



Optimized Features for Hybrid Neural Network Model in Online Character Recognition

Olusayo D. Fenwa^{*}, Modupe O. Alade, Elizabeth A. Amusan

Computer Science LAUTECH Ogbomosho,
Oyo State, Nigeria

Abstract: Recognition of handwritten characters has become a difficult problem because of the high variability and ambiguity in the character shapes written by individuals. One of the major problems encountered by researchers in developing character recognition system is selection of efficient features (optimal features), low recognition accuracy and high recognition time. In this paper, Particle Swarm Optimization (PSO) is proposed for optimal feature selection. However, various classification approaches have been used by many researchers to develop character recognition system such as template matching approach, statistical approach, structural approach and neural networks approach but there is need to further improve the performances of these systems. Also, it has been demonstrated in a number of applications that using more than a single classifier in a recognition task can lead to a significant improvement of the system's overall performance. Hence, in this paper, PSO is integrated with hybrid of Counterpropagation and modified Optical Backpropagation Neural Network (COMOB) model to enhancement the performance of the classifier in terms of recognition accuracy and recognition time.

Experiments were conducted on COMOB and PSO- Based COMOB 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 COMOB classifier in terms of the performance measures.

Keywords- Artificial Neural Network, Counterpropagation Neural Network, Character Recognition, Feature Extraction, Feature Selection, PSO, COMOB

I. 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 ([1], [2], [3]- [4]).

The use of neural network for handwriting recognition is a field that is attracting a lot of attention. As the computing technology advances, the benefits of using Artificial Neural Network (ANN) for the purpose of handwriting recognition become more obvious. Hence, new ANN approaches geared toward the task of handwriting recognition are constantly being studied. Character recognition is the process of applying pattern-matching methods to character shapes that has been read into a computer to determine which alpha-numeric character, punctuation marks, and symbols the shapes represent. The classes of recognition systems that are usually distinguished are online systems for which handwriting 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 ([5], [6], [7], [8], [9]). The online methods have been shown to be superior to their offline counterpart in recognizing handwriting characters due the temporal information available with the former ([9]). 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 ([10]).

Online handwriting recognition today has special interest due to increased usage of the hand held devices. The incorporation of keyboard being difficult in the hand held devices demands for alternatives, and in this respect, online method of giving input with stylus is gaining quite popularity ([11]). Recognition of handwritten characters with respect to any language is difficult due to variability of writing styles, state of mood of individuals, multiple patterns to represent a single character, cursive representation of character and number of disconnected and multi-stroke characters ([12]). Current technology supporting pen-based input devices include: Digital Pen by Logitech, Smart Pad by Pocket PC, Digital Tablets by Wacom and Tablet PC by Compaq ([13]). Although these systems with handwriting recognition capability are already widely available in the market, further improvements can be made on the recognition performances for these applications.

The challenges posed by the online handwritten character recognition systems are to increase the recognition accuracy and to reduce the recognition time ([14], [11]). Various approaches that have been used by many researchers to develop character recognition systems, these include; template matching approach, statistical approach, structural approach, neural networks approach and hybrid approach. Hybrid approach (combination of multiple classifiers) has become a very active area of research recently ([15], [16]). It has been demonstrated in a number of applications that using more than a single classifier in a recognition task can lead to a significant improvement of the system’s overall performance. Hence, hybrid approach seems to be a promising approach to improve the recognition rate and recognition accuracy of current handwriting recognition systems ([17]). However, Selection of a feature extraction method is probably the single most important factor in achieving high recognition performance in character recognition system ([8]). No matter how sophisticated the classifiers and learning algorithms, poor feature extraction will always lead to poor system performance ([18]). In furtherance, [2] developed a feature extraction technique for online character recognition system using hybrid of geometrical and statistical features. Thus, through the integration of geometrical and statistical features, insights were gained into new character properties, since these types of features were considered to be complementary.

