Web-based Writing Learning Application of Basic Hanacaraka Using Convolutional Neural Network Method

Dewi Candani Sulaiman[1], Teady Matius Surya Mulyana[2]
[1,2] Informatics, Faculty of Technology and Design, Universitas Bunda Mulia, Jakarta, Indonesia
[1]dewicandani8@gmail.com, [2]tmulyana@bundamulia.ac.id

Accepted 09 December 2022
Approved 29 December 2022

***Abstract*— The Javanese script, known as Hanacaraka, or Carakan, is one of the traditional Indonesian scripts developed and used on the island of Java. The government's efforts to preserve the use of Javanese language and script by making Javanese a compulsory subject of local content at the education level in Central Java and East Java. In the basic competence of writing, the Javanese script has a complicated shape so that students have difficulty writing and recognizing Javanese script writing. Through this research a web-based basic Javanese writing learning application was designed that can recognize handwriting digitally which aims to help learn basic hanacaraka writing for beginners, especially students at the basic education level in Central Java and East Java. Handwriting Recognition is a system that can recognize handwritten characters and convert them into text that can be read and understood by machines or computers. The handwriting recognition process in this study uses the Convolutional Neural Network (CNN) algorithm which has the capability and ability to recognize patterns in images. Based on the tests that have been carried out between the two architectural models that have been made, the performance of the CNN model that will be used from various experiments has an accuracy of 98.29% and a loss of 0.0746 on the training data. As well as producing an average accuracy value of 99.52%, an average error rate of 0.48%, an overall accuracy of 95.03% and an overall error rate of 4.97%.**

***Index Terms*— convolutional neural network; deep learning; hanacaraka; handwriting recognition**.

I. INTRODUCTION

The Javanese script, known as Hanacaraka, or Carakan, is one of the traditional Indonesian scripts developed and used on the island of Java, which is a derivative of the Brahmi script originating from India. Javanese script is used in the field of literature, and in the daily life of the Javanese people from the 17th century until now [1]. But as a result of the times, the application of Javanese script in everyday life began to decline or be limited. The Government's efforts to maintain and preserve the use of the Javanese language and script by making Javanese a compulsory subject of local content at the elementary to senior high school levels in Central Java and East Java. In the basic competencies of writing and reading Javanese script, it is a difficult part because it has complicated shapes and complex elements, so that students have difficulty remembering the forms or patterns of Javanese script writing.

In learning a language whose writing is in the form of characters or symbols, regular writing practice is needed. Practicing writing can improve long-term memory, and the ability to recognize things because when writing motor skills work which connects to the nerves of the brain [2]. In this era of digitization, so that writing practice activities such as basic Javanese script are still being carried out so learning applications are needed that can recognize handwriting digitally.

Applications for recognizing digital handwriting can be built with the help of Artificial Intelligence (AI) which can analyze patterns based on data known as deep learning. Deep learning architecture that can recognize patterns in an image is the Convolutional Neural Network (CNN) [3]. Convolutional Neural Network is a deep learning algorithm that can recognize a pattern in an image such as numbers, characters, and in the case of computer vision and natural language processing [4] [5]. Where CNN has good ability and capability in performing image recognition. The recognition system used is online handwriting recognition system.

Based on previous research conducted by Dewa C, Fadhilah A, Afiahayati A with the title "Convolutional Neural Networks for Handwritten Javanese Character Recognition" using the CNN and Multi-Layer Perceptron (MLP) architectures to compare the performance of Javanese script handwriting recognition. Conclusion from this research, the accuracy of handwriting recognition using CNN is higher than using MLP. Although in terms of the time required for training, CNN requires a longer time than MLP [6].

So in this research a comparison will be made between two CNN models. This research was conducted with the aim of knowing the accuracy and error rate of the convolutional neural network in recognizing hanacaraka handwriting.

