A Baseline Evaluation of OCR Segmentation and Classification Methods for Printed Javanese Script

Anastasia Rita Widiarti
Informatics Department, Sanata Dharma University, Yogyakarta, Indonesia
rita_widiarti@usd.ac.id (corresponding author)

Susilawati Endah Peni Adji
Department of Indonesian Literature, Sanata Dharma University, Yogyakarta, Indonesia
peni@usd.ac.id

Fransisca Tjandrasih Adji
Department of Indonesian Literature, Sanata Dharma University, Yogyakarta, Indonesia
nuning@usd.ac.id

Gerardus Kristha Bayu Indraputra
Informatics Department, Sanata Dharma University, Yogyakarta, Indonesia
g.kristhabayu1.0@gmail.com

Yulius Agung Trisnanto
Informatics Department, Sanata Dharma University, Yogyakarta, Indonesia
yuliusat079@gmail.com

FX Bima Yudha Pratama
Informatics Department, Sanata Dharma University, Yogyakarta, Indonesia
fx.bima.yudha.pratama@gmail.com

Received: 14 October 2025 | Revised: 8 November 2025 | Accepted: 24 November 2025

 DOI: https://doi.org/10.48084/etasr.15502

ABSTRACT

Optical Character Recognition (OCR) for Javanese script remains challenging due to its complex glyph structure and overlapping components. In this context, this study presents a pilot investigation into the segmentation and transliteration performance of printed Javanese text using a single-page dataset. The workflow begins with preprocessing, followed by segmentation and script-level classification using the k-Nearest Neighbor (k-NN) algorithm. Experiments were conducted with 91 and 281 scripts, utilizing 100 and 500 samples per script to evaluate system performance. As a result, the segmentation stage achieved an average accuracy of 63.5%, with some lines reaching above 80%. Moving to transliteration, accuracy comparisons were performed between segmentation output, model predictions, and ground truth, yielding average accuracies of 61.4%, 72.3%, and 39.2%, respectively. Further experiments using 3-NN and 11-NN configurations demonstrated that increasing training samples and script diversity improved recognition accuracy, achieving up to 68.75% on certain lines. This research provides an initial benchmark dataset and a systematic evaluation framework, establishing a baseline that bridges the gap between handwritten and printed OCR research. The findings offer empirical insights for developing robust OCR systems to support the digital preservation of Indonesia's written cultural heritage.

Keywords-Optical Character Recognition (OCR); k-Nearest Neighbor (k-NN); printed Javanese text; segmentation

I. INTRODUCTION

Regional languages constitute a vital component of Indonesia's cultural diversity. According to Grimes, as cited in [1], Indonesia is home to 672 regional languages distributed throughout the archipelago. Several of these languages are extinct, critically endangered, or at risk of extinction, highlighting the urgency of regional language preservation. Within this context, Javanese stands out as a regional language with widespread usage and a rich literary heritage.

The Javanese script, also called Hanacaraka, Carakan, or Dentawyanjana, started in the Hindu-Buddhist period of the region. It comes from the Brahmi script and has changed over time through local forms such as Pallava and Kawi. The Javanese script is an abugida, a writing system in which each sign usually shows a consonant with a basic vowel sound, which can be modified with extra marks. This script has been used to write religious texts, legal rules, epic poems, historical stories, and teaching materials. Saving and sharing scripts, including by writing them down, teaching others, and using tools like Optical Character Recognition (OCR)—which converts images of text into computer text—is important for protecting national culture. These efforts help keep Javanese identity, language, and knowledge alive.

The Javanese script exhibits a complex morphographic structure, characterized by several interacting groups of basic forms. Morphographic structure denotes the manner in which written symbols visually encode linguistic units, such as syllables or sounds. Typically, this system comprises 20 basic forms of nglegena script (basic letters) and their pasangan, which combine to form closed consonants. It also encompasses murda script (special capital-form letters employed for proper nouns or honorifics) and sandhangan (diacritics or modifiers applied to vowels and consonants). Collectively, these elements result in a syllabic writing system, wherein each script signifies a syllable. This inherent complexity poses considerable challenges for OCR development. Difficulties are particularly notable during segmentation and normalization, as morphological variation and minimal spacing between graphemes or units, such as letters or symbols, often lead to ambiguity in delineating written units [2].

