JURNAL UPI

Jurnal Pendidikan Multimedia (EDSENCE)

Journal homepage: https://ejournal.upi.edu/index.php/Edsence

Jurnal Pendidikan Multimedia EDSENCE Education, Science, and Creative Technology

Hanacaraka Javanese Handwriting Detection Using Recurrent Neural Network (RNN)

Ichsan Nur Rachmad Yusuf[1], Rin Rin Nurmalasari[2], Lia Kamelia[3]*

[1,2,3] UIN Sunan Gunung Djati, Bandung, Indonesia

Correspondence: E-mail: rinrinnm@uinsgd.ac.id

ABSTRACT

Hanacaraka Javanese script is a valuable Indonesian cultural heritage, but its use has declined due to a lack of knowledge and ability to read and write the script. The main challenge in detecting and recognizing handwritten Javanese script is the variation of its shape and writing style. This research aims to train computers to recognize Javanese script. Prior to this research, there have been several similar studies with different recognition methods. In this research, the Recurrent Neural Network (RNN) method is used. The process of detection and recognition of Javanese script letters is divided into three parts: input image of script images used as a dataset of 500 images for training data and 8 pictures for prediction test data, the process of creating a Recurrent Neural Network model, and the output of this design is the performance of the Recurrent Neural Network model. The test results show that the model has an overall accuracy of 96%, with an average precision, recall, and F1 score of 96% each. Sentences such as "Dhahara", "Jawanagara", "Malaca", and "Ramayana" were successfully detected completely correctly, although some sentences such as "Jayabaya", "Nyala", and "Palawa" experienced prediction errors.

ARTICLE INFO

Article History:
Submitted/Received 28 Feb 2024
First Revised 20 May 2024
Accepted 5 Jun 2024
First Available online 29 Jun 2024
Publication Date 29 Jun 2024

Keyword:
Javanese Script,
Handwriting Detection,
Hanacaraka,
Deep Learning,
Recurrent Neural Network (RNN).

1. INTRODUCTION

Language is vital in life, especially in the process of interaction between one human being and another. Language becomes a tool to convey ideas and ideas. A region has a mother tongue as an introduction to communication between communities (Rabiah, 2018). The writing and representation of regional languages are indicated by certain letters (Utami et al., 2023). Javanese is one of the regional languages that has a particular letter in its writing, known as Javanese script. Javanese Hanacaraka or Carakan script is one of the traditional scripts in Indonesia. Javanese script Hanacaraka was used by the Javanese community, especially in the royal courts such as Yogyakarta and Surakarta, to develop a written tradition in Javanese. Hanacaraka is generally used to write texts such as stories (serat), historical records (babad), ancient songs (kakawin), or predictions (primbon). Currently, writing in Javanese script has been widely abandoned, but that does not mean that Javanese script has completely disappeared. Javanese script is still preserved and can still be found in several regions on the island of Java. One of the ways to preserve this cultural heritage is by incorporating Javanese script lessons into the local content curriculum at school (Atina et al., 2023).

A number of studies on Javanese script pattern recognition continue to be conducted using various methods that focus on its unique characteristics. Researchers use image processing algorithms to deeply analyze the structure and features of Javanese characters, and apply machine learning and artificial intelligence approaches to improve pattern recognition accuracy. Image pattern recognition is one of the capabilities possessed by a computer. One of the fields of study that studies image processing is where both input and output are in the form of digital image files. Photos are an example of images that can be processed easily through certain software. Deep Learning is the center of attention in the development of machine learning because it provides optimal results in computer vision. Deep Learning is a branch of machine learning that uses algorithms inspired by the structure of the human brain (Fontanella et al., 2023).

Machine learning-based methods are commonly used techniques to solve these problems. One technique that has proven to produce the best results is the Recurrent Neural Network technique. Recurrent Neural Network (RNN) is one of the deep learning algorithms that can make predictions based on numerical time series data, which works by performing repeated processing. A recurrent Neural Network is an artificial neural network that modifies the output of the network to become the input of the recurrent neural network. The network is then used to generate new outputs (Caniago et al., 2023).

