Pattern Recognition of Batik Madura Using Backpropagation Algorithm

Abduh Riski* Ega Bandawa Winata, Ahmad Kamsyakawuni

Mathematics Department, Faculty of Mathematics and Sciences, University of Jember, Indonesia
Corresponding author. Email: riski.fmipa@unej.ac.id

ABSTRACT

Since October 2, 2009, UNESCO has acknowledged batik as one of Indonesia's intellectual properties. Throughout the archipelago, distinct and diverse batik motifs have emerged and been produced with the passage of time; Madura batik is one of them. The Backpropagation Algorithm is used to recognize Madura Batik Patterns in this research. Bunga Satompok, Manuk Poter, Pecah Beling, Rumput Laut, and Sekar Jagat are the motifs used in this study. To begin, resize the image to 200 × 200 pixels and convert it to a grayscale image. The Gray Level Co-occurrence Matrix (GLCM) approach is used to extract image features, and the Backpropagation Algorithm is used to recognize them. With GLCM, the angle orientations utilized in the feature extraction process are 0, 45, 90, and 135 degrees. There are 1, 3, and 5 hidden layers used throughout the training process, with hidden neurons in each layer of 8, 16, and 32. The highest accuracy is achieved when five hidden layers with 32 hidden neurons and one hidden layer with 32 hidden neurons in each layer are used in the testing process, which is 98 percent.

***Keywords:** Batik, Backpropagation, Gray level co-occurrence matrix, Neural network.*

1. INTRODUCTION

Indonesia is a multicultural country that has a diversity of cultures and arts. UNESCO has designated batik as one of the world's cultural heritage from Indonesia [1]. Batik has many motifs. Almost every region in Indonesia has its motives. One of them is a motif from Madura batik. Madura batik has a characteristic colour that tends to be firm and bold. Many previous studies have done pattern recognition, for example, facial recognition [2], fingerprints [3], Retina [4], and others. The algorithm used is also diverse, one of which is the Backpropagation algorithm. Backpropagation is one of the supervised algorithms on Artificial Neural Networks that can train networks to learn patterns used during training and recognize patterns of input similar to those used during training during testing [5]. In the recognition process, the Gray Level Co-occurrence Matrix (GLCM) method is used as a feature extraction method using four GLCM computation features, namely Contrast, Energy, Correlation, and Homogeneity. The results obtained in the feature extraction process will be used to calculate the accuracy generated as a measure of the algorithm, and the method used runs well.

2. BACKPROPAGATION ALGORITHM

Artificial neural network is a computational method with an interconnected network, so it is similar to human neural networks. Artificial neural networks have an adaptive nature, which can learn from previous data and recognize data that changes [4]. One of the most widely used artificial neural network algorithms is backpropagation.

Backpropagation is an algorithm that has a strong basic mathematical [5]. Backpropagation is an algorithm that uses weight adjustment to achieve the minimum error value between predicted output and actual output [6].

In Figure 1, it can be seen that what acts as the input layer is X, and Z as the hidden layer with V is the weight between the input layer and the hidden layer. Y is the output layer, W is the weight between the hidden and the output layers, and 1 is the bias.

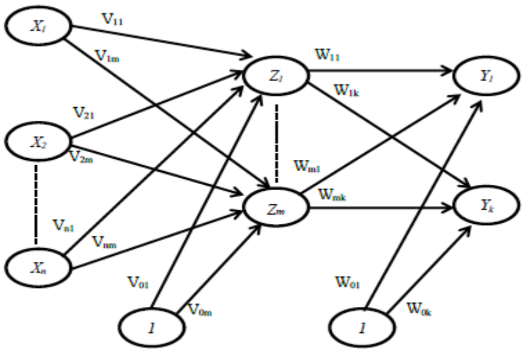


Figure 1 Backpropagation architecture.

2.1. Backpropagation Training Algorithm

The Backpropagation Artificial Neural Network training algorithm is as follows:

Step 0: Set the initial weight and bias to the lowest possible random value, set the minimum error.

Step 1: While the stopping condition is false, do steps 2 - 9.

Step 2: For each pair of training data, do steps 3-8.

Step 3: For each input unit (x_i, i = 1,2, ..., n), receive input x_i and spread the signal throughout the hidden layer neurons.

Step 4: Each hidden unit (z_j = 1,2, ..., p) sums the weight of the input signal with the following equation.

$$z_{in\_j} = V_{0j} + \sum_{i=1}^{p} x_i v_{ij}$$

with: v_{0j} = initial bias between the input layer and the hidden layer.

v_{ij} = initial weight between the input layer and the hidden layer.

