Handwriting Character Recognition Javanese Letters Based on Artificial Neural Network

Ariyono Setiawan[1]
Department of Air Transportation Management
Aviation Polytechnic of Surabaya
Surabaya, Indonesia, 60294
rmaryo4u@gmail.com

Achmad Setya Prabowo[2]
Department of Air Transportation Management
Aviation Polytechnic of Surabaya
Surabaya, Indonesia, 60294
setyo.atkp@gmail.com

Eva Y Puspaningrum[3]
Department of Informatics Engineering
Universitas Pembangunan Nasional "Veteran" Jatim
Surabaya, Indonesia, 60294
evapuspaningrum.if@upnjatim.ac.id

***Abstract*— Javanese character is one of Indonesia's cultural heritages that must be preserved. Manuscript form of the Javanese character is one of the priceless inheritances. Javanese characters are often called the Hanacaraka font. The development of technology especially in the field of image processing is one method to preserve the culture. The development of image processing methods for detecting characters from images with fewer errors is a great task. The purpose of this research is to create a system of Javanese character recognition from the handwriting into Latin character, so that the young generation can learn the form of Javanese character easily. The method used in this research is back-propagation as a classification method. The trial result of this recognition is to have accuracy 74 %.**

***Keywords*— Javanese Character, Recognition, Back-propagatio.**

I. Introduction

Digital image processing now includes character recognition techniques such as alphanumeric characters, handwritten characters, kanji characters, and others. Optical Character Recognition usually uses input in the form of images captured by the camera, text images, handwritten text, typed or printed [1]. Character recognition system or often called OCR is an effective solution for the process of converting printed documents into digital documents [2]. Much work has been done on the recognition of Latin characters, both of separated and of cursive script [3]. Work on recognizing java characters is still limited. Improvement of image processing can also be used in preserving science and culture. One application can be used in preserving the Javanese character, which until now is almost extinct because of the development of increasingly modern times. Javanese character is a part of Javanese language that is inherent in Javanese culture. The minimum use of Javanese character among the Javanese community, makes Javanese character rarely used for daily life. Javanese language users are reduced in number and few teenagers are familiar with Javanese script clearly. So it is feared that this ancestral heritage will be extinct. One of the salvation that we can do in a short time is to digitize the script of Javanese inheritance which is almost extinct. An advanced technique is the automatic translation of Javanese scripts into Latin characters

Using character recognition based information technology with image processing. It takes a long process to automatically produce Javanese script translations. An important part of the automatic translation system is the recognition of each of the basic Javanese characters [4]. In general the Javanese script is known as Hanacaraka font. Basic Javanese character has 20 main characters that are ha, na, ca, ra, ka, da, sa, wa, la, pa, dha, ja, yes, ma, ga, ba, ta, and nga.

Several attempts to create a Javanese Character pattern recognition system have been carried out by researchers but until now the results have not been good in accuracy [5]. Previously research Javanese character recognition by using LVQ algorithm. The accuracy obtained is 46.7% [6]. Another research used Multi-Layer Perceptron to perform Javanese character recognition. In his research and the total accuracy obtained is 38.1% [7]. The first process is preprocessing digital images, followed by segmentation and feature extraction. These features will be used as input for the recognition system.

This research is a transliteration process Javanese script into Latin letters. In this traditional method of OCR, there are three important steps which are segmentation, Feature Extraction, and Classification [8]. In this paper we have implemented the classification through neural network using back propagation algorithm. The degree of character recognition depends on the quality of the image i.e. the image resolution. The scanned image is more complicated because there are many possible variations in background, font and illumination. In image preprocessing, we apply various techniques to remove noise, and unwanted text. Images can be cropped, resized, enlarged, reduced and can also adjust the image resolution. We use a back-propagation algorithm where pixel value forms the input node and the output node is obtained from the database. Back-propagation identifies the character that suggests the winning neuron from the output node called the output neuron as the winning neuron signifies a character. The main point is to show that the Back-propagation network can be applied to image recognition problems without the need for large and complex preprocessing stages. Unlike most previous studies, learning networks are directly fed with images, not feature vectors, thus demonstrating the ability of the Back-propagation network to handle large amounts of low-level information.