II. Methodology

A. Javanese Script (Hanacaraka)

The Javanese script, known as Hanacaraka, or Carakan, is one of the traditional Indonesian scripts that is developing and actively used on the island of Java, especially in Central and East Java [7], and is a derivative of the Brahmi script that originated in India [1]. The existence of the Javanese script cannot be separated from ancient legends. The legend comes from a character named Ajisaka who wrote poetry for his two servants [1]. From this poem, a basic Javanese script was created which has 20 kinds of characters as shown in Figure (1) [6].

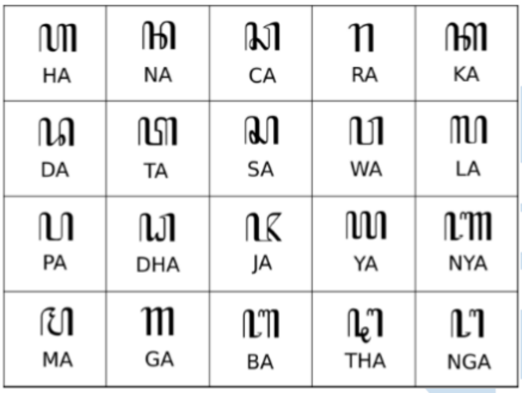

Fig. 1. Basic Javanese Script (Hanacaraka)

B. Handwriting Recognition System

Handwriting Recognition is a system that can recognize handwriting and convert it into text that can be read and understood by computers. There are two types of handwriting recognition, Offline handwriting recognition which is recognition handwriting derived from an image such as a scanned human handwriting, and Online handwriting recognition which is recognition handwriting derived from digital writing written using a touchpad or touch screen [8] [9]. The Handwriting Recognition process consists of preprocessing, feature extraction, and classification [10].

C. Convolutional Neural Network

Convolutional Neural Network (CNN) is a deep learning architecture that works based on multi-layer perceptron (MLP) [10]. The CNN architecture consists of a special layer for extracting the input image with a network model connected to the logistic regression classifier [6]. Weights, parameters, and bias play a role in the transformation from the original image to the vector to find out about the nature of the image during the training period. So that CNN can properly recognize the patterns contained in the image, the processed image will go through several stages. The process of CNN is divided into 3 parts, namely as follows [11]:

1) Convolutional Layer

The convolution stage is a filtering stage in the image using a kernel in the form of a matrix which will be multiplied by each selected pixel value. The mathematical representation of the convolution layer is contained in Equation (1) [12].

$$z_{i,j,k}^{1} = w_k^1 * x_{ij}^1 + b_k^1 \quad (1)$$

where $z_{i,j,k}^{l}$ is the result of convolution, where w_k^l is the weight, x_{ij}^l is the coordinate point value (i, j) in the lth layer, and b_k^l is the bias value of the kernel in l layer. After the convolution operation process, the convolution results will be processed using the activation function. The activation functions used are sigmoid, ReLU, tanh [13] [14].

2) Pooling Layer

The pooling layer is a layer that aims to process data to be smaller by reducing dimensions or reducing pixels in convolved images in order to speed up data processing in the next layer. There are 2 types of pooling layers, namely average pooling and max pooling. Between the two pooling, what is often used is max pooling [15].

3) Fully-Connected Layer

Fully-Connected Layer is the last layer on CNN, which is a neural network that is fully connected with the final result of the pooling layer being converted into a vector [15]. Then the output from the fully-connected layer will pass through the activation function, namely the softmax activation function [8]. The representation of the softmax activation function is contained in Equation (2):

$$s(x_n) = \frac{e^{x_n}}{\sum_{k=1}^{K} e^{x_k}} \quad (2)$$

where $s(x_n)$ is the probability value of each class that belongs to, e^{x_n} is the exponential value of x from the nth class, and $\sum_{k=1}^{K} e^{x_k}$ is the sum of all the exponentials of fully-connected output layer [6].

D. System Development Plan

In the development of a system required planning or design stages of the application to be made. Here are the steps needed:

1) Data Collection

The data needed for this system is a basic hanacaraka dataset or collection of images that will be used for the training and testing process in the form of open-source datasets obtained from Kaggle and additional personal handwriting from researchers, where the dataset is in the form of basic hanacaraka handwritten images consisting of the 20 typeface characters. Each character consists of 150 images. As much as 80 percent of the dataset will be used for training and as much as 20 percent will be used for testing.

