Journal of Information Engineering and Applications
ISSN 2224-5782 (print) ISSN 2225-0506 (online)
Vol.4, No.8, 2014
www.iiste.org

Using Optimized Features for Modified Optical Backpropagation Neural Network Model in Online Handwritten Character Recognition System

Fenwa O.D*, Adetunji A.B* and Ajala F.A*

Department of Computer Science and Engineering, Ladoke Akintola University of Technology, Ogbomoso, Nigeria

(*Email of Corresponding authors - fenwadeborah@yahoo.com, abadetunji@lautech.edu.ng, faajala@lautech.edu.ng)

Abstract

One major problem encountered by researchers in developing character recognition system is selection of efficient features (optimal features). In this paper, Particle Swarm Optimization (PSO) is proposed for feature selection. However, backpropagation algorithm has been reported to be an effective and most widely used supervised training algorithm for multi-layered feedforward neural networks but has the shortcomings of longer training time and entrapment into a local minimal. Several research works have been proposed to improve this algorithm but some of these research works were based on 'learning parameter' which in some cases slowed down the training process. Hence, this paper has focused on alleviating the problem of standard backpropagation algorithm based on 'error adjustment'. To this effect, PSO is integrated with a 'Modified Optical Backpropagation (MOBP)' neural network to enhancement the performance of the classifier in terms of recognition accuracy and recognition time. Experiments were conducted on MOBP neural network and PSO-based MOBP classifiers using 6,200 handwritten character samples (uppercase (A-Z), lowercase (a-z) English alphabet and 10 digits (0-9)) collected from 100 subjects using G-Pen 450 digitizer and the system was tested with 100 character samples written by people who did not participate in the initial data acquisition process. Experimental results show promising results for the PSO-based MOBP classifier in terms of the performance measures.

Keywords: Artificial Neural Network, Feature Extraction, Feature Selection, Particle Swarm Optimization, Modified Optical Backpropagation.

1. Introduction

Advancement in computing technology has greatly influenced the lives of human beings and the usage of computer is increasing at a tremendous rate. As computer systems become increasingly integrated into our everyday life, it is therefore necessary to make them more easily accessible and user friendly. The ease with which we can exchange information between user and computer is of immense importance today because input devices such as keyboard and mouse have limitations. Owing to these limitations, researchers for over decades have been attracted to device a quick and natural way of communication between computer systems and human beings (Anita and Dayashankar, 2010; Fenwa, Omidiora and Fakolujo, 2012a; Fenwa et.al., 2012b).

The classes of recognition systems that are usually distinguished are online systems for which handwritten data are captured during the writing process (which makes available the information on the ordering of the strokes) and offline systems for which recognition takes place on a static image captured once the writing process is over (Anoop and Anil, 2004; Liu, Stefan and Masaki, 2004; Mohamad and Zafar, 2004; Naser, Adnan, Arefin, Golam and Naushad, 2009; Pradeep, Srinivasan and Himavathi, 2011). The online methods have been shown to be superior to their offline counterpart in recognizing handwritten characters due the temporal information available with the former (Pradeep, Srinivasan and Himavathi, 2011). Handwriting recognition system can further be broken down into two categories: writer-independent recognition system which recognizes wide range of possible writing styles and a writer-dependent recognition system which recognizes writing styles only from specific users (Santosh and Nattee, 2009).

Statistical classifiers, Probabilistic classifiers, Artificial Neural Networks (ANN) are some of the widely used image classifiers. The major drawback of the statistical classifiers is its inability to classify accurately. On the other hand, probabilistic classifiers suffer from the setback of difficulty in estimating the conditional probabilities. Artificial intelligence has been a major contribution to the advancement of computer science in that it focuses on the creation of machines/systems that can mimic human thoughts, understand speech and countless other feat that were assumed never to be possible. However, ANNs outperform the other classifiers because of its flexibility, scalability, tolerance to faults, accuracy and learning. Generally speaking, the practical use of neural networks has been recognized mainly because of such distinguished features as: general nonlinear mapping between a subset of the past time series values and the future time series values. The capability of capturing essential functional relationships among the data, which is valuable when such relationships are not *a priori* known or are very difficult to describe mathematically and/or when the collected observation data are corrupted by noise universal function approximation capability that enables modeling of arbitrary nonlinear continuous functions to any degree of accuracy.