Based on the background that has been described, this research focuses on handwriting detection of Hanacaraka Javanese script using the Recurrent Neural Network (RNN) algorithm. The main objective is to recognize and classify Hanacaraka Javanese script based on unique handwriting characteristics. By utilizing advances in the field of artificial intelligence, this research aims to make a significant contribution to solving problems related to the identification and classification of Hanacaraka Javanese letters, which can have a positive impact on the preservation and use of Javanese writing culture. The results of this research are expected to not only improve the accuracy of Hanacaraka Javanese script pattern recognition in general but also provide a classification of the letters based on their type. With a handwriting detection system that can distinguish between the characters of the Hanacaraka Javanese characters.

DOI: https://doi.org/10.17509/edsence.v6i1.74727
p-ISSN:2685-2489 | e-ISSN:2685-2535

2. METHODS

2.1. System Design

The system design for the detection of Javanese hanacaraka handwriting consists of three stages, namely input, process, and output. The input of the handwriting detection of the Javanese hanacaraka script is the image of the Javanese script image which is used as a dataset to train and evaluate the performance of the Recurrent Neural Network model. The process stage of this design is the creation of a Recurrent Neural Network model. The output of this design is the performance of the Recurrent Neural Network model. The flowchart of this system design is shown in Fig.1.

2.2. Dataset Collection

The dataset in Fig 2 has a total of 500 dataset images from Kaggle (Nugroho,2024). The images are divided into 20 folders; the folders are named Ca, Ra, Ka, Da, Ta, La, Dha, Ja, Ya, Ma, Ga, Ba, Tha, Nga, Ha, Sa, Pa, Nya, Na, and Wa. Figure 2 shows the hanacaraka letter dataset folder.

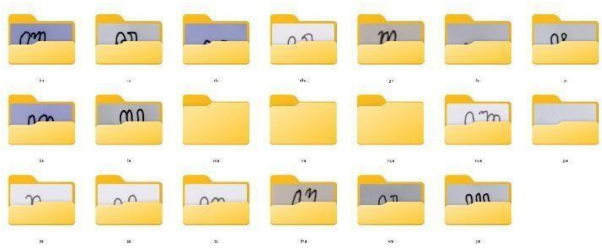

Figure 1. Hancaraka Javanese Script Dataset Folder

The dataset in Fig 2 is the Hannan Hunafa dataset consisting of 8 images grouped in a folder called predict test (Hannanhunafa,2022). This folder contains sentences in Hanacaraka Javanese script, namely: dhahara, jawanagara, jayabaya, malaca, nyala, palawa, paratama, and ramayana. Figure 2 is the sentence used for testing.

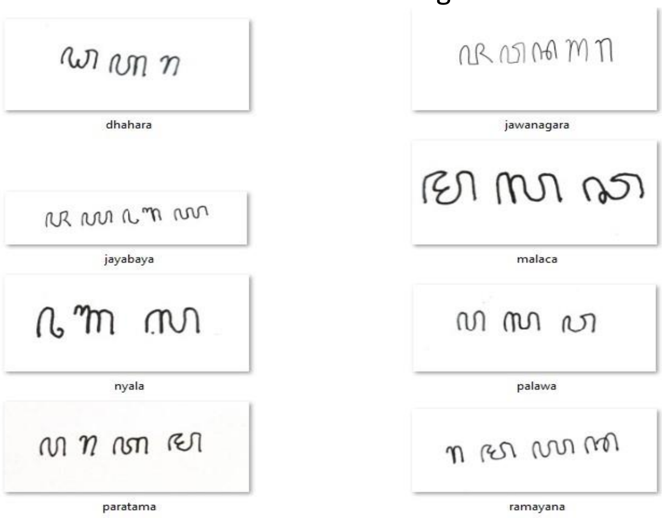

Figure 2. Sample testing data

DOI: https://doi.org/10.17509/edsence.v6i1.74727
p-ISSN:2685-2489 | e-ISSN:2685-2535

2.3 Recurrent Neural Network Model

The Recurrent Neural Network (RNN) architecture shown in Figure 3 represents a deep learning model for detecting Javanese Hanacaraka characters using Recurrent Neural Networks (RNN) with an LSTM layer. Here's a detailed breakdown of the architecture:

