Comparison of Multilayer Feed Forward Neural Network and Hopfield Neural Network upon Buddhist Iconographies Classification

Thin Thin Aye, Thiri Thitsar Khaing
Computer University, Myitkyina
thinthin30@gmail.com, thirithitsarkhaing@gmail.com

Abstract

This paper proposes two types of artificial neural networks to classify the Buddhist iconographies in order to compare the performances of two networks; multilayer feed forward network and Hopfield network. The quality of the image of Buddhist iconography is improved by smooth filtering. The smooth filtered image is transformed into intensity image. The resultant gray image is edge detected to obtain the abstract graph or the edge of the Buddhist iconography. The resultant image is resized into the defined pixel area. This is the final step of the image preprocessing stage. The resized image is applied into the two networks. Multilayer network is composed of three layers. The input layer accepts the preprocessed images and the certain values of the output layer indicate the type of the Buddhist iconography. Hopfield network consists of a single layer which serves as both input and output layers. The training time and accuracy of the two networks are then analyzed and compared in this proposed paper.

1. Introduction

Pattern recognition techniques are important subject of the intelligent systems and can be applied in data processing as well as in decision making systems. The pattern recognition can be effectively manipulated by the artificial neural networks. The ANNs(Artificial Neural Networks) can be used to train the machine for the pattern recognition.

To highlight the importance of the pattern recognition techniques, this paper proposes a Buddhist Iconography classification system that identifies out the type of Buddhist Iconography; Bumisparsa, Abhaya, Amida, Buddhapatta, Dharmacakra using two types of ANNs, multilayer feedforward and Hopfield networks. Some example images of Buddhist Iconographies are illustrated in Figure 1.

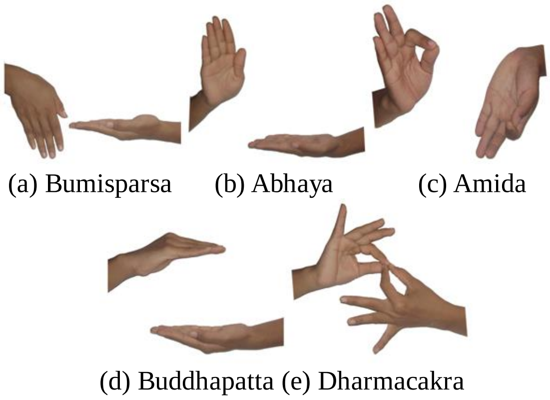


Figure 1. Basic buddhist iconographies

2. Related Works

Andrew B. Graham proposed the image processing algorithm for detection of two-dimensional visual code markers at Stanford University, Stanford, USA, in 2006[1]. The 3D building recognition using artificial neural network as the ICA workshop on generalization and multiple representation was carried out by Jagdish Lal, Liqiu Meng at Technical University of Munich, Germany, August in 2004[3]. M. M. Mahbubul Syeed, Fazlul Hasan Siddiqui and Abu Saleh Abdullah Al-Mamun developed the Bengali character recognition system using bidirectional associative memories (BAM) neural network at Islamic University of Technology, Board Bazar, Gazipur[5].

3.Image Processing Stages of the Proposed System

The images of the Buddhist Iconographies are acquired by a digital imaging device, such as, a digital camera or a digital scanner and are stored in the bitmap file format. The bitmap file format represents image by defining the colors of the pixels with different red, green and blue (RGB) values. Different colors can be accomplished by superposition of three additive primary colors of red, green and blue. [2]