2) CNN Model Design

In architectural design, there are 2 types of network architecture from the CNN model that will be used for this training process. Table (1) represents model A, with the simplest architecture consisting of only two convolution layers. Table (2) represents model B, with a higher number of convolution layers.

TABLE I. CNN ARCHITECTURE DETAILS OF MODEL A

Layer	**Size**	**Output Dimension**	**Parameter**
Input	(100, 100, 1)	-	
Convolution 2D + ReLU	32 (3 x 3) *filters; padding* "same"	(None, 100, 100, 32)	320
Pooling	(2 x 2) *max pool*	(None, 50, 50, 32)	
Convolution 2D + ReLU	48 (3 x 3) *filters; padding* "same"	(None, 50, 50, 48)	13872
Pooling	(2 x 2) *max pool*	(None, 25, 25, 48)	
Flatten		(None, 25 * 25 * 48)	
Dense + ReLU	256 *perceptrons*	(None, 256)	7680256
Dropout (rate=0.5)		(None, 256)	
Dense + Softmax	21 *perceptrons*	(None, 21)	5397
Total Trainable Parameters			7699845

TABLE II. COMPARISON BASED-ON ACCURACY

Layer	**Size**	**Output Dimension**	**Parameter**
Input	(100, 100, 1)	-	
Convolution 2D + ReLU	16 (3 x 3) *filters; padding* "same"	(None, 100, 100, 16)	160
Pooling	(2 x 2) *max pool*	(None, 50, 50, 16)	
Convolution 2D + ReLU	32 (3 x 3) *filters; padding* "same"	(None, 50, 50, 32)	4640
Pooling	(2 x 2) *max pool*	(None, 25, 25, 32)	
Convolution 2D + ReLU	32 (3 x 3) *filters; padding* "same"	(None, 25, 25, 32)	9248
Pooling	(2 x 2) *max pool*	(None, 12, 12, 32)	
Convolution 2D + ReLU	64 (3 x 3) *filters; padding* "same"	(None, 12, 12, 64)	18496
Pooling	(2 x 2) *max pool*	(None, 6, 6, 64)	
Convolution 2D + ReLU	64 (3 x 3) *filters; padding* "same"	(None, 6, 6, 64)	36928
Pooling	(2 x 2) *max pool*	(None, 3, 3, 64)	
Flatten		(None, 576)	
Dropout (rate=0.5)		(None, 576)	
Dense + ReLU	128 *perceptrons*	(None, 128)	73856
Dense + Softmax	21 *perceptrons*	(None, 21)	2709
Total Trainable Parameters			146037

3) System Planning

The main feature of this application is the character analyzer feature, and the quiz feature. Here is a detailed description of the main features of this application:

- *Character Analyser*

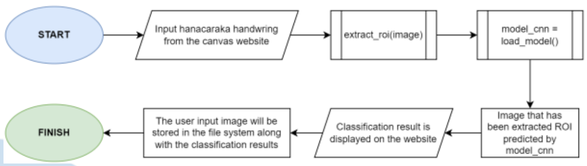

Fig. 2. Flowchart of handwriting image classification on canvas

Based on Figure (2) the handwriting written by the user on the canvas will be the input image into the system. After that, the image will go through a region of interest (ROI) search process from digital writing. The extracted image based on the ROI coordinates will be extracted for its features using the CNN model. Before making predictions, the weights of the CNN model will be entered into the system first. The output from CNN is in the form of predictions or classification results which are Javanese character class classifications from the image being analyzed. The digital writing image written by the user will be stored in the file system in the folder that matches the predicted character.