II. RELATED WORKS

Reference [1] implemented Counterpropagation for digit recognition. Reference [7] proposed the conventional CPN for the recognition of online upper case English alphabets and recognition rates of 60% to 93% were obtained for different sets of character samples. An improved PSO based modified CPN for abnormal MR Brain Image classification was implemented by [19] and better performance was obtained. Implementation of a Modified Counterpropagation Neural Network Model in Online Handwritten Character Recognition System was proposed by [4] and better recognition accuracies were reported. However, [3] proposed an hybrid of counter propagation and optical backpropagation for recognition of online handwritten characters and better results were reported. Hence, this paper has focused on integrating PSO with hybrid of Counterpropagation neural network and modified Optical Backpropagation neural network for recognition of online uppercase (A-Z), lowercase (a-z) English alphabets and digit (0-9).

III. METHODOLOGY

The five stages involved in developing the proposed character recognition system, which include data acquisition, pre-processing, character processing that comprises feature extraction and character digitization, training and classification using hybrid neural network model and testing, are as shown in Figure 3.1. Experiments were performed with 6200 handwriting character samples (English uppercase, lowercase and digits) collected from 50 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 different learning rates, different image sizes and different database sizes.

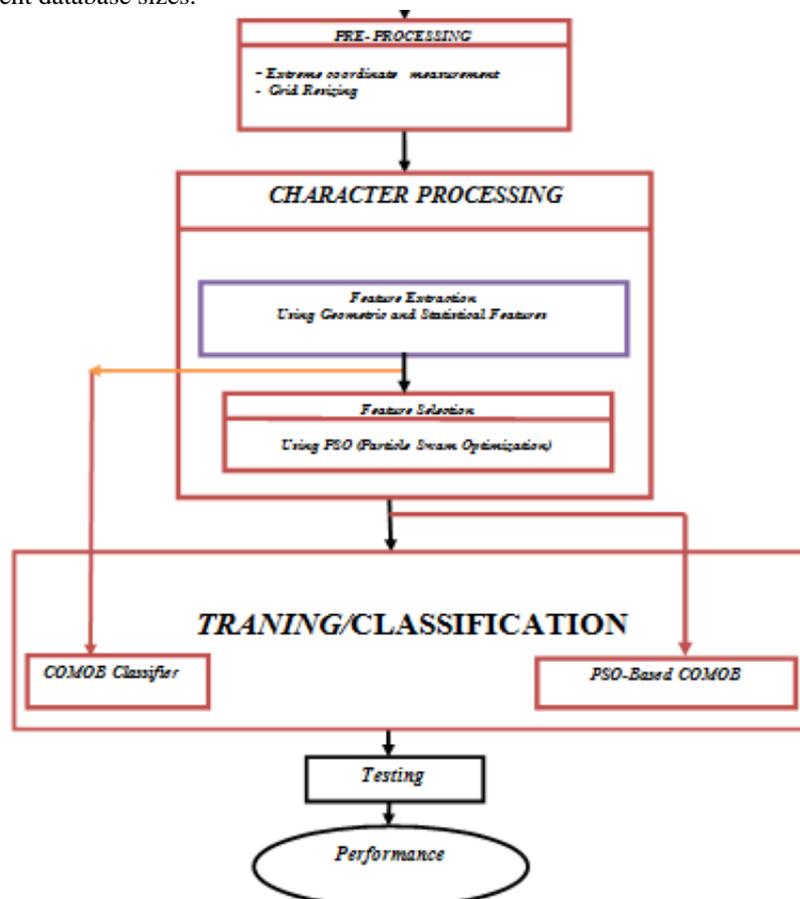


Figure 3.1: Block Diagram of the Developed Character Recognition System

3.1 Data Acquisition

The data used in this work were collected using Digitizer tablet (G-Pen 450) shown in Figure 3.2. It has an electric pen with sensing writing board. An interface was developed using C# to acquire data (character information) such as stroke number, stroke pressure, etc from different subjects using the digitizer tablet. 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 (62 x 2 x 50) were collected from 50 subjects as each individual was requested to write each of the characters 2 times (this is done to allow the network learn various possible variations of a single character and become adaptive in nature). This serves as the training data set which was the input data that was fed into the neural network. Sample character were shown in figure 3.3



