# Handwritten Javanese script recognition method based 12-layers deep convolutional neural network and data augmentation

**Ajib Susanto[1], Ibnu Utomo Wahyu Mulyono[1], Christy Atika Sari[1], Eko Hari Rachmawanto[1], De Rosal Ignatius Moses Setiadi[1], Md Kamruzzaman Sarker[2]**
[1]Department of Informatics Engineering, Dian Nuswantoro University, Semarang, Indonesia
[2]Department of Computing Science, University of Hartford, West Hartford, United States

**Article Info**

***Article history:***

Received Oct 17, 2022
Revised Oct 26, 2022
Accepted Dec 21, 2022

***Keywords:***

Convolution neural network
Data augmentation
Fourth keyword
Javanese script recognition
Small dataset

**ABSTRACT**

Although numerous studies have been conducted on handwritten recognition, there is little and non-optimal research on Javanese script recognition due to its limitation to basic characters. Therefore, this research proposes the design of a handwritten Javanese Script recognition method based on twelve layers deep convolutional neural network (DCNN), consisting of four convolutions, two pooling, and five fully connected (FC) layers, with SoftMax classifiers. Five FC layers were proposed in this research to conduct the learning process in stages to achieve better learning outcomes. Due to the limited number of images in the Javanese script dataset, an augmentation process is needed to improve recognition performance. This method obtained 99.65% accuracy using seven types of geometric augmentation and the proposed DCNN model for 120 Javanese script character classes. It consists of 20 basic characters plus 100 others from the compound of basic and vowels characters.



***Corresponding Author:***

Ajib Susanto
Department of Informatics Engineering, Dian Nuswantoro University
Semarang, Indonesia
Email: ajib.susanto@dsn.dinus.ac.id

## 1. INTRODUCTION

Indonesia is a country comprising numerous ethnic groups and various languages and cultures. One of the largest ethnic groups is the Javanese, who use the Javanese language originally written with the Javanese script. This language is currently rarely used by this ethnicity, therefore it needs to be preserved. Technology-based learning of the Javanese script is one way to re-popularize the writing of this language. This research proposed a highly accurate Javanese script recognition method. Many recognition methods have been proposed. Some are used for Javanese script recognition [1]–[4], as well as non-Latin languages, such as Arabic [5]–[7], Tamil [8], Bangla or Bengali [9]–[11], Kannada [12], Gurmukhi [13], Tifinagh [14], and Thai [15]. Non-Latin character recognition is usually more difficult due to limited research and datasets and the relatively complex shapes of the character. This is also proven in the study by [16] that certain algorithms have better accuracy when interpreting Latin characters than Javanese scripts.

Preliminary studies have been carried out on handwritten Javanese script recognition, such as those by [4] and [1]–[3], which are based on machine learning and deep learning, respectively. However, the results obtained are still unsatisfactory because they are limited to basic characters (*Carakan*). To make a good sentence with Javanese script, the basic (*Carakan*), vowels (*Sandhangan Swara*), and consonant scripts (*Sandhangan Panyigeg* and *Sandhangan Wyanjana*), including numbers, and punctuation, are required. The vowel, consonant, and basic scripts are used to turn off vocal reading. The vowel and consonant scripts are only used in the middle of words or sentences. The Javanese script is written from left to right, while that of *sandhangan* is different, i.e. left, right, top, and bottom. Figure 1 is an example of a typical script, lines 1 and 2 are Javanese scripts, while the subsequent one is a basic script compounded with vowels. This is because its recognition is more complicated than Latin characters. One of the most accurate studies on Javanese script recognition was carried out by [1], who proposed a convolutional neural network (CNN) method. This approach consists of three convolutional and pooling, as well as two fully connected layers, yielding a recognition accuracy of 94.57 percent for 20 basic Javanese scripts.

This study proposes a method to improve the recognition accuracy of Javanese script that is not limited to the basic script compounded with vowels using a deep convolutional neural network (DCNN) and data augmentation. Data augmentation is used to enrich the relatively small number of the dataset used in this research. This manuscript consists of five parts, namely section 1, the introduction. Section 2 centered on motivations and explained why DCNN and data augmentation were proposed, including related research, and the contributions were made. Meanwhile, section 3 describes the detailed steps of the proposed method. Sections 4 and 5 explain the results and analysis, including the implementation of the method and conclusion, respectively.