- *Handwriting Quiz*

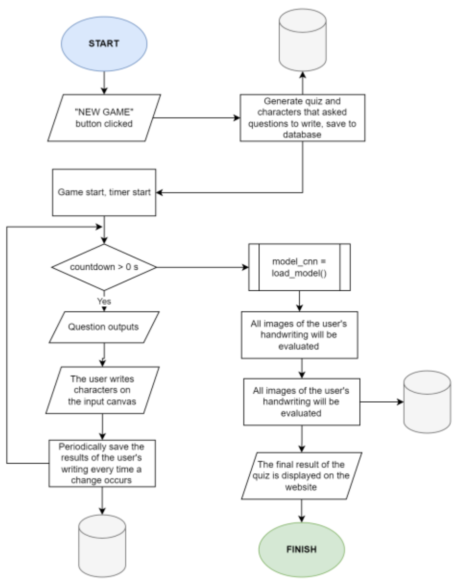

Fig. 3. Flowchart handwriting quiz

In Figure (3) the game will start when the user presses the "Start Game" button. After that the questions will be randomly generated, and the time system will run. Information about quiz questions, and the number of countdowns will be stored in the database. Users can fill in answers and change answers as long as the countdown count is still greater than 0. Any changes to answers that occur will be stored periodically by the system and update the information in the database. If the user has completed all the questions, then the user can submit answers. Quiz will auto-submit if the allotted time has expired. After that each answer will be evaluated using the CNN model, then the system will count the number of answers that were written correctly and the results will be stored in a database which will be displayed as game history.

- *Guess Word Quiz*

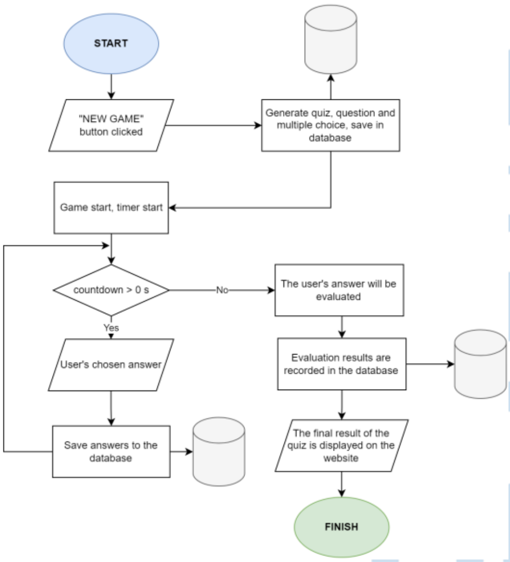

Fig. 4. Flowchart guess word quiz

In Figure (4) the game will start when the user presses the "Start Game" button. After that the questions and choices will be randomly generated, and the time system will run. Information about quiz questions, answer choices and countdown numbers will be stored in the database. Users can fill in answers and change answers as long as the countdown count is still greater than 0. All answers selected by the user will be directly stored in the database. If the user has completed the quiz, then he can submit an answer. If the given countdown time has ended, the system will auto submit. After that, all answers from users will be evaluated. The system will count the number of questions answered correctly, and the evaluation results will be stored in the database. User quiz results will be displayed on the history page on the website.

4) ERD

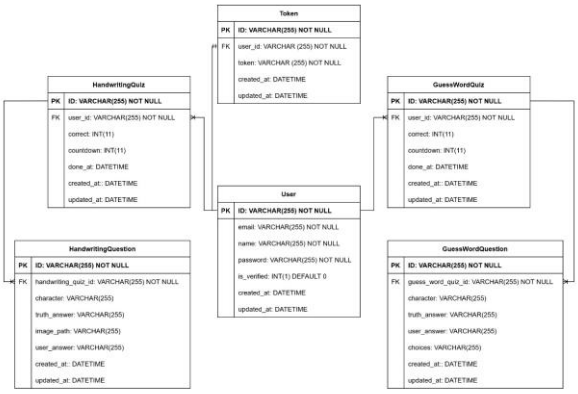

Fig. 5. Entity Relationship Diagram of the system

The designed application uses a database to store user data. The database used is the mySQL database. The following is a table used in the application where each table has an id attribute which is the primary key, the created_at and updated_at attributes are standard attributes for each entity to record when the data row was created and last modified:

- *User Table*
  The user table consists of the attributes name, email, password, and is_verified which is a sign to know whether the account has been verified.

