Knowledge Engineering and Data Science (KEDS)
Vol 4, No 2, December 2021, pp. 117–127

pISSN 2597-4602
eISSN 2597-4637

Recognition of Handwritten Javanese Script using Backpropagation with Zoning Feature Extraction

Anik Nur Handayani [a, 1, *], Heru Wahyu Herwanto [a, 2], Katya Lindi Chandrika [a, 3], Kohei Arai [b, 4]

[a] *Department of Electrical Engineering, Universitas Negeri Malang*
Jl. Semarang 5, Malang 65145, Indonesia

[b] *Department of Information Science, Saga University*
Saga Shi Honjou Machi Honjou, 840-8502, Japan

[1] *aniknur.ft@um.ac.id*; [2] heru_wh@um.ac.id; [3] katyachandrika@gmail.com, [4] arai@is.saga-u.ac.jp*
** corresponding author*

ARTICLE INFO

Article history:
Submitted 22 October 2021
Revised 5 November 2021
Accepted 21 November 2021
Published online 31 December 2021

Keywords:
Backpropagation
Feature extraction
Image processing
Javanese script
Pattern recognition

ABSTRACT

Backpropagation is part of supervised learning, in which the training process requires a target. The resulting error is transmitted back to the units below in its training process. Backpropagation can solve complicated problems because it consumes less memory than other algorithms. In addition, it also can produce solutions with a low error rate while executing less time. In image pattern recognition, backpropagation can be utilized for cultural preservation in many places worldwide, including Indonesia. It is used to recognize picture patterns in Javanese script writings. This study concluded that feature extraction approaches, zoning, and backpropagation could be utilized to distinguish handwritten Javanese characters. The best accuracy is attained at 77.00%, with the network architecture comprising 64 input neurons, 40 hidden neurons, a learning rate of 0.003, a momentum of 0.03, and an iteration of 5000.

I. Introduction

The backpropagation method is one of the approaches used in Artificial Neural Networks (ANN), separated into three layers: input, hidden, and output. Backpropagation can solve complicated problems since it consumes less memory than other algorithms and produces solutions with a low error rate and less time [1]. Moreover, this method is preferred because of its ability to distinguish incomplete or weak input patterns. There are three phases in the backpropagation training process, including the forward-backward and weight modification phases. As a result, backpropagation is frequently used in machine learning for a variety of tasks, including classification [2], prediction [3], forecasting [4], and image pattern recognition [5].

Backpropagation in image pattern recognition can be utilized to preserve cultures in diverse parts of the world, such as handwriting recognition of different regional languages in a country, particularly in Asia. Related studies discuss the identification and recognition of each printed Chinese character utilizing projection and zoning feature extraction [6]. In another study, the 990 most commonly used syllables were also used to introduce traditional Korean script, or Hangul [7]. In Thailand, the study identifies Thai letters that include 77 different character patterns [8]. In Japan, there is a study on a procedure to distinguish between Kanji and Kana writing styles for character recognition [9]. Moreover, due to the complexity of printed and handwritten Arabic letters, there is research on Arabic characters to synthesize the essential aspects of the Arabic writing style [10].

Because of the various writing styles, handwriting identification has been the subject of extensive and fascinating research over the last few decades. As a result, the focus of this study will be on using backpropagation to recognize the picture pattern of Javanese scriptwriting in Indonesia. Many similar studies have been conducted, including identifying each character in the Javanese script pattern using

https://doi.org/10.17977/um018v4i22021p117-127

the horizontal and vertical image extraction method transformed through Fourier, which produced an accuracy of 59.5% [11]. Then, another study using backpropagation ANN resulting an accuracy of 61% [12]. Also, the research on *Hanacaraka* Javanese script using the Backpropagation ANN and producing an accuracy of 74% [13].

From those studies, no one has applied a novel strategy to feature extraction, which is zoning developed by Elima Hussain [14] with backpropagation. Therefore, this study discusses the introduction of handwritten Javanese scripts using the backpropagation zoning feature extraction method. In comparison to prior studies, it is expected that the current results will yield a higher accuracy value rather than that did not apply zoning feature extraction on Backpropagation,

II. Method

The initial data collection step will be completed in this project. The obtained data is subsequently entered into the second step, which is the pre-processing data. The pre-processing aims to use data in the feature extraction procedure [15]. The image's features will be stored in the feature extraction procedure at the third step. The normalization procedure is followed by testing with selected algorithms, as well as evaluation and validation of the findings. Figure 1 depicts the progression of the research stages. In the following subchapter, the method and the details of the steps will be detailed in greater depth.

A. Data Collection

The image of the Javanese character *Nglegena* script was used in this study. The data was obtained by spreading the form to respondents. Children, adolescents, and adults who have studied Javanese script were chosen as respondents depending on their age categories: children's ages range from 5 to 11, adolescents' ages range from 12 to 25, and adults' ages range from 26 to 45 years. Each age group will have ten respondents, bringing the total number to 30. Each respondent will write a total of 20 Nglegena Javanese characters. There are 600 photos in all that have been collected. Figure 2 illustrates a sample of respondents' responses to the scanned Javanese script.