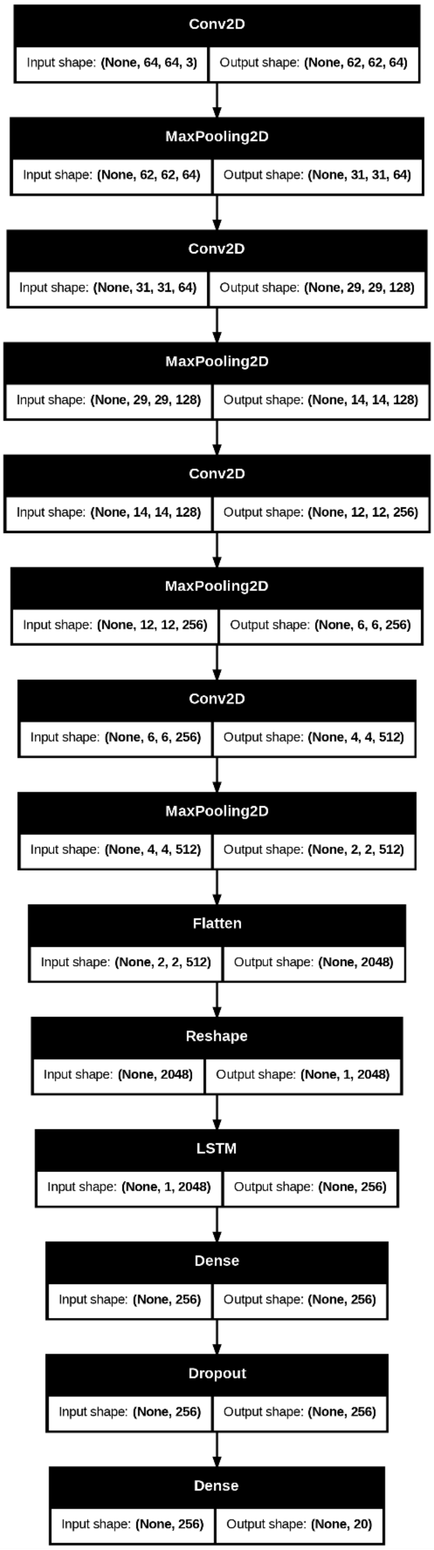

Figure 3. Model architecture

DOI: https://doi.org/10.17509/edsence.v6i1.74727
p-ISSN:2685-2489 | e-ISSN:2685-2535

a. Input Layer: Shape: (64, 64, 3), representing a 64x64 pixel image with 3 color channels (RGB).

b. Convolutional Layers: The model begins with four convolutional layers, each followed by a max-pooling layer.
   - Conv2D (64 filters): Extracts 64 feature maps from the image using 3x3 filters.
   - MaxPooling2D: Reduces the spatial dimensions (downsampling) by a factor of 2, from (62, 62, 64) to (31, 31, 64).
   - Conv2D (128 filters): Further extracts more complex features using 128 filters.
   - MaxPooling2D: Reduces the dimensions to (14, 14, 128).
   - Conv2D (256 filters): Increases the depth with 256 filters, extracting even higher-level features.
   - MaxPooling2D: Reduces the dimensions to (6, 6, 256).
   - Conv2D (512 filters): The final convolution layer, with 512 filters.
   - MaxPooling2D: Reduces the final spatial dimensions to (2, 2, 512).

c. Flatten Layer: Converts the 3D feature maps into a 1D vector of size 2048, preparing the data for the LSTM layer.

d. Reshape Layer: This function reshapes the flattened vector into a sequence of length 1 with 2048 features, compatible with the LSTM layer.

e. LSTM Layer: 256 units: A Recurrent Neural Network (RNN) layer with Long Short-Term Memory (LSTM) cells, which helps capture temporal dependencies in the data. This is crucial for detecting sequential patterns in the Javanese characters.

f. Dense Layers:
   - A fully connected layer with 256 units and ReLU activation.
   - A Dropout layer (0.2) is applied to prevent overfitting by randomly dropping 20% of the units during training.
   - Another Dense layer with 20 output units, corresponding to the 20 classes of Javanese characters, using softmax activation to output probabilities for classification.

g. Output: The final output is a probability distribution over 20 classes (the Javanese characters), allowing the model to predict the correct character. This architecture leverages the feature extraction power with the sequential modeling capabilities of LSTMs, making it effective for recognizing and classifying complex characters like Javanese Hanacaraka.