Research by authors in [3] indicates that the combination of 20 nglegena base letters, their paired forms, and sandhangan produces more than 11,000 distinct syllabic units. The breadth of these combinations underscores the need for OCR systems designed for Javanese script to accommodate extensive feature spaces and considerable visual variation. Consequently, preliminary investigations, such as the present study, are necessary for evaluating segmentation and classification approaches on limited datasets before applying them to large-scale manuscript collections. Structural complexity further necessitates adaptive methodology in feature extraction and pattern classification to achieve optimal OCR accuracy for Javanese script [2, 3].

The development of OCR technology has progressed for Latin scripts; however, significant obstacles remain for traditional scripts, such as Javanese. The primary challenge involves segmenting and classifying overlapping syllabic forms, which complicates accurate separation and subsequent recognition. Therefore, preliminary studies on controlled datasets are necessary to examine segmentation and classification methods prior to expanding to broader manuscript collections.

Segmentation is the most crucial step in the OCR process for Javanese manuscripts. It determines the system's ability to accurately recognize syllable units. Segmentation involves separating lines, syllables, or scripts in text images so that each can be individually recognized. The primary challenge before classification is extracting each syllable form from the manuscript. These manuscripts are typically compact and contain many additional elements, such as pasangan and sandhangan.

In printed Javanese script, line and syllable spacing enable segmentation based on profile projection. This approach is relatively effective because the symbols' shapes and spacing are more uniform than in handwritten manuscripts [2, 4-7]. Nevertheless, the segmentation challenges posed by Javanese script differ from those encountered in Latin scripts. Due to its syllabic structure, a single unit may encompass multiple vertically overlapping components. Examples include pasangan beneath syllables or sandhangan above them. These features can obscure syllable boundaries and cause over- or under-segmentation [3, 8]. Robust segmentation, therefore, requires both spatial analysis and a comprehensive understanding of Javanese script morphology. Systems must discern characteristic patterns, including the positions of pasangan and combinations of sandhangan. Integrating projection-based segmentation techniques with morphological image analysis methods—such as connected-component analysis—remains essential for improving syllable-separation accuracy.

After segmentation, the next challenge is automatic transliteration, which translates Javanese syllables into the Latin script. Prior research notes that combinations of nglegena, pasangan, and sandhangan create over 11,000 distinct syllabic forms [3]. This large number of classes complicates classification. Visual similarities, such as script of ka, kha, and kra, further increase difficulty. Effective recognition demands good feature extraction and suitable classifier algorithms to reduce misclassification [9].

In this context, the k-Nearest Neighbor (k-NN) algorithm is a simple yet effective approach for handling automatic transliteration of Javanese script. k-NN operates based on the principle of proximity between feature vectors (numerical representations of a script's visual traits), enabling it to classify syllable forms with high similarity while controlling the error rate [10]. Authors in [2] demonstrated that k-NN can recognize Javanese script with competitive accuracy compared to other complex methods, provided the segmentation and feature extraction stages are performed well. Its non-parametric nature and lack of intensive training make k-NN a viable alternative in pilot studies of Javanese script OCR, especially when the amount of data per class is limited and deep learning models cannot be optimally implemented. According to authors in [11], the use of k-NN in research is also considered relevant for capturing subtle differences among printed images, thereby supporting more accurate and adaptive classification under varying printing conditions.

This study presents a pilot evaluation of OCR methods for printed Javanese script, an area that remains underexplored compared to handwritten script research. Using a representative single-page dataset, it provides an initial benchmark and empirical evaluation framework for segmentation and classification. The study establishes a baseline that bridges the gap between handwritten and printed OCR and analyzes how segmentation quality influences recognition performance. These findings form a foundation for developing robust OCR systems to support the preservation of Indonesia's written cultural heritage.

II. METHODOLOGY

The overall research methodology includes an experimental approach with a workflow-based implementation, as illustrated in Figure 1. This workflow consists of three main stages: (1) database development, (2) modeling, and (3) transliteration. It is designed to demonstrate the performance of the proposed method for recognizing and transliterating printed Javanese script images.

In the database development stage, a printed Javanese script dataset is prepared by collecting and segmenting script images from single-page source documents. The modeling stage involves constructing and training a recognition model. The transliteration stage involves transliterating recognized script markers into Latin script according to standard Javanese transliteration rules.

This methodological design allows for systematic evaluation from data preparation to model inference, ensuring that each stage contributes to the overall transliteration accuracy and robustness of the system.