B. Data Pre-processing

The picture data will be turned into an image used for feature extraction before further processing [16]. The steps at this process, including changing the image to a gray image (grayscaling), converting the image to a binary image (binarization), cleaning noise (noise removal), equalizing position (crop edge), and equalizing image size, are the stages involved in processing (resizing).

- Grayscaling is the initial level of image processing in this study. The percentage value of each pixel R, G, and B is added to obtain the color picture transition to gray image [17]. Equation (1) shows the percentage of each value in this case.

$$Gray = Y = (0{,}2989 \cdot R) + (0{,}5870 \cdot G) + (0{,}1140 \cdot B) \tag{1}$$

- Binarization, the threshold value determines whether a grayscale pixel value changes from 0 to 255 or black and white [18]. The value chosen as a threshold is 128 [19]. This value is determined by dividing the white pixel value range by the number of gray pixels, which is 255.

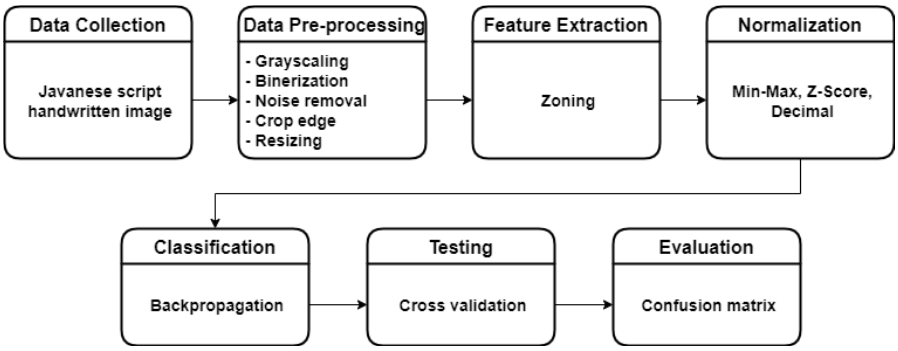
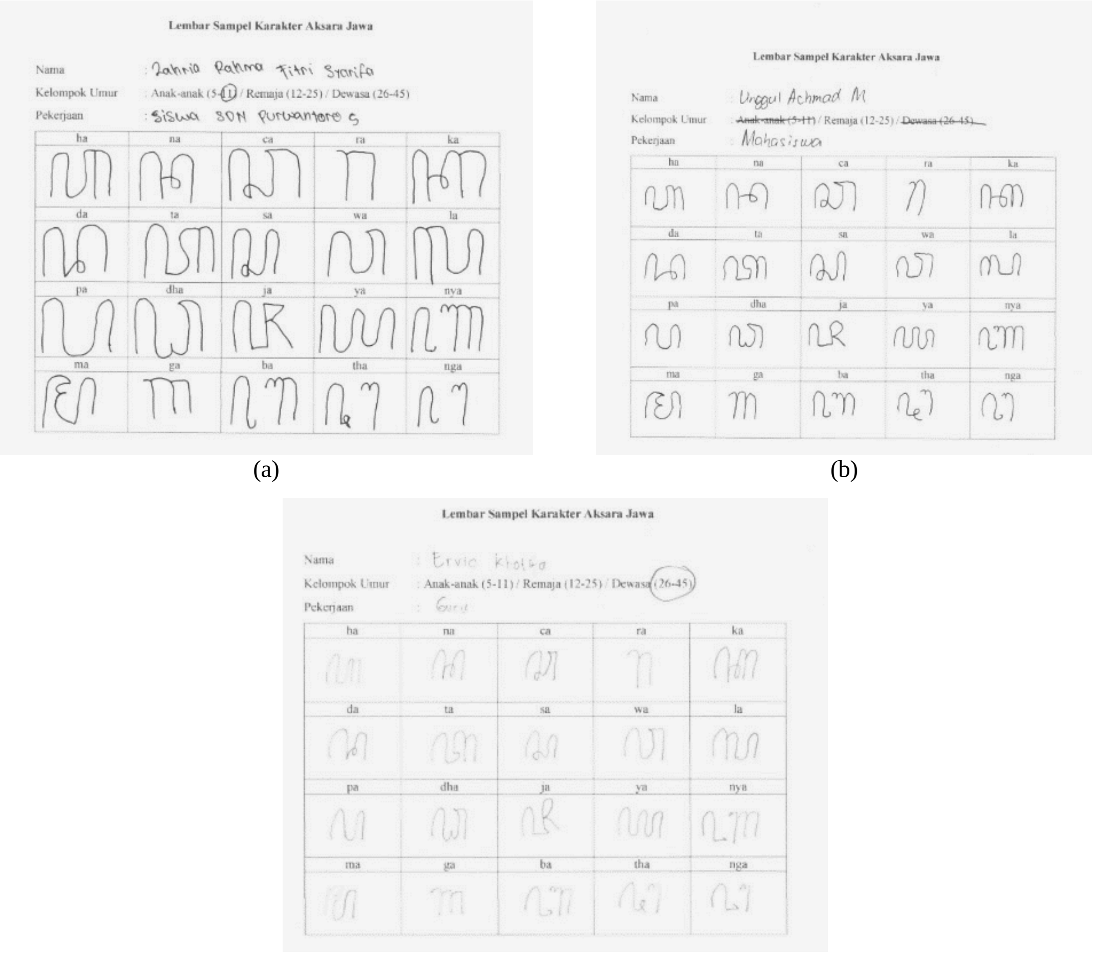


