International Journal of Innovative Computing, Information and Control
Volume **16**, Number **1**, February **2020**

ICIC International ©2020 ISSN 1349-4198
pp. 301–313

VISUAL TRACKING-BASED HAND GESTURE RECOGNITION USING BACKPROPAGATION NEURAL NETWORK

Budi Sugandi, Siska Eka Octaviani and Nanta Faqih Pebrianto

Electrical Engineering Department
Politeknik Negeri Batam
Jl. Ahmad Yani, Batam Center, Batam 29461, Indonesia
budi_sugandi@polibatam.ac.id

Received April 2019; revised August 2019

Abstract. *This article proposed a hand gesture recognition using backpropagation neural network based on visual tracking. The proposed algorithm is passed through three stages, namely, hand region extraction, hand feature extraction and hand gesture recognition. The hand region is extracted from the background using skin color detection based on YCbCr color filter. The extracted hand region is then converted to gray scale and binary image to speed up processing time. The hand feature is obtained from the binary image of the detected hand region by dividing the feature into six regions. The last stage is the recognition of the hand gesture which is performed based on the backpropagation neural network. The six regions of the hand feature of each gesture become the input data of the neural network. In order to accommodate the six regions of the hand feature, the proposed method implemented six nodes on the input layer. Number of the hidden layers is two with ten nodes on each. The output layer has four nodes to accommodate ten output of the recognitions process. The hand gestures to be recognized are a set of ten hand gestures, namely: Stop, One, Two, Three, Four, Hello, Yes, No, OK and Call. The experiments on each hand gesture showed that our proposed algorithm can reach the good performance of recognition rate with minimum result of 80%, maximum result of 94% and average result of 86.67%.*
Keywords: Hand gesture, Color filter, Visual tracking, Backpropagation neural network

1. **Introduction.** The hand gesture recognition is one of the image processing areas which are developed in many applications such as sign language for disabled people [1], virtual or augmented reality [2,3], robot control [4], and game application [5]. The hand gesture can be a symbol to represent a word or expression such as sign language. The sign language can assist the disabled people to communicate to other persons. There are many sign languages in use around the world today such as American Sign Language, British Sign Language, and Japanese Sign Language. Each country generally has its own native sign language.

The current studies of hand gesture recognition are proposed by many scientists. Badi et al. [6] proposed feature extraction technique for static hand gesture recognition. In their article, the system could recognize six hand gestures. They proposed three stages of hand gesture recognition, namely, pre-processing, feature extraction and classification. Lekova and Adda [7] proposed hand gesture recognition based on signal cross-correlation. They presented the framework for hand gesture featuring, profiling and recognizing based on signal processing and cross correlation of detected signals. They recognized the hand gesture by matching to the hand joint profiles in the database by fast signals cross-correlation. Elakkiya et al. presented the hand gesture application for human computer

DOI: 10.24507/ijicic.16.01.301

interaction [8]. They developed a system to be interacted with computers without using mouse click and keystroke. They used fuzzy neural network to do the recognition and classification of the gesture. The similar systems was developed by Sharma and Verma [9] and Hariaa et al. [10]. In another study, Jin et al. [11] presented mixed hand gesture and its application. They used state-based spotting algorithm to distinguish gesture from random sequence. Zhang et al. implemented fuzzy c-means algorithm (FCM) to classify the hand gesture which is extracted based on unequal-probabilities background difference [12].

One of important stages in recognition is feature extraction in which the goal is to determine the discriminating data to be processed for the next stage. The most common feature used in image processing is color feature, shape, texture and spatial data. The color feature [13,14] presents the method to detect hand based on skin color using some common color filters such as: RGB, HSV and YCbCr. The shape feature [15] is used to extract the shape of the object based on contour and region extraction method. The texture feature is extracted based on Gabor filter and wavelet transforms [16]. The spatial feature usually used convolution mask that is applied to the image to get the object region of interest. Some of the studies combine each feature to detect the object [17].

The artificial neural network (ANN) is commonly used for classification and recognition process. ANN is one of the smart systems which have capability to learn unknown relationship which is available in prior condition. One type of ANN is backpropagation neural network (BPNN). BPNN is supervised learning algorithm which has many layers (multi layers neural network). BPNN used error output to change the weights in backward phase. In order to obtain the error, the forward phase should be done first [18]. There are three phases in BPNN namely, forward phase, backward phase and weight modification. In forward phase, input pattern is calculated forwardly from input layer till output layer. In backward phase, each output node receives output pattern related to input pattern to be calculated the error between them. This error will propagate backwardly. Weight modification aims at reducing the error between output and input. The three phases are iterated till the stop condition is achieved.

Some related studies on hand recognition based on BPNN have already been proposed in the past. Murthy and Jadon [19] proposed a BPNN method for classifying hand gestures into ten categories. They converted the image into binary image and created the 3D Euclidian space as input of BPNN. They achieved 89% of correct results. Hasan and Kareem [20] proposed a static hand gesture recognition using BPNN method to recognize six static hand gestures. They extracted the contour of hand gesture and used it as a feature for BPNN. They implemented contour scaling and translation to reduce the problem. They achieved the performance of 86.38% recognition rate. Badi [21] proposed the similar method to recognize six hand gestures. They proposed the complex moment algorithms to describe hand gesture and treat the rotation problem. They achieve better performance of 86.90% recognition rate.