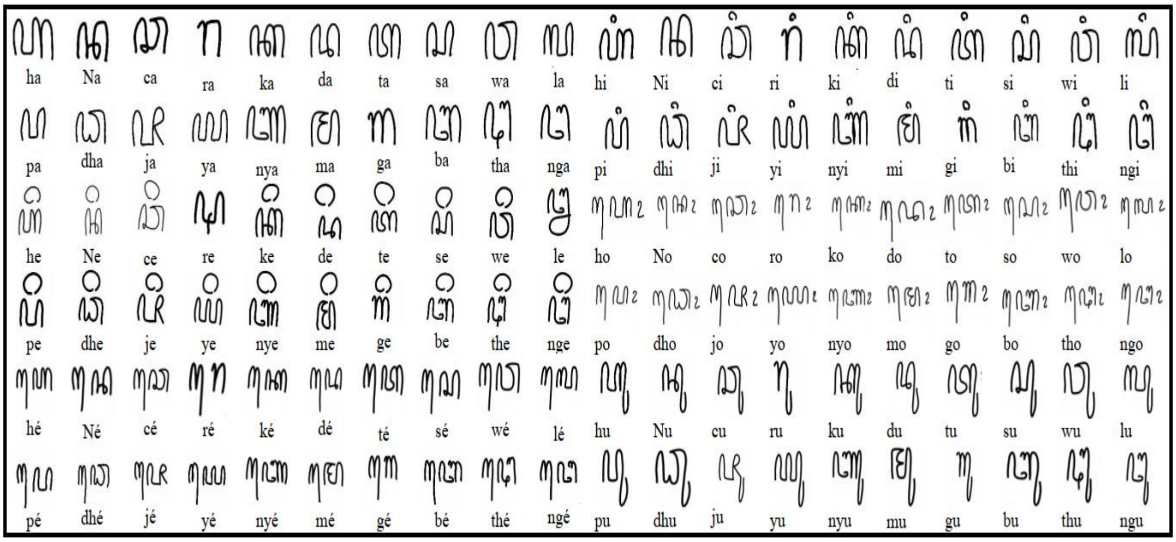



Figure 1. Javanese script characters

## 2. MOTIVATION AND CONTRIBUTION

Several studies on handwritten Javanese Script recognition have been carried out, including the one by [3]. It involved using several artificial neural network methods to recognize 20 basic characters, four vowel scripts, and seven numbers. The handwritten Javanese script image was initially read and converted to grayscale. Besides, several pre-processing procedures such as slope detection and correction are carried out and then segmented by thresholding and skeletonizing. After the area of the character has been obtained, it is divided into 4×5 zones, where feature extraction is carried out on each of them with the image centroid and zone (ICZ) as well as zone centroid and zone (ZCZ) methods. Additionally, 40 ICZ-ZCZ was realized and used for ANN input classification. There are several classification methods, such as the counter propagation network (CPN), backpropagation neural network (BPNN), and evolutionary neural network (ENN), as well as a combination of the Chi2 and BPNN, approaches. It was reported that these methods produced the best classification with an accuracy of 73.71%.

The research carried out by [4] used the k-nearest neighbor (KNN) classifier method combined with roundness and eccentricity feature extraction to recognize 20 basic Javanese scripts with relatively few datasets consisting of 240 images. However, the proposed method has an accuracy of approximately 87.5%. The performance of the recognition process is fairly good because the pre-processing stage consists of binarization, median filter, and dilation.

In the research carried out by [1], one of the deep learning algorithms employed to recognize 20 basic Javanese scripts is CNN, and 11,000 datasets were used. The proposed method is divided into two models. Model 1 consists of two convolutions, three pooling, and one fully connected layer. In contrast, model 2 uses similar layers and an additional fully connected layer at the end before the classification process. Each model was tested with 0.006 and 0.01 learning rates and 0.0005 and 1.00E-004 regularization. It was discovered that model 2, which uses a 0.01 learning rate and 0.0005 regularizations, had the best performance with an accuracy of 94.57%.

Based on some preliminary studies, the research on Javanese script recognition still has a great opportunity to be improved. Interestingly, recognition is mostly limited to basic characters, and the performance of its method still needs to be re-optimized. Considering the recognition of other scripts, such as Tamil and Bengali, which have similar writing systems with abugida, and are both derived from Brahmi script, the results tend to be better. In the research carried out by [8], the CNN method was used to recognize handwritten Tamil characters. A total of 82,929 images were extracted from the online version with linear interpolation and constant thickening factor and were normalized by resizing to 64×64. All the images were processed by the CNN algorithm, which consists of five convolution, two max-pooling, and fully connected layers. Several hyperparameters are also used in this method, namely initialization=Xavier, batch size=64, optimizer=Adam, epoch=100, learning rate 0.001 and activation function= rectified linear unit (ReLU). As a result, this approach has an accuracy of relatively 97.7% in terms of testing the data on 156 handwritten Tamil character classes.

In the research carried out by [10], the method for performing Bangla character recognition using DCNN and squeeze and excitation (SE)-ResNeXt, was proposed. The dataset used is BanglaLekha-Isolated (Biswas *et al.* 2017), which consists of 50 basic, 10 numeric, and 24 compound characters. The image in the dataset has a size of 150×150 to 185×185, which is further normalized and resized to 32×32 pixels. Additionally, all data are then processed using six process layers. The first is a 3×3 convolution block with 64 filters, the second layer consists of SE-ResNeXt Block-1 with 64 filters, and the third is a SE-ResNeXt Block-2 with 128 filters. The fourth, fifth and sixth are SE-ResNeXt Block-3, AVG global pooling, and fully connected layers. This approach has an accuracy of relatively 99.82%.

Another deep learning recognition method was used to decipher the Gurmukhi character by [13]. This research used a combination of both offline and online learning features to recognize Gurmukhi handwriting. A pre-training model was adopted in the learning architecture on offline data to classify images consisting of simple lines with classes. Therefore, only the lower-level layers were used to study low-level features in the image. The processed results are passed to two of the fully connected layer with 512 neurons, 40% dropout and a ReLU activation layer. The SoftMax activation layer is used in the output, while the root mean squared propagation (RMSprop) optimizer was adopted to perform multiclass classification. Three blocks of the CNN layer are used based on the online aspect. The first one has two 1D convolution layers with 64 filters and 1D max-pooling. The second block has two layers of 1D convolution with 128 filters and 1D max-pooling. The third has 1D convolution with 128 filters and 1D max-pooling. The CNN layers output is flattened before passing to the fully connected layer with 512 neurons and drops out by 30%. Like the offline aspect, the online aspect also uses the ReLU and SoftMax activation layers and RMSprop optimizer. The best accuracy was relatively 97.44%, with 90% training and 10% testing data based on the test results.

Another study that has similar objects is [17]. This research employed a combination of the multi augmentation technique (MAT), adaptive Gaussian thresholding, convolutional autoencoder (AGCA) and CNN to recognize Balinese script. Augmentation improves recognition performance on a relatively small dataset, namely 1197 Balinese character images written in the papyrus manuscript and 18 classes. The MAT-AGCA method produced 3159 datasets consisting of 2835 training, 216 validation, and 108 tests. This method has the highest accuracy of 96.29%, with MobileNetV2 as the pre-trained model. The augmentation model provides high accuracy, with recognition of 40.74%.