However, backpropagation algorithm has been reported to be an effective and most widely used supervised training algorithm for multi-layered feedforward neural networks but has the shortcomings of longer training time and entrapment into a local minimal (Freeman and Skapura, 1992). Several research proposals have been made to improve this algorithm. Some of these research works were based on ‘learning parameter’ which in some cases slowed down the training process (Minai, 1990; Riedmiller and Braun, 1993; Otair and Salameh, 2005). Thus, this paper will employ a modified optical backpropagation proposed by Fenwa et al. 2012 which focused on alleviating the problem of standard backpropagation algorithm based on ‘error adjustment’.

2. Related Work

Otair and Salameh (2005) proposed an online handwritten character recognition system using an Optical Backpropagation network. Two neural networks were developed and trained to recognize handwritten characters. Also two algorithms were use, namely classical Backpropagation (BP) and Optical Backpropagation (OBP), which applies a non-linear function on the error from each output unit before applying the Backpropagation phase. The OBP aimed at speeding up the training process and escape from local minima and was successful at that.

In 2012, Fenwa et.al proposed a modified Optical Backpropagation neural network model in online handwritten character recognition system using hybrid of geometrical and statistical features. The proposed system showed better performance when compared with the existing Optical backpropagation neural network in literature.

In this paper, Particle Swarm Optimization (PSO) algorithm is integrated with a 'Modified Optical Backpropagation (MOBP)' neural network to enhancement the performance of the classifier in terms of recognition accuracy and recognition time.

3. Research Methodology

In this paper, four stages of development of the proposed character recognition system which include; data acquisition, pre-processing, character processing which consists of feature extraction and feature selection and classification using MOBP, and PSO-based MOBP classifiers as shown in Figure 3.1. Experiments were performed with 6,200 handwriting character samples (uppercase (A-Z) lowercase (a-z) English alphabet) and digits (0-9) collected from 100 subjects using G-Pen 450 digitizer and the system was tested with 100 character samples written by people who did not participate in the initial data acquisition process. The performance of the system was evaluated based on convergence time and recognition accuracy.

3.1 Character Acquisition

The data used in this work were collected using Digitizer tablet (G-Pen 450). The G-Pen has an electric pen with sensing writing board. An interface was developed using C# to acquire geometrical features such as stroke number, pressure used in writing the strokes of the characters, number of junctions and the location in the written characters and the horizontal projection count of the character from different subjects using the Digitizer tablet.

Characters considered were 26 upper case (A-Z), 26 lower case (a-z) English alphabets and 10 digits (0-9) making a total number of 62 characters. 6,200 characters were collected from 100 and this serves as the training data set which was the input data that was fed into the neural network. Sample data used were as shown in figure 3.2

3.2 Feature Extraction

This research focuses on a feature extraction technique that combined three characteristics of the handwritten character to create a global feature vector. A hybrid feature extraction algorithm was developed using Geometrical and Statistical features. Integration of Geometrical and Statistical features was used to highlight different character properties, since these types of features are considered to be complementary. Eleven features from two categories (Geometrical features and Statistical features) were used in this work.

The Hybrid (Geom-Statistical) Feature Extraction Algorithm proposed by Fenwa, Omidiora and Fakolujo, 2012a was used:

Step 1: Get the stroke information of the input characters from the digitizer (G-pen 450)

These include:

(i) Pressure used in writing the strokes of the characters

(ii) Number (s) of strokes used in writing the characters

(iii) Number of junctions and the location in the written characters

(iv) The horizontal projection count of the character

Step 2: Apply Contour tracing algorithm to trace out the contour of the characters

Step 3: Run Hybrid Zoning algorithm on the contours of the characters

Step 4: Feed the outputs of the extracted features of the characters into the digitization stage in order to convert all the extracted features into digital forms

The Hybrid Zoning Algorithm: Hybrid of Modified ICZ and Modified ZCZ

Input: Pre-processed character image

Output: Features for Classification and Recognition

Method Begins

Step 1: Divide the input image into 25 equal zones.

