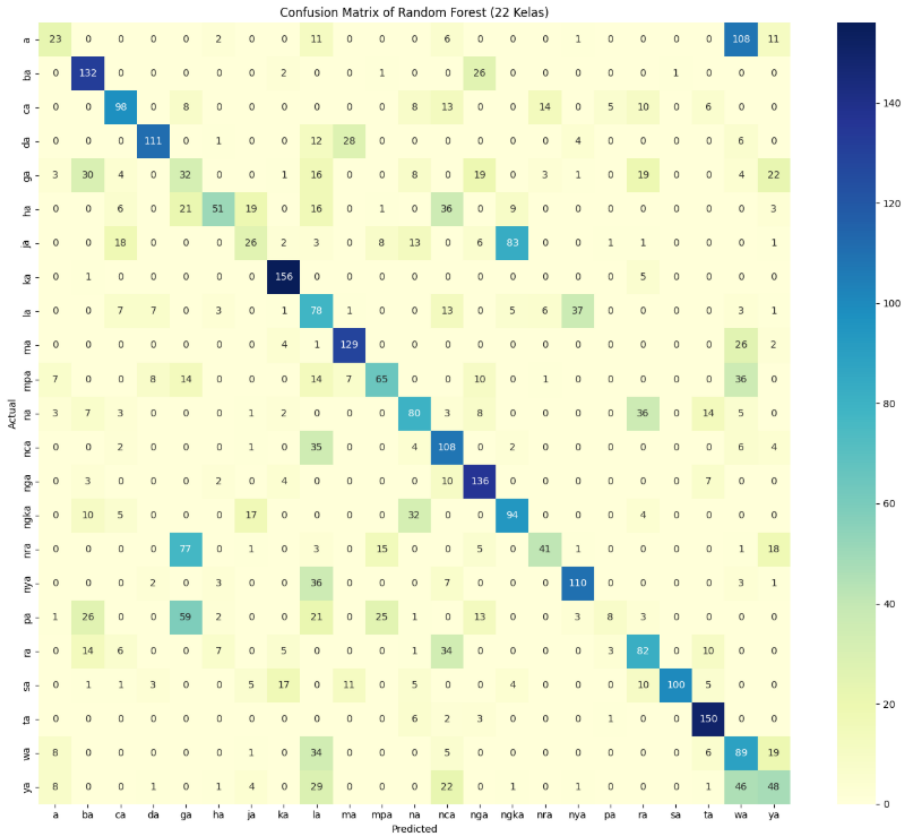

Figure 5. Confusion Matrix in Random Forest Method

Table 3. Dataset Performance K-NN

K-NN	Accuracy	Precision	Recall
2	0,95	0,95	0,95
…	…	…	…
6	0,98	0,98	0,98
…	…	…	…
9	0,97	0,97	0,97

The results show that higher k values generally produce more stable and higher performance. At k = 2, accuracy, precision, and recall are each 0.95. As k increases from 3 to 9, accuracy remains consistent within the range of 0.97 to 0.98, with precision and recall also increasing to reach 0.98 at k = 6 and k = 7. The highest performance is achieved at k = 6 and k = 7, where all evaluation metrics are at 0.98, indicating that the model can classify data very well and balanced[27]. Based on these results, it can be concluded that selecting the appropriate k value significantly impacts model performance, and k = 6 or k = 7 provides the best performance.or k = 7 provides the best performance [28].