Figure 3.2: The snapshot of Genius Pen (G-Pen 450) Digitizer for character acquisition

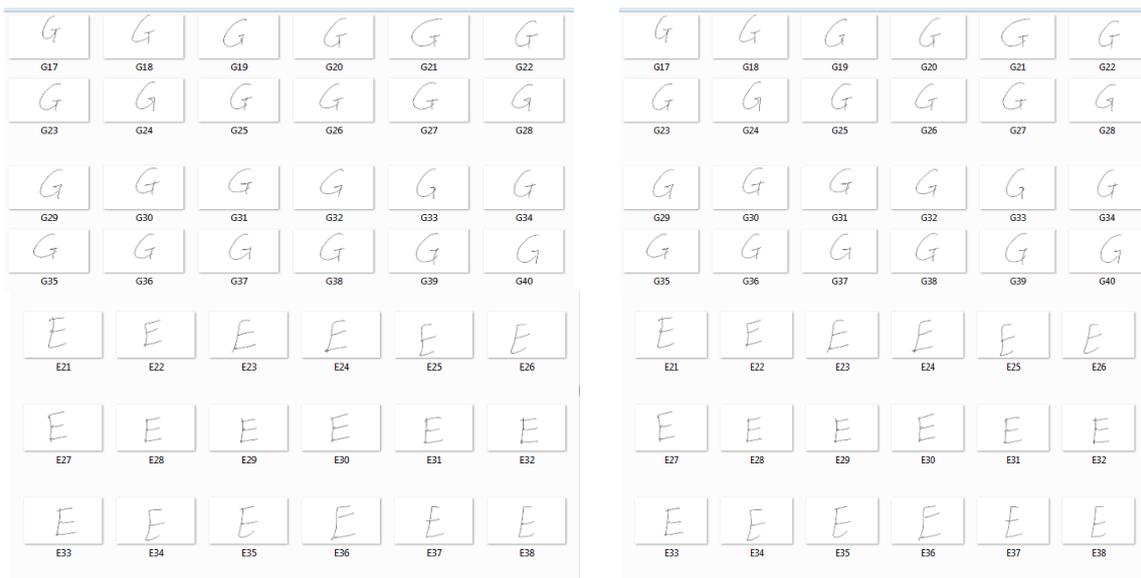


Figure 3.3: Sample Data collected using G-pen 450 Digitizer

3.2 Data Preprocessing

Pre-processing is done prior to the application of feature extraction algorithms. Pre-processing aims to produce clean character images that are easy for the character recognition systems to operate more accurately. Feature extraction stage relies on the output of this process. The pre processing techniques used in this research work is Grid Resizing.

3.3 Resizing Grid

From the interface that was provided, there wasn't any degree of measurement to determine how small/big the input character should be. Hence, the character written is resized to a matrix size of 5 by 7, 10 by 14 and 20 by 28 for all input characters. This is used to get the universe of discourse which is the shortest matrix that fit the entire character skeleton. The universe of discourse is measured to easily get a uniform matrix size in multiple of 5 by 7, 10 by 14 and 20 by 28 respectively. Any character measured smaller than the required size is considerably resized to the multiple of 5 by 7, 10 by 14 and 20 by 28 conversely any character measured larger than the required size will also be resized to the multiple of 5 by 7, 10 by 14 and 20 by 28. This implies that rows and columns of Zero's are added or subtracted from the resized image matrix to achieve the required multiple of 5 by 7, 10 by 14 and 20 by 28.