A. Dataset Development

The Javanese script image dataset for this research was obtained from the printed manuscript titled "Hamong Tani" [12]. This manuscript was chosen because it contains comprehensive and representative content, including various forms of Javanese script.

Training data were obtained through image segmentation of the "Hamong Tani" manuscript, resulting in image fragments of individual scripts. The classification model was initially trained with 91 script classes, comprising nglegena, sandhangan, and pasangan. The segmented image data were saved in JPG format with uniform resolution after preprocessing. The amount of data per class was unbalanced, with the largest class reaching 654 images. Manual labeling was carried out on these data, where each image was labeled according to the script class based on the Javanese script dictionary, including the following scripts: ha, na, ca, ra, ka, da, ta, sa, wa, la, pa, dha, ja, ya, nya, ma, ga, ba, tha, nga, _ha, _he', h, hing, hu, _sa, _sar, _si, cu, d_tu, dhang, g_dha, ga, ge', gi, ing, je', je'ng, kang, ku, kung, l, l_la, l_li, lang, li, m, m_bu, m_ma, me', mi, mu, mung, n, n__na, n_da, n_ga, n_ka, n_la, n_na, n_ni, n_ru, n_ta, n_te', n_tu, ne', ngi, ngri, ni, pu, pe', pi, pra, pu, re', reng, s_ka, s_la, s_ra, s_tu, s_wa, s_wi, si, t_bu, taling, tarung, te', te'ng, ti, tu, we', wi, wu, ya_ng, and padalingsa.

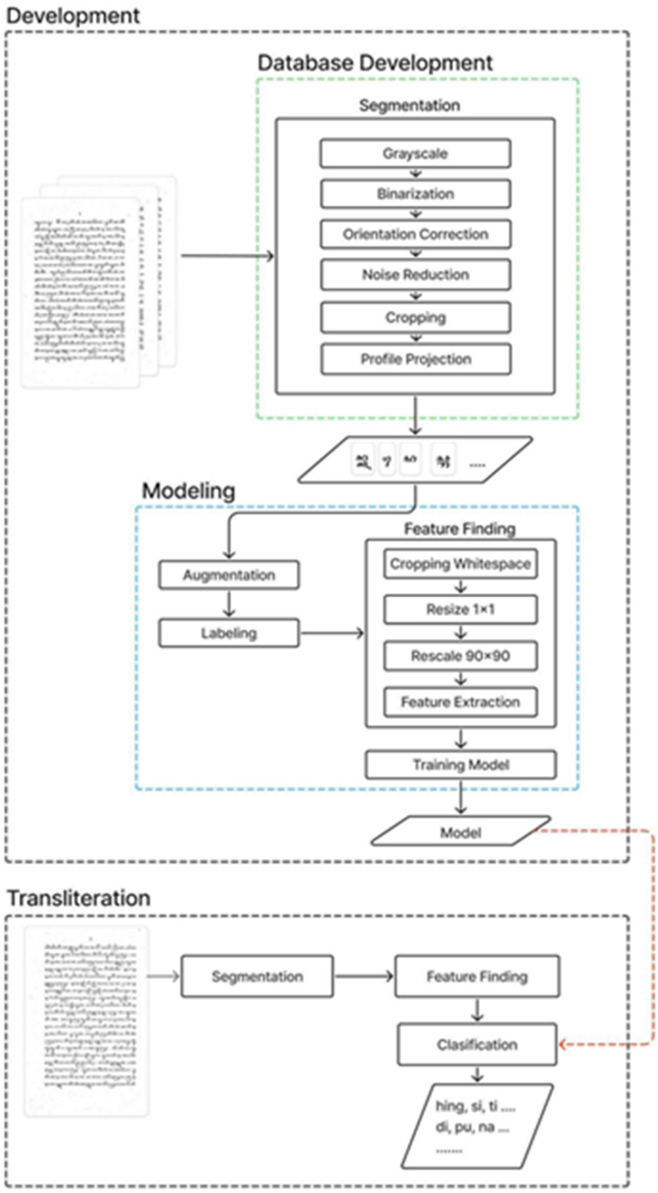

Fig. 1. Overall research methodology workflow.

Two custom-built applications were employed to support the dataset preparation process: Scinaja, which automates line and script segmentation, and Lasaja, which facilitates the manual labeling of segmented Javanese scripts. Both applications were developed by the authors specifically for this study and are available at https://drive.google.com/file/d/1xPCM_OrflnrZ_AnyUu-AChwvwf6fT4UT/view?usp=sharing.