3. RESULT AND ANALYSIS

3.1. Epoch Testing and Analysis

This test is the process of setting and determining the number of iterations or rounds performed by the algorithm when training the model on the dataset. One epoch means one time the model looks at the entire training dataset. The optimal number of epochs is usually determined through experimentation and validation. Table 1 shows the results of five training times.

DOI: https://doi.org/10.17509/edsence.v6i1.74727
p-ISSN:2685-2489 | e-ISSN:2685-2535

Table 1. Model Testing by Epoch

No.	Epoch	Loss (%)	Validation Loss (%)	Accuracy (%)	Validation Accuracy (%)
1.	Epoch 1	281.23	169.54	10.76	42.92
2.	Epoch 2	50.58	32.99	82.87	91.42
3.	Epoch 3	5.97	18.87	98.26	94.63
4.	Epoch 4	2.41	32.04	99.24	93.37
5.	Epoch 5	1.28	27.92	99.65	93.33

Overall, from epoch one to five, the model showed a significant increase in accuracy and a decrease in loss values on both the training and validation data. The model learns very well and has good generalization ability, as seen from the increased accuracy and decreased loss values on the validation data.

3.2. Testing and Analysis of Recurrent Neural Network Loss and Accuracy

Loss Evolution refers to the change in the loss value during the training process of a machine learning model. The loss value measures the error made by the model in prediction, and the goal of training is to minimize this loss value. Accuracy Evolution refers to the change in the accuracy value during the model training process. Accuracy measures how good the model is at making correct predictions on training and validation data. The accuracy value is also calculated at each epoch during training.

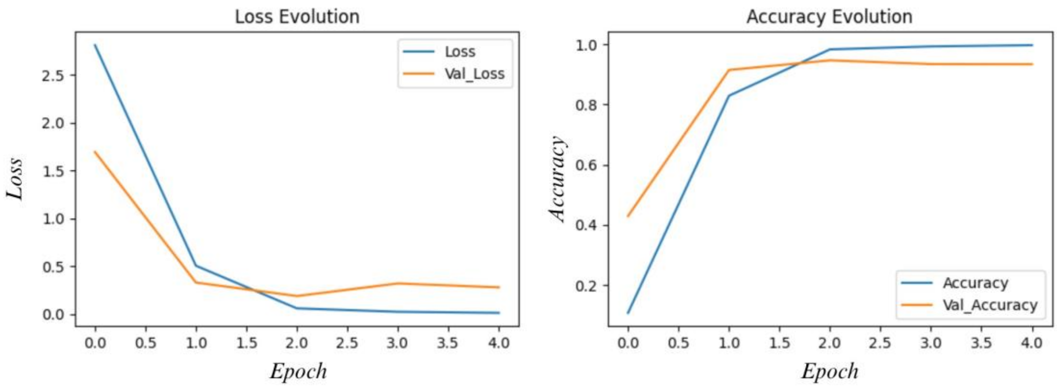

Figure 4. Graph of Loss Evolution and Accuracy Evolution

The Loss Evolution graph shows that the loss value on the training data (blue curve) decreases consistently as epochs increase. This shows that the model learns from the training data well, reducing its prediction error. Meanwhile, the loss value on the validation data (orange curve) also decreases but then tends to stabilize after a particular epoch. This shows

DOI: https://doi.org/10.17509/edsence.v6i1.74727
p-ISSN:2685-2489 | e-ISSN:2685-2535

that the model also generalizes the validation data well, but after a certain point, additional improvement becomes minimal.

The Accuracy Evolution graph shows that the accuracy value on the training data (blue curve) increases significantly as the epoch increases, indicating that the model is getting better at making correct predictions on the training data. The accuracy value on the validation data (orange curve) also increases, although not as fast as the increase in training data. This shows that the model generalizes well and is able to make correct predictions on untrained data.

The conclusion in Figure 4 shows that the model is learning well. The decreasing loss value indicates that the prediction error is decreasing, while the increasing accuracy value indicates that the model is getting better at making correct predictions. The consistency between the values in the training data and validation data shows that the model does not experience significant overfitting, meaning that the model is able to generalize well to new data.