3.4 Feature Extraction Development

The goal of feature extraction is to extract a set of features, which maximizes the recognition rate with the least amount of elements. Many feature extraction techniques have been proposed to improve overall recognition rate;

however, most of the techniques used only one property of the handwritten character. This research focuses on a feature extraction technique that combined three characteristics (stroke information, contour pixels and zoning) of the handwritten character to create a global feature vector. Hence, a hybrid feature extraction algorithm was developed using Geometrical and Statistical features as shown in Figure 3.4. Integration of Geometrical and Statistical features was used to highlight different character properties, since these types of features are considered to be complementary. Feature Extraction

This paper 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 [2] was used in this work and the model is as shown in figure 3.4

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 feature selection stage (using PSO algorithm) to select the best features.

Step 5: Feed the outputs of the selected features (from PSO algorithm) of the characters into the digitization stage in order to convert all the extracted features into digital forms

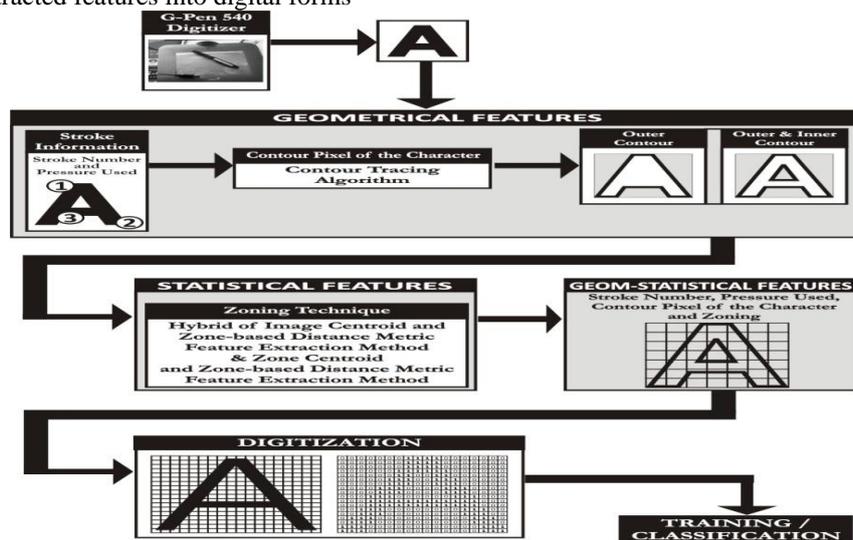


Figure 3.4: The Developed Hybrid Feature Extraction Model [2](Fenwa et al 2012)

3.5 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.

3.5.1 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. 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 $i = 1, 2, \dots, p$.
Randomly initialize particle velocities $v_0(i)$
for $i = 1, 2, \dots, p$.
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
$$v_{k+1}(i) = w^*(v_k(i)) + c_1r_1*(pbest_k(i) - x_k(i)) + c_2r_2*(gbest_k - x_k(i))$$

(1)
Update particle position vector using the following equation
$$x_{k+1}(i) = x_k(i) + v_{k+1}(i)$$

(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, 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

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

(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.

3.6 Development of Hybrid Neural Network Model

In this research work, hybrid approach with serial combination scheme was adopted. Hybrid of modified Counterpropagation and modified Optical Backpropagation neural network was developed and the architecture of the neural network with three layers is illustrated in Figure 3.5. The output of i^{th} layer is given by (3) except the output layer which uses maxsoft function:

$$a^i = \text{logsig}(w^i a^{i-1} + b^i)$$

(3)

where $i = (2, 3)$ and $a^0 = P$, $a^1 = E$ where E is the Euclidean distance between the weight vector and the input vector.

w^i = Weight vector of i^{th} layer

a^i = Output of i^{th} layer

b^i = Bias vector for i^{th} layer.

The input vector 'P' is represented by the solid vertical bar at the left. The dimensions of 'P' are displayed as 35 x 1, indicating that the input is a single vector of 35 elements (i.e. the image size). These inputs go to weight matrix 'W¹', which has 86 rows (i.e. 86 neurons in the first hidden layer) and 35 columns. A constant '1' enters the neuron as input and is multiplied by a bias 'b¹'. The net input to the transfer function (Euclidean distance) in the Kohonen (hidden layer) is 'n¹', which is given as the Euclidean distance between the weight vector w^1 and the input vector P. The neurons output 'a¹' serves as inputs to the second hidden layer. These inputs go to weight matrix 'W²', which has 86 rows (i.e.86 neurons in the first hidden layer) and 35 columns. A constant '1' enters the neuron as input and is multiplied by a bias 'b²'.