In this paper, we present a hand gesture recognition based on BPNN algorithm. The proposed algorithm has three stages which combine the hand detection using YCbCr color filter, the hand gesture feature extraction using six regions of detected hand in binary image and the hand gesture classification and recognition using BPNN. The first stage is the hand region detection which is extracted from the background using skin color detection based on YCbCr color filter. The extracted hand region is then converted to gray scale and binary image to speed up processing time. The hand feature is extracted from the binary image of the detected hand region by dividing it into six regions. The last stage is the recognition of the hand gesture which is performed based on the BPNN algorithm.

The combination of detection method using YCbCr color filter, feature extraction of hand region using six regions feature and recognition of ten hand gestures using BPNN algorithm are the main contributions of the study. Unlike the previous methods, we implemented the hand detection using YCbCr color filter that is robust to the change of light condition. We find that the six regions of the hand region can be considered as extracted feature to be recognized using BPNN. We implemented our work in the real time processing based on visual sensing by camera.

This paper is organized as follows. The detail of our proposed methods and some image processing techniques for detecting and recognizing the hand gesture are presented in Section 2. Experimental results employing our proposed method are illustrated in Section 3. Finally conclusion and discussion are given in Section 4.

2. Method.

This article proposed hand gesture recognition based on back propagation neural network. We divide the procedure into three steps: hand region extraction, hand feature extraction and hand gesture recognition. In the first step, we implemented the skin color extraction using YCbCr color filter. The YCbCr color filter has advantage to extract the skin color compared to another color. The YCbCr color filter is implemented after the image is captured by camera in RGB image. The input image can be still image, recorded video or real time video. The filtering image is then converted to gray scale image and finally becomes binary image. In the second step, we employed a spatial feature to be extracted from the extracted hand gesture. The feature extraction is performed by dividing the detected hand region in binary image into six regions. The pixel value of each six regions in binary image is then counted to be the training data for BPNN. Therefore each hand gesture will have six sets of data. And finally in the last step, the hand gesture is recognized using BPNN. The input data is obtained from the extracted hand of the six regions in the binary image. In order to accommodate data from extracted feature which have six sets of data, we implemented the BPNN with six nodes in input layer and two hidden layers with ten nodes. Four nodes in output layer implemented to accommodate ten output of hand gestures recognition or four digit of binary number. The overall system proposed in this paper is described in Figure 1.

2.1. Hand gesture region extraction.

The main aim of the hand gesture extraction is to extract the hand region separated from background region. Firstly, the hand region is extracted using YCbCr color filter. YCbCr is a color filter that is widely used in computer vision especially to detect the skin color.

The YCbCr color consists of luminance information (Y) and color information which is stored as two components (Cb and Cr). The Cb component is the color component that represents the difference between a blue component and reference color. The Cr component represents the difference between red and reference color. The color conversion from RGB to YCbCr [22] is described in (1).

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.299 & -0.587 & 0.886 \\ 0.701 & -0.587 & -0.114 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1}$$

In order to speed up the processing time, the extracted hand region is then converted to gray image and binary image. Binary image only has two values black and white, in which the white color pixel belongs to the hand region. The color conversion to gray and

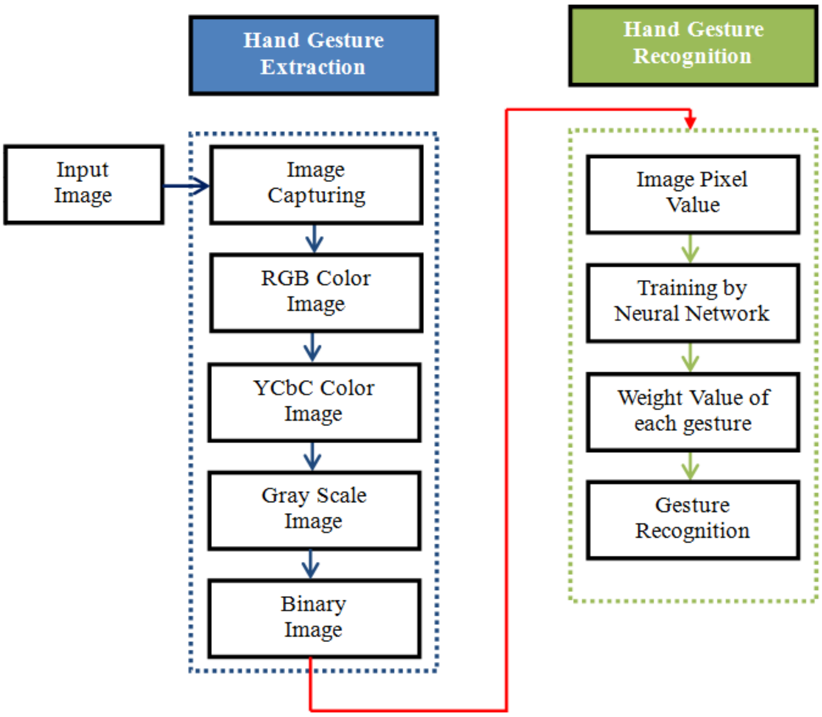


FIGURE 1. Overall system flow

binary image are described in (2).

$$\begin{gathered} I_{Gray} = 0.299 I_R + 0.587 I_G + 0.144 I_B \\ I_{Binary} = \begin{cases} 1, & \text{if } I_{Gray} \geq Threshold \\ 0, & \text{if } I_{Gray} < Threshold \end{cases} \end{gathered} \tag{2}$$

The result of hand extraction using YCbCr color filter and its conversion in gray and binary are shown in Figure 2. The left figure shows the result of YCbCr color filter which extracts only the hand region. The middle and the right figure show the conversion image to gray scale and binary image, respectively.

