Application of Backpropagation Algorithm for Handwriting Recognition

by Sustin Farlinda

Application of Backpropagation Algorithm for Handwriting Recognition

Sustin Farlinda[1], Muhammad Yunus[2*], Mudafiq Riyan Pratama[3], Bakhtiyar Hadi Prakoso[4], Angga Rahagiyanto[5], Dyah Susilowati[6], Ni Gusti Ayu Dasriani[7], Lukmanul Hakim[8], Khairil Anam[9]

[1,2,3,4,5]Program Studi Rekam Medik, Jurusan Kesehatan, Politeknik Negeri Jember
[6,7,8]Program Studi Ilmu Komputer, Fakultas Teknik dan Desain, Universitas Bumigora Mataram
[9]Program Studi Teknik Informatika, Fakultas Teknik, Universitas Maarif Hasyim Latif Sidoarjo

*m.yunus@polije.ac.id

Abstract. Handwritten letter recognition is one form of pattern recognition. The introduction of letters looks simple to humans, but it becomes a tough task for computer programs to complete. In recognizing someone's handwriting, a computer program trained first. This study discusses how a computer recognizes a digital image pattern in the form of handwritten letters using the backpropagation algorithm. The backpropagation algorithm used in making this system is backpropagation with momentum. The network architecture used consists of an input screen with 64 neurons, a hidden screen consisting of 46 neurons, and an output screen with five neurons. Before the recognition process, input image with image format (jpg) processed first, which includes scaling, grayscale, and binarization. The trial results show that the system can recognize handwriting with an accuracy of up to 65%.

Keywords: handwriting recognition, algorithm, backpropagation

1. Introduction

The development of technology today has a lot of influence on the development of science, one of which is in terms of pattern recognition. Pattern recognition is a science to classify or describe something based on quantitative measurements of features (features) or the main properties of an object [1]. Pattern recognition is widely applied to detect fingerprints, writing, signatures, and even a person's face. One area of study in the field of pattern recognition is Optical Character Recognition (OCR). The letter recognition system, or often called OCR, is a computer system that can read letters, both originating from a printer (printer or typewriter) or originating from handwriting [2]. Handwritten letter recognition is a technique in which data input in the form of scanned paper sheets uses a scanner and produces images on a computer that recognized as dots (bitmaps), these bitmaps are then further processed using specific algorithms into characters so that they can be known and transformed into information [3].

A handwritten letter recognition system is needed when essential documents written by hand need to be changed and saved into records in the form of text, but lazy to retype. Also, storing documents in the way of writing becomes more effective and safe because it can be stored easily on a computer. The problem that arises in the process of handwriting letter recognition is how a recognition technique can recognize various types of letters with different sizes, thicknesses, and shapes because there are limitations on the image matching method and statistical approach, then we need another way that allows the letter recognition system will give better results.

One solution to overcome this problem is to design a system that can recognize handwriting using the backpropagation method. The backpropagation method is the right method to identify a handwriting pattern because this method uses repetitive training so that it can guarantee the accuracy of the data. With some training in pattern recognition, it hoped that this method could recognize handwriting. Backpropagation is one of the supervised training methods. Backpropagation trains the network to strike a balance between the ability of the system to recognize patterns used during training and the strength of the system to provide correct responses to patterns of input similar to those used during training.