Fig. 1. Research flow

- Noise removal removes the noise from the marker ink and dirt that is simultaneously being scanned. This noise removal process uses the Wiener median filtering method, as illustrated in Equation (2) [20].

$$W(f_1, f_2) = H*(f_1,f_2)S_{xx}(f_1,f_2) / |H(f_1,f_2)|^2S_{xx}(f_1,f_2)+S_{\eta\eta}(f_1,f_2) \qquad (2)$$

- Crop Edge is used to remove unneeded bits of Javanese characters before selecting them. Character cutting is done by searching for the highest and lowest x and y point values with black pixels [21].
- Resizing, at this stage, the image size is equalized to 120×120 pixels [22]. Figure 3 presents the difference between the raw image and the resized image.



Lembar Sampel Karakter Aksara Jawa

Nama : Zahria Rahma Fitri Syarifa
Kelompok Umur : Anak-anak (5-[✓ 11]) / Remaja (12-25) / Dewasa (26-45)
Pekerjaan : Siswa SDN Purwantoro 5

ha
na
ca
ra
ka
da
ta
sa
wa
la
pa
dha
ja
ya
nya
ma
ga
ba
tha
nga

(a)

Lembar Sampel Karakter Aksara Jawa

Nama : Unggul Achmad M
Kelompok Umur : ~~Anak-anak (5-11)~~ / Remaja (12-25) / ~~Dewasa (26-45)~~
Pekerjaan : Mahasiswa

ha
na
ca
ra
ka
da
ta
sa
wa
la
pa
dha
ja
ya
nya
ma
ga
ba
tha
nga

(b)

Lembar Sampel Karakter Aksara Jawa

Nama : Ervic Kholifa
Kelompok Umur : Anak-anak (5-11) / Remaja (12-25) / Dewasa [✓ (26-45)]
Pekerjaan : Guru

(c)

Fig. 2. Samples of scanned data sheets for Javanese script; (a) children; (b) adolescents; and (c) adults

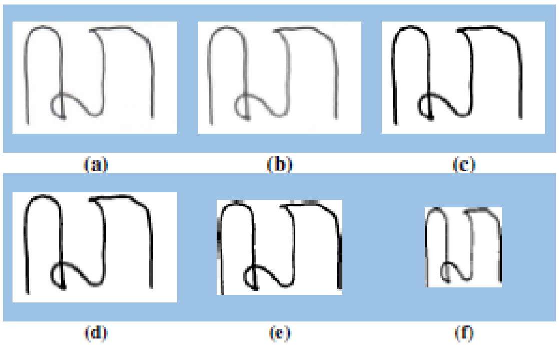
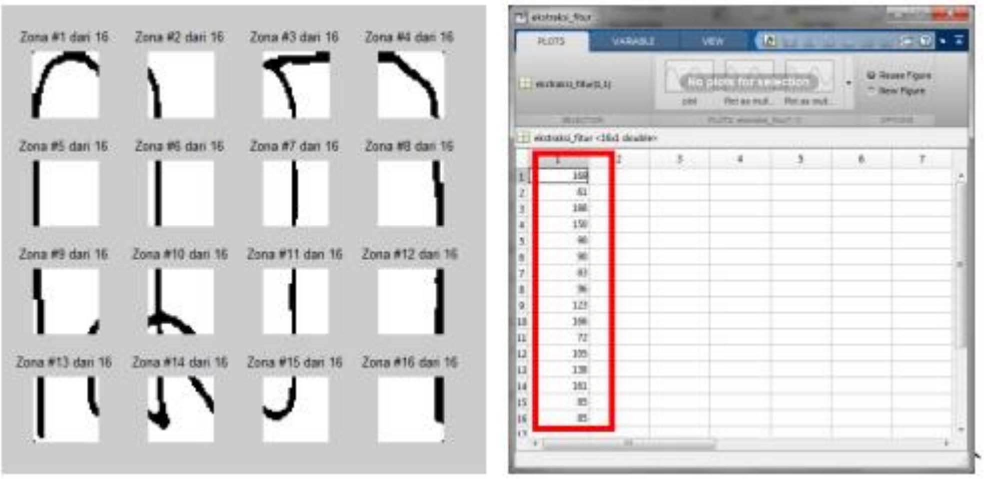


Fig. 3. Differences in character image; (a) before image pre-processing; (b) after grayscalling; (c) after binarization; (d) after crop edge; (e) after resizing; and (f) final resized image



	1
1	169
2	61
3	186
4	150
5	96
6	90
7	83
8	96
9	123
10	196
11	72
12	105
13	138
14	161
15	85
16	85

Fig. 4. Division of 16 zones on the character "ca" and the calculation results of black pixels for each zone

C. Extraction Feature

In this study, feature extraction is done using a novel method called zoning, developed by Elima Hussain [14]. The 120×120 character image will be broken into 16, 25, 36, and 64 parts. This number of divisions was chosen since these values may entirely divide the image size. The zoning feature produces numerical data on the number of black pixels in each zone. Figure 4 shows an illustration of the zone division.