For the next training, the number of classes increased to 306. Increasing the number of classes allows testing in real-world systems, where texts use varied sandhangan and pasangan.

The test data come from a single page of the "Hamong Tani" manuscript, processed in its entirety without manual segmentation. The purpose of the testing was to determine the OCR model's ability to recognize scripts in the context of a real manuscript, starting with automatic segmentation and classification, and finally, transliteration into Latin script.

B. Database Development

Figure 1 illustrates the implemented segmentation workflow. The development of the Javanese script database from printed manuscripts began with acquiring high-resolution manuscript images, followed by a structured segmentation pipeline designed to isolate individual scripts.

The segmentation process first converts the RGB images into grayscale. This simplifies subsequent operations and reduces computational complexity [13]. After the grayscale transformation, the images are binarized using the Sauvola thresholding method. This approach adapts the threshold based on the local mean and standard deviation. It enhances the visibility of script boundaries and minimizes background interference. In this study, the Sauvola parameters were set to a window size of 75 and k = 0.2 [14, 15]. To ensure uniform text alignment, an orientation correction was applied using the skew-detection method based on image moments [16]. This rotation ensures that all text lines are horizontally aligned before further processing. Noise and small artifacts were then removed using a median filter with a 5 × 5 kernel size [15, 17, 18]. Cropping followed noise removal to focus on relevant manuscript regions. Cropping boundaries were determined using histogram-based projection, with an additional 10% padding to ensure the text area remained complete.

Finally, line and script segmentation were conducted using the projection profile method. Horizontal projection profiles detected text lines; with valleys indicating line boundaries. Similarly, vertical projection profiles within each line were analyzed to identify script boundaries. In these profiles, valleys corresponded to spaces between adjacent scripts. This process produced segmented script fragments that were subsequently used in the feature extraction and modeling stages. The entire segmentation procedure followed and adapted methodologies previously applied to handwritten Javanese manuscripts [15].

C. Modeling Stage

The modeling stage begins with data augmentation, which increases the variety of script images and addresses dataset limitations. The augmentation process randomly rotates each Javanese script by −15° to 15° to generate new samples. Afterward, each script is assigned a label based on its class.

The next stage is feature discovery, which involves several processes. The first step is whitespace cropping to remove space around scripts. The images are then normalized to a uniform resolution of 1 × 1. Previous research showed that feature extraction worked for images measuring 90 × 90 pixels, so all scripts were resized to this size beforehand.

After image enhancement, the Javanese script images undergo feature extraction using several methods, including zoning [19, 20], projection profiles, Hu moments [21], bounding-box, and shape statistics [22, 23]. This process produces a numerical representation of each script's shape, which is then used in classification.

The zoning method divides the image into 8 × 8 pixel areas, generating 64 features. Horizontal and vertical projection profiles yield 32 features. Hu moments provide 7 features, whereas bounding-box and shape statistics give BBox ratio, pixel density, perimeter/area ratio, and solidity. There are 6 features from these methods. Altogether, the extraction process yields 109 features representing each Javanese script.

The extracted features train a k-NN model using the Euclidean (L2) metric. The resulting model is then ready for transliteration.

D. Transliteration Stage

The third stage of the research is the implementation of the model on a new manuscript. The manuscript image used as test data undergoes a series of segmentation processes, followed by feature discovery, similar to the database development stage. Next, the segmented letters are classified by the k-NN model, resulting in a transliteration in Latin text, for example, "hing, si, ti, ta, m_tu."

III. RESULTS AND DISCUSSION

A. Discussion of Modeling Development

Using 100 training datasets per class, from 91 classes, we conducted an experiment to find the best k-value for implementation in the manuscript image transliteration system, as shown in Table I. The data used for processing consisted of 9,100 vectors resulting from feature extraction from each dataset. The trends for accuracy, F1-score, and processing time were plotted based on Table I, as illustrated in Figure 2.

TABLE I. EXPERIMENTAL RESULTS OF K-NN CLASSIFIER BASED ON ACCURACY, F1-SCORE, AND TRAINING TIMES

k (number of neighbors)	Accuracy	F1-score	Training time (s)
3	0.954	0.953	0.32
5	0.929	0.926	0.33
7	0.897	0.892	0.30
9	0.875	0.867	0.31
11	0.854	0.844	0.30
21	0.723	0.711	0.32
31	0.657	0.643	0.35

A summary of the trends in Figure 2, based on Table I, shows that at k = 3, the accuracy and F1-score reached their highest value, approximately 0.9533. At k = 5, both accuracy and F1-score decreased significantly, and continued to decline at k = 7, 9, and 11.