II. Literature review

A. Javanese Character

The Javanese script, or aksara Jawa, is used for writing the Javanese language, the native language of one of the peoples of Java, known locally as basa Jawa [9]. The Javanese script is also used for writing Sanskrit, Jawa Kuna (a kind of Sanskritized Javanese), and transcriptions of Kawi (the Kawi script itself is not unified with Javanese). Javanese script was in current use in Java until about 1945. Traditional Javanese texts are written on palm leaves, books of these bound together are called lontar. Javanese character has 20 main characters as shown in Fig. 1.

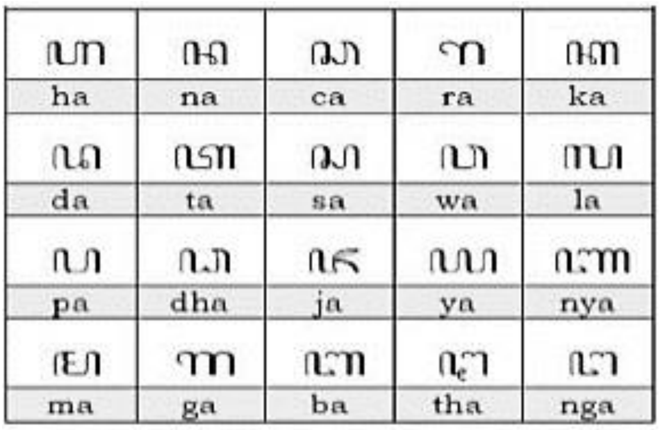


Fig. 1. Javanese Character (hanacaraka)

B. Back propagation Back Propagation Neural Network

Back propagation is a learning method that is usually used by perceptron with many layers to change the weights associated with neurons in the hidden layer. Back propagation method uses its error output to change its weighted value in back-forward. To get this error, the feed-forward stage must be performed. According to Fausett [10] the following is a back propagation network training algorithm for one hidden layer:

1. Initialize the weights by giving a random value
2. As long as the condition stops false, do steps 3-9
3. For each pair of training data (x_setb, tb) where b = 1, ... i, do 4-8
4. Starting the forward process, each input unit (Xi, i = 1, ..., n) receives the input signal xi and continues to the hidden layer, each hidden unit (Zj, j = 1, ... p) adds up the input signals weighted

   $$Z = V0_j + \sum_{i=j}^{n} X_i V_{ij} \quad (1)$$

   Use the activation function to calculate the signal the output,

   $$Z_i = f(Z_{ni}) \quad (2)$$

   and continue the signal to all units in the upper layer (output layer).

5. Each output unit (Yk, k = 1, ..., m) sums the weighted input signals,

   $$y_{nk} = w0_k + \sum_{i=1}^{p} Z_j W_{jk}$$
   (3)

   Use the activation function to calculate the output signal, and proceed to the backward process

6. Each unit of output (yk, k = 1, ..., m) receives a target pattern related to the learning input pattern, calculates error information k,

   $$\delta_k = (t_k - y_k) f'(y_{nk}) \quad (4)$$

   Calculate the weight correction wjk, calculate the correction bias w0k, and send the error information value to the lower layer

7. Each hidden unit sums the product times the error information with Weight

   $$\delta_{nj} = \sum_{k=1}^{m} \delta_k w_{jk}$$
   (5)

   calculate error information j,

   $$\delta_j = \delta_{ni} f'(Z_{nj}) \quad (6)$$

   Correction of vij weights, and correction bias v0j, and proceed to the weight update stage.

8. Each unit of output (Yk, k = 1, ..., m) fixes its weight and bias (j = 0, ..., p),

   $$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk} \quad (7)$$

   Each hidden unit (Zj, j = 1,… p) fixes its weight and bias (i = 0, ..., n),

   $$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij} \quad (8)$$

9. Test the condition, if true then the training stops.

III. METHODOLOGY

First preparation of Javanese script data. At this stage it is determined what will be done as prepared to make handwriting. In this study handwriting using a pen. The pen was chosen because the ink used will produce hand writing that is clearer and better than a charcoal pencil. This will affect the image processing. The handwriting will be scanned. The scan results will be input to this system. In this study the data used are Javanese script which will become a dataset. Javanese script used in the form of an image data or image size of 30 × 30 pixels totaling 20 pieces. The input data needed is Javanese script that is handwritten on a scanned paper shown in Fig. 2. The preprocessing can be seen in Fig. 3

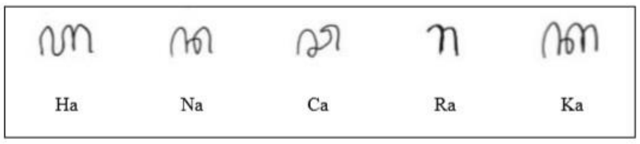


Fig. 2. Sample Data

Scanning images are transformed into grayscale images. After that the results of the grayscale process will be detected the edge of each character that exists using the Canny algorithm. After getting the edges, the process of threshold and changing the image into black and white form. Then thickening the edges of the image. This is done to reinforce the shape of the image that has been threshold. After that the image will be resized to 30 × 30 pixels. The rest will be processed in the Back-propagation algorithm.