D. Normalization

Normalization is the process of scaling attribute values to fit within a given range [23]. Since the range of data acquired in this study is so vast, normalization is required. For example, a zone size of 10 by 10 pixels will have a black pixel value range of 0 to 100. Several normalizing strategies are presented; the normalization techniques used in this study are:

- Min-Max Normalization is the process of scaling data from one range to another [24]. This method uses the formula stated in Equation (3) to change the data.

$$normalized(x) = \frac{minRange + (x - minValue)(maxRange - minRange)}{maxValue - minValue} \tag{3}$$

- Z-Score Normalization is accomplished by removing the value from the data's average and dividing it by its standard deviation [25]. Equation (4) shows the formula for this method.

$$new\ value = \frac{old\ value - mean}{stdev} \tag{4}$$

- Decimal Scaling Normalization is a method of normalization that divides a variable by the power of 10 [26]. This approach is presented in Equation (5).

$$new\ value = \frac{old\ value}{10} \tag{5}$$

E. Classification

Backpropagation ANN is the classification algorithm employed in this study. Backpropagation is part of supervised learning because the training procedure requires a target. Backpropagation is named after the resulting error propagates back to the units below it throughout the training process [27]. The network architecture used in this study is a Multi-Layer Net, which is a network design with multiple layers. An input layer, a hidden layer, and an output layer make up the network. There are 16 neurons in the input layer that contain the value retrieved from the feature. The number of neurons in the hidden layer is modified to find the best outcomes [8]. The output layer has 20 neurons, which matches the expected number of Javanese characters, which is 20. The architecture employed in this investigation is illustrated in Figure 5. Backpropagation's pseudocode is shown in the code snippet below.

```
# Calculate neuron activation for an input
def activate(weights, inputs):
  activation = weights[-1]
  for i in range(len(weights)-1):
  activation += weights[i] * inputs[i]
  return activation

# Transfer neuron activation
def transfer(activation):
  return 1.0 / (1.0 + exp(-activation))

# Forward propagate input to a network output
def forward_propagate(network, row):
  inputs = row
  for layer in network:
    new_inputs = []
    for neuron in layer:
      activation = activate(neuron['weights'], inputs)
      neuron['output'] = transfer(activation)
      new_inputs.append(neuron['output'])
    inputs = new_inputs
  return inputs

# Calculate the derivative of an neuron output
def transfer_derivative(output):
  return output * (1.0 - output)

# Backpropagate error and store in neurons
def backward_propagate_error(network, expected):
  for i in reversed(range(len(network))):
    layer = network[i]
    errors = list()
    if i != len(network)-1:
      for j in range(len(layer)):
        error = 0.0
        for neuron in network[i + 1]:
          error += (neuron['weights'][j] * neuron['delta'])
        errors.append(error)
    else:
       for j in range(len(layer)):
         neuron = layer[j]
         errors.append(expected[j] - neuron['output'])
    for j in range(len(layer)):
       neuron = layer[j]
       neuron['delta']=errors[j]*transfer_derivative(neuron['output'])
```

F. Testing

During the testing phase, the collected data will be separated into two categories, including training data and testing data. The k-fold cross-validation approach will be used to divide the data. The study employed 5, 10, and 15 as the number of k [28]. In this experiment, 600 image data will be separated into ten groups, each with 60 data. The data from the nine groups were utilized as training data, while one group was used as test data. Figure 6 presents the process of partitioning the dataset using the k-fold cross-validation approach.

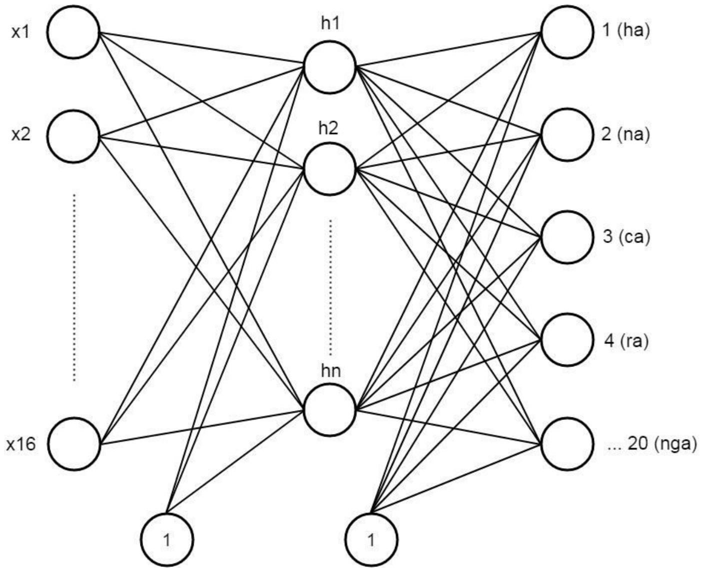
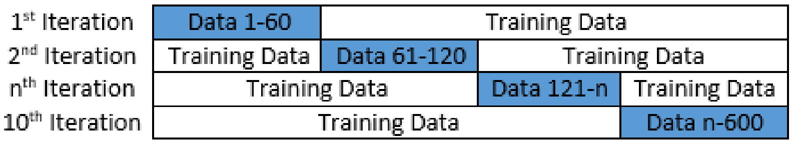