Step 2: Compute the input image centroid

Step 3: Compute the distance between the image centroid to each pixel present in the zone.

Step 4: Repeat step 3 for the entire pixel present in the zone.

Step 5: Compute average distance between these points.

Step 6: Compute the zone centroid

Step 7: Compute the distance between the zone centroid to each pixel present in the zone.

Step 8: Repeat step 7 for the entire pixel present in the zone

Step 9: Compute average distance between these points.

Step 10: Repeat the steps 3-9 sequentially for the entire zones.

Step 11: Finally, 2xn (50) such features were obtained for classification and recognition.

Method Ends

3.3 Feature Selection

Feature selection refers to the problem of dimensionality reduction of data, which initially consists of large number of features. The objective is to choose optimal subsets of the original features which still contain the information essential for the classification task while reducing the computational burden imposed by using many features. In this work, Particle Swarm Optimization is proposed for feature selection.

Particle Swarm Optimization (PSO)

The PSO method is a member of wide category of Swarm Intelligence methods for solving the optimization problems. It is a population based search algorithm where each individual is referred to as particle and represents a candidate solution. In this paper the PSO algorithm proposed by Anita and Jude 2010 was employed. Each single candidate solution is "an individual bird of the flock", that is, a particle in the search space. Each particle makes use of its individual memory and knowledge to find the best solution. All the particles have fitness values, which are evaluated by fitness function to be optimized and have velocities which direct the movement of the particles. The particles move through the problem space by following a current of optimum particles. The initial swarm is generally created in such a way that the population of the particles is distributed randomly over the search space. At every iteration, each particle is updated by following two "best" values, called *pbest* and *gbest*. Each particle keeps track of its coordinates in the problem space, which are associated with the best solution (fitness value). This fitness value is called *pbest*. When a particle takes the whole population as its topological neighbor, the best value is a global best value and is called *gbest*. The detailed algorithm is given as follows:

Step 1: Set the constants k_{max}, c_1, c_2, r_1, r_2, w.
Randomly initialize particle positions $x_0(i)$ for i = 1, 2………p.
Randomly initialize particle velocities $v_0(i)$ for i = 1, 2……..p.
Step 2: Set k =1.
Step 3: Evaluate function value f_k using design space coordinates $x_k(i)$
If $f_k \geq f_{pbest}$, then pbest(i) = $x_k(i)$
If $f_k \geq f_{gbest}$, then gbest= $x_k(i)$
Step 4: Update particle velocity using the following equation

$$v_{k+1}(i)= w*(v_k(i)) + c_1r_1*(pbest_k(i) - x_k(i)) + c_2r_2(gbest_k – x_k(i)) \quad (1)$$

Update particle position vector using the following equation

$$x_k + 1(i) = x_k(i) + v_k + 1(i) \quad (2)$$

Step 5: Increment i. If i > p, then increment k and set i = 1.
Step 6: Repeat steps 3 to 5 until k_{max} is reached.
The notations used in this algorithm are:
k_{max} = maximum iteration number
w = inertia weight factor
c_1, c_2 = cognitive and social acceleration factors
r_1, r_2 = random numbers in the range (0, 1).

In this paper, each of the eleven features are represented by a chromosome (string of bits) with 11 genes (bits) corresponding to the number of features. An initial random population of 20 chromosomes is formed to initiate the genetic optimization. The initial coding for each particle is randomly generated. The order of position of the features in each string is pressure of the stroke, stroke number, horizontal projection count, contour pixel, image centroid, zone centroid, distance between zone centroid, distance between image centroid, horizontal centre of gravity and vertical centre of gravity respectively. A suitable fitness function is estimated for each individual.

The fittest individuals are selected and the crossover and the mutation operations are performed to generate the new population. This process continues for a particular number of generations and finally the fittest chromosome is calculated based on the fitness function. The features with a bit value "1" are accepted and the features with the bit value of "0" are rejected. The fitness function used in this work is given by

$$\text{Fitness } (\alpha * \gamma) + \beta * \left(|c| - |r| / |c| \right) \quad (3)$$

where γ = classification accuracy

c = total number of features

r = length of the chromosome (number of '1's)

$\alpha \in [0, 1]$ and $\beta = 1 - \alpha$

This formula shows that the classification accuracy and the feature subset length have different significance for feature selection. A high value of α assures that the best position is at least a rough set reduction. The goodness of each position is evaluated by this fitness function. The criteria are to maximize the fitness values. An optimal solution is obtained at the end of the maximum iteration. This value is binary coded with eleven bits. The bit value of "1" represents a selected feature whereas the bit value of "0" represents a rejected feature. Thus an optimal set of features are selected from the PSO technique. Out of the eleven features extracted, seven optimal set of features are selected from the PSO algorithm.