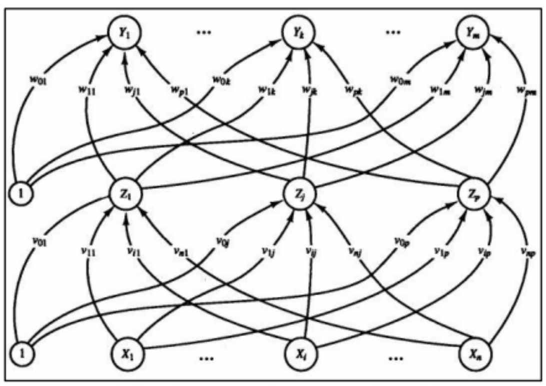

Figure 1. Backpropagation network architecture

The picture above is a backpropagation network architecture image with n inputs (plus refraction), a hidden screen consisting of p units (plus a preference), and m output units. V_ji is the line weight from the input unit X_i to the dark screen unit Z_j (V_jo is the line weight that connects the bias in the input unit to the hidden screen unit Z_j). W_kj is the weight of the dark screen unit Z_j to the Y_k output unit (W_ko is the weight of the dark screen refraction to the Y_k output unit) [4]. In general, the following are backpropagation algorithm steps [5]:

a. Initialize all weights with small random numbers
b. Determine parameters such as maximum epoch, target, error, learning rate, and momentum
c. As long as the stop conditions met, the loop repeated. Stopping diseases can include reaching the number of epochs or reaching the desired Mean Squared Error (MSE)

Phase I: Propagation going forward

d. Each input unit receives a signal and passes it to the hidden unit above it
e. For each secret group, (z_j where j = 1, ..., p), add up the weighted input signal (including refraction) with the following formula:

$$z_{net_j} = v_{j0} + \sum_{i=1}^{n} x_i \, v_{ji}$$

Where :

v_{j0} = refraction in hidden units
x_i = input value to i

f. v_{ji} = weights from the unit I input to hidden group i

Then calculate the output signal from the secret unit according to the activation function that has set:

$$z_j = f(z\_net_j)$$

Then the signal will be sent to all units in the next layer, namely the output unit.

g. For each output unit, (y_k where k = 1, ..., m) will do the calculation :

$$y_{net_k} = w_{k0} + \sum_{j=1}^{p} z_j \, w_{kj}$$

Where w_k_0 = refraction unit output value to k

h. z_j = input value to j (results of calculation of step 5)

w_{kj} = the weight of the hidden unit k to the output unit j

Then calculate the output signal from the output unit according to the activation that has set:

$$y_k = f(y\_net_k)$$

Phase II: Backward Propagation

i. For each output unit, (y_k where k = 1, ..., m) an error will be calculated between the output values generated by the network and the target with the following formula:

$$\delta_k = (t_k - y_k)\, f'(y_{net_k})$$

Then calculate the error correction (Δw_{kj}) which used to correct w_kj with the following formula:

$$\Delta w_{kj} = \alpha \delta_j z_k$$

Then the correction biased will be calculated (Δw_{k0}) which used to correct w_k0 using the following formula:

$$\Delta w_{k0} = \alpha \delta_0$$

Factor δ is then sending to layers that are in the next stage.

j. Calculate the hidden unit factor based on the error in each secret unit Zj (j=1,2,...,p)

$$\delta\_net_j = \sum_{k=1}^{m} \delta_k \, w_{kj}$$

The result of δ_net_j is multiplied by the derivation of the activation function used to get the error δ_j error, where:

$$\delta_j = \delta\_net_j \, f'(z\_net_j)$$

After that weights will be calculated (to correct v_ji)

$$\Delta v_{ji} = \alpha \delta_j x_i \qquad ; j = 1, 2 \ldots, p\ ;\ i = 0, 1, \ldots, n$$

Phase III: Change in weights

k. Calculate all weight changes

Changes inline weights leading to the output unit:

$$w_{kj}(new) = w_{kj}\,(long) + \Delta w_{kj}\ (k = 1, 2, \ldots, m\ ;\ j = 0, 1, \ldots, p)$$

Changes inline weights leading to hidden units:

$$v_{ji}(new) = v_{ji}\,(long) + \Delta v_{ji}\ (j = 1, 2, \ldots, p\ ;\ i = 0, 1, \ldots, n)$$

l. If the stop conditions have been met (such as a maximum epoch or target error), then neural network training can be stopped. Testing can do through advanced propagation with the following steps:

1. Provide weight initialization based on training results

2. For each input vector, work through steps 3,4, and 5 below
3. For i=1,...,n:
4. Set the activation of the selected unit for x_i input
5. For j=1,...,p:

$$z_{net_j} = v_{j0} + \sum_{i=1}^{n} x_i \, v_{ji}$$

Where :
v_{j0} = *bias in hidden units*
x_i = input value to i
v_{ji} = weights from unit j input to hidden unit i
Then count :

$$z_j = f(z\_net_j)$$

6. For k=1,...,m:
For each output unit, (y_k where k = 1, ..., m) will do the calculation:

$$y_{net_k} = w_{k0} + \sum_{j=1}^{p} z_j \, w_{kj}$$

Where :
w_{k0} = the bias value of the unit output to k
z_j = input value to j (results of calculation of step 4)
w_{kj} = the weight of the hidden unit k to the output unit j

Then calculate the output:

$$y_k = f(y\_net_k)$$

Activation Function

In backpropagation, the activation function that used must meet several conditions, namely continuous, easily differentiated, and is a function that does not go down [4]. Several activation functions used. The following is an activation function on a neural network:

1. Binary Sigmoid
   The binary sigmoid function produces positive numbers between 0 and 1.
   $f(x) = \frac{1}{1+e^{-x}}$ with derivatives $f'(x) = f(x)(1 - f(x))$

2. *Hyperbolic* Tangent

$$f_{AN}(net-0) = \frac{e^{(net-0)} - e^{-(net-0)}}{e^{(net-0)} + e^{-(net-0)}}$$

   or

$$f_{AN}(net-0) = \frac{2}{1 + e^{-(net-0)}} - 1$$

   The output of hyperbolic tangent is range (-1,1).

3. *Softmax*
   The activation function applied to neuron output. In multiclass classification problems, the output layer usually has more than one neuron. In this case, the activation function commonly used is the softmax function. For example $\mathrm{a} = [\mathrm{a}_1 \ldots\ldots \mathrm{a}_m]$ *is a vector with m elements, and Softmax is defined as follows:*

$$\sigma(a_j) = \frac{\exp(a_j)}{\sum_{k=1}^{m} \exp(a_k)}$$

Can check that $\sum_{j=1}^{m} \sigma(a_j) = 1$ for *softmax*.

***Mean Squared Error* (MSE)**

The mean squared error (MSE) value in one training cycle is the average error value (error = output value - input value) of all records that are presented to ANN and formulated as:

$$MSE = (\frac{\sum error^2}{jumlah\ rekord})$$

The smaller the MSE, the lower the ANN error in predicting the class of the new record. Thus, ANN training intended to reduce MSE from one cycle to the next cycle until the difference in MSE value in this cycle with the previous period is smaller or equal to the minimum limit given.

***Optical Character Recognition* (OCR)**

Text recognition technology is the technology that can recognize text in digital images and transfer them to digital documents. Currently, the basic concept of OCR used in several text recognition applications.

The following picture shows an illustration of the OCR process:

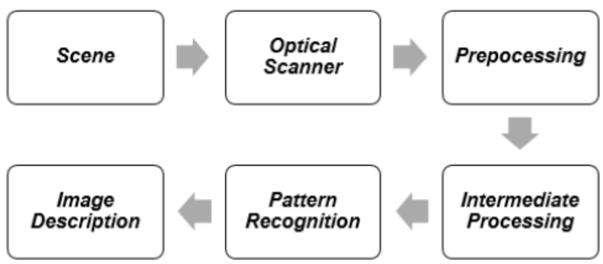

Figure 2. Process of OCR [6]

The working principle of the OCR application is as follows:

1. Insert a document containing text (machine printed text) into an optical device (scanner) so that an image file obtained
2. The image file is processed using text recognition application software, where this device performs the process of recognizing the characters in the image file
3. The output of this text recognition application software is in the form of text files containing characteristics that have been identified and are ready for further processing.