Fig. 5. Backpropagation Neural Network Architecture Java Script Recognition



1st Iteration	Data 1-60	Training Data		
2nd Iteration	Training Data	Data 61-120	Training Data	
nth Iteration	Training Data		Data 121-n	Training Data
10th Iteration	Training Data			Data n-600

Fig. 6. Illustration of K-fold cross validation with total K=10

G. Evaluation

The confusion matrix approach is used to evaluate the Backpropagation classification algorithm. This evaluation approach is currently being used to test two classes, but it can be changed to handle many class classifications. Table 1 shows the confusion matrix for 20 classes [29]. The accuracy formula presented in Equation (6) [30] is used to calculate the evaluation to determine the correctness of the data.

$$Accuracy = \frac{\Sigma\, N\, correct}{\Sigma\, \text{N}} \tag{6}$$

where Σ *N correct* is the number of image data that are classified correctly, and Σ *N* is the number of available image data.

Table 1. Confusion matrix for handling 20 classes

	Classification Results										
	K1	**K2**	**K3**	**K4**	**K5**	**K6**	**K7**	**K8**	**K9**	**…**	**K20**
K1	X										
K2		X									
K3			X								
K4				X							
K5					X						
K6						X					
K7							X				
K8								X			
K9									X		
…										X	
K20											X

III. Results and Discussions

Backpropagation architecture parameters used in the training process include architecture, number of neurons, activation function, learning rate value, momentum value, maximum iteration, and learning algorithm. Table 2 lists the requirements for these parameters.

In this study, there were various stages to the experiments. The first step is to figure out which normalization results are the best. Based on these findings, the next stage is to determine the best learning rate and momentum values. The number of neurons in the hidden layer is determined at the third stage. The number of input neurons will be changed in the fourth stage to discover the optimal zone in this study.

A. Determination of the Best Normalization Results

Each experiment has a set design at this point, with 16 neurons in the input layer, 20 neurons in the hidden layer, and 20 neurons in the output layer, with a maximum of 5000 iterations. The number of zones formed is equal to the number of input layers. A total of 16 zones were employed in this experiment. In the third stage, the value of the zone or input layer will be adjusted. There is a range of parameter values in each experiment. Raw data, data from the Min-max Normalization process, data from the Z-score Normalization process, and data from the Decimal Normalization process are all used. Experiments at this stage will use K-Fold Cross Validation with a K value of 10 until the fourth stage. Table 3 shows the findings of the first experiment.

According to Table 3, data normalized using the Z-score Normalization approach had the highest overall accuracy. The range of values produced using Z-score Normalization data is 56.33% to 60.00%. This value range is higher than the accuracy results produced from raw data, data adjusted using the Min-Max approach, and data standardized using Decimal Normalization.

B. Determination of the Best Learning Rate and Momentum Values

Each experiment has the same architecture and settings as the previous experiment. The data used has been adjusted using the Z-Score Normalization method. Table 4 shows the results of determining the best learning rate and momentum values.

Table 4 shows that the best accuracy results are obtained when the learning rate is 0.003 with the momentum at 0.03. The accuracy of recognizing Javanese characters reaches 60.00% with these two values. The experimental stages of calculating the number of hidden neurons will be carried out using this learning rate and momentum value.

Table 2. Specifications of backpropagation neural network

Characteristic	**Specification**
Input Neuron	16
Hidden Neuron	20, 30, dan 40
Output Neuron	20
Activation Function	Sigmoid Biner (logsig)
Learning Rate	0.003; 0.005; 0.008
Momentum	0.005; 0.01; 0.03; 0.05; 0.08; 0.1
Maximum Iteration	5000

Table 3. Best normalization result

No.	**Learning Rate**	**Momentum**	**Iteration**	**Raw**	**Min-Max**	**Z-Score**	**Decimal**
1	0.003	0.03	5000	43.33%	23.00%	60.00%	9.17%
2	0.003	0.005	5000	43.67%	20.50%	59.33%	9.83%
3	0.003	0.05	5000	46.83%	22.00%	59.67%	9.67%
4	0.005	0.1	5000	40.17%	44.83%	57.33%	10.17%
5	0.008	0.01	5000	29.83%	55.17%	57.33%	27.00%
6	0.008	0.08	5000	32.67%	56.17%	56.17%	29.83%

C. Determination of the Number of Hidden Neurons

The third step is to determine how many neurons are in the hidden layer. The number of neurons used varies between 20, 30, and 40. The learning rate parameter is 0.003, the momentum parameter is 0.03, and the iterations are 5000. Table 5 shows the results of the experiment.

Table 5 shows the maximum result of 64.00%, achieved with as many as 40 neurons in the hidden layer. The test findings of various network architectures are still of low value, as can be shown. Therefore, increasing the number of input neurons can help enhance backpropagation testing outcomes [31]. In this scenario, increasing the number of input neurons means increasing the number of zones, which provides more precise information about the network.

D. Determination of the Number of Input Neurons

In the experiment, there are 40 hidden neurons in the backpropagation architecture. According to the zone partition plan, 16, 25, 36, and 64 input neurons are used. The best learning rate and momentum levels are 0.003 and 0.03. Table 6 shows the results of determining the number of input neurons.