(a) Hand region in YCbCr color image

(b) Hand region in gray scale image

(c) Hand region in binary image

FIGURE 2. (color online) Hand region extraction on YCbCr, gray and binary image

2.2. **Hand gesture feature extraction.** One of important stages in recognition is feature extraction in which the goal is to determine the discriminating data to be processed for the next stage. We employed a spatial feature to be extracted from the extracted hand gesture. The feature extraction is performed by dividing the detected hand region in binary image into six regions as described in Figure 3. Region 1 is the top-left region, region

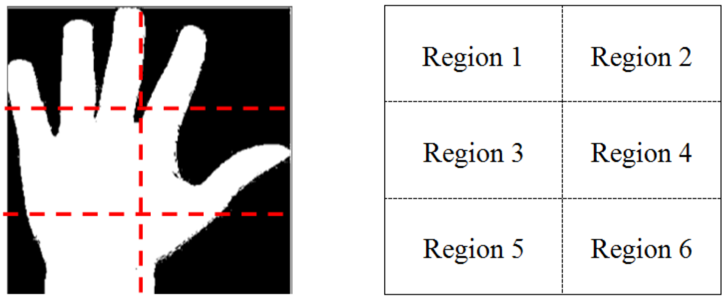


FIGURE 3. Hand gesture feature extraction

2 is the top-right region, region 3 is the middle-left region, region 4 is the middle-right region, region 5 is the bottom-left region and region 6 is the bottom-right region. The pixel value of each region is saved into data base as reference data for training process in hand gesture recognition. Each sample data will have six sets data to be matched with each training data.

2.3. **Hand gesture recognition.** In this study, a BPNN algorithm is employed to recognize the hand gesture. BPNN is widely used in computer vision to recognize the object based on object pattern. BPNN is one of the supervised learning algorithms used in multilayer perceptron. Each weight of the neuron is changed in each hidden layer. BPNN algorithm used output network error to change the weight in the backward direction. In order to obtain the backward error, the forward propagation should be done first. In the forward propagation the neurons are activated using sigmoid activation function as described in (3).

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

where, e is epsilon and $x = \sum_i^n X_i(p) \cdot W_{ij}(p)$, with X_i is the input of node i and W_{ij} is weight of the input i that transfer to node j in the next layer and n is number of neurons. The explanation is described in Figure 4. The sigmoid function will accept the single value and will change the value within 0-1.

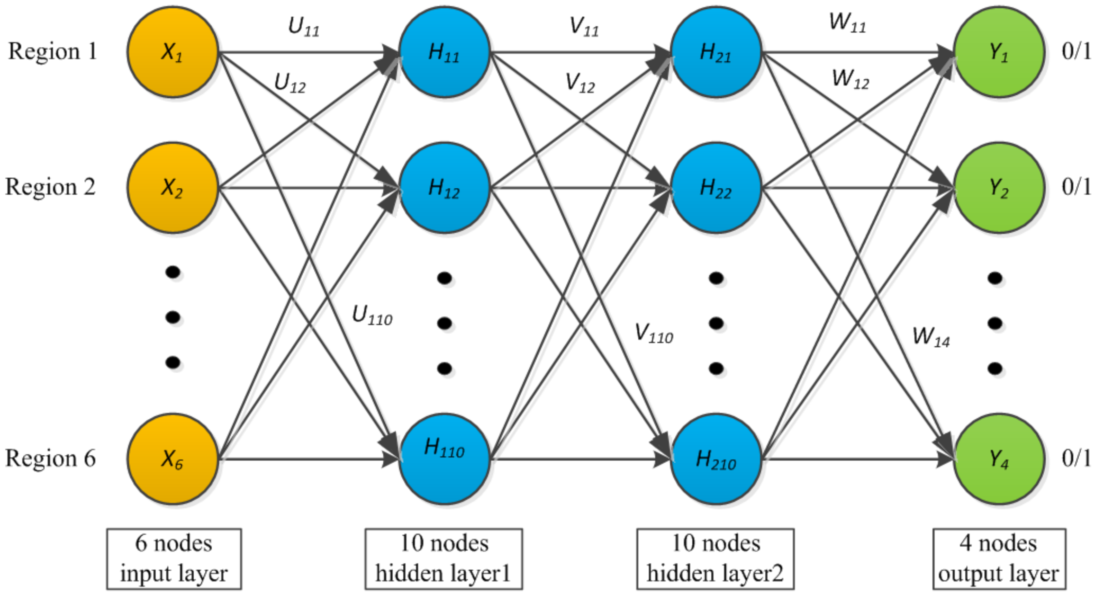


FIGURE 4. BPNN architecture using two hidden layers

The architecture of BPNN used in this paper is described on Figure 4. We have 6 nodes of input layers to accommodate six regions of hand gesture feature in which region 1 is input of X_1, region 2 is input of X_2 and region 3 is input of X_3 and so on. The hidden layer consists of 2 layers with 10 nodes in each. And the output layer has 4 nodes which consist of binary output 0 or 1 to accommodate ten hand gestures to be recognized.

The algorithm of BPNN used in this paper is described as the following.