Table I shows that the training time was relatively stable, ranging from 0.30 s to 0.33 s. These data suggest that computational complexity is not a significant issue.

The accuracy trend at low k values, specifically k = 1, 3, and 5, demonstrated relatively high accuracy but was prone to overfitting due to its sensitivity to similar test data. At intermediate k values (k = 7–15), the accuracy tended to stabilize and did not drop significantly, indicating a bias–variance balance point. At large k values (k > 20), the accuracy begins to decline significantly because the model becomes too smooth (increasingly biased), thus ignoring information about the minority class. This pattern shows that a k value that is too small provides good accuracy but risks undergeneralization, whereas a k value that is too large decreases accuracy due to oversmoothing.

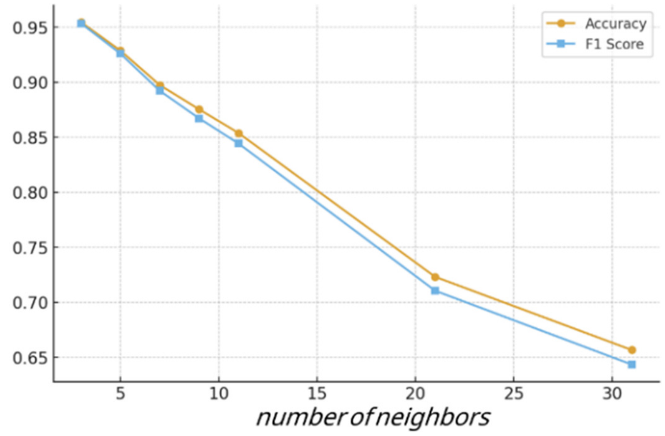

Fig. 2. Trends of accuracy and F1-score for different k values in the k-NN classifier.

The F1-score, as a balance indicator, follows the accuracy trend, meaning precision and recall are relatively balanced. The F1-score peak occurs at mid-range k values (around 7–13), indicating a compromise between misclassification and successful minority class recognition.

Judging from the variance of the results, fluctuations between folds are typically higher at small k values because the model relies heavily on nearest neighbors. At mid-range k values, variance decreases, indicating that the model is more consistent. At large k values, despite the low variance, the average performance decreases.

Based on the analysis of accuracy and F1-score values, testing will be conducted with a value of k = 3 before implementing the transliteration model in a real-world system, as it achieves the highest accuracy and F1-score. This is expected to maximize overall accuracy. The choice of the single-data test with k = 3 was not only due to its highest accuracy but also because Table I indicates no other more balanced "sweet spot." Over time, all k values are relatively similar, so this factor does not influence the choice.

However, considering the k values in the range of 7–13, this range is optimal for accuracy. Furthermore, if the focus is on stable model conditions, subsequent experiments may use a slightly larger k value within that range, namely k = 11. This also supports practical implementation, as k = 11 can be used as an initial choice and then further validated with cross-validation.

B. Discussion of the Transliteration Model Results

From the previous stage, a classification model was identified that has the potential to be applied in the automatic transliteration of Javanese script into Latin script. Using data from the "Hamong Tani" book on page 2, which consists of 20 lines, Table II shows a statistical summary of the results of the transliteration experiment using the 3-NN model.

TABLE II. PERFORMANCE EVALUATION OF SEGMENTATION AND TRANSLITERATION MODELS

Statistic	Segmentation accuracy (%)	Transliteration accuracy (%)		
		vs segmentation	vs system	vs ground truth
Minimum	40.00	27.27	33.33	16.67
Maximum	82.35	87.50	100.00	56.25
Mean	63.66	59.13	73.68	36.09
Median	64.21	62.50	71.43	35.25
Std. dev	12.31	17.12	18.83	10.83

Table II presents a summary of the performance evaluation results of the proposed model. The segmentation accuracy rate is obtained by dividing the number of correct segmentations by the total number of original data segments. The transliteration accuracy rate displays statistical test results for two training configurations: 100 datasets per class and 500 datasets per class. Accuracy is calculated based on the segmentation results, system output, and the original data.

Of the 355 Javanese scripts on the page, 225 were correctly cropped, yielding a segmentation accuracy of 63.38%. This limited accuracy represents a major bottleneck, as the system loses substantial information at the segmentation stage.