Then calculate the output value using the activation function used with the following equation.

$$z_j = f(z_{in_j})$$

And sends a signal to all output layer units.

Step 5: Each unit of output neuron (y_k, k = 1,2, ..., m) sum the weight of the output signal.

$$y\_in_k = w_{0k} + \sum_{j=1}^{m} z_j w_{jk}$$

with: w_{0k} = initial bias between the hidden layer and output layer.

w_{jk} = initial weight between the hidden layer and output layer.

Calculate the output signal with the activation function in the following equation.

$$y_k = f(y_{in_k})$$

Step 6: For each output (y_k, k = 1,2, ..., m) that receives a target pattern that corresponds to the input pattern and then calculates the error value with the following equation.

$$\delta_k = (t_k - y_k) f'(y\_in_k)$$

with: t_k= Target Output

Calculate the weight correction using the following equation.

$$\Delta w_{jk} = \alpha \delta_k z_j$$

with: α = learning rate used in training.

Calculate the correction bias with the following equation.

$$\Delta w_{jk} = \alpha \delta_k$$

And send the signal to the previous layer.

Step 7: Calculate the hidden unit factor based on the error in each hidden unit (z_j, j = 1,2, ..., p).

$$\delta\_in_j = \sum_{k=1}^{m} \delta_k w_{jk}$$

Factor δ for hidden layer .

$$\delta_j = \delta\_in_j f'(z_{in\_j})$$

calculate the weight correction with the following equation.

$$\Delta v_{ij} = \alpha \delta_j x_i$$

Calculate the bias correction with the following equation.

$$\Delta v_{0j} = \alpha \delta_j$$

Step 8: Each neuron output (y_k, k = 1,2, ..., m) update bias and weight (j = 0.1, ..., p) with the following equation.

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

Each hidden unit (z_j, j = 1,2, ..., p) updates its bias and weight (I = 0,1, ..., n) with the following equation.

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

Step 9: Test the stopping condition based on epoch and error with equation 1 [7].

$$\text{MSE} = \frac{1}{2}\sum_{j=1}^{m}\left(t_{kj} - y_{kj}\right)^2 \qquad (1)$$

3. GRAY LEVEL CO-OCCURENCE MATRIKS (GLCM)

GLCM is a matrix whose elements are the probability of the appearance of gray levels i and j from two different pixels at a distance (d) and angular direction (θ = [0°, 45°, 90°, 135°]) [8]. A neighboring pixel with a distance d between them can be located in eight different directions, and this is shown in Figure 2.

To produce a concurrency matrix, first calculate the relationship between pixels based on the gray value of the grayscale image with the angular direction as shown in Figure 2, so that the resulting matrix concurrency with distance d and angle θ. Figure 3 shows how to calculate the concurrency matrix.

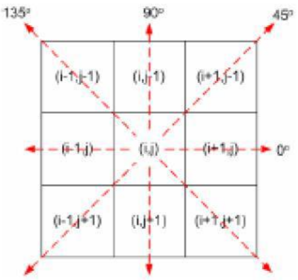
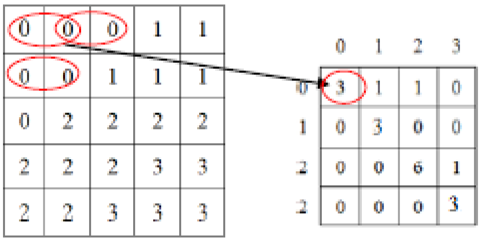


Figure 2 Neighbor relationship between pixels.



0	0	0	1	1
0	0	1	1	1
0	2	2	2	2
2	2	2	3	3
2	2	3	3	3

	0	1	2	3
0	3	1	1	0
1	0	3	0	0
2	0	0	6	1
2	0	0	0	3

(a) (b)

Figure 3 (a) Grayscale image values with 4 levels of gray level, (b) The concurrency matrix with the distance *d* = 1 and the angular direction *θ* = 0°.