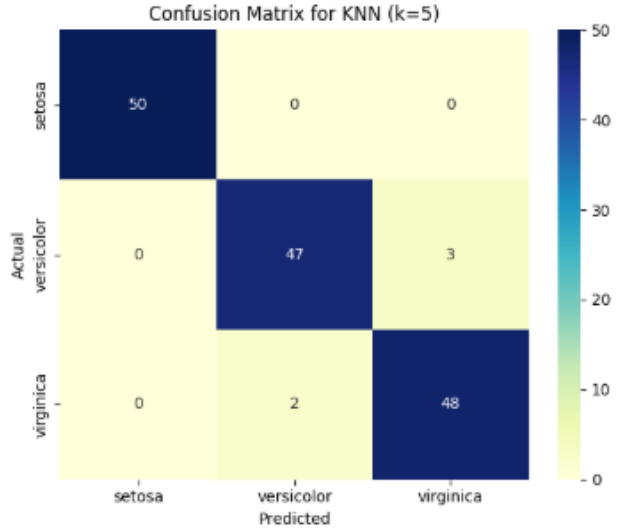

Figure 6. Confusion Matrix in K-NN

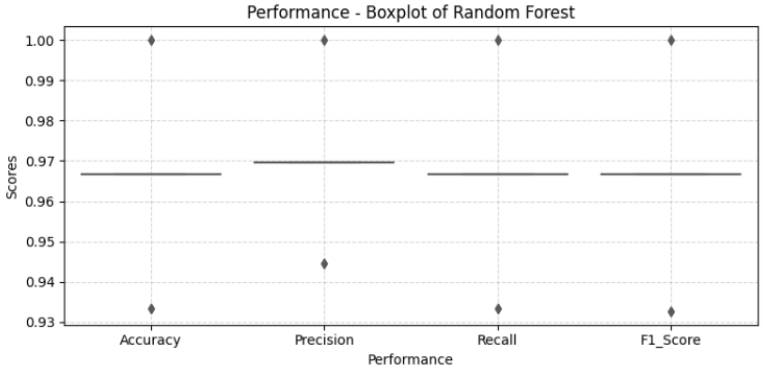

Figure 7. Comparison of Each Cross-Validation in the Random Forest Method

In **Figure 7**, the performance results were obtained from the Random Forest method, where the highest accuracy in the fifth cross-validation was 1.00, the highest precision was 1.00, the highest recall was 1.00, and the highest F1-score also reached 1.00. The average performance results from the entire cross-validation process show an accuracy value of 0.968, precision of 0.970, recall of 0.967, and F1-score of 0.967.

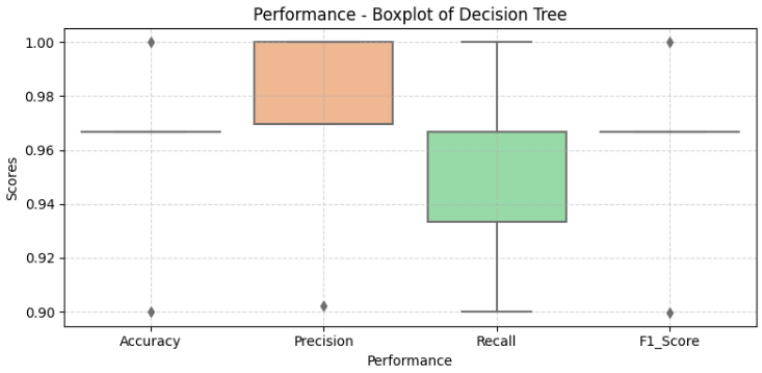

Figure 8. Comparison of Each Cross-Validation in the Decision Tree Method

In **Figure 8**, the performance results are obtained from the Decision Tree method, where the highest accuracy in the 5th cross validation is 1.00, the highest precision is 1.00, the highest recall is 1.00, and the highest F1-score also

reaches 1.00. The average performance results of the entire cross-validation process showed an accuracy value of 0.9667, precision of 0.9623, recall of 0.9533, and F1-score of 0.9599.

4. Conclusion

This research proves that classification methods such as Decision Tree, Random Forest, and K-NN can be used effectively in the process of recognizing and classifying Lontara script[21]. Based on the evaluation results, the Random Forest method provides the best performance with accuracy, precision, recall, and F1-score of 0.96. The Decision Tree method also showed excellent performance with an accuracy of 0.96 and F1-score of 0.95. Meanwhile, the K-NN algorithm showed stable performance, especially at values of k = 6 and k = 7, with accuracy, precision, and recall reaching 0.98. These results show that the selection of the right classification method is very influential in improving the accuracy of Lontara script recognition [29]. This research contributes to the preservation and digitization of regional scripts with a machine learning-based technology approach. In the future, this research can be further developed by adding a variety of features or deep learning methods to achieve higher accuracy in traditional script recognition systems [30].

Acknowledgments

Praise and gratitude are due to God Almighty for all His mercy and grace so that this article can be completed properly. The author expresses his deepest gratitude to his beloved parents for their endless prayers, moral support, and encouragement. The author would like to thank the faculty of computer science, especially thanks to the laboratory of the faculty of computer science for the support of facilities and funding that has been provided during this research process. Thanks also go to the first and second supervisors for their guidance, input, and cooperation which greatly helped in the completion of this article. Last but not least, the author appreciates the participation of all those who have contributed, both directly and indirectly, in supporting the success of this research

References:

[1] N. Altwaijry and I. Al-Turaiki, "Arabic handwriting recognition system using convolutional neural network," *Neural Comput. Appl.*, vol. 33, no. 7, pp. 2249–2261, 2021, doi: 10.1007/s00521-020-05070-8.

[2] W. Astuti, D. P. I. Putri, A. P. Wibawa, Y. Salim, Purnawansyah, and A. Ghosh, "Predicting Frequently Asked Questions (FAQs) on the COVID-19 Chatbot using the DIET Classifier," *3rd 2021 East Indones. Conf. Comput. Inf. Technol. EIConCIT 2021*, no. December, pp. 25–29, 2021, doi: 10.1109/EIConCIT50028.2021.9431913.