1) Weight initialization which is taken from initial weight with small random.
2) While not Stop-Criterion do:

Feed forward:

a. Each node of input layer (X_i, $i = 1, 2, 3, \ldots, 6$) receives X_i signal to be transferred to the upper layer (hidden layer).

b. Each node of hidden layer 1 (H_{1j}, $j = 1, 2, 3, \ldots, 10$) sums the weighted input signal as described in (4).

$$H_{1\_inj} = \sum_{i=1}^{6} X_i \cdot U_{ij} \tag{4}$$

The sigmoid activation function is used to calculate the output signal of hidden layer in (5).

$$H_{1j} = f(H_{1\_inj}) \tag{5}$$

c. Each node of hidden layer 2 (H_{2j}, $j = 1, 2, 3, \ldots, 10$) sums the weighted hidden layer 1 output signal as described in (6).

$$H_{2\_inj} = \sum_{i=1}^{10} H_{1i} \cdot V_{ij} \tag{6}$$

The sigmoid activation function is used to calculate the output signal of hidden layer as described in (7).

$$H_{2j} = f(H_{2\_inj}) \tag{7}$$

d. Each node of output layer (Y_k, $k = 1, 2, 3, 4$) sums the weighted hidden layer 2 signal output as calculated in (8).

$$Y\_in_k = \sum_{i=1}^{10} H_{2i} \cdot W_{ik} \tag{8}$$

The sigmoid activation function is used to calculate the output signal of output layer as described in (9).

$$Y_k = f(Y\_in_k) \tag{9}$$

e. Each node of output layer (Y_k, $k = 1, 2, 3, 4$) receives the target pattern (T_k) which is related to the input learning pattern. The gradient error is calculated using (10).

$$\delta_k = (T_k - Y_k) \cdot Y_k \cdot (1 - Y_k) \tag{10}$$

The gradient error of each hidden layer is calculated using (11).

$$\delta_j = \delta_k \cdot W_{jk} \tag{11}$$

Backpropagation:

a. Updating weight from input layer to hidden layer 1 using (12).

$$U_{ij}(current) = U_{ij}(old) \cdot \mu \cdot \delta \cdot X_i \tag{12}$$

μ is a learning rate.

b. Updating weight from hidden layer 1 to hidden layer 2 using (13).

$$V_{ij}(current) = V_{ij}(old) \cdot \mu \cdot \delta \cdot H_{1j} \tag{13}$$

c. Updating weight from hidden layer 2 to output layer using (14).

$$W_{ij}(current) = W_{ij}(old) \cdot \mu \cdot \delta \cdot H_{2j} \tag{14}$$

3) End While.

3. **Experimental Results.** In order to evaluate our proposed method, the experiments were done to extract and to recognize the hand gesture region. The first experiment was to extract ten hand gestures in the real time application. The extracted data will be the sample data for BPPN. The BPPN will then save the sample data to be matched to test data and recognize it into ten hand gestures.

3.1. **Sample data set.** The sample data set is taken within 1000 data for 10 hand gestures. Each gesture has 100 input data. The 10 hand gestures are *Stop*, *One*, *Two*, *Three*, *Four*, *Hello*, *Yes*, *No*, *OK*, *Call* as described in Figure 5.

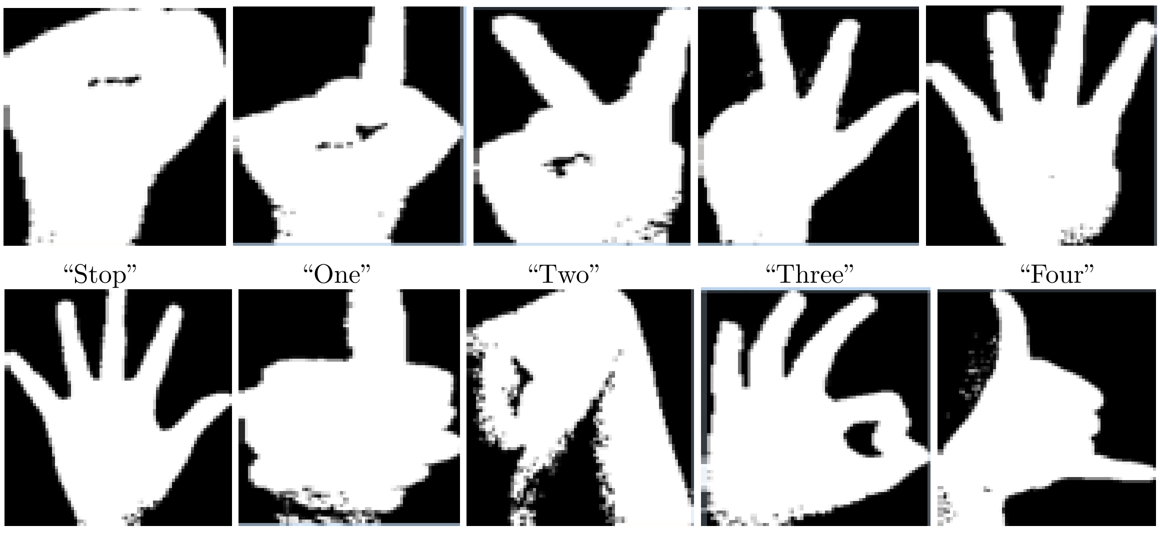


Figure 5. Ten hand gestures characteristic

The pixels number of each gesture is too many to be able to be trained in the neural network framework. In order to speed up the processing time of the neural network in the training and recognition process, the region of the hand gesture is normalized. The region of the hand gesture is divided into six regions. The learning data of each part is taken ten times. The sample result of normalized pixel value of the hand gesture is described in Figure 6. The X axis shows the region 1 until region 6 of each gesture. The Y axis shows the number of pixels on each region which has been normalized. Sample 1 until sample 10 show the training number of each gesture in one region. Each gesture has significant difference training data that can be the feature of each gesture.