3.4 Classification Method

In this work, two types of neural network classifiers are used and these are: the Modified Optical Backpropagation neural network and the PSO-Based Modified Optical Backpropagation neural network.

3.4.1 The Optical Backpropagation Neural Network:

The difficulty encountered in the standard Backpropagation algorithm is when the actual value $f^o_k(net^o_{pk})$ approaches either extreme value, the factor $f^o_k((net^o_{pk}) \cdot (1 - f^o_k(net^o_{pk})))$ makes the error signal very small. The Optical Backpropagation algorithm (an enhanced Backpropagation) focused on this delay of the convergence that is caused by the derivative of the activation function. However, a slight modification of the error signal function of the standard Backpropagation algorithm has resolved this shortcoming and indeed greatly accelerates the convergence to a solution. The adjustment of the new algorithm (OBP) is described to improve the performance of the *BP* algorithm. The convergence speed of the training process was improved significantly by OBP through maximizing the error signal, which was transmitted backward from the output layer to each unit in the intermediate layer.

In *BP,* the error at a single output unit is defined according to equation (4) as:

$$\delta^o_{pk} = (Y_{pk} - O_{pk}) \cdot f^{o'}_k(net^o_{pk}) \quad \textbf{(4)}$$

where the subscript "p" refers to the p_{th} training vector, and "k" refers to the k_{th} output unit. In this case, Y_{pk} is the desired output value, and O_{pk} is the actual output from k_{th} unit, then δ^o_{pk} will propagate backward to update the output-layer weights and the hidden-layer weights while the error at a single output unit in adjusted OBP is given as (Otair, & Salameh, 2005):

$$\text{New } \delta^{o}_{pk} = (1+e^{(Ypk - Opk)2} \,.\, f^{o'}_{k}(net^{o}_{pk})), \quad \text{if } (Y - O) >= \text{zero} \tag{5a}$$

$$\text{New } \delta^{o}_{pk} = -(1+e^{(Ypk - Opk)2} \,.\, f^{o'}_{k}(net^{o}_{pk})), \quad \text{if } (Y - O) < \text{zero} \tag{5b}$$

An OBP uses two forms of Newδ^{o}_{pk}, because the *exponential* function always return *zero* or *positive* values, while adapts operation for many output units need to decrease the actual outputs rather than increasing it. The Newδ^{o}_{pk} will propagate backward to update the output-layer weights and the hidden-layer weights. This Newδ^{o}_{pk} minimized the errors of each output unit more quickly than the old δ^{o}_{pk}, and the weights on certain units change very *large* from their starting values.

The steps of an OBP (Otair and Salameh, 2005)

1. Apply the input example to the input units.
2. Calculate the net-input values to the hidden layer units.
3. Calculate the outputs from the hidden layer.
4. Calculate the net-input values to the output layer units
5. Calculate the outputs from the output units
6. Calculate the error term for the output units, using the Newδ^{o}_{pk} (using equations 5a and 5b) instead of δ^{o}_{pk} in equation (4)
7. Calculate the error term for the hidden units, through applying New δ^{o}_{pk}, also

   $$\text{New}\delta^{h}_{pj} = f^{h'}_{j}(net^{h}_{pj}) \,.\, (\textstyle\sum_{k=1}^{M} New\,\delta^{o}_{pk} \,.\, W^{o}_{kj}) \tag{6}$$

8. Update weights on the output layer.

   $$W^{o}_{kj}(t+1) = W^{o}_{kj}(t) + (\eta \,.\text{New}\delta^{o}_{pk} \,.\, i_{pj}) \tag{7}$$

9. Update weights on the hidden layer.

   $$W^{h}_{ji}(t+1) = W^{h}_{ji}(t) + (\eta \,.\, \text{New}\delta^{o}_{pj} \,.\, X_i) \tag{8}$$

10. Repeat steps from step 1 to step 9 until the error ($Y_{pk} - O_{pk}$) is acceptably small for each training vector pairs.