Based on related research, it was concluded that the deep learning method, especially convolution, has been proven to have excellent performance for handwriting recognition and various derivatives of the Brahmi script. Currently, preliminary studies on the Javanese script are limited to basic characters. Therefore, this research was carried out to optimize Javanese script recognition accuracy by designing an appropriate DCNN model. This study recognizes the basic characters, compound vowels script, and 120 classes. The number of classes is relatively much more than the previous Javanese script recognition research. The dataset used is quite limited, and a data augmentation process was carried out to improve learning performance in this research.

## 3. PROPOSED METHOD

The research proposes a recognition method that uses DCNN as the main algorithm. This approach has proven to have good performance in various image classification process, especially for handwritten, printed, and digital text recognitions, both in modern Latin characters and traditional scripts of various languages [5], [6], [14], [18]–[24]. Before carrying out the convolution and learning processes, the image dataset is pre-processed to ensure accuracy, including the grayscaling, cropping, negative image, resizing, and data augmentation processes. Data augmentation is carried out to ensure the datasets vary. Besides, it is conducted to improve the classification accuracy performance [17], [25]–[29]. Figure 2 shows the method proposed in this research, further described in detail in subsections 3.1 to 3.3.

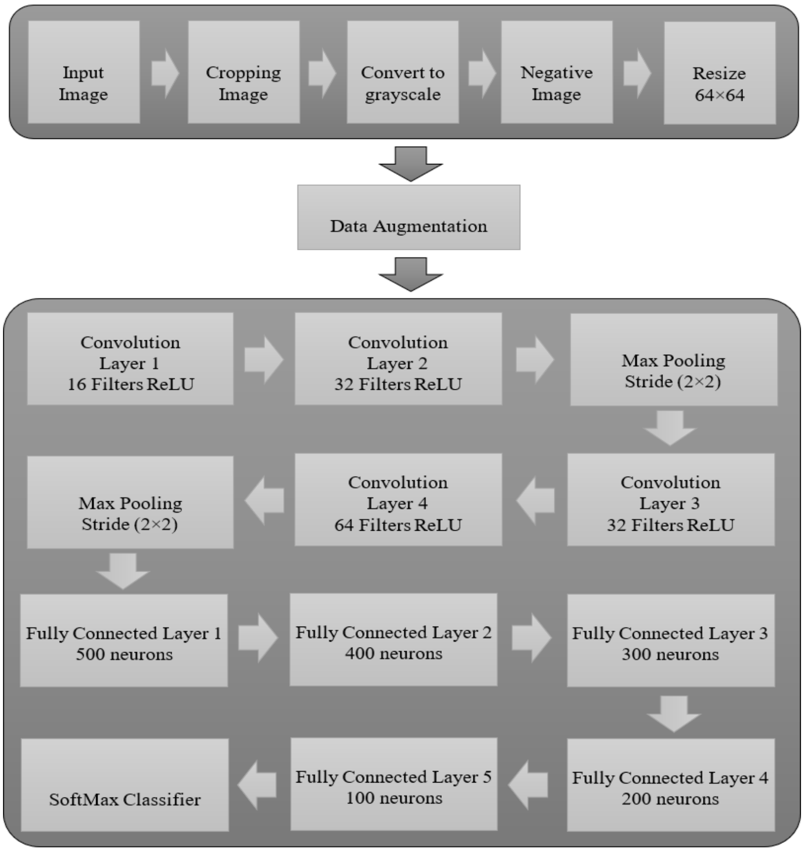



Figure 2. Proposed method

### 3.1. Pre-processing

The image datasets used in this research need to be normalized to improve the classification performance. Several processes are carried out in the pre-processing stage, and the first is the conversion to grayscale. This simplifies the image and reduces computational complexity because the calculations are only carried out on one layer. The handwritten image is relatively not concerned with color features in terms of deciphering its meaning because the writing patterns only consist of lines and dots. The text could be any color, but one that contrasts with the background is recommended. The second aspect is the image cropping process. This procedure has a square shape with a size of N×N. It aims to reduce the empty writing area and not change its shape during the resizing process. The third is converting the image to a negative one. This is carried out because there is a binarization procedure in most segmentation processes where the object and its background are generally converted to 1 (white) and 0 (black), respectively. This concept is also widely applied to the recognition methods to change the images to their complementary form [4], [19], [30]. The four of them are resized to 64×64 pixels. The aim is to reduce the computational process considering that deep learning requires expensive resources and computations.

### 3.2. Data augmentation

Data augmentation processes such as recognition, and classification, are mostly employed in the learning process. The goal is to increase the dataset to learn more, thereby improving the accuracy performance. This is chiefly carried out in various deep learning processes that have relatively small datasets, such as in research [17], [26], [31], [32]. Image augmentation can be performed in various ways, namely rotation, scaling, width, and height shifts, filtering, flipping, stretching, squeezing, affine, and projection transformation, gamma correction, noise injection, and color augmentation. [17], [33], [34]. It can also be performed during various image processing. Although, in recognition of handwritten objects, some augmentations are not used in this research because they have the potential to reduce recognition performance and alter the meaning of the writing, noise injection, flipping, and rotation to a certain degree.

In contrast, noise injection and gamma correction are not used because the image dataset was directly taken from the smartphone notepad handwriting application to ensure it has good quality. The augmentation processes used in this research are rotation, scaling, affine and projection transformations, and squeezing.

However, these are liable to change the image's size, and this led to the carrying out of several normalization procedures afterwards. In a more detailed manner, each augmentation process is performed, as shown in Table 1. The $t$ value in Table 1 is the transform form matrix used in geometric transformation. In this type of augmentation, rotation and squeezing produce two images. This is due to the different rotation directions, whereas shifting and squeezing depend on the width or height. Therefore, seven types of augmentation are used in this study.