- *Token Table*
  The token table consists of the user_id attribute which is a foreign key and has a relationship with the user table, and the token attribute which contains the token generated by the system for user account verification.

- *HandwritingQuiz Table*
  The handwriting quiz table consists of the user_id attribute which is a foreign key and has a relationship with the user table, the correct attribute to store the number of correct answers to quiz questions, the countdown attribute to store the remaining time to complete the quiz, and the done_at attribute to record when the quiz was completed by the user.

- *HandwritingQuestion Table*
  The HandwritingQuestion table consists of the handwriting_quiz_id attribute which is a foreign key and has a relationship with the HandwritingQuiz table, the character attribute which stores the Javanese characters in question, the truth_answer attribute to store answers to questions, the image_path attribute to store the storage location of each handwritten image from the user, and the user_answer attribute to store user answer data. The HandwritingQuiz table has a one-to-many relationship with the HandwritingQuestion table.

- *GuessWordQuiz Table*
  The guesswordquiz table consists of the user_id attribute which is a foreign key and has a

relationship with the user table, the correct attribute to store the number of correct answers to quiz questions, the countdown attribute to store the remaining time to complete the quiz, and the done_at attribute to record when the quiz was completed by the user.

- *GuessWordQuestion Table*
  The GuessWordQuestion table consists of the guess_word_quiz_id attribute which is a foreign key and has a relationship with the GuessWordQuiz table, the character attribute which stores the Javanese characters in question, the truth_answer attribute to store answers to questions, the user_answer attribute to store user answer data, and the choices attribute to store choices. answers to quiz questions. The GuessWordQuiz table has a one-to-many relationship with the GuessWordQuestion table.

III. Results and Discussion

This section covers the interface implementation of the application, and the test results of the system that has been developed

A. *Interface Implementation*

The interface of this application consists of a login page, sign-up page, verification page, home page (character analyzer), handwriting history page, handwriting quiz page, guess word history page, guess word quiz page. The following is the result of the display of the main features of this application:

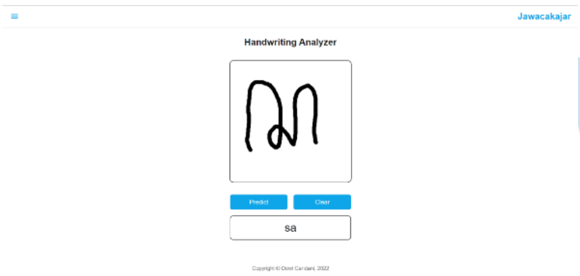

Fig. 6. Home Page Display (character analyzer)

On the home page as shown in Figure (6) there is a canvas to write the basic hanacaraka that will be predicted, the predict button to predict the posts written by the user, and the predicted results will appear in the column below the predict and clear buttons, as well as the clear button to delete the writing on canvas and in the predicted output column.

Fig. 7. Handwriting Quiz view

On the handwriting quiz page as shown in Figure (7) there are 5 questions which if clicked will display the questions as well as the canvas containing the user's answers (canvas write hanacaraka), the canvas where the user's answers are written (canvas write here), the clear button to delete the answers on the canvas write here, and there is also a submit answer button that will appear when all the questions have been answered.

Fig. 8. Guess Word Quiz View

On the guess word quiz page as shown in Figure (8) there are 10 questions which if clicked will display the questions and their choices. There is also a submit answer button that will appear when all the questions have been answered.

B. *Method Test Results*

Tests were carried out to determine the performance of the developed CNN model. Where there are several factors that will affect how the performance of the CNN model in classifying. These factors are the number of layers of the network, the type of pooling layer used, and so on. There are also other factors such as hyperparameters such as learning rate and epoch.