The success rate of text recognition application software depends significantly on several factors, as follows [7] :

a. The image quality of the text in the document read and the level of complexity (size, text format, color, background)
b. The variety of optical devices used (scanners). The quality of the text recognition application software.

2. Methodology

The system development method used in this study is a waterfall which consists of several stages, namely requirements analysis, system design, unit testing and application, system implementation and testing, and maintenance.

Requirements Analysis

The system created is a system for handwriting recognition using the backpropagation method. In this handwriting recognition, the input data compared with the target to achieve. Handwritten data divided into two, namely training data and testing data. Training data used for the training process and testing data used for the testing process, where the results will be in the form of text characters.

Following is the workflow of the handwriting recognition system:

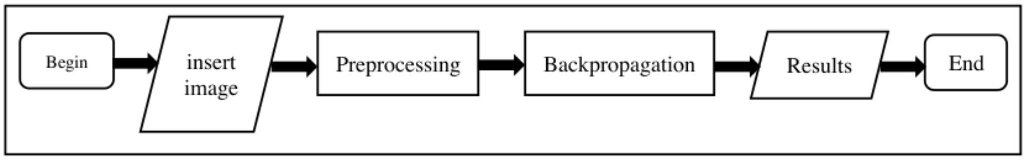

Figure 3. Flowchart Handwriting Recognition System

System Design

The following is an object-based system design of the system created:

Use Case Diagrams

The following is the use case diagram of the system created:

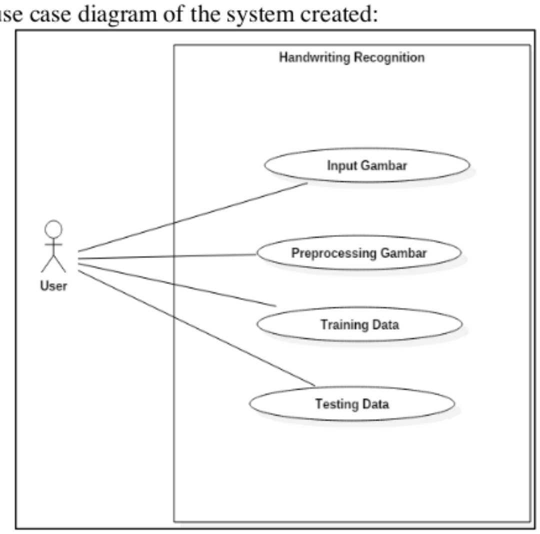

Figure 5. Use Case Diagrams

Activity Diagram

1. *Activity* Diagram of *Preprocessing*

   The figure below shows the sequence of activities of the user during preprocessing.

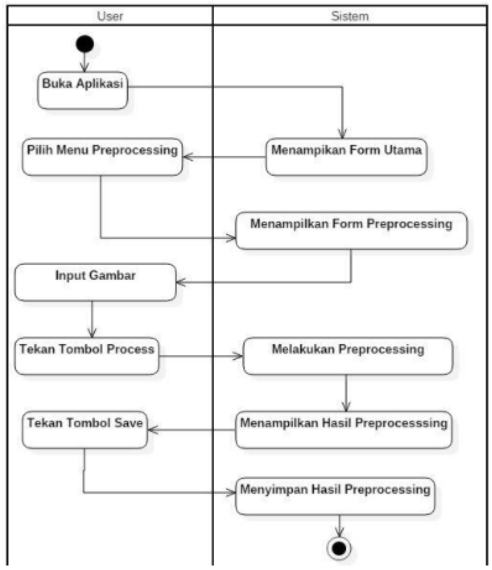

Figure 6. *Activity* Diagram of *Preprocessing*

2. *Activity* Diagram of Data *Training*
   This activity diagram illustrates the sequence of activities of the user when data training.