Based on each gesture training data, we have hand gesture pattern which can be distinguished each other. The BPNN compares each data of the gesture which has the similar feature. In order to classify each gesture, each output neural network is defined using Table 1. The gesture classification is obtained from output of BPNN that has 4 nodes in output layer. We can classify each gesture in four bit numbers such as Stop 0000, One 0001, Two 0010, and Three 0011.

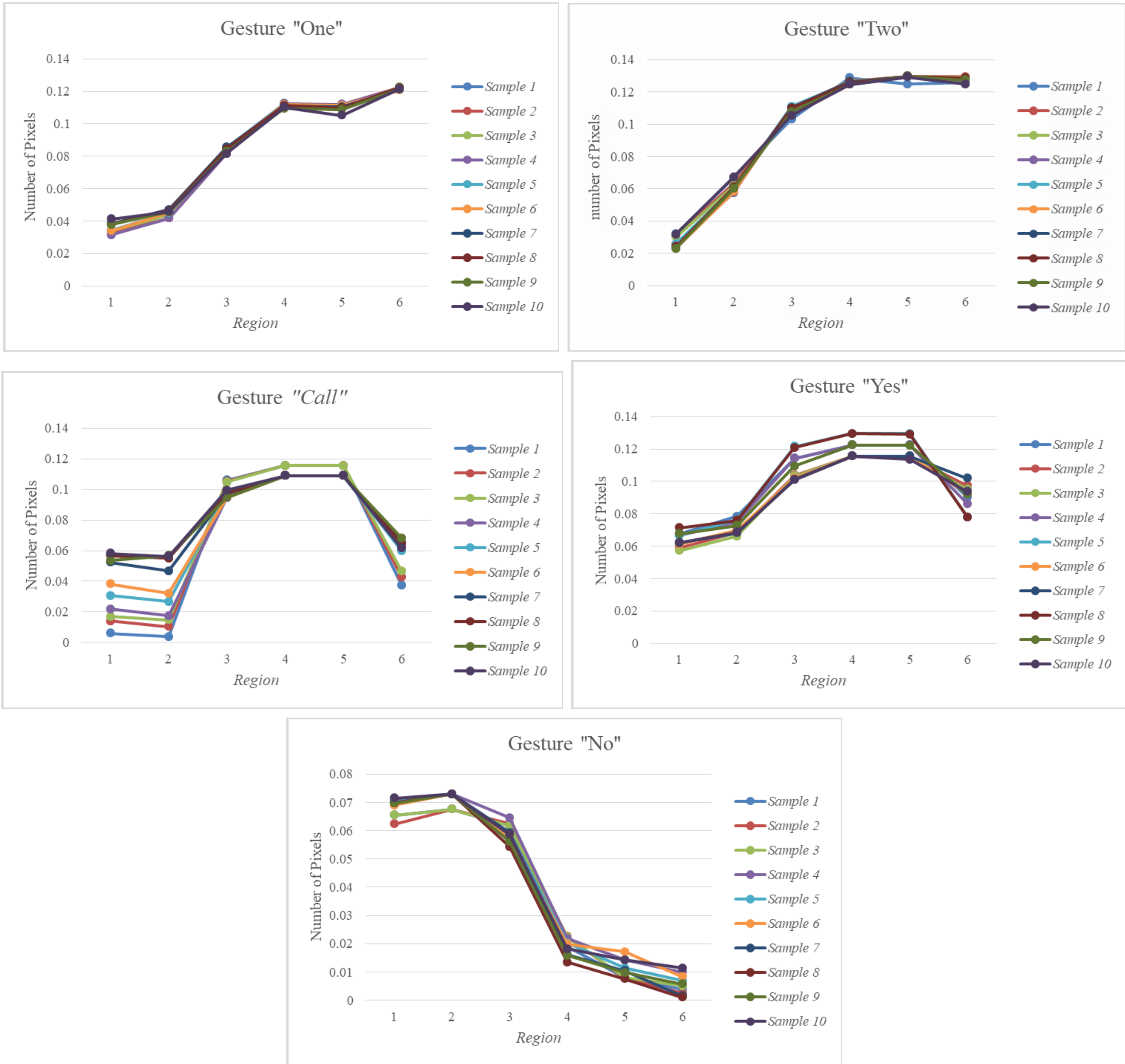


FIGURE 6. (color online) Five of ten hand gestures data

TABLE 1. Gesture classification

No	Gesture	Output setting			
1	*Stop*	0	0	0	0
2	*One*	0	0	0	1
3	*Two*	0	0	1	0
4	*Three*	0	0	1	1
5	*Four*	0	1	0	0
6	*Hello*	0	1	0	1
7	*Yes*	0	1	1	0
8	*No*	0	1	1	1
9	*OK*	1	0	0	0
10	*Call*	1	0	0	1

3.2. **Training parameters.** In order to obtain the optimal parameters of neural network, the parameters of neural network are evaluated. The parameters were obtained by experiment in several times to get the optimal result. Based on the experiment, the number of hidden layer 1 and hidden layer 2 which get optimal result are 10. We did the training three times using that number of hidden layer 1 and hidden layer 2 and we obtained the parameter of the neural network as shown in Table 2. Figure 7 shows the GUI (Graphic User Interface) of the application to calculate the training parameters of BPNN. By entering the value of each parameter on the top-left of application such as number of input node 6, output node 4 and the training parameter in Table 2, the system calculates the number of iterations and error of the BPNN and displays it on the bottom-right of application. For example, on training number 1 we obtain number of iterations 33211000 and error 0.0000999945 as well as on training number 2 and 3 we obtain number of iterations 33432000 and error 0.00000998701 and number of iterations 33645000 and error 0.00000981745, respectively.