In this test it will be seen how accuracy and loss are based on the results of the training process. Testing the CNN model will be carried out using the data that has been collected which consists of 20 basic hanacaraka character classes and 1 unknown class (characters that are not basic hanacaraka). Based on the CNN model that has been designed, the results are obtained in the form of accuracy values, and losses from the training and testing processes as shown in the table in the form of Figure (9).

No.	Model	LR	Epoch	Accuracy	Loss	Testing Accuracy	Testing Loss	Training Time
1	A	1.00E-03	30	0.9578	0.1257	0.9441	0.1603	384 seconds
2	**B**	**1.00E-03**	**30**	**0.9815**	**0.0496**	**0.9940**	**0.0570**	**341 seconds**
3	A	2.00E-03	30	0.9525	0.1478	0.9521	0.1707	409 seconds
4	B	2.00E-03	30	0.9688	0.1010	0.9840	0.0755	361 seconds
5	A	1.00E-03	50	0.9696	0.0866	0.9641	0.1692	745 seconds
6	B	1.00E-03	50	0.9886	0.0361	0.9780	0.0900	914 seconds

Fig. 9. Accuracy and Loss Value in the Training and Testing Process

From several training processes, it was found that the loss value was relatively low compared to other training trials, namely in the 2nd and 6th training, which is model B. From these results, model B shows relatively better performance based on the loss value and its accuracy if compared to model A.

The most important factors that affect the learning capacity of a deep learning model are the number of layers and the number of perceptrons that a model has. A model that has a sufficient number of layers and perceptrons will produce good performance. The more layers that can represent data, the more feature information will be extracted. This is the reason model B has better performance than model A.

Of the two candidate models trained, namely the 2nd and 6th training trial model, the training results from the 2nd model will be selected to carry out the hanacaraka classification process on the system of the application to be developed. Figure (10) is a graph of accuracy and loss from the training results of the 2nd experimental model and Figure (11) is a graph of accuracy and loss from the training results of the 6th experimental model.

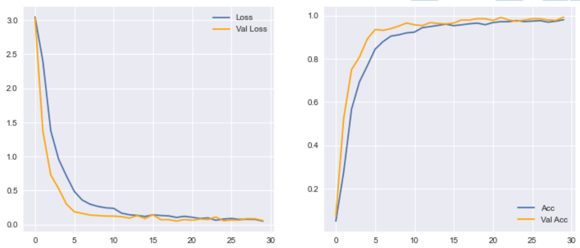

Fig. 10. Graph of loss (left) and accuracy (right) from the results of the 2nd training

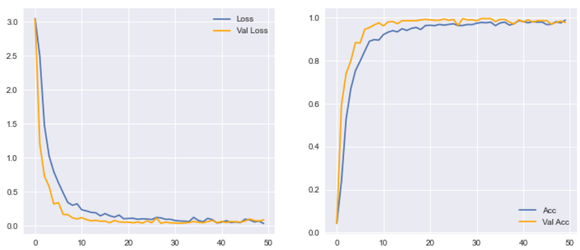

Fig. 11. Graph of loss (left) and accuracy (right) from the results of the 6th training

If you look at the graph in Figure (11), it can be seen that the accuracy and loss values during the training epoch have begun to stabilize and stagnate when entering the 30th epoch, and there are no longer significant changes in the accuracy and loss values. Therefore, if the training process is continued, it has a high potential to cause the model to "memorize" the given training dataset which in the end if it is continued it can cause overfitting. A significant decrease in the loss value can be seen in the epoch range from the 1st to the 10th epoch, and then the decreasing gradient starts to slope from the 10th to the 20th epoch. Due to the reasons above, for this study it was concluded that the CNN model from the results of the 2nd training experiment will be used in the Javanese character recognition system which will be implemented in learning applications.

Calculation of the accuracy and error rate of the CNN model in handwriting classification will use a confusion matrix for multiclass classification cases. To find out the accuracy and error rate of a model, you can do it by comparing the results of the predicted class with the actual class. All details regarding the true positive, true negative, false positive, false negative, accuracy and error rate of each class of character classification results from the 2nd training result model can be seen in Table (3), and the 6th training in Table (4).