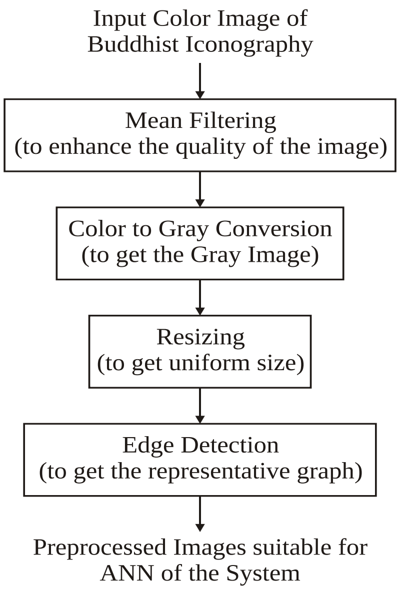


Figure 2. Image Processing Steps of the System

Figure 3. Mean filtered images of buddhist iconographies

The images of Buddhist Iconographies are different in qualities from the image processing point of view. The image acquisition step may also reduce the quality of the image. To get more precise images, the quality of acquired images is enhanced by means filtering image processing method. To get the mean or average color, a window of 3 x 3 pixels matrix slides through all over the image. While sliding the window through all over the image, the mean or average color of 3 x 3 pixels within the window is calculated out. This mean color value is the representative color of the current 3 x 3 pixels window and center pixel of the current window is replaced by this mean color. This sliding window mean filtering image enhancement reduces the color variations of the image and leads to remarkable enhancement to the image. [2]

Figure 4. Resultant images of color to gray conversion

The gray or intensity image contains the pixels only with 256 different gray values. Therefore, the mean filtered image is transformed into gray scale using the following equation:

Gray = 0.299*Red + 0.5876*Green +0.114*Blue

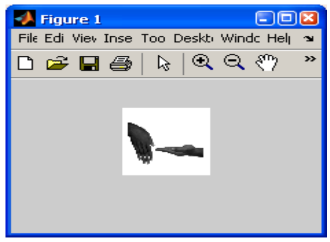


Figure 5. Resized gray images buddhist iconographies

To get uniform sized images of Buddhist Iconographies in order to construct a unique architecture of the ANN of the system, the gray images are resized into the predetermined pixel matrix. In this proposed system, the predetermined size is taken as 64 x 64 pixel matrix. This size has to be adjusted between the accuracy and speed of the system. If this size is small, though the system becomes faster, the system becomes lack of accuracy. If the size is large, the system's accuracy becomes increase but remarkable reduces the execution rate.

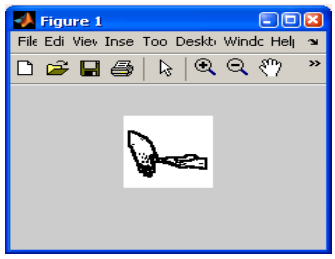


Figure 6. Edge images of buddhist iconographies

The Sobel edge detector is applied upon the resized gray images of Buddhist Iconographies to get the abstract or representative graph of the Buddhist Iconographies. This is the last stage of the image preprocessing portion of the system. After image preprocessing portion, the uniform sized edge images of Buddhist Iconographies are obtained to train as well as to test the ANNs of the System.

The Sobel edge detector is a nonlinear edge enhancement technique. It is another simple variation of the discrete differencing scheme for enhancing edges.

Let $\mathbf{a} \in \mathbb{R}^{X}$ be the source image, and *a*0, *a*1, …, *a*7 denote the pixel values of the eight neighbors of (*i*, *j*) enumerated in the counterclockwise direction as follows:

a_3	a_2	a_1
a_4	(i, j)	a_0
a_5	a_6	a_7

The Sobel edge magnitude image $m \in R^X$ is given by

$$M(i,j) = (u^2+v^2)^{1/2}$$

Where

$$u = (a_5+2a_6+a_7) - (a_1+2a_2+a_3)$$

and

$$v = (2a_0+a_1+a_7) - (a_3+2a_4+a_5)$$

4.Multilayer Feed Forward Neural Network of the Proposed System

In this proposed system, multilayer network is designed with three layers; input layer, hidden layer and output layer. The input layer accepts the prepared images of the Buddhist Iconographies from the image processing steps. The input layer has 4097 nodes to accept the pixel values from the 64 x 64 pixel matrix from image processing steps and extra bias node. The hidden layer memorizes the trained types of Buddhist Iconographies in order to recognize them. The hidden layer is constructed with 100 neurons to get the memory improvement of the ANN. The output layer is composed of five neurons since there are five basic types of Buddhist Iconographies, one output node for each Iconography. The corresponding certainty values of the neurons in the output layer decide the type of the Iconography.