Table 1. Data augmentation details

| Augmentation Type | Specification |
|---|---|
| Affine 2D-transform | $t = [1\ 0.3\ 0; 0.1\ 1\ 0; 0\ 0\ 1]$ |
| Projective 2D-transform | $t = [1\ 0 - 0.002; 0.3\ 1\ -0.0002; 0\ 0\ 1]$ |
| Rotation | 3° and -3° |
| Scaling | From 64 pixels to 54 pixels in width and height |
| Squeezing | From 64 pixels to 44 pixels in width or height |

### 3.3. DCNN model

After the pre-processing and augmentation stages, the dataset is divided into two groups, namely training and testing with DCNN. This is feature extraction and deep learning method widely used in image recognition and classification [11], [35], [36]. Various DCNN models have been proposed, while the one used in this study has 12 layers. It consists of four layers in which every two convolution layers (C) are inserted in the max-pooling (MP), followed by five fully connected (FC) and softmax classifier (SC) layers. The first six layers, namely 2-C, 1-MP, 2-C, and 1-MP, are used to perform feature extraction, in which the two initial convolution layers are given a dropout value of 0.2 with 16 and 32 filters, respectively. The MP layer uses the feature map function to reduce output size and control overfitting. Next, two more convolution layers were performed with 32 and 64 filters with a dropout value of 0.3 each, followed by an MP layer. Each convolution layer uses the ReLU activation function to perform thresholding, where every $(x)$ value less than 0 is converted to 0 and is calculated using (1) [14].

$$f(x) = max\ (0, x) \tag{1}$$

The FC layer processes the data, which aims to transform and classify its dimensions linearly. Each neuron in the convolution layer needs to be transformed into one-dimensional data before it can be entered into an FC layer. Meanwhile, five FC layers were proposed in this research to carry out the learning process in stages to achieve better outcomes. Each FC layer is given a dropout value of 0.2, 0.3, 0.3, 0.2, and 0.2, whose names are generated from the best test results. This is inspired by several studies utilizing multiple FC layers to maximize learning. The data is entered into the SoftMax classifier to obtain the recognition results in the last stage. SoftMax classifier was selected because it provides more intuitive results and is also used to obtain a good probabilistic interpretation. SoftMax is used to calculate the probabilities for all labels. From the existing ones, a vector was taken and converted into a one with a value between zero and one, which, when added up, is equivalent to one. Additionally, it needs to be noted that the proposed DCNN model is combined with an adaptive moment estimation (ADAM) optimizer with a learning rate of 0.001.

### 3.4. Evaluation

The proposed method was evaluated using several stages. The first one compares the recognition process based on the split ratio between the training and testing data. Besides, three split ratios were used, namely 70:30, 80:20, and 90:10. After obtaining the best, several optimizers were evaluated to prove that the selected one is the best for the proposed model. Two popular comparison optimizers, namely root mean square propogation (RMSprop) and stochastic gradient descent (SGD), were compared to ADAM. The last evaluation was carried out by changing the classifier with several popular ones, including reducing and adding the number of FC layers used. This proves that the proposed method uses the most optimal classifier.

## 4. RESULTS AND DISCUSSION

This study employed a private handwritten Javanese script dataset using a notepad application. The Javanese script consists of 120 classes constituting basic characters (*Carakan*) and those compounded with vowel scripts (*Sandhangan Swara*), namely e, é, i, o, and u. Meanwhile, the vowel a has been integrated into the basic character, as shown in Figure 1. The dataset used consists of 480 images written with two different text thickness levels, which are written in bolder as Figure 3(a) and thinner as Figure 3(b). Next, several preprocessing steps were carried out, namely cropping, converting a grayscale, negative image, and resizing to produce a size of 64×64, which are respectively shown in Figures 4(a) to 4(c).

(a) (b)

Figure 3. "*Ha*" sample character of Javanese script (a) written with a thick line and (b) written with a thin line

(a) (b) (c)

Figure 4. Sample pre-processing results (a) cropped image, (b) complement image, and (c) resized image

After the pre-processing stage, several image augmentations, such as affine and projective 2-dimensional transforms, resizing 10 pixels smaller, squeezing width and height, and rotating 3° and -3°. It is important to note that some of these processes cause changes in size, namely affine, projective 2-dimensional transforms, and rotation. In this case, the augmented image is resized to 64×64. In the resize augmentation process, because the size is reduced by 10 pixels, then 5 paddings are added above, below, left, and right, with a value of zero. In the squeezing of width, the image is compressed vertically from 64 pixels to 44 pixels, enabling 10 pixels of padding to be added on the right and left. This is also performed for the squeezing height, although only horizontal compression is employed in this case. Figures 5(a) to 5(g) shows sample image augmentation results.

(a) (b) (c) (d) (e) (f) (g)

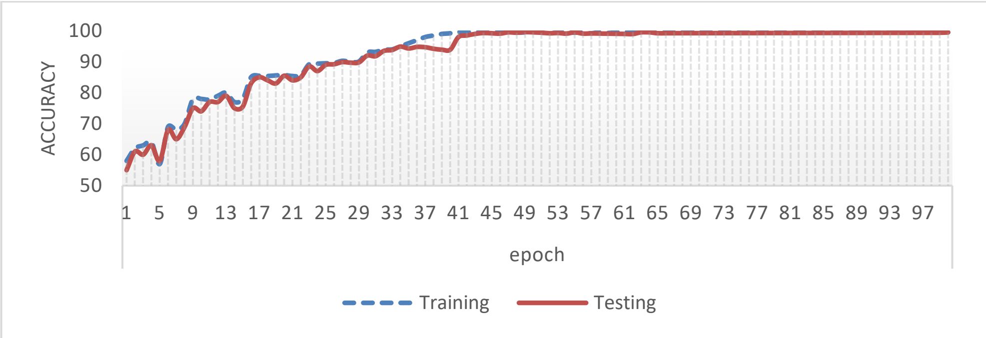
Figure 5. Sample Augmentation results (a) afine2D, (b) projective2D, (c) resize, (d) rotate 3°, (e) rotate -3°, (f) squeezing width, and (g) squeezing height