Approximately one-third of the scripts (36.6%) failed to be cropped correctly due to over-segmentation, under-segmentation, or misdetection, as illustrated in Figure 3. Over-segmentation occurs due to the presence of destructive pixels, which break a single script into multiple components. Ligature errors, where a single glyph represents a combination of letters, cause the segmentation system to treat several letters as one letter, a condition known as under-segmentation. The third factor, baseline misalignment, occurs when a subscript above or below the main letter is not detected.

Fig. 3. Representation of segmentation errors in Javanese script recognition.

In the recognition stage, translating Javanese into Latin, out of 225 correctly segmented scripts, only 150 were correctly recognized. Therefore, the recognition accuracy after correct segmentation was 66.67%. When evaluated for end-to-end accuracy, i.e., from the original image and calculated against a total of 355 ground truth scripts, the accuracy was 42.25%. This means that only about 42% of the scripts on the page were correctly recognized from input to final output. Very low end-to-end accuracy is common in OCR of complex scripts like Javanese, as the two stages of the process—segmentation and recognition—both contribute to errors.

Figure 4 shows an example of a transliteration error in which the Javanese script "ta" was mistakenly recognized as "ka." This error occurred due to the high visual similarity between the two glyphs, especially in printed manuscripts with subtle stroke variations. Of the ten samples tested, eight (80%) were misclassified. This shows that this pair of letters is among the most frequently misrecognized in the transliteration process in this study.

Fig. 4. Example of transliteration error between similar glyphs.

The experiment continued by using 11 neighbors to evaluate the effect of a larger k on classification accuracy, while also increasing the number of classes from 91 to 281. The results are presented in Table III.

TABLE III. ACCURACY COMPARISON OF K-NN (K = 3 AND K = 11) FOR DIFFERENT NUMBERS OF SCRIPTS PER LINE AND TRAINING SAMPLE SIZES

Row	**Number of scripts**	**Accuracy (%)**		
		3-NN (91 scripts, 100 samples / script)	**3-NN (91 scripts, 500 samples / script)**	**11-NN (281 scripts, 500 samples / script)**
1	16	56.250	56.250	68.750
2	17	29.412	35.294	41.176
3	17	35.294	35.294	41.176
4	20	35.000	35.000	45.000
5	18	50.000	50.000	61.111
6	17	35.294	35.294	52.941
7	19	36.842	36.842	36.842
8	18	33.333	33.333	44.444
9	18	55.556	61.111	55.556
10	19	26.316	31.579	36.842
11	19	26.316	26.316	42.105
12	17	52.941	52.941	52.941
13	17	17.647	23.529	52.941
14	19	21.053	21.053	36.842
15	16	25.000	25.000	68.750
16	18	33.333	33.333	38.889
17	18	16.667	16.667	16.667
18	18	50.000	50.000	66.667
19	17	41.176	41.176	47.059
20	17	35.294	41.176	52.941

Table IV summarizes OCR performance results for the 3-NN and 11-NN classifiers under different training configurations, specifically highlighting accuracy outcomes. Table III contains the comprehensive training and evaluation data that serve as the basis for this summary. The experiments used 91 and 281 scripts, each with 100 or 500 samples per script. Accuracy values, averaged over 10-fold cross-validation, represent the model's performance stability.

In general, the 3-NN configuration produced the highest average accuracy compared to the other k values (k = 5, 7, 9, 11, 13, 21, 31), and was therefore selected as the primary model. Meanwhile, the 11-NN model was retained in the comparison test because it demonstrated accuracy stability across a larger range of scripts and datasets.

TABLE IV. SUMMARY OF OCR CLASSIFICATION PERFORMANCE

Statistic	**Accuracy (%)**		
	3-NN (91 scripts, 100 samples / script)	**3-NN (91 scripts, 500 samples / script)**	**11-NN (281 scripts, 500 samples / script)**
Minimum	16.67	16.67	16.67
Maximum	56.25	61.11	68.75
Mean	35.93	37.45	48.62
Median	35.29	35.29	47.06
Std. dev	10.69	11.63	11.67

The test results showed that increasing the number of samples per script from 100 to 500 at 91 scripts did not significantly improve the average accuracy per line of text. However, when the number of scripts was expanded to 281, the 11-NN model provided better stability on lines with varying script complexity. This indicates that adding scripts expands the distinction between classes, but also increases the classification challenge, especially for scripts with high shape similarity, such as pasangan and sandhangan.