Table 2. BPNN training parameters

Training number	Training parameters					
	Hidden layer 1	Hidden layer 2	Target error	Learning rate	Number of iterations	Error
1	10	10	0.00001	0.3	33211000	0.00000999945
2	10	10	0.00001	0.2	33432000	0.00000998701
3	10	10	0.00001	0.1	33645000	0.00000981745

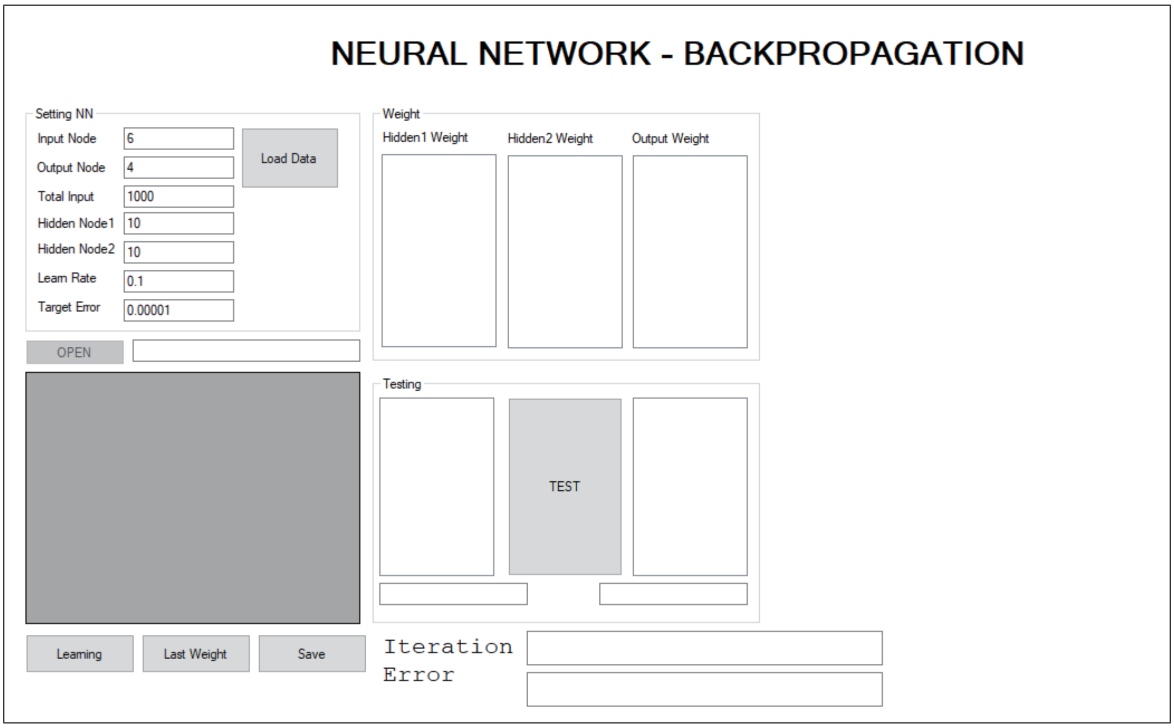
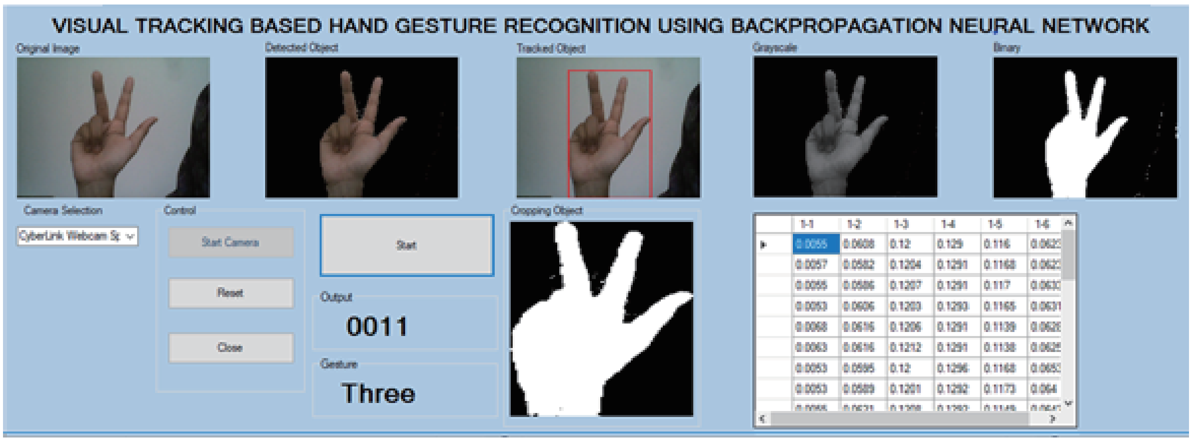
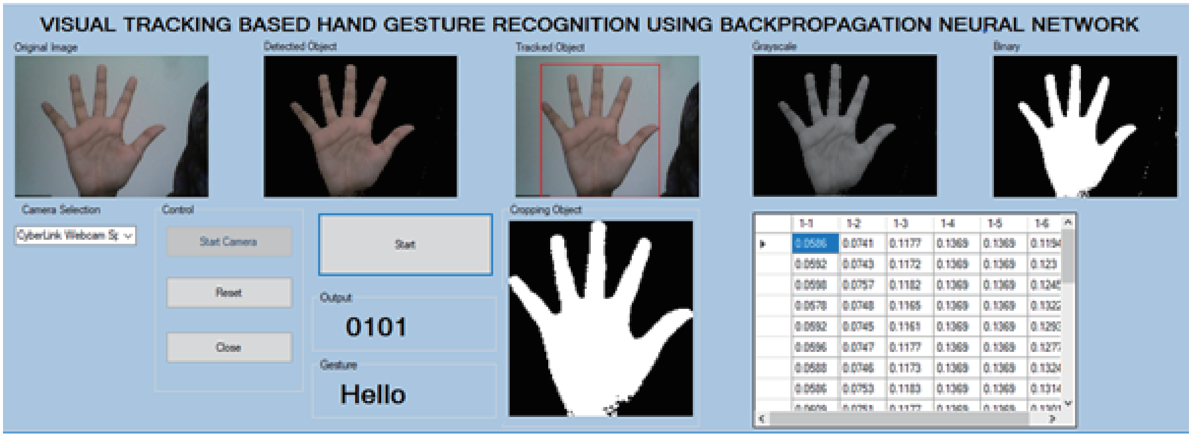