With 480 original images added to each of the seven augmented ones, 3,360 datasets were obtained. In the next stage, the recognition process is carried out with DCNN. The proposed method, as shown in Figure 2, comprises some tuning hyperparameters. The testing process was carried out severally to obtain different hyperparameter values, as shown in Table 2. In the training and testing processes, the dataset is decomposed into two parts, namely training and testing data. The composition of training data and testing data include 70%:30%, 80%:20%, and 90%:10%. Based on this, the most accurate result of 80:20 processes were obtained from 100 epochs, as shown in Figure 6.

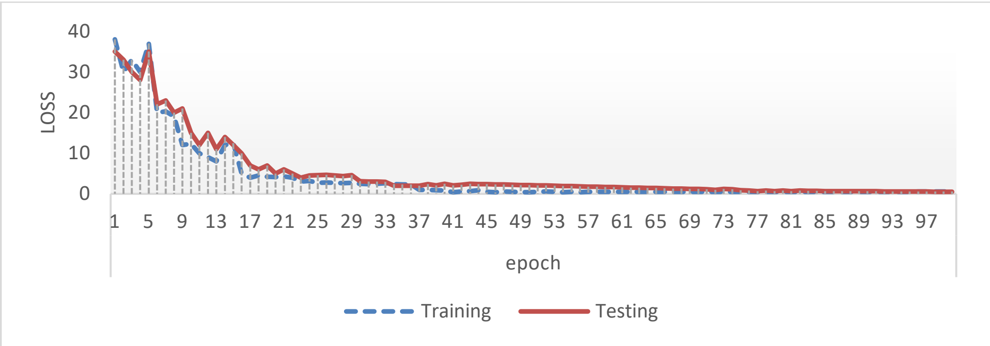
Table 2. Summary of tunning hyperparameters

| Hyperparameters | Tested Values | Optimal Values |
|---|---|---|
| Input image Dimension | 64×64 | 64×64 |
| Optimizer | Adam, RMSprop, SGD | Adam |
| Dropout | 0.2, 0.3, 0.4 | Combination 0.2 and 0.3 |
| Activation functions | ReLU, Sigmoid | ReLU |
| Batch Size | 32×32, 64×64 | 64×64 |
| Learning Rate | 0.0001, 0.001, 0.1 | 0.001 |

Figure 6(a) shows that the maximum accuracy generated from the training and testing data are 99.73% and 99.65%, respectively. Meanwhile, the minimum loss generated from the training and testing data are 2.01%, and 3.1%, respectively, see Figure 6(b). These results prove that the proposed method is effective without overfitting for Javanese Script recognition. In a more detailed manner, the recognition results based on the split ratio are shown in Table 3.




(a)




(b)

Figure 6. Recognition results (a) accuracy and (b) loss

Table 3. Accuracy and loss recognition results based on split ratio

| Split Ratio | Accuracy (%) | | Loss (%) | |
|---|---|---|---|---|
| | Training | Testing | Training | Testing |
| 70:30 | 98.88 | 98.13 | 1.31 | 1.95 |
| 80:20 | 99.73 | 99.65 | 0.23 | 0.14 |
| 90:10 | 99.27 | 99.48 | 0.71 | 0.56 |

Table 3 shows that the 80:20 split ratio performed on the data was used to obtain the most accurate results. Further analysis was also carried out to examine the accuracy and extent of influence on the augmented data. Please note that the method's accuracy was relatively 88.95% before the data augmentation was used. This is because the dataset is too small, consisting of four samples for each character, thereby leading to a total of 480 images. Furthermore, a split ratio of 75:25 is used, representing three training and one test data. It was concluded that the enlarged data significantly affects the recognition accuracy of this small dataset. As stated in section 3, the ADAM optimizer was used in this research, and further tests were also carried out on two other widely-used optimizers, namely RMSprop and SGD. Figure 7 shows that the accuracy results are pretty different. Although the same learning rate was not employed, the ADAM and RMSprop employed a learning rate of 0.001, while the SGD used 0.1. The following learning rate was selected based on the best trial values of 0.0001, 0.001, and 0.1. Some comparisons were made to prove that the proposed method has a good performance by changing the classifier, such as support vector machine (SVM), random forest (RF), multilayer perceptron (MLP), and modifying the layer utilized. Recognition experiments were carried out using several other approaches to test the effectiveness of CNN feature extraction and compare the classifier performance. Table 4 shows the comparison of the recognition results.

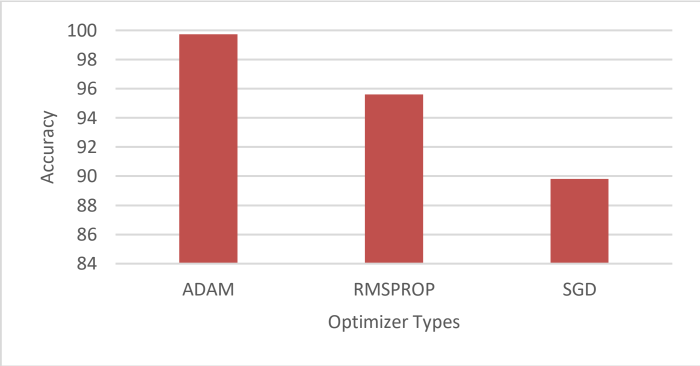



Figure 7. Comparison of optimizer used

Table 4. Comparison of recognition results with different methods

| Method | Accuracy (%) |
|---|---|
| Random Forest (RF) | 85.00 |
| Multilayer Perceptron (MLP) | 81.00 |
| Support Vector Machine (SVM) | 88.00 |
| Proposed Model with 1 FC Layer (500 Neurons) | 95.23 |
| Proposed Model with 3 FC Layer (500, 300, and 100 Neurons) | 96.35 |
| Proposed Model with 7 FC Layer (500, 400, 300, 200, 150, 100, 50 Neurons) | 98.75 |
| Proposed Model without Data Augmentation | 88.95 |
| Proposed Model witth Data Augmentation | 99.65 |