After obtaining the concurrency matrix, the symmetric matrix is calculated by summing the concurrency matrix with the transpose, as follows.

$$\begin{bmatrix} 2 & 2 & 1 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 2 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}$$

After obtaining the symmetric matrix, normalize the symmetric matrix by dividing each element by the sum of all elements as follows.

$$\begin{bmatrix} \frac{4}{24} & \frac{2}{24} & \frac{1}{24} & \frac{0}{24} \\ \frac{2}{24} & \frac{4}{24} & \frac{0}{24} & \frac{0}{24} \\ \frac{1}{24} & \frac{0}{24} & \frac{6}{24} & \frac{1}{24} \\ \frac{0}{24} & \frac{0}{24} & \frac{1}{24} & \frac{2}{24} \end{bmatrix}$$

After obtaining the normalization matrix, the texture value calculation is carried out using 14 GLCM features [9]. In general, research on pattern recognition using GLCM as a feature extraction method uses only five features because these five features are the main features in GLCM [10]. The five features are as follows:

a. Entropy, the entropy value shows the randomness of the gray degree distribution of an image. Entropy is defined as:

$$\text{Entropy} = -\sum_i \sum_j P_{ij} \log P_{ij}$$

b. Energy, commonly called ASM (Angular Second Moment), measures the uniformity of an image or value used to calculate the concentration of pair intensity on a matrix.

$$\text{ASM} = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (P_{ij})^2$$

c. Contrast, contrast is a measure of the spread of matrix elements in an image. Contrast is a value that is used to calculate the inaccuracy of the intensity of an image.

$$\text{Contrast} = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (i-j)^2 . P_{ij}$$

d. Homogeneity, homogeneity or Inverse Different Moment shows the homogeneity of an image that has a similar gray.

$$\text{IDM} = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} \frac{1}{1+(i-j)^2} . P_{ij}$$

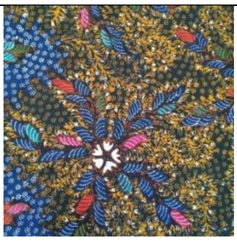
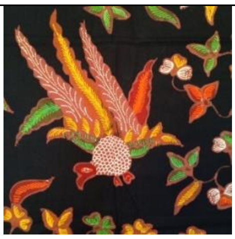
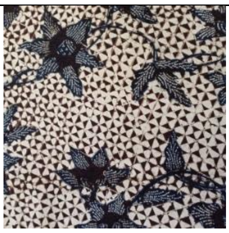
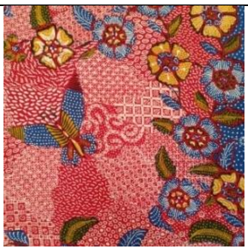
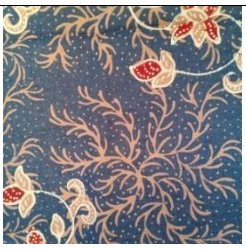
e. Correlation, correlation is a linear dependency relationship between values on the gray level in the image.

$$\text{corr} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{(i-\mu_i)(j-\mu_j)p_{ij}}{\sigma_i \sigma_j}$$

4. DATA

The data used in this study are five different Madura batik motifs, where each motif will be taken as many as 30 different images. In order to obtain 150 images of Madura batik motifs. Of the 150 images divided by 100 for training images and 50 as test images. Table 1 is an example of Batik data used for each motif.

Table 1. Madura batik motifs and labeling

No.	Name of Batik	Picture	Class	Label
1	Batik Bunga Satompok		Class 1	[1;0;0]
2	Batik Manuk Poter		Class 2	[0;1;0]
3	Batik Pecah Beling		Class 3	[0;0;1]
4	Batik Sekar Jagat		Class 4	[1;1;0]
5	Batik Rumput Laut		Class 5	[1;0;1]

5. RESULT AND DISCUSSION

The recognition of Madura batik patterns with the Backpropagation Algorithm starts with the preprocessing stage, feature extraction with GLCM, and classification using the Backpropagation algorithm. The data used are five classes of Madura batik motifs taken using a smartphone camera. The preprocessing stage includes the scaling and gray scaling processes. Figure 4 is an example of an image that has been carried out in the preprocessing stage.

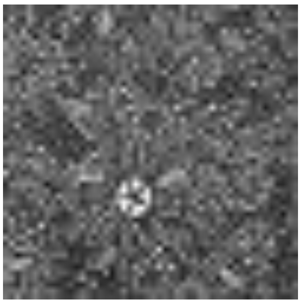


Figure 4 Preprocessing image.

The next step is feature extraction using the GLCM method. This stage produces values from four GLCM features with distance $d = 1$ and angular direction (θ = [0 °, 45 °, 90 °, 135 °]) as in Table 2.

Table 2. GLCM value

	0°	45°	90°	135°
Contrast	0,943	1,231	0,86	1,321
Correlation	0,866	0,825	0,879	0,812
Energy	0,067	0,062	0,070	0,061
Homogeneity	0,754	0,734	0,767	0,725

The training stages use the data in Table 2 as input. Table 3 shows some of the neural network architectures that will be tested.