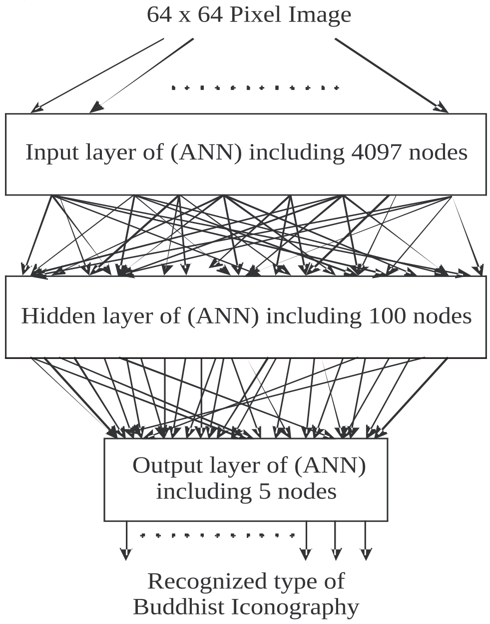


Figure 7. The architecture of ANN of the proposed system

The solution to a pattern recognition problem initially involves the collection of a database of training patterns. If these patterns have labels associated with them a supervised learning procedure is used to train the neural networks. The weights are adapted so as to create a mapping from input patterns to output values, such that the latter approximate the desired values (target values) as closely as possible over the whole training set. The recognition performance of the trained network is then evaluated on test data. If the network is able to recognize a large proportion of these new patterns correctly, then it is said to be capable of generalization. The aim is to achieve the best possible generalization performance, given the available training database.

The ANN of the system is initially trained with a training set using the initial small random weights. Primarily, one certain training set is used to train the network. The random weight values are set with small values between -0.5 and +0.5. According to the weights, the accepted pixel values as the input to the hidden layer of the ANN is changed.

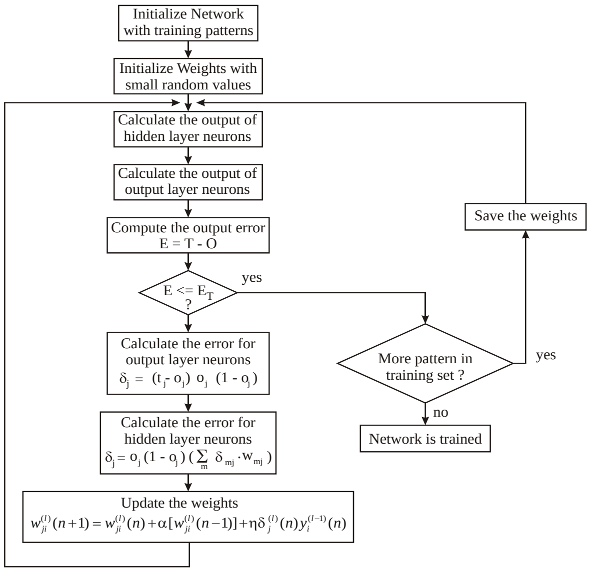


Figure 8. Flow chart of the ANN of the system.

There are 100 neurons in the hidden layer and each neuron in hidden layer has 4097 inputs from the input layer. The total net input for each neuron in the hidden layer is calculated by the following formula.[8]

$$v_k = \sum_{j=0}^{m} w_{kj} x_j$$

where j is the number of input neurons
k is the number of hidden neurons
V_k is the total net input to hidden layer
W_{kj} is the weight from input to hidden layer
X_j is the value from the input layer