Figure 7. The GUI to calculate the BPNN parameters

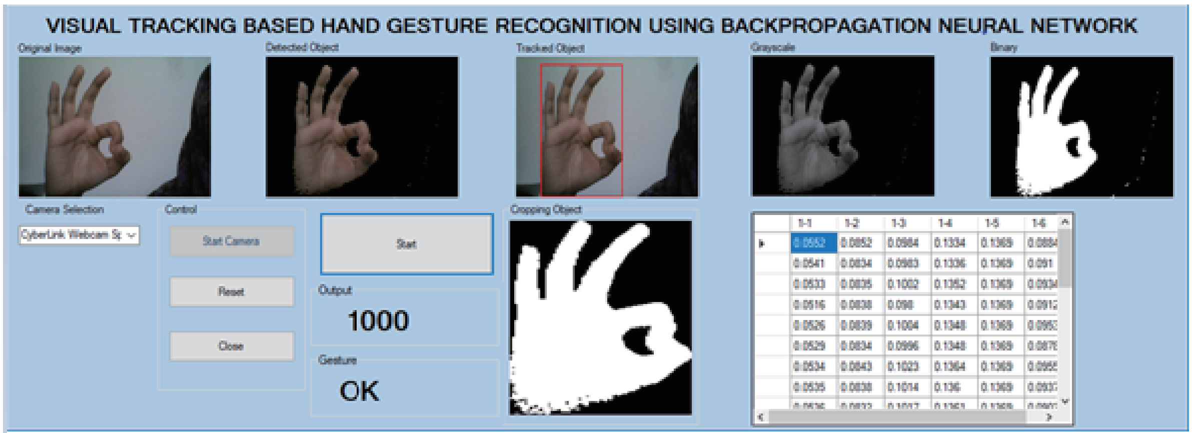
3.3. **Evaluation data.** After the neural network is trained using parameters in Table 2, the recognition process is then performed. The sample GUI of the gesture recognition application is shown in Figure 8. The GUI is real time-based application for hand gesture recognition using camera as input to capture the image. When the image of hand is detected as shown on picture box on the top-left, the recognition process is started with pushing the button Start. Then the image of hand will be detected and tracked by system



(a) Gesture "Three"



(b) Gesture "Hello"



(c) Gesture "OK"

FIGURE 8. Some of GUI of hand recognition application and experimental results

as shown on picture box of detected object and tracked object, respectively. The detected and tracked object will be converted to gray scale and binary image as shown on picture box of grayscale and binary image, respectively. The binary image is then cropped and extracted to obtain the hand feature of each gesture. The extracted feature is shown on the table on the bottom-right of GUI. If the feature is matching with the image feature in data base, then the gesture is recognized as gesture of data base as shown in Figure 5. The output of recognition is shown using 4 digit binary number such as 0011 and the recognized gesture is shown as the gesture such as "Three" as shown in Figure 8(a). In the next figures (Figure 8(b) and Figure 8(c)), the recognition process are the same and we obtain the recognition result as shown on those figures.

The recognition results are summarized in Table 3 and Table 4 for evaluation distance of 30 cm and 35 cm from camera, respectively with each learning rate. The higher accuracy is obtained when the learning rate is lower and object distance is closer to camera. Gesture recognition accuracy on distance of 35 cm and 30 cm has recognition accuracy in average of 82.7% and 90.7%. The highest accuracy is obtained on the distance from camera is 30 cm with recognition accuracy of 94% and the lowest accuracy is obtained on the distance from camera is 35 cm with recognition accuracy of 80%. The failure of hand recognition was caused especially by the motion of hand and a shadow of the hand to be recognized. The hand motion causes the test data is changed. Therefore it fails to be recognized as similar gestures. The other reason is that some detected gestures have similar feature which has difficulties to be recognized.