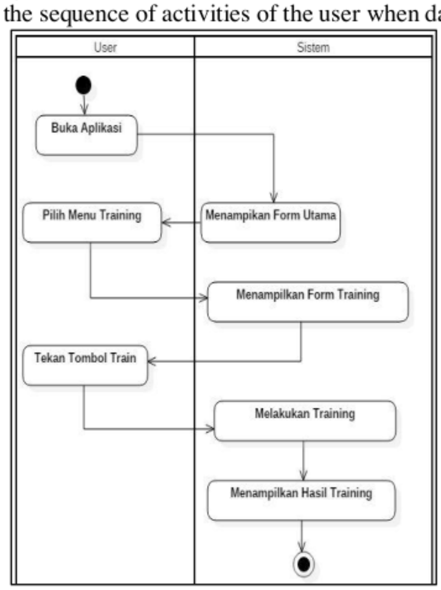

Figure 7. *Activity* Diagram of Data *Training*

3. *Activity* Diagram of Data *Testing*
   This activity diagram illustrates the sequence of activities of users when data testing.

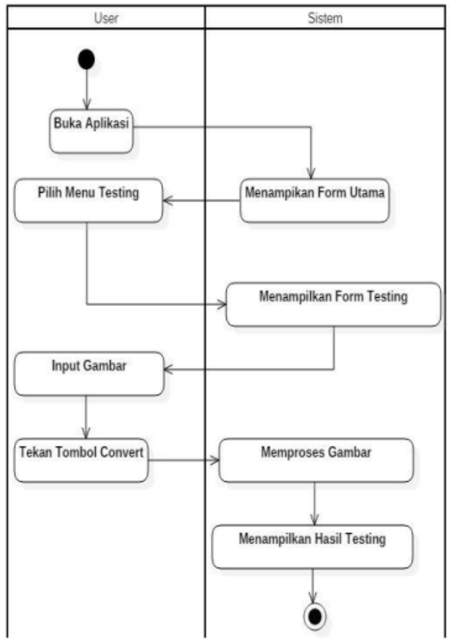

Figure 8. *Activity* Diagram of Data *Testing*

3. Result and Discussion

The test conducted on 104 character letters consisting of capital letters A-Z. Each letter is amounting to 4 letters. Examples of image processing before the image enters the artificial neural network are in Figure 9 below:

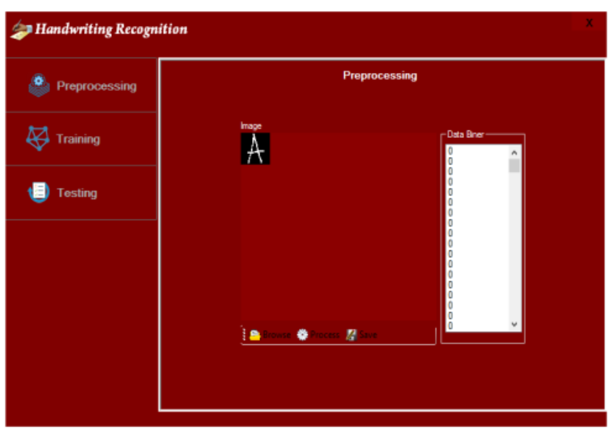

Figure 9. Preprocessing Results

The final result of the image will be the training set on the artificial neural network has a size of 8x8 pixels. Based on these measurements, the number of inputs in the input neural network is 64, 46 units for the hidden layer, and five units for the output layer. Here is the training result data.