Specifically, transliteration accuracy values varied between 16.67% and 68.75%, with a decreasing trend in rows containing more than 18 scripts. This phenomenon indicates the influence of segmentation noise and script shape variations between row positions.

The best results were achieved in the 3-NN configuration with 100 samples per script in row 1 (68.75%), whereas the lowest accuracy was found in row 17 (16.67%). These findings confirm that the amount of training data is not the sole dominant factor; rather, segmentation quality and the balance of class distribution are more important determinants of transliteration performance.

IV. CONCLUSION

Segmentation performance directly determines the accuracy of Javanese script transliteration. Errors such as over-segmentation, under-segmentation, and baseline misalignment significantly reduce accuracy, making segmentation the primary source of failure in the Optical Character Recognition (OCR) and transliteration pipeline. These findings confirm that improving segmentation, especially for overlapping glyphs and print variations, is essential. Enhancing this stage is expected to increase overall transliteration accuracy in future studies.

Although recognition accuracy in this pilot study is below 60%, the results represent an important initial evaluation. The highest training accuracy reached 95.4%, demonstrating that the k-Nearest Neighbors (k-NN) model effectively learns the features of printed Javanese scripts. The lower test accuracy is mostly due to the limited sample size and segmentation imperfections in the current dataset. Consequently, this study serves as a baseline experiment to inform future work on dataset expansion and OCR pipeline optimization.

Building on the previously highlighted need for improved segmentation and dataset expansion, future research could strengthen this study by optimizing feature selection and advancing computational methods. Prior work by authors in [24] demonstrated that dimensionality reduction enhances text classification performance, suggesting potential benefits for visual feature optimization in OCR. New models, such as the Quantum Dilated Convolutional Neural Network (QDCNN) [25], may further improve recognition accuracy and computational efficiency by leveraging quantum methods, supporting Javanese OCR systems in handling high-resolution data while reducing training complexity. Future research will focus on applying deep learning and quantum models to traditional script recognition in resource-limited settings.

ACKNOWLEDGMENT

This study was supported by the LPPM Sanata Dharma University Grants 016 PeneI./LPPM-USD/II/2025.

REFERENCES

[1] H. Alwi, "Kebijakan Bahasa Daerah," in *Bahasa Daerah dan Otonomi Daerah*, H. Alwi and A. R. Zaidan, Eds. Jakarta, Indonesia: Pusat Bahasa, 2001, pp. 38–47.

[2] A. R. Widiarti, A. Harjoko, Marsono, and S. Hartati, "Preprocessing Model of Manuscripts in Javanese Characters," *Journal of Signal and Information Processing*, vol. 5, no. 4, pp. 112–122, Oct. 2014, https://doi.org/10.4236/jsip.2014.54014.

[3] A. R. Widiarti, "Model Transliterasi Otomatis Citra Naskah Aksara Jawa," Ph.D. dissertation, Computer Science Department, Universitas Gadjah Mada, Yogyakarta, Indonesia, 2015.

[4] A. W. Mahastama and L. D. Krisnawati, "Improving Projection Profile for Segmenting Characters from Javanese Manuscripts," in *Proceedings of the 1st International Conference on Intermedia Arts and Creative Technology*, Yogyakarta, Indonesia, 2019, pp. 77–82, https://doi.org/10.5220/0008526900770082.

[5] Y. Sugianela and N. Suciati, "Character Image Segmentation of Javanese Script using Connected Component Method," *Jurnal Ilmu Komputer dan Informasi*, vol. 12, no. 2, pp. 67–74, July 2019, https://doi.org/10.21609/jiki.v12i2.677.

[6] A. F. Ganai and A. Koul, "Projection profile based ligature segmentation of Nastaleeq Urdu OCR," in *2016 4th International Symposium on Computational and Business Intelligence*, Olten, Switzerland, 2016, pp. 170–175, https://doi.org/10.1109/ISCBI.2016.7743278.

[7] T. A. Tofiq and J. A. Hussein, "Kurdish Text Segmentation using Projection-Based Approaches," *UHD Journal of Science and Technology*, vol. 5, no. 1, pp. 56–65, May 2021, https://doi.org/10.21928/uhdjst.v5n1y2021.pp56-65.

[8] L. Liliana, S. M. Soephomo, G. S. Budhi, and R. Adipranata, "Segmentation of Hanacaraka Characters Using Double Projection Profile and Hough Transform," in *Big Data Technologies and Applications: 8th International Conference, BDTA 2017*, Gwangju, South Korea, 2017, pp. 29–37, https://doi.org/10.1007/978-3-319-98752-1_4.