From the calculated total net input, the output value of each neuron in the hidden layer is calculated by means of sigmoid function. [8]

$$\varphi(v) = \frac{1}{1 + \exp(-v)}$$

where $\varphi(v)$ is the output value of a neuron in the hidden layer

V is the total net input to hidden layer

The output of the hidden layer is applied as the input to the output layer. The total net input to the output layer as well as the output value of a neuron in the output layer are calculated out as the same for the hidden layer as described above. Then the output values of the neurons from the output layer of ANN are obtained and these values are in the range from 0 to 1. There are five output nodes in the output layer of the ANN, each for one Iconography. These values are the certainty values of each of the five Buddhist Iconographies. The targeted output value from the expected output node is examined and its value must be close to output value 1. If the certainty value of a neuron is high (close to 1) and all the other certainty values of the other neurons are low (close to 0), then the ANN is said to be well trained to recognize the corresponding Iconography. Hence, in the training stage, the obtained output values from the neurons in the output layer may somewhat different from the target values. These differences are the errors of the ANN and the error values are determined by the following formula.[8]

E = T – O

where, E is the error

T is the target output value

O is the current output value of the ANN

If the errors are not negligible, the applying weights are needed to adjust to be the errors negligible. The changes in errors of the output neurons and hidden neurons are computed. Depending upon the computed changes in errors, the weight values are adjusted. The adjusted weights are applied in the ANN and the above steps are repeated until the ANN becomes well trained with the Buddhist Iconographies.

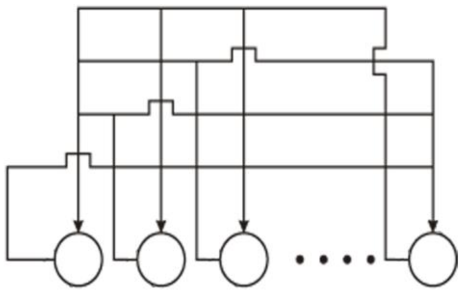
If the errors become negligible, the ANN is said to be trained with a single Buddhist Iconography. Then another Iconography from the current training set is used to train the ANN as above steps till all the Iconographies in the training set become well trained by the ANN.

5. The Hopfield Network of the Proposed System

The Hopfield network consists of a set of N interconnected neurons which update their activation values asynchronously and independently of other neurons. All neurons are both input and output neurons. The activation values are binary. In this system there are 4096 neurons to accept the pixel values from the image preprocessing portion of the system. The Hopfield network is an autoassociative network. Hopfield network has the following characteristics:

- A single layer of units.
- All units connect to every other unit, but a unit does not connect to itself.
- Only one unit updates at a time.
- Units are updated in random order, but each unit must be updated at the same average rate.
- The output of a unit is restricted to 0 or 1.

The Hopfield network is recurrent in that, for a given input pattern, the network's output is recirculated as input until a stable state is reached. [6]



4096 neurons

Figure 9. Architecture of the hopfield network of the system

Binary inputs can be used with Hopfield network but the presentation given here will use +1 to denote an ‘on’ state for a unit and -1 to denote an ‘off’ state. The net input to a unit is calculated as:

$$net_j = \sum_{i=1}^{n} s_i w_{ij}$$

where s_i is the state of unit i. when a unit is updated it will have its state modified according to the rule

$$s_j = \begin{cases} +1 & \text{if } net_j > 0 \\ -1 & \text{if } net_j < 0 \end{cases}$$

The above relation is known as the signum function and can be abbreviated to

$$s_j = \mathrm{sgn}(netj)$$

If the net input is zero then the unit stays in the state it was in prior to the update. An input vector provides the initial state of each unit. A unit is selected at random to be updated. The selected unit receives a weighted signal from all other units and updates its state. The network has converged when no unit changes state if selected for updating.

The Hopfield network acts as a memory and the procedure for storing a single vector is to take the outer product of the vector itself. This procedure produces a matrix which defines the weights for a Hopfield network provided all of the diagonal elements have been set to zero because the diagonal elements define the self-connections and a unit does not connect to itself. So the weight matrix to store a vector x is given by:

$$W = x^T x$$

6.Analytical Conclusion of the Comparison of Multilayer Perceptron Network and Hopfield Network

To evaluate the performance of the system, two experiments with MLP and HP have been carried out. We compare training time and classification rates between two methods. The result of training time with 20 images of each class of Buddhist Iconographies is depicted in Table (1).