The proposed algorithm as classical BP is stopped when the squares of the differences between the actual and target values summed over units and all patterns are acceptably small.

3.4.2 Modified Optical Backpropagation Neural Network

The error function defined in Optical Backpropagation earlier is proportional to the square of the Euclidean distance between the desired output and the actual output of the network for a particular input pattern. As an alternative, other error functions whose derivatives exist and can be calculated at the output layer can replace the traditional square error criterion (Haykin, 2003). In this research work, error of the third order (Cubic error) had been adopted to replace the traditional square error criterion used in Optical Backpropagation. The equation of the cubic error is given as:

$$\delta^o_{pk} = -3(Y_{pk} - O_{pk})^2 \cdot f^{o'}_k(net^o_{pk}) \quad (9)$$

The cubic error in equation (6) was manipulated mathematically and this further maximized the error signal of each output unit which was transmitted backward from the output layer to each unit in the intermediate layers (Fenwa, Omidiora, Fakolujo and Ganiyu, 2012).

The derived equations were as shown in equations (10a) and (10b) below:

$$\text{Modified } \delta^o_{pk} = 3((1+e^t)^2 \cdot f^{o'}_k(net^o_{pk})) \qquad \text{If } (Y_{pk} - O_{pk})^2 >= 0 \quad (10a)$$

$$\text{Modified } \delta^o_{pk} = -3((1+e^t)^2 \cdot f^{o'}_k(net^o_{pk})) \qquad \text{If } (Y_{pk} - O_{pk})^2 <=0 \quad (10b)$$

where Y_{pk} = Target or Desired output

O_{pk} = Network output

$t = (Y_{pk} - O_{pk})^2$

However, one of the ways to reduce the training time is through the use of momentum, as it enhances the stability of the training process. The momentum was used to keep the training process going in the same general direction (Haykin, 2003). In the modified Optical Backpropagation network, momentum was introduced. Hence, equation (6) becomes

$$W^o_{kj}(t+1) = W^o_{kj}(t) + \mu W^o_{kj}(t) + (\eta \cdot \text{Modified } \delta^o_{pk} \cdot i_{pj}) \quad (11)$$

where μ is the momentum coefficient typically about 0.9 and η is the learning rate.

The Modified Optical Backpropagation Algorithm:

Modifications of the algorithm are in terms of Error Signal Function

With the introduction of Cubic error function and Momentum, the modified Optical Backpropagation is given as:

1. Apply the input example to the input units.
2. Calculate the net-input values to the hidden layer units.
3. Calculate the outputs from the hidden layer.
4. Calculate the net-input values to the output layer units
5. Calculate the outputs from the output units
6. Calculate the error term for the output units, using equation (10a) and (10b) instead of equations (5a) and (5b)
7. Calculate the error term for the hidden units, through applying improve δ^o_{pk} also

$$\text{Modified } \delta^h_{pj} = f^{h'}_j(net^h_{pj}) \cdot \left(\sum_{k=1}^{M} \textit{Modified } \delta^o_{pk} \cdot W^o_{kj}\right) \quad (12)$$

8. Update weights on the output layer.

$$W^o_{kj}(t+1) = W^o_{kj}(t) + \mu W^o_{kj}(t) + (\eta \,.\, \text{Modified } \delta^o_{pk} \,.\, i_{pj}) \quad (13)$$

9. Update weights on the hidden layer.

$$W^h_{ji}(t+1) = W^h_{ji}(t) + (\eta \,.\, \text{Modified } \delta^h_{pj} \,.\, X_i) \quad (14)$$

Repeat steps from step 1 to step 9 until the error ($Y_{pk} – O_{pk}$) is acceptably small for each of the training vector pair. The proposed algorithm as OBP is stopped when the cubes of the differences between the actual and target values summed over units and all patterns were acceptably small.