3.3. Testing and Analysis of Recurrent Neural Network Confusion Matrix

This test is performed using the program code to train the recurrent neural network model. Recurrent neural network models are often used in natural language and time series processing tasks due to their ability to handle sequential data. By applying a confusion matrix, it is possible to evaluate how well the recurrent neural network model predicts the classes and make necessary adjustments to improve its performance. Classes of datasets that can be classified will first be converted into numbers 0 to 19, this aims to facilitate the process of writing classes in confusion matrix testing. The confusion matrix evaluation of the recurrent neural network model is shown in Figure 5.

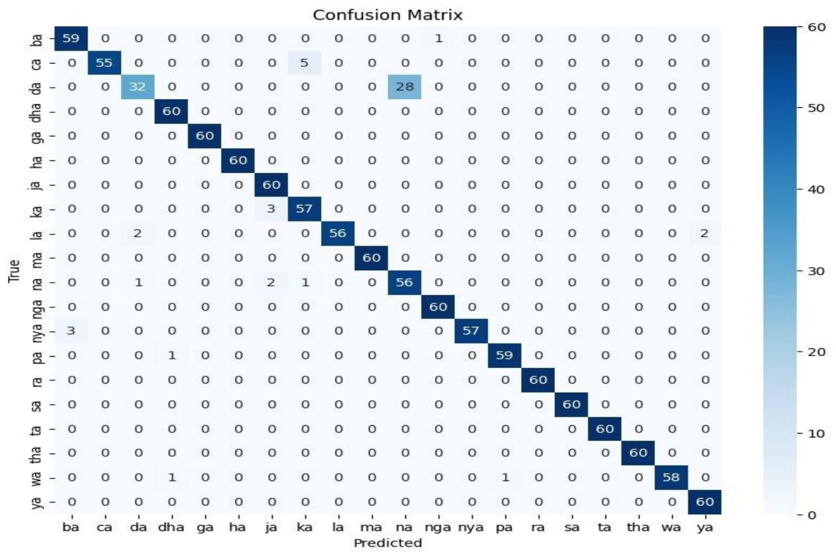

Figure 5. Confusion Matrix Recurrent Neural Network

DOI: https://doi.org/10.17509/edsence.v6i1.74727
p-ISSN:2685-2489 | e-ISSN:2685-2535

Figure 6 shows that the letters dha, ga, ha, ja, ma, nga, ra, sa, ta, tha, and ya were correctly predicted without error, each reaching 60 on the main diagonal of the matrix. This reflects the model's excellent ability to recognize these letters. However, letters such as ba, ca, da, ka, la, na, nya, pa, and wa experienced some errors, with the number of correct predictions being 59, 58, 57, 56, 55, and 32 out of 60, respectively. In particular, the letter da showed the lowest performance, indicating difficulty for the model in recognizing this letter.

	precision	*recall*	*fl-score*	*support*
ba	1.00	0.97	0.98	60
ca	1.00	0.98	0.99	60
da	0.95	1.00	0.98	60
dha	0.92	1.00	0.96	60
ga	1.00	0.98	0.99	60
ha	1.00	1.00	1.00	60
ja	1.00	1.00	1.00	60
ka	0.95	0.63	0/76	60
la	1.00	0.97	0.98	60
ma	0.98	0.92	0.95	60
na	0.74	0.95	0.83	60
nga	0.97	1.00	0.98	60
nya	1.00	1.00	1.00	60
pa	0.87	0.92	0.89	60
ra	1.00	1.00	1.00	60
sa	0.98	1.00	0.99	60
ta	1.00	1.00	1.00	60
tha	1.00	0.98	0.99	60
wa	1.00	0.93	0.97	60
ya	0.94	1.00	0.97	60
Accuracy			0.96	1200
Macro avg	0.97	0.96	0.96	1200
Weighted avg	0.97	0.96	0.96	1200

Figure 6. Classification Report

Based on Figure 6, The support which indicates the actual number of instances of each class in the dataset, was 60 for each class, indicating a balanced distribution of the data. The overall accuracy of the model is 0.96, meaning 96% of all predictions are correct. The macro averages for precision, recall, and f1-score are all 0.96, indicating a steady performance across all classes. The weighted average is also 0.96, confirming that the overall result is not much affected by the support distribution.