Table 6 shows that the accuracy improves when the input neurons are increased up to 64 neurons. When the number of input neurons is increased to 100, the accuracy attained drops to 73.00%. It validates that the highest accuracy results are achieved with 64 input neurons, which is 77.00%. Low test scores can be caused by a lack of diversity in handwritten Javanese writing patterns used as training data.

E. Determination of the Number of K in K-Fold Cross-Validation

The last experiment was to identify the amount of K in K-Fold Cross-Validation. K's value will be adjusted to 5, 10, and 15, respectively. Table 7 shows the results of the K number on the K-Fold Cross-Validation.

According to Table 7, the test with a K number of 5 has a 75.17% accuracy, whereas the test with a K number of 15 has a 76.67% accuracy. These two results are low compared to the accuracy results achieved when the K value is 10. The accuracy rating can reach 77.00% with a K value of 10.

Table 4. The best learning rate and momentum values

No.	Learning Rate	Momentum	Iteration	Accuracy
1	0.003	0.03	5000	60.00%
2	0.003	0.005	5000	59.33%
3	0.003	0.05	5000	59.67%
4	0.005	0.1	5000	57.33%
5	0.008	0.01	5000	57.33%
6	0.008	0.08	5000	56.17%

Table 5. The best hidden neuron number results

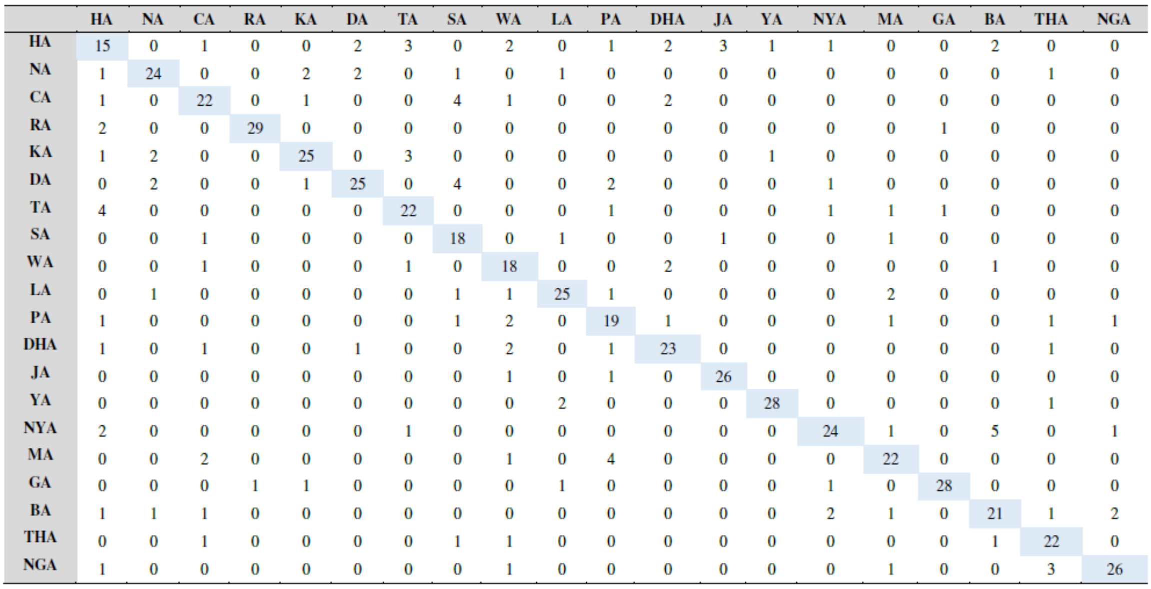
No.	Learning Rate	Momentum	Hidden Neuron	Iteration	Accuracy
1	0.003	0.03	20	5000	60.10%
2	0.003	0.03	30	5000	62.50%
3	0.003	0.03	40	5000	64.00%

Table 6. The best number of input neurons

No.	Input Neuron	Learning Rate	Momentum	Hidden Neuron	Iteration	Raw
1	16	0.003	0.03	40	5000	64.00%
2	25	0.003	0.03	40	5000	64.83%
3	36	0.003	0.03	40	5000	76.17%
4	64	0.003	0.03	40	5000	77.00%
5	100	0.003	0.03	40	5000	73.00%

Table 7. Result of best normalization

No.	Number of K	Input Neuron	Learning Rate	Momentum	Hidden Neuron	Iteration	Accuracy
1	5	64	0.003	0.03	40	5000	75.17%
2	10	64	0.003	0.03	40	5000	77.00%
3	15	64	0.003	0.05	40	5000	76.67%