3.4.3 The PSO-Based Modified Optical Backpropagation

The second classifier used in this work is PSO-Based MOBP classifier. The objective for using the optimization algorithm is two folds: (i) dimensionality reduction which improves the convergence rate and (ii) elimination of insignificant features which improves the classification accuracy. In this work Particle Swarm Optimization algorithm is used for optimal feature selection. The extracted features are subjected to this optimization technique which finally yields the optimal feature set. The number of neurons used in the input layer for this PSO-Based MOBP classifier is reduced since the number of optimal features is lesser than the complete feature set. Also, the mathematical calculations are minimized because of the reduced size of the weight matrix. Hence, a significant reduction in the time period is achieved for the weight adjustment of the hidden layer neurons. Thus, the PSO-Based MOBP neural network is better than MOBP neural network.

4. Experiment

Experiments were carried out on a Hewlett Packard system with the configuration: 64 bits operating system, 4.00G RAM and Intel(R) CORE(TM) i5-3210M CPU @ 2.50GHz processor. The system was implemented using C# programming language.

5. Results and Discussions

Experiments were performed with 6,200 handwriting character samples (uppercase (A-Z) lowercase (a-z) English alphabet) and digits (0-9) collected from 100 subjects using G-Pen 450 digitizer and the system was tested with 100 character samples written by people who did not participate in the initial data acquisition process. The performance of the system was evaluated based on convergence time and recognition accuracy.

It was shown in Table 1 that the more the dimensional input vector (character matrix size), the more the number of epochs. Usually, the complex and large sized input sets require a large topology network with more number of iterations (Epochs). The epochs is directly proportional to the training time, this implies that the larger the image size, the more the training time. Three different image sizes (5 by 7, 10 by 14 and 20 by 28) were considered in this paper and it was shown from the results that the higher the image size, the higher the number of epochs required to train the network due to increment in vector space to be processed by the network. From the above table, it is evident that the PSO-Based MOBP is superior to MOBP classifier in terms of convergence rate.

From table 2, the training time of PSO-Based MCPN is smaller when compared with the MOBP classifier due to its ability to achieve dimensionality reduction and removal of irrelevant features of character images.

Classification accuracy is the ratio of the number of correctly classified images to the total number of images. Convergence time period is the time taken for training process and testing process. From table 3, it is clearly understood that the architecture of the PSO-Based MOBP is highly simplified and is less prone to error in classification than the MOBP classifier. It also reveals the less number of mathematical computational operations involved in PSO-Based MOBP. The PSO-Based MOBP classifier has better classification accuracy than MOBP classifier.

6. Conclusion and Future Work

This paper explores the need for optimization algorithms to enhance the performance of the classifiers. In this work, PSO is used as the optimization algorithm and it is used along with the modified Optical Backpropagation classifier. Experimental results suggest better improvement in the classification accuracy for the PSO-Based MOBP than the other classifier (MOBP). However, an increase in the convergence rate is also achieved by the PSO-based MOBP classifier which is highly essential for real-time applications. Therefore an optimization technique is highly essential irrespective of the classifiers under consideration.

Finally, the application of PSO optimization algorithm for performance improvement of the neural classifier has been explored in the context of online character image classification. Future can be tailored towards hybridization of other classifiers to further enhance the performance of the system. The work can also be extended by using different optimization algorithms to estimate the performance of the classifiers. However, different set of features can be used to improve the classification accuracy and experiments can be carried out on a different set of database in order to generalize the technique. Irrespective of the modifications and the systems used, this paper has been able to present the significance of optimization algorithm for accurate and quick image classification systems.

7. References

Anita, P. and Dayashankar S. (2010): ''Handwritten English Character Recognition using Neural Network'', International Journal of Computer Science and Communication, 1(2): 141-144.

Anoop, M. N. and Anil K.J. (2004): ''Online Handwritten Script Recognition'', IEEE Trans. PAMI, 26(1): 124-130.

Fenwa O.D., Omidiora E.O., Fakolujo O.A. (2012a): ''Development of a Feature Extraction Technique for Online Character Recognition System'', Journal of Innovation System Design and Engineering, 3(3):10-23.