It should be noted that the results of the comparison in Table 3 were all obtained using augmentation data with the same split ratio of 80:20. The results of the proposed model had the best accuracy. This is because more FC layers are used to smooth the learning stage. In addition, dropouts at each FC layer tend to reduce overfitting and improve recognition accuracy. Afterwards, some commonly used CNN models, such as Alex Net with five convolutional layers and VGGNet-16, were also compared. AlexNet and VGGNet were compared because these CNN models are effectively used for various classification processes. Based on Table 5, the tested method has similar accuracy to the two previous CNN models. The proposed method has a training process speed that is much faster than the two CNN models. This shows that it has an excellent performance. Additionally, the results obtained using the proposed method are consistent with several previous studies on Handwritten Javanese Script Recognition shown in Table 6. Based on these results, few methods use datasets with 120 classes 120. The accuracy obtained in this method is better. This is influenced by a combination of deep learning and augmentation methods. With fewer datasets, the best accuracy is obtained.

Table 5. Comparison of recognition results with different methods

| CNN Model | Accuracy (%) | Training Time (in seconds) |
|---|---|---|
| AlexNet | 99.63 | 665 |
| VGGNet-16 | 99.79 | 1033 |
| Ours | 99.65 | 457 |

Tabel 6. Comparison of Recognition Results with Previous Research

| Method | Number of Dataset Class | Total Dataset Record | Accuracy (%) |
|---|---|---|---|
| Method [2] | 20 | 2470 | 70.22 |
| Method [16] | 20 | 2000 | 80.65 |
| Method [4] | 20 | 240 | 87.50 |
| Method [1] | 20 | 11500 | 94.57 |
| Method [3] | 31 | 620 | 98.00 |
| Method [37] | 120 | 5880 | 97.50 |
| Ours | 120 | 3360 | 99.65 |

## 5. CONCLUSION

Based on the test results in this research, it is proven that the proposed method works effectively for recognizing Javanese scripts. Furthermore, basic characters compounded with vowel scripts, totalling 120 classes, were investigated. This excellent breakthrough is limited due to the inadequate recognition research on high-accuracy Javanese script and many classes with compound characters. It was proven that the proposed method has an accuracy of 99.65%. The data augmentation process has also been proven to improve recognition by relatively 10% significantly. This shows that it also plays an essential role in recognizing small datasets. Future research must be carried out on more complex datasets combined with consonant scripts to improve its accuracy.

## ACKNOWLEDGEMENTS

The authors are grateful for the support and funding provided by the Ministry of Research and Technology/National Research and Innovation Agency of Indonesia with grant number 6/061031/PG/SP2H/JT/2021.

## REFERENCES


[1] M. A. Wibowo, M. Soleh, W. Pradani, A. N. Hidayanto, and A. M. Arymurthy, "Handwritten Javanese character recognition using descriminative deep learning technique," *Proceedings - 2017 2nd International Conferences on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2017*, vol. 2018-Janua, pp. 325–330, 2018, doi: 10.1109/ICITISEE.2017.8285521.

[2] Rismiyati, Khadijah, and A. Nurhadiyatna, "Deep learning for handwritten Javanese character recognition," *Proceedings - 2017 1st International Conference on Informatics and Computational Sciences, ICICoS 2017*, vol. 2018-January, pp. 59–63, 2017, doi: 10.1109/ICICOS.2017.8276338.

[3] G. S. Budhi and R. Adipranata, "Handwritten Javanese character recognition using several artificial neural network methods," *Journal of ICT Research and Applications*, vol. 8, no. 3, pp. 195–212, 2015, doi: 10.5614/itbj.ict.res.appl.2015.8.3.2.

[4] C. A. Sari, M. W. Kuncoro, D. R. I. M. Setiadi, and E. H. Rachmawanto, "Roundness and eccentricity feature extraction for Javanese handwritten character recognition based on K-nearest neighbor," *2018 International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2018*, pp. 5–10, 2018, doi: 10.1109/ISRITI.2018.8864252.

[5] A. Qaroush, A. Awad, M. Modallal, and M. Ziq, "Segmentation-based, omnifont printed Arabic character recognition without font identification," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 3025–3039, 2022, doi: 10.1016/j.jksuci.2020.10.001.

[6] A. Lamsaf, M. A. Kerroum, S. Boulaknadel, and Y. Fakhri, "Recognition of Arabic handwritten words using convolutional neural network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 26, no. 2, pp. 1148–1155, 2022, doi: 10.11591/ijeecs.v26.i2.pp1148-1155.

[7] R. H. Finjan, A. S. Rasheed, A. A. Hashim, and M. Murtdha, "Arabic handwritten digits recognition based on convolutional neural networks with resnet-34 model," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 1, pp. 174–178, 2021, doi: 10.11591/ijeecs.v21.i1.pp174-178.

[8] B. R. Kavitha and C. Srimathi, "Benchmarking on offline handwritten Tamil character recognition using convolutional neural networks," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 4, pp. 1183–1190, 2022, doi: 10.1016/j.jksuci.2019.06.004.