[3] S. K. Jadwaa, "X-Ray Lung Image Classification Using a Canny Edge Detector," *J. Electr. Comput. Eng.*, vol. 2022, 2022, doi: 10.1155/2022/3081584.

[4] D. M. Elbourhamy, A. H. Najmi, and A. I. M. Elfeky, "Students' performance in interactive environments: an intelligent model," *PeerJ Comput. Sci.*, vol. 9, p. e1348, May 2023, doi: 10.7717/peerj-cs.1348.

[5] Herman, H. Nasir, M. N. Megat Mohamed Noor, T. Hasanuddin, D. Indra, and H. B. Lumentut, "Exploration of CNN Parameters to Measure Performance of LeNet-5 Architecture in Toraja Carving Classification," in *2024 IEEE 8th International Conference on Signal and Image Processing Applications (ICSIPA)*, 2024, pp. 1–6. doi: 10.1109/ICSIPA62061.2024.10686353.

[6] H. Azis, Nirmala, L. Syafie, Herman, F. Fattah, and T. Hasanuddin, "Unveiling Algorithm Classification Excellence: Exploring Calendula and Coreopsis Flower Datasets with Varied Segmentation Techniques," in *2024 18th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, Jan. 2024, pp. 1–7. doi: 10.1109/IMCOM60618.2024.10418246.

[7] M. D. Adane, J. K. Deku, and E. K. Asare, "Performance Analysis of Machine Learning Algorithms in Prediction of Student Academic Performance," *J. Adv. Math. Comput. Sci.*, vol. 38, no. 5, pp. 74–86, 2023, doi: 10.9734/jamcs/2023/v38i51762.

[8] E. Priyanto, E. I. Sela, L. A. Latumakulita, and N. Islam, "Decision Tree C4.5 Performance Improvement using Synthetic Minority Oversampling Technique (SMOTE) and K-Nearest Neighbor for Debtor Eligibility Evaluation," *Ilk. J. Ilm.*, vol. 15, no. 2, pp. 373–381, 2023, [Online]. Available: https://jurnal.fikom.umi.ac.id/index.php/ILKOM/article/view/1676

[9] X. Ji, H. Guo, and M. Hu, "Features Extraction and Classification of Wood Defect Based on Hu Invariant

Moment and Wavelet Moment and BP Neural Network," in *Proceedings of the 12th International Symposium on Visual Information Communication and Interaction*, in VINCI '19. New York, NY, USA: Association for Computing Machinery, 2019. doi: 10.1145/3356422.3356459.

[10] Al Danny Rian Wibisono, Syahrul Hidayat, Humam Maulana Tsubasanofa Ramadhan, and Eva Yulia Puspaningrum, "Comparison of K-Nearest Neighbor and Decision Tree Methods using Principal Component Analysis Technique in Heart Disease Classification," *Indones. J. Data Sci.*, vol. 4, no. 2, pp. 90–100, 2023, doi: 10.56705/ijodas.v4i2.70.

[11] J. Basavaiah and A. Arlene Anthony, "Tomato Leaf Disease Classification using Multiple Feature Extraction Techniques," *Wirel. Pers. Commun.*, vol. 115, no. 1, pp. 633–651, 2020, doi: 10.1007/s11277-020-07590-x.

[12] M. Shanbehzadeh, H. Kazemi-Arpanahi, M. Bolbolian Ghalibaf, and A. Orooji, "Performance evaluation of machine learning for breast cancer diagnosis: A case study," *Informatics Med. Unlocked*, vol. 31, no. March, p. 101009, 2022, doi: 10.1016/j.imu.2022.101009.

[13] C. D. Suhendra, E. Najwaini, E. Maria, and E. Faizal, "A Machine Learning Perspective on Daisy and Dandelion Classification: Gaussian Naive Bayes with Sobel," *Indones. J. Data Sci.*, vol. 4, no. 3, pp. 151–159, 2023, doi: 10.56705/ijodas.v4i3.112.

[14] A. Sharma and P. K. Mishra, "Performance analysis of machine learning based optimized feature selection approaches for breast cancer diagnosis," *Int. J. Inf. Technol.*, vol. 14, no. 4, pp. 1949–1960, 2022, doi: 10.1007/s41870-021-00671-5.

[15] M. R. Amiarrahman and T. Handhika, "Analisis dan Implementasi Algoritma Klasifikasi Random Forest Dalam Pengenalan Bahasa Isyarat Indonesia (BISINDO)," *Semin. Nas. Inov. Teknol.*, pp. 83–88, 2018.