Fenwa, O.D., Omidiorah, E.O., Fakolujo, O.A. and Ganiyu, R.A. (2012b): ''Development of a Writer-Independent Online Handwritten Character Recognition System using modified Hybrid Neural Network Model'', Accepted Manuscript for Publication in Journal of Computer Engineering and Intelligent System, International Institute of Science, Technology and Education, New York, USA, Paper ID: 2011471827.

Freeman, J.A. and Skapura, D.M. (1992): Backpropagation Neural Networks Algorithm Applications and Programming Techniques, Addison-Wesley Publishing Company: 89-125.

Haykin, S. (2003): Neural Networks: A Comprehensive Foundation, PHI, New –Delhi, India

Hecht-Nielsen, R. (1990): Neurocomputing: Addison-Wesley Publishing Company.

Jude, D. H, Vijila, C. K. and Anitha, J (2010): ''Performance Improved PSO Based Modified Counterpropagation Neural Network for Abnormal Brain Image Classification'', International Journal Advanced Soft Computing Application, 2(1): 65-84.

Liu, C.L., Nakashima, K., Sako, H. and Fujisawa, H. (2004): ''Handwritten Digit Recognition: Investigation of Normalization and Feature Extraction Techniques'', Pattern Recognition, 37(2): 265-279.
Minai, A.A. and Williams, R.D. (1990): ''Acceleration of Backpropagation through learning Rate Momentum Adaptation'', Proceeding of the International Joint Conference on Neural Networks: 1676-1679.
Mohamad, D. and Zafar, M.F. (2004): ''Comparative Study of Two Novel Feature Vectors for Complex Image Matching Using Counterpropagation Neural Network'', Journal of Information Technology, FSKSM, UTM, 16(1): 2073-2081.
Morita M., Sabourin R., Bortolozzi F., Suen C. Y. (2003): '' A Recognition and Verification Strategy for Handwritten Word Recognition'', ICDAR'03), Edinburgh-Scotland: 482-486.
Naser, M.A., Adnan, M., Arefin, T.M., Golam, S.M. and Naushad, A. (2009): Comparative Analysis of Radon and Fan-beam based Feature Extraction Techniques for Bangla Character Recognition, IJCSNS International Journal of Computer Science and Network Security, 9(9): 120-135.
Otair, M.A. and Salameh, W.A. (2005): '' Speeding up Backpropagation Neural Network'' Proceedings 0f the 2005 Informing Science and IT Education Joint Conference 2: 167-173.
Pradeep, J., Srinivasan, E. and Himavathi, S. (2011): ''Diagonal Based Feature Extraction for Handwritten Alphabets Recognition using Neural Network'', International Journal of Computer Science and Information Technology (IJCS11), 3(1): 27-37.
Riedmiller, M. and Braun, H. (1993): '' A Direct Adaptive Method for Faster Backpropagation learning the PROP Algorithm'', Proceedings of the IEEE International Conference on Neural Networks (ICNN), 1: 586-591, Francisco.
Santosh, K.C. and Nattee, C. (2009): ''A Comprehensive Survey on Online Handwriting Recognition Technology and Its Real Application to the Nepalese Natural Handwriting'', Kathmandu University Journal of Science, Engineering Technology, 5(1): 31-55.