	HA	NA	CA	RA	KA	DA	TA	SA	WA	LA	PA	DHA	JA	YA	NYA	MA	GA	BA	THA	NGA
HA	15	0	1	0	0	2	3	0	2	0	1	2	3	1	1	0	0	2	0	0
NA	1	24	0	0	2	2	0	1	0	1	0	0	0	0	0	0	0	0	1	0
CA	1	0	22	0	1	0	0	4	1	0	0	2	0	0	0	0	0	0	0	0
RA	2	0	0	29	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
KA	1	2	0	0	25	0	3	0	0	0	0	0	0	1	0	0	0	0	0	0
DA	0	2	0	0	1	25	0	4	0	0	2	0	0	0	1	0	0	0	0	0
TA	4	0	0	0	0	0	22	0	0	0	1	0	0	0	1	1	1	0	0	0
SA	0	0	1	0	0	0	0	18	0	1	0	0	1	0	0	1	0	0	0	0
WA	0	0	1	0	0	0	1	0	18	0	0	2	0	0	0	0	0	1	0	0
LA	0	1	0	0	0	0	0	1	1	25	1	0	0	0	0	2	0	0	0	0
PA	1	0	0	0	0	0	0	1	2	0	19	1	0	0	0	1	0	0	1	1
DHA	1	0	1	0	0	1	0	0	2	0	1	23	0	0	0	0	0	0	1	0
JA	0	0	0	0	0	0	0	0	1	0	1	0	26	0	0	0	0	0	0	0
YA	0	0	0	0	0	0	0	0	0	2	0	0	0	28	0	0	0	0	1	0
NYA	2	0	0	0	0	0	1	0	0	0	0	0	0	0	24	1	0	5	0	1
MA	0	0	2	0	0	0	0	0	1	0	4	0	0	0	0	22	0	0	0	0
GA	0	0	0	1	1	0	0	0	0	1	0	0	0	0	1	0	28	0	0	0
BA	1	1	1	0	0	0	0	0	0	0	0	0	0	0	2	1	0	21	1	2
THA	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	22	0
NGA	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	3	26

Fig. 7. Confusion matrix

F. Evaluation

Based on the previous experimental stages, the maximum accuracy is obtained using a network architecture that comprises 64 input neurons, 40 hidden neurons, and 20 neurons in the output layer. The learning rate is 0.003, the momentum is 0.3, and the number of iterations is 5000. Also, the test uses K-Fold Cross Validation with a K value of 10. The confusion matrix in Figure 7 is used to analyze the categorization findings from the testing stage.

The accuracy value for 20 classes is calculated by adding all the correctly predicted data and dividing by the number of data tested using Equation 6. As a result of the evaluation, the resulting accuracy value is 462/600 × 100% = 77.00%, with a neuron architecture of 64-40-20.

IV. Conclusion

There are multiple stages to creating the backpropagation architecture, including identifying the appropriate learning rate and momentum values, number of hidden layers, and number of input neurons. The network architecture includes 64 input neurons, 40 hidden neurons, and 20 neuron outputs, achieving the highest accuracy of 77.00%. The learning rate is 0.003, the momentum is 0.03, and the number of iterations is 5000. The recognition accuracy is affected by increasing the number of input neurons (in this case, adding zones) in the backpropagation architecture. Variations in Javanese script handwriting patterns that are more reproduced as backpropagation training data and research data development utilizing Javanese scriptwriting in the form of words or sentences can be used for future research.

Declarations

Author contribution

All authors contributed equally as the main contributor of this paper. All authors read and approved the final paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of interest

The authors declare no known conflict of financial interest or personal relationships that could have appeared to influence the work reported in this paper.

Additional information

Reprints and permission information are available at http://journal2.um.ac.id/index.php/keds.

Publisher's Note: Department of Electrical Engineering - Universitas Negeri Malang remains neutral with regard to jurisdictional claims and institutional affiliations.

References

[1] A. Suliman and Y. Zhang, "A Review on Back-Propagation Neural Networks in the Application of Remote Sensing Image Classification," *J. Earth Sci. Eng.*, vol. 5, no. 1, Jan. 2015.

[2] M. Muladi, D. Lestari, D. T. Prasetyo, A. P. Wibawa, T. Widiyaningtiyas, and U. Pujianto, "Classification of Locally Grown Apple Based On Its Decent Consuming Using Backpropagation Artificial Neural Network," in *2019 International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, 2019, pp. 96–100.

[3] H. Aini and H. Haviluddin, "Crude Palm Oil Prediction Based on Backpropagation Neural Network Approach," *Knowl. Eng. Data Sci.*, vol. 2, no. 1, p. 1, 2019.

[4] P. Purnawansyah, H. Haviluddin, H. Darwis, H. Azis, and Y. Salim, "Backpropagation Neural Network with Combination of Activation Functions for Inbound Traffic Prediction," *Knowl. Eng. Data Sci.*, vol. 4, no. 1, p. 14, Aug. 2021.

[5] S. Afroge, B. Ahmed, and F. Mahmud, "Optical character recognition using back propagation neural network," in *2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)*, 2016, pp. 1–4.

[6] A. Khawaja, S. Tingzhi, N. M. Memon, and A. Rajpar, "Recognition of printed Chinese characters by using Neural Network," in *2006 IEEE International Multitopic Conference*, 2006, pp. 169–172.

[7] S.-B. Cho and J. H. Kim, "Recognition of large-set printed Hangul (Korean script) by two-stage backpropagation neural classifier," *Pattern Recognit.*, vol. 25, no. 11, pp. 1353–1360, Nov. 1992.

[8] B. Kijsirikul and S. Sinthupinyo, "Approximate ILP Rules by Backpropagation Neural Network: A Result on Thai Character Recognition," in *9th International Workshop on Inductive Logic Programming*, 2003, pp. 162–173.