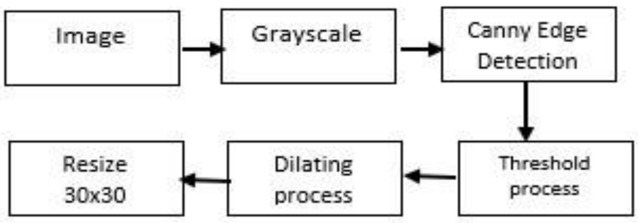


Fig. 3. Preprocessing

After the pre-processing is complete, the binary data from the image will be input in each neuron in the input layer. Before starting the training process, initials are carried out for a number of variables, such as learning rate, which functions to calculate changes in weight between the output layer and hidden layer, then error tolerance which functions to keep the weight according with what is desired, then there are many targets of training, and the last is the maximum number of iterations to limit the length of the training process. After all initialization, then create a loop error with iteration or error tolerance. Then, check the amount of data who will be trained.

In the process of recognizing Javanese script patterns, it also requires a feature extraction process for input images in the form of scanned Javanese scripts. Subsequent scans will act as input. By using the weight value of the training results stored in the dataset and the method of calculating ANN, the Javanese script will be known.

At this recognition stage after getting the binary image system number will only call the feed-forward function. Next is the process of checking the output activation from the feed-forward stage to the target output of the training results The result of reducing the target output in the training process and the result of activating the output of the feed-forward process that has the closest value to 1 will be the result of recognition

IV. RESULT AND DISCUSSION

In this research the number of nodes used for input is 900 nodes. Because the size of the extracted feature used as input is 30 × 30 pixels. The number of nodes in the output layer is 20. This corresponds to the probability of each Javanese script being trained. The result of feature extraction from each character of Javanese script will be counted so that the output will be in the form of a Javanese script matrix and determined how the Latin letters will be. Each pixel used as input will be checked whether it has a value of 1 or 0. When all nodes are known, the value will be matched with the input dataset, it will have a matrix corresponding to the character of Java.

Image data used are 200 pieces for the training process and 100 for the testing process. The details are for training each of the 10 characters, while for testing as many as 5 each. So the total data used is 300 images. From the training results obtained by running the system as much as 6 times obtained from the total number of outputs that match the system or have reached the target of 179 characters. Testing data is taken from several people with different handwriting. An example of the handwriting used for testing data can be seen in Fig.4.

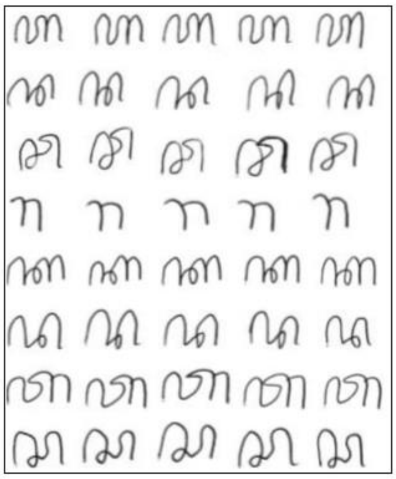
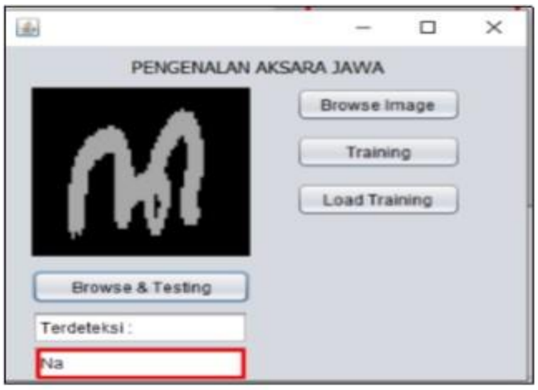


Fig. 4. Sampel Of Data Testing

Recognition of Javanese script with handwritten data other than the author. The testing process used 100 characters of data. Each character type 5 image. Display for trial Java characters can be seen in fig.5.