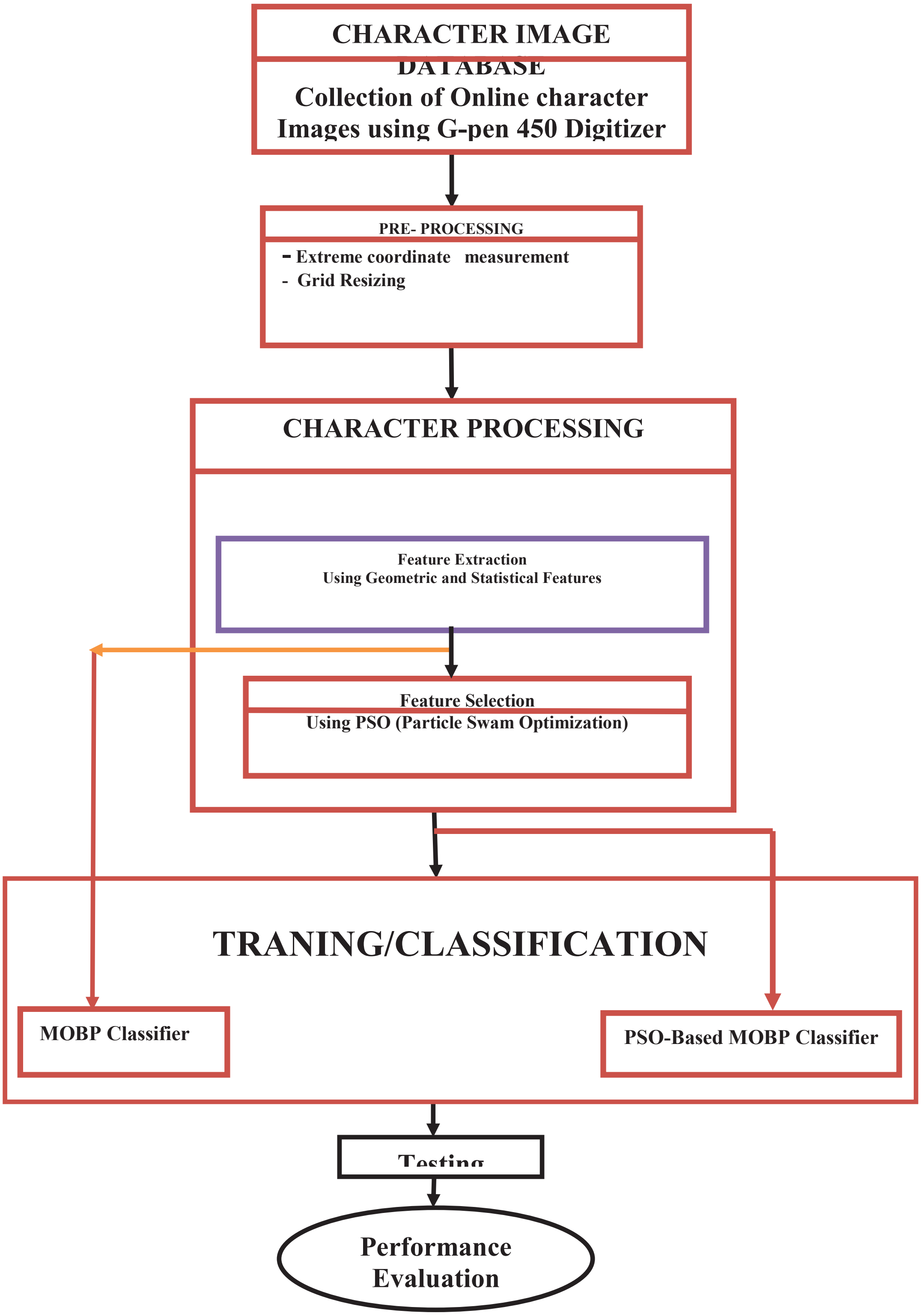

FIG. 3.1: THE BLOCK DIAGRAM OF THE PROPOSED PSO-BASED MOBP CHARACTER RECOGNITION SYSTEM

ISSN 2224-5782 (print) ISSN 2225-0506 (online)
Vol.4, No.8, 2014

Figure 3.2: Sample Data collected using G-pen 450 Digitizer

Table 1: Epochs Values with different Image Sizes for the two Classifiers

Image Sizes	MOBP	PSO-Based MOBP
5 by 7	1120	920
10 by 14	1600	1200
20 by 28	1864	1417

Table 2: The Training Time (in minutes) of the two classifiers under different datasets

Character Samples	MOBP Training Time	PSO-Based MOBP Training Time
1,200	**0.84**	**2.28**
2,480	**7.01**	**4.01**
3,720	**9.87**	**6.36**
4,960	**18.69**	**13.98**
6,200	**52.68**	**36.79**

Table 3: Classification Accuracies of the three Classifiers

Character Samples	**CR (%)**	**FR (%)**	**RF (%)**	**CR (%)**	**FR (%)**	**RF (%)**
1,240	75	6	2	79	3	1
2,480	80	3	1	85	2	1
3720	86	1	1	89	2	0
4,960	92	1	0	95	1	0
6,200	95	1	0	98	1	0

CR: Correct Recognition, FR: False Recognition, RF: Recognition Failure