Table 1. Training time of the two networks upon five basic buddhist iconographies

Buddhist Iconographies	Training Time	
	MLP	HP
Bumisparsa	2 sec	1.5 sec
Abhaya	2 sec	1.5 sec
Amida	2 sec	1.5 sec
Buddhapatta	2 sec	1.5 sec
Dharmacakra	2 sec	1.5 sec
Average	2 sec	1.5 sec

To estimate the robustness of the proposed systems, noises such as gaussian, speckle, poisson are first added to test images and then classified with two methods.

Table 2 . Recognition rates of the two network with noise added images upon five basic buddhist iconographies

Buddhist Iconographies	Recognition Rates	
	MLP	HP
Bumisparsa	100%	60 %
Abhaya	100%	70%
Amida	100%	70%
Buddhapatta	100%	80%
Dharmacakra	100%	100%
Average	100%	76%

According to the above results, they show us that the proposed approach based on MLP can recognize the images more efficiently than HP and give the imformation exactly to the user.

Architecturally, the MLP network, as its name implies, is a multi layer network. The MLP network can be constructed with at least three layers; input, hidden and output layers. The architecture of the MLP network is flexible and there may contain two hidden layers to get more powerful recognition of certain complex systems. Being multi layer network, there are many connections among input nodes and hidden neurons as well as among the hidden neurons and output nodes with corresponding weights assigned. This fact makes the MLP network more powerful in pattern recognition.

The architecture of the HP network is more restricted and can contain only a single layer. Connections established with this single layer. Therefore, there are fewer weight assignments. This fact makes the HP network lesser powerful in pattern recognition.

Training stage may take a remarkable long time for MLP network and may take short time for HP network.

If the system to be developed is small and unique, HP network is a favorite one to be used to implement the system. If the system to be developed is a moderately large and complex, MLP network should be used to get more accuracy of the system.

6.1 Further Extensions of the System

The variants of the images of the Buddhist Iconographies are of limited in this system. In this system five training sets, that is, 50 Buddhist Iconography images are used to train the ANN. The more number of images with different perspectives, distortions should be used to train the ANN as the further extension of the system. From the above conclusion results, if the future system intends to improve the intelligibility, MLP network should be chosen, otherwise, if the future system intends to reduce the training time, the HP network is a favor one to be applied.

Another techniques, such as, Fuzzy logic should also be applied in the future system in order to get the recognition of Buddhist Iconographies. A separate Fuzzy logic system as well as ANN associated with Fuzzy logic should be developed for further analysis.

7. References

[1] Andrew B. Graham, "Image Processing Algorithm for Detection of Two-Dimensional Visual Code Markers", Stanford University, Stanford, USA, 2006.

[2] Ian T. Young, Jan J. Gerbrands, Lucas J. van Vliet, "*Fundamentals of Image Processing*", Version 2.2, McGraw-Hill International Inc., 1998.

[3] Jagdish Lal, Liqiu Meng, "3D Building Recognition using Artificial Neural Network", *ICA Workshop on*

Generalization and Multiple representation, Technical University of Munich, Germany, August 2004.

[4] Laurene Fauseff, "*Fundamentals of Neural Neworks (Architectures, Algorithms And Applications)*", McGraw-Hill International Inc., 2000.

[5] M. M. Mahbubul Syeed, Fazlul Hasan Siddiqui, Abu Saleh Abdullah Al-Mamun, "Bengali Character Recognition using Bidirectional Associative Memories (BAM) Neural Network", Islamic University of Technology, Board Bazar, Gazipur.

[6] Patrick Van Der Smagt, "*An Introduction to Neural Networks*", Eighth edition, Prentice Hall Inc., 1996.

[7] Rafael C. Gonzalez, Richard E. Woods, "*Digital Image Processing*", Second Edition, Prentice Hall Inc., 2002.

[8] Simon Haykin, "*Neural Networks (A Comprehensive Foundation)",* Second Edition, Prentice Hall Inc., 1999.