Terdeteksi :

Na

Fig. 5. Interface System

From the results of character testing where each character consists of 5 pieces with different handwriting. After testing it was obtained the results can be seen in Table 1.

TABLE I. TESTING RESULT

Latin	Javanese	Accuracy
Ha	ꦲ	40%
Na	ꦤ	80%
Ca	ꦕ	20%
Ra	ꦫ	100%
Ka	ꦏ	100%
Da	ꦢ	80%
Ta	ꦠ	40%
Sa	ꦱ	20%
Wa	ꦮ	80%
La	ꦭ	80%
Pa	ꦥ	100%
Dha	ꦝ	80%
Ja	ꦗ	100%
Ya	ꦪ	60%
Nya	ꦚ	100%
Ma	ꦩ	80%
Ga	ꦒ	100%
Ba	ꦧ	60%
Tha	ꦛ	60%
Nga	ꦔ	100%
Average		74%

From the test results it can be seen that there are some images that cannot reach the target. The image did not reach the target because there are similar values between the script pattern and each other. There are some images that have a shape that is somewhat similar so that the characters are not recognized. The trial results obtained an average accuracy of 74%.

V. Conclusion

From trials conducted using 100 test data taken from several people with different handwriting in the test results. In this system the iteration is 100000. Which means the training process stops at iteration 100000 with an error target close to 0 for each character trained. From the test results it can be seen that there are some images that cannot reach the target. The image did not reach the target because there are similar values between the script pattern and each other. There are some images that have a shape that is somewhat similar so that the characters are not recognized. The trial results obtained an average accuracy of 74%.

Acknowledgment

Thanks to the co-author who helped with the completion of writing this paper. And thanks to Aviation Polytechnic of Surabaya for helping to finance this paper so that this paper can be published.

References

[1] M. Cai, J. Song and M. R. Lyu. A “New Approach for Video Text Detection”, In Proc. of International Conference On ImageProcessing, Rochester, New York,USA, pp. 117-120, 2002.

[2] Sutojo, T, Mulyanto, E, Suhartono, V, Nurhayati, O, D, Wijanarto. “Teori Pengolahan Citra Digital”. Yogyakarta: Andi. 2009

[3] Somaya Al-Ma’adeed, Dave Elliman, and Colin Higgins, “A Data Base for Arabic Handwritten Text Recognition Research”, The International Arab Journal of Information Technology, Vol. 1, No. 1, January 2004

[4] M Agung Wibowo, M Soleh, W Pradani, A Nizar Hidayanto , Aniati M Arymurthy. “Handwritten Javanese Character Recognition using Descriminative Deep Learning Technique”, 2nd International Conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), IEEE. Publised 2017

[5] A. R. Himamunanto and A. R. Widiarti, “Javanese character image segmentation of document image of Hamong Tani,” Proc. Digit. 2013 - Fed. 19th Int’l VSMM, 10th Eurographics GCH, 2nd UNESCO Mem. World Conf. Plus Spec. Sess. fromCAA, Arqueol. 2.0 al., vol. 1, pp. 641–644, 2013.

[6] A. C. Agustina, S. Suwarno, and U. Proboyekti, “Pengenalan Aksara Jawamenggunakan Learning Vector Quantization (Lvq),” no. 1, 2009.

[7] M. Christian Wibowo, I. Dewa Gede Rai Mardiana, and S. Wirakusuma, “Pengenalan pola tulisan tangan aksara jawa menggunakan multi layer perceptron,” Semin. Nas. Teknol. Inf. dan Multimed., p. (3.8)1-6, 2015

[8] Swapnil Desai, Ashima Singh.” Optical character recognition using template matching and back propagation algorithm”, International Conference on Inventive Computation Technologies (ICICT), IEEE, Published 2017

[9] Michael Everson,” Proposal for encoding the Javanese script in the UCS”, Universal Multiple-Octet Coded Character SetInternational Organization for StandardizationOrganisation Internationale de Normalisation or consideration by JTC1/SC2/WG2 and UTC. 2008.

[10] L. Fausett, “Fundamentals of Neural Networks: Architectures, Algorithms, and Applications”, Upper Saddle River: Prentice-Hall, 1994