Table 3. Neural network architectures

Hidden Layers	Hidden Neurons in Each Layer		
	Architecture 1	Architecture 2	Architecture 3
5	32	16	8
3	32	16	8
1	32	16	8

Based on Table 3, then testing is carried out using data different from the training data. The testing results for five hidden layers and 32 neurons are displayed with the Confusion Matrix in Table 4.

Table 4. Confusion matrix

Classes	1	2	3	4	5
1	10	0	0	0	0
2	0	10	0	0	0
3	0	0	10	0	0
4	0	1	0	9	0
5	0	0	0	0	10

The accuracy is as follows:

$$accuracy = \frac{49}{50} \times 100\% = 98\%$$

The accuracy of testing from all architectures is shown in Table 5.

Table 5. Accuracy from all architectures

Hidden Layers	Hidden Neurons In Each Layer			Average
	32	16	8	
5	98%	96%	94%	96%
3	96%	96%	94%	96,33%
1	98%	96%	96%	96,67%
Average	97,33%	96%	96,67%	

The number of hidden layers and hidden neurons and the initial weights and biases used in training affect the process of image recognition. In this study, hidden layers and hidden neurons are determined manually by the researcher. In Table 5, it can be seen that the architecture using five hidden layers with 32 hidden neurons in each layer has the highest accuracy of 98%.

6. CONCLUSION

The highest accuracy is obtained in architecture using five hidden layers with 32 hidden neurons in each layer equal to 98%. The lowest accuracy is obtained in architecture using five hidden layers with eight hidden neurons and three hidden layers with eight hidden neurons equal to 94%.

REFERENCES

[1] Y. Brasilka, Ernawati, D. Andreswari, Klasifikasi Citra Batik Besurek Berdasarkan Ekstraksi Fitur Tekstur Menggunakan Jaringan Saraf Tiruan Self Organizing MAP (SOM) (in Indonesian), Jurnal Rekursif, 2015, vol. 3, no. 2, pp. 132-145.

[2] A. Desiani, Kajian Pengenalan Wajah dengan Menggunakan Metode Face-Arg dan Jaringan Saraf Tiruan Backpropagation (in Indonesian), Media Informatika, 2007, vol. 5, no. 2, pp. 99-111.

[3] Sulistiani, E. Winarko, Klasifikasi Pola Sidik Jari Menggunakan Jaringan Saraf Tiruan Backpropagation (in Indonesian), Berkala MIPA, 2014, vol. 24, no. 4, pp. 215-224.

[4] B. Poernomo, Y. Melita, Pengolahan Citra Digital Untuk Pengenalan Retina dengan Jaringan Saraf Tiruan Hopfield Diskrit (in Indonesian), Jurnal Ilmiah Teknologi dan Informasi ASIA, 2013, vol. 7, no. 1, pp. 37-47.

[5] S. Redjeki, Perbandingan Algoritma Backpropagation dan K-Nearest Neighbor (K-NN) untuk Identifikasi Penyakit (in Indonesian), in: Proceedings of the Seminar Nasional Aplikasi Teknologi Informasi (SNATI), 2013.

[6] A. A. Kasim, A. Harjoko, Klasifikasi Citra Batik Menggunakan Jaringan Saraf Tiruan Berdasarkan Gray Level Co-occurence Matrices (GLCM) (in Indonesian), in: Proceedings of the Seminar Nasional Aplikasi Teknologi Informasi (SNATI), 2014.

7] L. Fausett,. Fundamental Of Neural Network: Architectur, Algoritm, and Aplication, New Jersey: Prentice Hall, 1993.

[8] T. W. A. Putra, K. Adi, R. R. Isnanto, Pengenalan Wajah Dengan Koonkurensi Aras Keabuan dan Jaringan Saraf Tiruan Probabilistik (in Indonesian). Jurnal Sistem Informasi Bisnis 02, 2013, vol. 3, no. 2.

[9] H. Wibawanto, A. Susanto, T. S. Widodo, S. M. Tjokronegoro, Identifikasi Citra Massa Kistik Berdasarkan Fitur Gray-Level Co-occurrence Matrix (in Indonesian), in: Proceedings of the Seminar Nasional Aplikasi Teknologi Informasi 2008 (SNATI 2008), 2008.

[10] X. Liu, K. Clarke, M. Herold, Popular Density and Image Texture: A Comparison Study America, Photogrammetric Engineering & Remote Sensing, 2006, vol. 72, no. 2, pp. 187-196.