TABLE III. ACCURACY AND ERROR RATE OF EACH CHARACTER CLASS CLASSIFICATION RESULTS OF THE 2ND TRAINING MODEL

Lable	TP	TN	FP	FN	Akurasi	Error
ba	24	478	2	0	0.996031746	0.003968254
ca	24	480	0	0	1	0
da	24	478	2	0	0.996031746	0.003968254
dha	24	480	0	0	1	0
ga	24	479	1	0	0.998015873	0.001984127
ha	19	479	1	5	0.988095238	0.011904762
ja	22	480	0	2	0.996031746	0.003968254
ka	24	476	4	0	0.992063492	0.007936508
la	24	477	3	0	0.994047619	0.005952381
ma	24	480	0	0	1	0
na	19	480	0	5	0.990079365	0.009920635
nga	24	480	0	0	1	0
nya	24	480	0	0	1	0
pa	24	474	6	0	0.988095238	0.011904762
ra	21	480	0	3	0.994047619	0.005952381
sa	24	480	0	0	1	0
ta	24	478	2	0	0.996031746	0.003968254
tha	24	480	0	0	1	0
unknown	22	477	3	2	0.990079365	0.009920635
wa	16	480	0	8	0.984126984	0.015873016
ya	24	479	1	0	0.998015873	0.001984127

TABLE IV. ACCURACY AND ERROR RATE OF EACH CHARACTER CLASS CLASSIFICATION RESULTS OF THE 6TH TRAINING MODEL

Label	TP	TN	FP	FN	Akurasi	Error
ba	24	480	0	0	1	0
ca	24	480	0	0	1	0
da	24	480	0	0	1	0
dha	24	479	1	0	0.998015873	0.001984127
ga	23	480	0	1	0.998015873	0.001984127
ha	19	480	0	5	0.990079365	0.009920635
ja	24	480	0	0	1	0
ka	23	479	1	1	0.996031746	0.003968254

la	20	479	1	4	0.990079365	0.009920635
ma	23	480	0	1	0.998015873	0.001984127
na	21	479	1	3	0.992063492	0.007936508
nga	22	480	0	2	0.996031746	0.003968254
nya	23	480	0	1	0.998015873	0.001984127
pa	24	469	11	0	0.978174603	0.021825397
ra	23	479	1	1	0.996031746	0.003968254
sa	23	479	1	1	0.996031746	0.003968254
ta	23	480	0	1	0.998015873	0.001984127
tha	24	480	0	0	1	0
unknown	22	467	13	2	0.970238095	0.029761905
wa	19	480	0	5	0.990079365	0.009920635
ya	21	479	1	3	0.992063492	0.007936508

From the calculation of accuracy, error rate, true positive, true negative, false positive, and false negative values of each class resulting from the 2nd and 6th training model classification, it can then be calculated average accuracy, average error rate, overall accuracy, and overall error rate as a metric from the confusion matrix which will be used in this test to compare the performance of the two models. The results of calculating the metrics from the confusion matrix used can be seen in Table (5).

TABLE V. CONFUSION MATRIX METRICS FROM TESTING CLASSIFICATION MODEL RESULTS OF THE 2ND AND 6TH TRAINING

Test No	Average Accuracy	Average Error Rate	Overall Accuracy	Overall Error Rate
2	**0.9952**	**0.0048**	**0.9503**	**0.0497**
6	0.9941	0.0059	0.9385	0.0615

Based on Table (5), it can be seen that the classification performance of the 2nd training model has a higher overall accuracy rate, and a lower overall error rate when compared to the 6th training model. It is from these results that support the reason why the 2nd training result model was selected and implemented as a model for introducing basic Javanese script in the application being developed.

IV. CONCLUSIONS

Based on the tests that have been carried out, researchers can draw several conclusions, namely:

1. The use of deep learning methods is quite effective in solving complex classification cases such as the use of convolutional neural networks to develop basic hanacaraka character recognition systems.

2. The performance of the CNN model used in the application is obtained after various experiments with an accuracy of 98.29%, and a loss of 0.0746 on the training data. In data testing obtained an accuracy of 97.82%, and a loss of 0.0924.