[9] S. D. Budiwati, J. Haryatno, and E. M. Dharma, "Japanese character (Kana) pattern recognition application using neural network," in *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*, 2011, pp. 1–6.

[10] H. A. A., "Back Propagation Neural Network Arabic Characters Classification Module Utilizing Microsoft Word," *J. Comput. Sci.*, vol. 4, no. 9, pp. 744–751, Sep. 2008.

[11] I. Prihandi, I. Ranggadara, S. Dwiasnati, Y. S. Sari, and Suhendra, "Implementation of Backpropagation Method for Identified Javanese Scripts," *J. Phys. Conf. Ser.*, vol. 1477, no. March, pp. 1–6, Mar. 2020.

[12] N. Nurmila, A. Sugiharto, and E. A. Sarwoko, "Algoritma Back Propagation Neural Network untuk Pengenalan Pola Karakter Huruf Jawa," *J. Masy. Inform.*, vol. 1, no. 1, pp. 1–10, 2010.

[13] A. Setiawan, A. S. Prabowo, and E. Y. Puspaningrum, "Handwriting Character Recognition Javanese Letters Based on Artificial Neural Network," *Int. J. Comput. Netw. Secur. Inf. Syst.*, vol. 1, no. 1, pp. 39–42.

[14] H. W. Herwanto, A. N. Handayani, K. L. Chandrika, and A. P. Wibawa, "Zoning Feature Extraction for Handwritten Javanese Character Recognition," in *2019 International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, 2019, pp. 264–268.

[15] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," in *2014 Science and Information Conference*, 2014, pp. 372–378.

[16] G. Kumar and P. K. Bhatia, "A Detailed Review of Feature Extraction in Image Processing Systems," in *2014 Fourth International Conference on Advanced Computing & Communication Technologies*, 2014, pp. 5–12.

[17] T. Kumar and K. Verma, "A Theory Based on Conversion of RGB image to Gray image," *Int. J. Comput. Appl.*, vol. 7, no. 2, pp. 5–12, Sep. 2010.

[18] K. Y. Kok and P. Rajendran, "A Descriptor-Based Advanced Feature Detector for Improved Visual Tracking," *Symmetry (Basel).*, vol. 13, no. 8, p. 1337, Jul. 2021.

[19] M. H. Ali, S. Kurokawa, and K. Uesugi, "Vision based measurement system for gear profile," in *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*, 2013, pp. 1–6.

[20] A. K. Ghosh and A. A. Ansari, "To Analysis and Implement Image De-Noising Using Fuzzy and Wiener Filter in Wavelet Domain," *Int. J. Trend Res. Dev.*, vol. 8, no. 3, pp. 320–373, 2021.

[21] S. Zhu, S. Dianat, and L. K. Mestha, "End-to-end system of license plate localization and recognition," *J. Electron. Imaging*, vol. 24, no. 2, p. 023020, Mar. 2015.

[22] R. Samad and H. Sawada, "Edge-based facial feature extraction using Gabor wavelet and convolution filters," *Proc. 12th IAPR Conf. Mach. Vis. Appl. MVA 2011*, pp. 430–433, 2011.

[23] J. C. Caicedo *et al.*, “Data-analysis strategies for image-based cell profiling,” *Nat. Methods*, vol. 14, no. 9, pp. 849–863, Sep. 2017.

[24] T. Jayalakshmi and A. Santhakumaran, “Statistical Normalization and Back Propagationfor Classification,” *Int. J. Comput. Theory Eng.*, vol. 3, no. 1, pp. 89–93, 2011.

[25] D. Singh and B. Singh, “Investigating the impact of data normalization on classification performance,” *Appl. Soft Comput.*, vol. 97, p. 105524, Dec. 2020.

[26] A. S. Eesa and W. K. Arabo, “A Normalization Methods for Backpropagation: A Comparative Study,” *Sci. J. Univ. Zakho*, vol. 5, no. 4, p. 319, Dec. 2017.

[27] A. P. Markopoulos, S. Georgiopoulos, and D. E. Manolakos, “On the use of back propagation and radial basis function neural networks in surface roughness prediction,” *J. Ind. Eng. Int.*, vol. 12, no. 3, pp. 389–400, Sep. 2016.

[28] S. Hulu, P. Sihombing, and Sutarman, “Analysis of Performance Cross Validation Method and K-Nearest Neighbor in Classification Data,” *Int. J. Res. Rev.*, vol. 7, no. April, pp. 69–73, 2020.

[29] G. Bueno *et al.*, “Automated Diatom Classification (Part A): Handcrafted Feature Approaches,” *Appl. Sci.*, vol. 7, no. 8, p. 753, Jul. 2017.

[30] A. Bogoliubova and P. Tymków, “Accuracy assessment of automatic image processing for land cover classification of St. Petersburg protected area,” *Acta Sci. Pol. Geod. Descr. Terrarum*, vol. 13, no. January 2014, pp. 5–22, 2014.

[31] O. Krestinskaya, K. N. Salama, and A. P. James, “Learning in Memristive Neural Network Architectures Using Analog Backpropagation Circuits,” *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 66, no. 2, pp. 719–732, Feb. 2019.