[9] 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*, vol. 7, no. 3, pp. 309–316, Aug. 2022, https://doi.org/10.22219/kinetik.v7i3.1491.

[10] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Hoboken, NJ, USA: John Wiley & Sons, 2000.

[11] A. Rahman and M. N. A. Khan, "A Classification Based Model to Assess Customer Behavior in Banking Sector," *Engineering, Technology & Applied Science Research*, vol. 8, no. 3, pp. 2949–2953, June 2018, https://doi.org/10.48084/etasr.1917.

[12] K. F. Holle, *Hamong Tani (in Javanese, translated from Dutch by F. L. Winter)*. Batavia, Indonesia: Landsdrukkerij, 1876.

[13] Z. N. Khudhair *et al.*, "Color to Grayscale Image Conversion Based on Singular Value Decomposition," *IEEE Access*, vol. 11, pp. 54629–54638, 2023, https://doi.org/10.1109/ACCESS.2023.3279734.

[14] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern Recognition*, vol. 33, no. 2, pp. 225–236, Feb. 2000, https://doi.org/10.1016/S0031-3203(99)00055-2.

[15] G. K. Indraputra and A. R. Widiarti, "Implementation of 4-Directional Depth First Search and Projection Profile for Javanese Manuscript Image Segmentation," *Journal of Informatics and Telecommunication Engineering*, vol. 9, no. 1, pp. 218–228, July 2025, https://doi.org/10.31289/jite.v9i1.15433.

[16] M. Ramanan, "A Hybrid Approach for Skew Detection and Correction in the Multi-script Scanned Document," *Asian Journal of Research in Computer Science*, vol. 4, no. 2, pp. 1–8, Nov. 2019, https://doi.org/10.9734/ajrcos/2019/v4i230112.

[17] V. Chauhan and P. Malhotra, "Reduction of Noise in Restoration of Images Using Mean and Median Filtering Techniques," *International Journal for Research in Applied Science and Engineering Technology*, vol. 9, no. 9, pp. 301–313, Sept. 2021, https://doi.org/10.22214/ijraset.2021.37965.

[18] S. Anand and L. Priya, "Digital Image Fundamentals," in *A Guide for Machine Vision in Quality Control*, Boca Raton, FL, USA: Chapman and Hall/CRC, 2019.

[19] 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*, Denpasar, Indonesia, 2019, pp. 264–268, https://doi.org/10.1109/ICEEIE47180.2019.8981462.

[20] M. T. Graciello, A. N. Handayani, and A. P. Wibawa, "Optimization of Nglegena Javanese Script Recognition With Machine Learning Based on Zoning And Normalization of Feature Extraction," *Indonesian Journal of Data and Science*, vol. 6, no. 2, pp. 281–293, July 2025, https://doi.org/10.56705/ijodas.v6i2.256.

[21] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, Feb. 1962, https://doi.org/10.1109/TIT.1962.1057692.

[22] G. A. Robby, A. Tandra, I. Susanto, J. Harefa, and A. Chowanda, "Implementation of Optical Character Recognition using Tesseract with the Javanese Script Target in Android Application," *Procedia Computer Science*, vol. 157, pp. 499–505, Jan. 2019, https://doi.org/10.1016/j.procs.2019.09.006.

[23] M. Anandhalli, A. Tanuja, and P. Baligar, "Geometric invariant features for the detection and analysis of vehicle," *Multimedia Tools and Applications*, vol. 81, no. 23, pp. 33549–33567, Sept. 2022, https://doi.org/10.1007/s11042-022-12919-8.

[24] S. R. Basha, J. K. Rani, and J. J. C. P. Yadav, "A Novel Summarization-based Approach for Feature Reduction Enhancing Text Classification Accuracy," *Engineering, Technology & Applied Science Research*, vol. 9, no. 6, pp. 5001–5005, Dec. 2019, https://doi.org/10.48084/etasr.3173.

[25] R. Basha, P. Pathak, M. Sudha, K. V. Soumya, and J. Arockia Venice, "Optimization of Quantum Dilated Convolutional Neural Networks: Image Recognition With Quantum Computing," *Internet Technology Letters*, vol. 8, no. 3, May 2025, Art. no. e70027, https://doi.org/10.1002/itl2.70027.