[16] T. P. Prathibha and P. M. Arabi, "Computer Aided Classification of Lung Cancer, Ground Glass Lung and Pulmonary Fibrosis Using Machine Learning and KNN Classifier," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 7, pp. 1145–1151, 2024, doi: 10.14569/IJACSA.2024.01507111.

[17] S. Anraeni, E. R. Melani, and H. Herman, "Ripeness Identification of Chayote Fruits using HSI and LBP Feature Extraction with KNN Classification," *Ilk. J. Ilm.*, vol. 14, no. 2, pp. 150–159, 2022, doi: 10.33096/ilkom.v14i2.1153.150-159.

[18] V. Çetin and O. Yıldız, "A comprehensive review on data preprocessing techniques in data analysis," *Pamukkale Univ. J. Eng. Sci.*, vol. 28, no. 2, pp. 299–312, 2022, doi: 10.5505/pajes.2021.62687.

[19] R. J. Samworth, "Optimal weighted nearest neighbour classifiers," *Ann. Stat.*, vol. 40, no. 5, pp. 2733–2763, 2012, doi: 10.1214/12-AOS1049.

[20] E. Alpaydin, "Voting over Multiple Condensed Nearest Neighbors," *Artif. Intell. Rev.*, vol. 11, no. 1–5, pp. 115–132, 1997, doi: 10.1007/978-94-017-2053-3_4.

[21] I. P. Adi Pratama, E. S. Jullev Atmadji, D. A. Purnamasar, and E. Faizal, "Evaluating the Performance of Voting Classifier in Multiclass Classification of Dry Bean Varieties," *Indones. J. Data Sci.*, vol. 5, no. 1, pp. 23–29, 2024, doi: 10.56705/ijodas.v5i1.124.

[22] Purnawansyah, N. A. Supriadi, A. R. Manga, R. Adawiyah, Harlinda, and T. Hasanuddin, "Application of Ensemble Machine Learning for DDoS Detection in Complex Network Environments," in *2025 19th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 2025, pp. 1–7. doi: 10.1109/IMCOM64595.2025.10857516.

[23] A. Rachman, "Utilization of Deep Learning YOLO V9 for Identification and Classification of Toraja Buffalo Breeds," vol. 17, no. 1, pp. 12–19, 2025.

[24] Harlinda, A. Rendi, H. Azis, D. Indra, L. N. Hayati, and N. Kurniati, "Classification of Cia-cia Letters Using MobileNetV2 and CNN Methods," *Proc. 2025 19th Int. Conf. Ubiquitous Inf. Manag. Commun. IMCOM 2025*, no. January, pp. 1–6, 2025, doi: 10.1109/IMCOM64595.2025.10857478.

[25] Purnawansyah, A. P. Wibawa, T. Widyaningtyas, H. Darwis, and H. Azis, "An In-depth Exploration of Supervised and Semi-Supervised Learning on Face Recognition," vol. 1, no. 11, pp. 1–32, 2016.

[26] N. Rismayanti, A. Naswin, U. Zaky, M. Zakariyah, and D. A. Purnamasari, "Evaluating Thresholding-Based Segmentation and Humoment Feature Extraction in Acute Lymphoblastic Leukemia Classification using

Gaussian Naive Bayes," *Int. J. Artif. Intell. Med. Issues*, vol. 1, no. 2, pp. 74–83, 2023, doi: 10.56705/ijaimi.v1i2.99.

[27] Nurul Rismayanti and Aulia Putri Utami, "Improving Multi-Class Classification on 5-Celebrity-Faces Dataset using Ensemble Classification Methods," *Indones. J. Data Sci.*, vol. 4, no. 2, pp. 124–133, 2023, doi: 10.56705/ijodas.v4i2.78.

[28] W. Astuti, A. P. Wibawa, H. Haviluddin, and H. Darwis, "DIET Classifier Model Analysis for Words Prediction in Academic Chatbot," *Ilk. J. Ilm.*, vol. 16, no. 1, pp. 59–67, 2024, doi: 10.33096/ilkom.v16i1.1598.59-67.

[29] A. R. Papua, T. Hasanuddin, and M. Hasnawi, "Decision Support System for Ranking Active Waste Bank in Makassar City Using TOPSIS and VIKOR Methods," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 13, no. 2, pp. 267–273, 2024, doi: 10.32736/sisfokom.v13i2.2158.

[30] R. Satra, I. A. Dahlan, H. Darwis, Purnawansyah, S. Mujaddid, and F. Fattah, "A Comparison of Accuracy: KNN, TabNet, and Wide & Deep Learning for DDoS Attack Detection in Software Defined Network," in *2025 19th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 2025, pp. 1–8. doi: 10.1109/IMCOM64595.2025.10857511.