3. Tests carried out using the multiclass confusion matrix produced an average accuracy value of 99.52%, an average error rate of 0.48%, an overall accuracy of 95.03% and an overall error rate of 4.97%.

REFERENCES

[1] A. Arismadhani, U. L. Yuhana, and I. Kuswardayan, “Aplikasi Belajar Menulis Aksara Jawa Menggunakan Android,” *Jurnal Teknik Pomits*, vol. 2, no. 1, 2013.

[2] A. M. Silva and R. Limongi, “Writing to Learn Increases Long-term Memory Consolidation: A Mental-chronometry and Computational-modeling Study of ‘Epistemic Writing,’” *J Writ Res*, vol. 11, no. vol. 11 issue 1, pp. 211–243, Jun. 2019, doi: 10.17239/jowr-2019.11.01.07.

[3] M. R. Alwanda, R. P. K. Ramadhan, and D. Alamsyah, “Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle,” *Jurnal Algoritme*, vol. 1, no. 1, pp. 45–56, Oct. 2020, doi: 10.35957/algoritme.v1i1.434.

[4] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, “Evolving Deep Convolutional Neural Networks for Image Classification,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, 2020, doi: 10.1109/TEVC.2019.2916183.

[5] K. Wróbel, M. Karwatowski, M. Wielgosz, M. Pietroń, and K. Wiatr, “Compression of Convolutional Neural Network for Natural Language Processing,” *Computer Science*, vol. 21, no. 1, 2020, doi: 10.7494/csci.2020.21.1.3375.

[6] C. K. Dewa, A. L. Fadhilah, and A. Afiahayati, “Convolutional Neural Networks for Handwritten Javanese Character Recognition,” *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 12, no. 1, 2018, doi: 10.22146/ijccs.31144.

[7] T. Matius Surya Mulyana, D. Widyaningrum, P. Studi Informatika, F. Teknologi dan Desain, U. Bunda Mulia Jl Raya Lodan, and J. Utara, “OCR HURUF JAWA DENGAN FITUR KODE RANTAI DAN LEVENSHTEIN DISTANCE.”

[8] R. Plamondon and S. N. Srihari, “Online and off-line handwriting recognition: a comprehensive survey,” *IEEE Trans Pattern Anal Mach Intell*, vol. 22, no. 1, pp. 63–84, 2000, doi: 10.1109/34.824821.

[9] P. C. Vashist, A. Pandey, and A. Tripathi, “A Comparative Study of Handwriting Recognition Techniques,” in *2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM)*, IEEE, Jan. 2020, pp. 456–461. doi: 10.1109/ICCAKM46823.2020.9051464.

[10] I. Khandokar, M. Hasan, F. Ernawan, S. Islam, and M. N. Kabir, “Handwritten character recognition using convolutional neural network,” in *Journal of Physics: Conference Series*, 2021. doi: 10.1088/1742-6596/1918/4/042152.

[11] Z. J. Wang *et al.*, “CNN Explainer: Learning Convolutional Neural Networks with Interactive Visualization,” *IEEE Trans Vis Comput Graph*, vol. 27, no. 2, 2021, doi: 10.1109/TVCG.2020.3030418.

[12] J. Briliantio, N. Santosa, G. Ardian, and L. Hakim, “Penerapan Convolutional Neural Network untuk Handwriting Recognition pada Aplikasi Belajar Aritmatika Dasar Berbasis Web,” *Jurnal Teknik Informatika Unika St. Thomas*, vol. 05, no. 02, 2020.

[13] M. Z. Alom *et al.*, “The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches,” Mar. 2018.

[14] Y. Tian, “Artificial Intelligence Image Recognition Method Based on Convolutional Neural Network Algorithm,” *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3006097.

[15] S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, “Conceptual Understanding of Convolutional Neural Network-A Deep Learning Approach,” in *Procedia Computer Science*, 2018. doi: 10.1016/j.procs.2018.05.069.