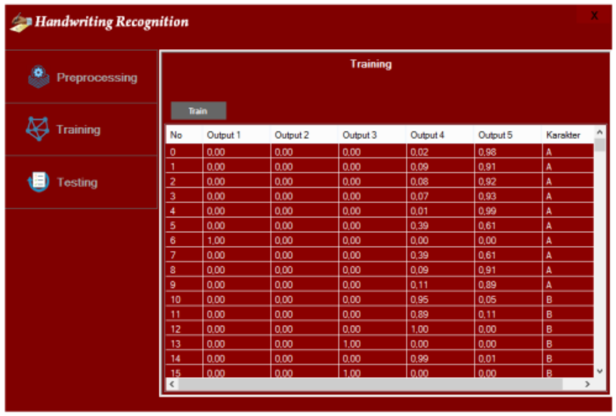

Figure 10. Training Results

The results of the training will be the output targeted for testing data. Here is an example of the effects of testing the system.

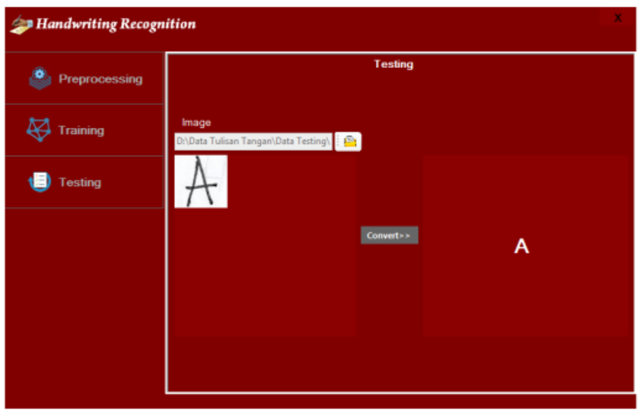

Figure 11. Testing Results

Meanwhile, the following are the complete results of testing the handwriting recognition system with the backpropagation algorithm:

Table 1. Testing Results

NO	DATA	Input	Results	Conclusions
1	A1	A	A	**True**
2	A2	A	A	**True**
3	A3	A	A	**True**
4	A4	A	A	**True**
5	B1	B	B	**True**
6	B2	B	B	**True**
7	B3	B	B	**True**
8	B4	B	B	**True**
9	C1	C	C	**True**
10	C2	C	C	**True**
11	C3	C	C	**True**
12	C4	C	C	**True**

NO	DATA	Input	Results	Conclusions
13	D1	D	D	**True**
14	D2	D	D	**True**
15	D3	D	D	**True**
16	D4	D	D	**True**
17	E1	E	E	**True**
18	E2	E	E	**True**
19	E3	E	E	**True**
20	E4	E	E	**True**
21	F1	F	F	**True**
22	F2	F	F	**True**
23	F3	F	F	**True**
24	F4	F	F	**True**
25	G1	G	C	False
26	G2	G	C	False
27	G3	G	E	False
28	G4	G	C	False
29	H1	H	H	**True**
30	H2	H	H	**True**
31	H3	H	H	**True**
32	H4	H	H	**True**
33	I1	I	I	**True**
34	I2	I	I	**True**
35	I3	I	I	**True**
36	I4	I	A	False
37	J1	J	J	**True**
38	J2	J	J	**True**
39	J3	J	J	**True**
40	J4	J	J	**True**
41	K1	K	C	False
42	K2	K	C	False
43	K3	K	C	False
44	K4	K	C	False
45	L1	L	L	**True**
46	L2	L	L	**True**
47	L3	L	L	**True**
48	L4	L	L	**True**
49	M1	M	M	**True**
50	M2	M	M	**True**
51	M3	M	M	**True**
52	M4	M	M	**True**
53	N1	N	J	False
54	N2	N	C	False
55	N3	N	C	False
56	N4	N	N	**True**
57	O1	O	C	False
58	O2	O	C	False
59	O3	O	C	False
60	O4	O	C	False
61	P1	P	P	**True**
62	P2	P	P	**True**
63	P3	P	P	**True**
64	P4	P	P	**True**