[9] A. Sufian, A. Ghosh, A. Naskar, F. Sultana, J. Sil, and M. M. H. Rahman, "BDNet: Bengali handwritten numeral digit recognition based on densely connected convolutional neural networks," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 2610–2620, 2022, doi: 10.1016/j.jksuci.2020.03.002.

[10] M. M. Khan, M. S. Uddin, M. Z. Parvez, and L. Nahar, "A squeeze and excitation ResNeXt-based deep learning model for Bangla handwritten compound character recognition," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 3356–3364, 2022, doi: 10.1016/j.jksuci.2021.01.021.

[11] T. Ghosh *et al.*, "Bangla handwritten character recognition using mobilenet v1 architecture," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 6, pp. 2547–2554, 2020, doi: 10.11591/eei.v9i6.2234.

[12] N. Shobha Rani, N. Manohar, M. Hariprasad, and B. R. Pushpa, "Robust recognition technique for handwritten Kannada character recognition using capsule networks," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 1, pp. 383–391, 2022, doi: 10.11591/ijece.v12i1.pp383-391.

[13] S. Singh, A. Sharma, and V. K. Chauhan, "Online handwritten Gurmukhi word recognition using fine-tuned deep convolutional neural network on offline features," *Machine Learning with Applications*, vol. 5, p. 100037, 2021, doi: 10.1016/j.mlwa.2021.100037.

[14] L. Niharmine, B. Outtaj, and A. Azouaoui, "Tifinagh handwritten character recognition using optimized convolutional neural network," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 4, pp. 4164–4171, 2022, doi: 10.11591/ijece.v12i4.pp4164-4171.

[15] K. Khunratchasana and T. Treenuntharath, "Thai digit handwriting image classification with convolution neuron networks," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 27, no. 1, pp. 110–117, 2022, doi: 10.11591/ijeecs.v27.i1.pp110-117.

[16] L. L. Zhangrila, "Accuracy level of $p algorithm for Javanese script detection on android-based application," *Procedia Computer Science*, vol. 135, pp. 416–424, 2018, doi: 10.1016/j.procs.2018.08.192.

[17] N. P. Sutramiani, N. Suciati, and D. Siahaan, "MAT-AGCA: Multi augmentation technique on small dataset for Balinese character recognition using convolutional neural network," *ICT Express*, vol. 7, no. 4, pp. 521–529, 2021, doi: 10.1016/j.icte.2021.04.005.

[18] A. Khalil, M. Jarrah, M. Al-Ayyoub, and Y. Jararweh, "Text detection and script identification in natural scene images using deep learning," *Computers and Electrical Engineering*, vol. 91, 2021, doi: 10.1016/j.compeleceng.2021.107043.

[19] D. Gupta and S. Bag, "CNN-based multilingual handwritten numeral recognition: A fusion-free approach," *Expert Systems with Applications*, vol. 165, 2021, doi: 10.1016/j.eswa.2020.113784.

[20] A. K. Bhunia, S. Mukherjee, A. Sain, A. K. Bhunia, P. P. Roy, and U. Pal, "Indic handwritten script identification using offline-online multi-modal deep network," *Information Fusion*, vol. 57, pp. 1–14, 2020, doi: 10.1016/j.inffus.2019.10.010.

[21] A. A. A. Ali and S. Mallaiah, "Intelligent handwritten recognition using hybrid CNN architectures based-SVM classifier with

dropout," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 3294–3300, 2022, doi: 10.1016/j.jksuci.2021.01.012.
[22] P. Mishra and P. V. V. S. Srinivas, "Facial emotion recognition using deep convolutional neural network and smoothing, mixture filters applied during preprocessing stage," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 4, pp. 889–900, 2021, doi: 10.11591/ijai.v10.i4.pp889-900.
[23] P. A. W. Santiary, I. K. Swardika, I. B. I. Purnama, I. W. R. Ardana, I. N. K. Wardana, and D. A. I. C. Dewi, "Labeling of an intra-class variation object in deep learning classification," *IAES International Journal of Artificial Intelligence*, vol. 11, no. 1, pp. 179–188, 2022, doi: 10.11591/ijai.v11.i1.pp179-188.
[24] O. Sudana, I. W. Gunaya, and I. K. G. D. Putra, "Handwriting identification using deep convolutional neural network method," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 18, no. 4, pp. 1934–1941, 2020, doi: 10.12928/TELKOMNIKA.V18I4.14864.
[25] F. J. Moreno-Barea, J. M. Jerez, and L. Franco, "Improving classification accuracy using data augmentation on small data sets," *Expert Systems with Applications*, vol. 161, 2020, doi: 10.1016/j.eswa.2020.113696.
[26] K. Nugroho, E. Noersasongko, Purwanto, Muljono, and D. R. I. M. Setiadi, "Enhanced Indonesian ethnic speaker recognition using data augmentation deep neural network," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 7, pp. 4375–4384, 2022, doi: 10.1016/j.jksuci.2021.04.002.
[27] O. A. Shawky, A. Hagag, E. S. A. El-Dahshan, and M. A. Ismail, "Remote sensing image scene classification using CNN-MLP with data augmentation," *Optik*, vol. 221, 2020, doi: 10.1016/j.ijleo.2020.165356.
[28] Y. Fu, X. Li, and Y. Ye, "A multi-task learning model with adversarial data augmentation for classification of fine-grained images," *Neurocomputing*, vol. 377, pp. 122–129, 2020, doi: 10.1016/j.neucom.2019.10.002.
[29] M. S. Jarjees, S. S. M. Sheet, and B. T. Ahmed, "Leukocytes identification using augmentation and transfer learning based convolution neural network," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 20, no. 2, pp. 314–320, 2022, doi: 10.12928/TELKOMNIKA.v20i2.23163.
[30] H. Yao, Y. Tan, C. Xu, J. Yu, and X. Bai, "Deep capsule network for recognition and separation of fully overlapping handwritten digits," *Computers and Electrical Engineering*, vol. 91, 2021, doi: 10.1016/j.compeleceng.2021.107028.
[31] D. C. Li, L. S. Lin, and L. J. Peng, "Improving learning accuracy by using synthetic samples for small datasets with non-linear attribute dependency," *Decision Support Systems*, vol. 59, no. 1, pp. 286–295, 2014, doi: 10.1016/j.dss.2013.12.007.
[32] F. F. Alkhalid, A. Q. Albayati, and A. A. Alhammad, "Expansion dataset COVID-19 chest X-ray using data augmentation and histogram equalization," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 2, pp. 1904–1909, 2022, doi: 10.11591/ijece.v12i2.pp1904-1909.
[33] Y. D. Zhang *et al.*, "Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation," *Multimedia Tools and Applications*, vol. 78, no. 3, pp. 3613–3632, 2019, doi: 10.1007/s11042-017-5243-3.
[34] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.
[35] I. A. M. Zin, Z. Ibrahim, D. Isa, S. Aliman, N. Sabri, and N. N. A. Mangshor, "Herbal plant recognition using deep convolutional neural network," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 5, pp. 2198–2205, 2020, doi: 10.11591/eei.v9i5.2250.
[36] M. A. Rasyidi and T. Bariyah, "Batik pattern recognition using convolutional neural network," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 4, pp. 1430–1437, 2020, doi: 10.11591/eei.v9i4.2385.
[37] G. Abdul 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, 2019, doi: 10.1016/j.procs.2019.09.006.