Table 3. Gesture recognition accuracy on distance of 35 cm

Gesture	Accuracy (%)		
	Learning rate = 0.3	Learning rate = 0.2	Learning rate = 0.1
Stop	80	90	90
One	70	80	80
Two	70	80	90
Three	80	70	70
Four	90	90	90
Hello	80	80	80
Yes	80	90	90
No	70	70	80
OK	90	90	90
Call	90	90	90
Average each learning rate	80	83	85
Average	82.7		

Table 4. Gesture recognition accuracy on distance of 30 cm

Gesture	Accuracy (%)		
	Learning rate = 0.3	Learning rate = 0.2	Learning rate = 0.1
Stop	100	100	100
One	80	90	90
Two	80	80	80
Three	80	80	90
Four	90	100	100
Hello	90	90	90
Yes	90	90	90
No	80	100	100
OK	90	90	100
Call	90	90	10
Average each learning rate	87	91	94
Average	90.7		

4. **Conclusions.** This article proposed a hand gesture recognition using BPNN based on visual tracking in real time environment. The algorithm has successfully implemented to recognize the hand gesture into ten gestures, namely: *Stop*, *One*, *Two*, *Three*, *Four*, *Hello*, *Yes*, *No*, *OK* and *Call*. The experiments on each hand gesture showed that our proposed algorithm can reach the good performance of recognition rate with minimum result of 80%, maximum result of 94% and average result of 86.67%. The highest accuracy was obtained on the distance 30 cm with recognition accuracy of 94% and the lowest was obtained on the distance 35 cm with recognition accuracy of 80%. However, the proposed methods still have limitations especially when the distance is far. Some gestures have similar features; therefore, they were recognized as wrong gesture. In real time system, the motion of the object and other environment condition such as object shadow have to be considered when designing a recognition system. These problems can be improved by a better feature selection method which can reduce the similar feature and reduce the change in environment. Taking them into consideration could lead to some improvement to our methods. These are remaining for the future works.

REFERENCES

[1] M. M. Hasan and P. K. Misra, Brightness factor matching for gesture recognition system using scaled normalization, *International Journal of Computer Science & Information Technology*, vol.3, no.2, 2011.

[2] A. Clark and D. Moodley, A system for a hand gesture-manipulated virtual reality environment, *International Conf. of the South African Institute of Computer Scientists and Information Technologists*, 2016.

[3] J. Weissmann and R. Salomon, Gesture recognition for virtual reality applications using data gloves and neural networks, *Proc. of the International Joint Conference on Neural Networks*, 1999.

[4] P. Wu and Q. Meng, Hand gesture recognition and its application in robot control, *International Journal of Computer Science Issues*, vol.13, no.1, 2016.

[5] C. Tan, S. Chin and W. Lim, Game-based human computer interaction using gesture recognition for rehabilitation, *IEEE International Conference on Control System, Computing and Engineering*, 2013.

[6] H. Badi, S. A. Kareem and S. Husien, Feature extraction technique for static hand gesture recognition, in *Recent Trends in Hand Gesture Recognition*, A. Chaudhary (ed.), Science Gate Publishing, Xanthi, Greece, 2015.

[7] M. Adda and A. Lekova, Hand gesture recognition based on signals cross-correlation, in *Recent Trends in Hand Gesture Recognition*, A. Chaudhary (ed.), Science Gate Publishing, Xanthi, Greece, 2015.

[8] R. Elakkiya, K. Selvamami, S. Kanimozhi, R. Velumadhava and A. Kannan, Intelligent system for human computer interface using hand gesture recognition, *International Conf. on Modeling, Optimization and Computing*, 2012.

[9] R. P. Sharma and G. K. Verma, Human computer interaction using hand gesture, *International Multi-Conference on Information Processing*, 2015.

[10] A. Hariaa, A. Subramaniana, N. Asokkumara, S. Poddara and J. Nayak, Hand gesture recognition for human computer interaction, *International Conf. on Advances in Computing & Communications*, 2017.

[11] S. Jin, Z. Yang, Y. Zheng, Y. Li, W. Chen and X. Zheng, Mixed hand gesture recognition system and its application, *International Journal of Innovative Computing, Information and Control*, vol.9, no.12, pp.4711-4723, 2013.

[12] Q. Zhang, J. Lu, H. Wei, M. Zhang and H. Duan, Dyanamic hand gesture segmentation method based on uneqal-probabilities background difference and improved FCM algorithm, *International Journal of Innovative Computing, Information and Control*, vol.11, no.5, pp.1823-1834, 2015.

[13] D. Avinash, K. Ghosh and S. Ari, Color hand gesture segmentation for images with complex background, *International Conf. on Circuits, Power and Computing Technologies*, 2013.

[14] P. Trindade, J. Lobo and J. P. Barreto, Hand gesture recognition using color and depth images enhanced with hand angular pose data, *International Conf. on Multisensor Fusion and Integration for Intelligent Systems*, 2012.

[15] P. Meenakshi, Hand gesture recognition based on shape parameters, *International Conf. on Computing, Communication and Applications*, 2012.

[16] X. L. Wang, X. Wang and L. Hu, A fast texture feature extraction method based on Gabor wavelets, *International Conf. on Comp. Vision and Pattern Recognition*, 2016.

[17] D. Vishwakarma, Hand gesture recognition using shape and texture evidences in complex background, *International Conf. on Inventive Computing and Informatics*, 2017.

[18] B. Krose and P. Smagt, *An Introduction to Neural Network*, The University of Amsterdam, Netherland, 1996.

[19] G. Murthy and R. Jadon, Hand gesture recognition using neural networks, *IEEE the 2nd International Advance Computing Conference (IACC)*, pp.134-138, 2010.

[20] H. Hasan and S. A. Kareem, Static hand gesture recognition using neural network, *Artificial Intelligent Review*, vol.41, pp.147-181, 2014.

[21] H. Badi, Recent method in vision-based hand gesture recognition, *International Journal of Data Science and Analytics*, vol.1, no.2, pp.77-87, 2016.

[22] S. Machekpsoshtia, M. Soltani, K. Raahemifarc, E. Bidakid and M. Sadeghie, PASI area and erythema scoring using YIQ color space, *Journal of Dermatology Research and Skin Care*, vol.1, no.1, 2017.