NO	DATA	Input	Results	Conclusions
65	Q1	Q	Q	**True**
66	Q2	Q	A	False
67	Q3	Q	A	False
68	Q4	Q	P	False
69	R1	R	R	**True**
70	R2	R	R	**True**
71	R3	R	R	**True**
72	R4	R	F	False
73	S1	S	C	False
74	S2	S	C	False
75	S3	S	C	False
76	S4	S	C	False
77	T1	T	T	**True**
78	T2	T	T	**True**
79	T3	T	T	**True**
80	T4	T	T	**True**
81	U1	U	C	False
82	U2	U	P	False
83	U3	U	R	False
84	U4	U	A	False
85	V1	V	V	**True**
86	V2	V	V	**True**
87	V3	V	V	**True**
88	V4	V	V	**True**
89	W1	W	C	False
90	W2	W	C	False
91	W3	W	C	False
92	W4	W	C	False
93	X1	X	X	**True**
94	X2	X	X	**True**
95	X3	X	X	**True**
96	X4	X	X	**True**
97	Y1	Y	X	False
98	Y2	Y	Y	**True**
99	Y3	Y	Y	**True**
100	Y4	Y	Y	**True**
101	Z1	Z	Z	**True**
102	Z2	Z	F	False
103	Z3	Z	J	False
104	Z4	Z	B	False
Percentage of Success				**65%**

Based on the test results in table 4.1 for handwritten characters, the system can recognize 68 handwritten characters out of a total of 104 characters tested with a success rate of 65%.

4. Conclusion

Based on the research results obtained in the previous chapter, it concluded that:

1. Based on the backpropagation system built to prove that backpropagation is suitable for handwriting recognition.
2. The accuracy of the backpropagation system in recognizing the character of the handwritten letters tested was 65%.

References

[1] D. Putra. *Pengolahan Citra Digital*. Yogyakarta : Andi. 2010.

[2] S. Hartanto, A. Sugiharto, and S. Endah, "OPTICAL CHARACTER RECOGNITION MENGGUNAKAN ALGORITMA TEMPLATE MATCHING CORRELATION," JURNAL MASYARAKAT INFORMATIKA, vol. 5, no. 9, pp. 1-12, Apr. 2015. https://doi.org/10.14710/jmasif.5.9.1-12g

[3] S. KC and C. Nattee, "A Comprehensive Survey on On-line Handwriting Recognition Technology and its Real Application to the Nepalese Natural Handwriting," KUSET, vol. 5, no. 1, pp. 31-55, 1.

[4] J.J. Siang. *Jaringan Syaraf Tiruan dan Pemrograman dengan MATLAB*. Yogyakarta : Andi. 2005

[5] S. Kusumadewi, and S. Hartati. *Neuro-Fuzzy Integrasi Sistem Fuzzy & Jaringan Syaraf Edisi 2*. Yogyakarta: Graha Ilmu. 2010.

[6] T. B. Nurwanto, "FUZZY DENGAN PENDEKATAN NEURAL NETWORKS BACK PROPAGATION CHARACTER HANDWRITING RECOGNITION USING FUZZY LOGIC WITH NEURAL NETWORK," 2007.

[7] N. I. Manik, "PERANCANGAN PROGRAM APLIKASI PENGENALAN TEKS," vol. 2010, no. semnasIF, pp. 34–41, 2010.

Application of Backpropagation Algorithm for Handwriting Recognition

ORIGINALITY REPORT

7%	0%	6%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Sitti Amalia. "Identification Of Number Using Artificial Neural Network Backpropagation", MATEC Web of Conferences, 2018 Publication	2%
2	A P Widowo, E A Sarwoko, Suhartono. "Performance comparison of machine learning methods for prediction of estimating water production", Journal of Physics: Conference Series, 2019 Publication	2%
3	Submitted to University of Lincoln Student Paper	2%
4	S, . "Artificial Neural Networks - Architectures and Algorithms", Computational Intelligence Paradigms Theory & Applications using MATLAB, 2010. Publication	2%

Exclude quotes	On	Exclude matches	< 2%
Exclude bibliography	On		