## BIOGRAPHIES OF AUTHORS

**Ajib Susanto** received a Bachelor's degree from the Department of Informatics Engineering, Dian Nuswantoro University, Semarang Indonesia, in 2004 and a Master's degree in Department of Informatics Engineering, Dian Nuswantoro University, Semarang, Indonesia, in 2008. He is currently the Lecturer and researcher at the Faculty of Computer Science, Dian Nuswantoro University, Semarang, Indonesia. His research interests include image processing, machine learning, and data mining. He can be contacted by email: ajib.susanto@dsn.dinus.ac.id.

**Ibnu Utomo Wahyu Mulyono** received a Bachelor's degree at the Department of Informatics Engineering Dian Nuswantoro University, Semarang, Indonesia, in 2001 and a Master's degree in the Department of Informatics Engineering, Dian Nuswantoro University, Semarang, Indonesia, in 2013. He is currently the Lecturer and researcher at the Faculty of Computer Science, Dian Nuswantoro University, Semarang, Indonesia. His research interests include image processing, machine learning, and data mining. He can be contacted at email: ibnu.utomo.wm@dsn.dinus.ac.id

**Christy Atika Sari** received a Bachelor's degree at the Department of Informatics Engineering Dian Nuswantoro University, Semarang Indonesia, in 2010 and a dual Master's degree in the Department of Informatics Engineering, Dian Nuswantoro University, Semarang, Indonesia and in Faculty of Computer Science and Information, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia, in 2012. She is currently the Lecturer and researcher at the Faculty of Computer Science, Dian Nuswantoro University, Semarang, Indonesia. She has authored or co-authored more than 90 refereed journal and conference papers. She is also a reviewer of more than 10 Scopus-indexed journals indexed by Scopus. Her research interests include image processing, especially image steganography, image watermarking, cryptography, and image classification. She can be contacted at email: atika.sari@dsn.dinus.ac.id.

**Eko Hari R Rachmawanto** received a Bachelor's degree at the Department of Informatics Engineering Dian Nuswantoro University, Semarang, Indonesia, in 2009 and a dual Master's degree in the Department of Informatics Engineering, Dian Nuswantoro University, Semarang, Indonesia and in Faculty of Computer Science and Information, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia, in 2012. He is currently the Lecturer and researcher at the Faculty of Computer Science, Dian Nuswantoro University, Semarang, Indonesia. He has authored or co-authored more than 100 refereed journal and conference papers. He is also a reviewer of more than 10 Scopus-indexed journals indexed by Scopus. His research interests include image processing, especially steganography, watermarking, cryptography, and image classification. He can be contacted at email: eko.hari@dsn.dinus.ac.id.

**De Rosal Ignatius Moses Setiadi** received a Bachelor's degree at the Department of Informatics Engineering Soegijaprana Catholic University, Semarang Indonesia, in 2010 and a Master's degree in the Department of Informatics Engineering Dian Nuswantoro University, Semarang, Indonesia, in 2012. He is currently the Lecturer and researcher at the Faculty of Computer Science, Dian Nuswantoro University, Semarang, Indonesia. He has authored or co-authored more than 138 refereed journal and conference papers indexed by Scopus. He is one of the academic editors in the Security and Communication Journal and Journal of Computer Networks and Communications Hindawi. He is also a reviewer of more than 50 Scopus-indexed journals. His research interests include image steganography, watermarking, cryptography, and image recognition. He can be contacted at email: moses@dsn.dinus.ac.id.

**Md Kamruzzaman Sarker** is working as a tenure track assistant professor at the Department of Computing Sciences at the University of Hartford. He obtained his Ph.D. in computer science focusing on Artificial Intelligence in 2020 from Kansas State University. After his Ph.D., he also worked as a postdoc at the Center for Artificial Intelligence and Data Science at the same university. He obtained his M.S. in Computer Science in 2018 from Wright State University and a B.Sc. in Computer Science and Engineering from Khulna University of Engineering & Technology. He also worked at Intel Corporation and Samsung Electronics. Furthermore, he has authored more than 30 peer-reviewed papers and edited a book. He can be contacted at email: mdkamruzzamansarker@gmail.com.