3.4. Testing and Analysis of Hancaraka Javanese Script Classification

Using the recurrent neural network model, it was tested using script sentences namely dhahara, jawanagara, jayabaya, malaca, nyala, palawa, paratama and ramayana. Table 2 Table 2 shows the results of testing the Javanese script Hanacaraka using several samples of Javanese script writing.

DOI: https://doi.org/10.17509/edsence.v6i1.74727
p-ISSN:2685-2489 | e-ISSN:2685-2535

Table 2. Results of the Hancaraka Javanese Script Classification Test

No	Sentence Script	Detection Results	number of letters	Correct Prediction	Incorrect Prediction
1	Dhahara	Dha, Ha, Ra	3	3	-
2	Jawanagara	Ja, Wa, Na, Ga, Ra	5	5	-
3	Jayabaya	Ja, Ya, Da, Ga, Ha	3	2	3
4	Malaca	Ma, La, Ca	3	3	-
5	Nyala	Ra, Ga, La, Ya	2	1	3
6	Palawa	Pa, Ya, Pa, Pa	3	1	3
7	Paratama	Pa, Ra, Ha, Sa	4	2	2
8	Ramayana	Ra, Ma, Ya, Na	4	4	-

Overall, the test results on table 2 show that the model has varying performance depending on the complexity and length of the sentences. Some sentences were detected ideally, while others showed significant error rates. This shows that although the model has good potential, further enhancements are still needed to improve the accuracy of hanacaraka Javanese script detection, especially for more complex sentences.

4. CONCLUSION

The design and implementation of a Recurrent Neural Network (RNN) model for Hanacaraka Javanese script image classification involves several essential steps. The first step is the collection and processing of a dataset consisting of Hanacaraka Javanese script images, which is then used to train an RNN model specifically designed to recognize patterns in handwritten sequences. Model implementation includes building the RNN architecture, training the model with the processed dataset, and evaluating the model performance based on metrics such as accuracy, precision, recall, and F1-score. As a result, the developed RNN model is able to classify Javanese script with an overall accuracy of 96%, and evaluation metrics such as precision, recall, and F1- score each average 96%, indicating that the model has good performance in detecting Javanese script handwriting. Some sentences such as "Dhahara", "Jawanagara", "Malaca", and "Ramayana" were successfully detected entirely correctly.

REFERENCES

Atina, V., Palgunadi, S., & Widiarto, W. (2016). Program Transliterasi Antara Aksara Latin Dan Aksara Jawa Dengan metode FSA. Jurnal Teknologi & Informasi ITSmart, 1(2), 60. https://doi.org/10.20961/its.v1i2.592

Caniago, A. I., Kaswidjanti, W., & Juwairiah, J. (2021). Recurrent neural network with gate recurrent unit for stock price prediction. Telematika, 18(3), 345. https://doi.org/10.31315/telematika.v18i3.6650

Fontanella, F., Colace, F., Molinara, M., Scotto Di Freca, A., & Stanco, F. (2020). Pattern recognition and artificial intelligence techniques for Cultural Heritage. Pattern Recognition Letters, 138, 23–29. https://doi.org/10.1016/j.patrec.2020.06.018

Hannanhunafa. (2022). Javanese script classification + recognition. Kaggle. https://www.kaggle.com/code/hannanhunafa/javanese-script-classification-recognition

Nugroho, R. P. (2024). Aksara Jawa / hanacaraka. Kaggle. https://www.kaggle.com/datasets/vzrenggamani/hanacaraka

Rabiah, S. (2018). Language as a Tool for Communication and Cultural Reality Discloser. https://doi.org/10.31227/osf.io/nw94m

Utami, G. C., Widiawati, C. R., & Subarkah, P. (2023). Detection of Indonesian food to estimate nutritional information using YOLOV5. Teknika, 12(2), 158–165. https://doi.org/10.34148/teknika.v12i2.636.

DOI: https://doi.org/10.17509/edsence.v6i1.74727
p-ISSN:2685-2489 | e-ISSN:2685-2535