The net input to the transfer function (log sigmoid) in the second (hidden layer) is 'n²', which is the sum of the bias 'b²' and the product 'W²a¹'. The output 'a²' is a single vector of 86 elements and this serves as inputs to the output layer 'a³'. These inputs go to weight matrix 'W³', which has 86 rows (i.e. 86 neurons from the second hidden layer) and 62 columns (26 uppercase + 26 lowercase + 10 digits). The net input to the transfer function (maxsoft) in the output layer is 'n³', which is the sum of the bias 'b³' and the product 'W³a²'. The neurons output 'a³' is a single vector of 62 elements and this serves as the final output of the neural network.

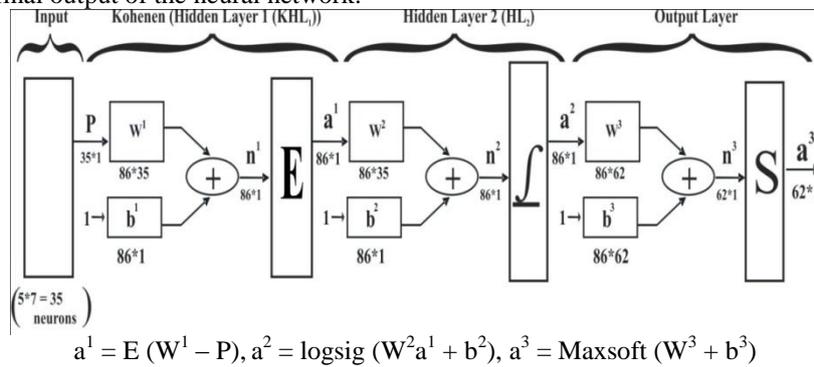


Figure 3.5: The Hybrid Neural Network Model([3])

This research work adopted the modified Counterpropagation network (CPN) developed by [19]). The training algorithm involves the following two phases:

(i) Weight adjustment between the input layer and the hidden layer (Kohonen layer)

The weight adjustment procedure for the hidden layer weights is same as that of the conventional CPN. It follows the unsupervised methodology to obtain the stabilized weights. After convergence, the weights between the hidden layer and the output layer are calculated.

(ii) Weight adjustment between the hidden layer and its output layer

The weight adjustment procedure employed in this work is significantly different from the conventional CPN. The weights are calculated in the reverse direction without any iterative procedures. Normally, the weights are calculated based on the criteria of minimizing the error. But in this work, a minimum error value is specified initially and the weights are estimated based on the error value. Thus without any training methodology, the weight values are estimated. This technique accounts for higher convergence rate since one set of weights are estimated directly. It is this output that served as the input to the modified optical backpropagation algorithm.

Modified Optical Backpropagation Neural Network

In the standard backpropagation, the error at a single output unit is defined according to Equation as:

$$\delta_{pk}^o = (Y_{pk} - O_{pk}) \cdot f'_{k}(\text{net}_{pk}^o) \quad (4)$$

where the subscript "p" refers to the pth training vector, and "k" refers to the kth output unit, Y_{pk} is the desired output value, O_{pk} is the actual output from kth unit, then δ_{pk}^o will propagate backward to update the output-layer weights and the hidden-layer weights. In the Optical backpropagation (OBP), error at a single output unit is adjusted according to [20]Otair and Salameh (2005) as:

$$\text{New } \delta_{pk}^o = (1 + e^{(Y_{pk} - O_{pk})^2}) \cdot f'_{k}(\text{net}_{pk}^o), \quad \text{if } (Y - O) \geq \text{zero} \quad (5a)$$

$$\text{New } \delta_{pk}^o = - (1 + e^{(Y_{pk} - O_{pk})^2}) \cdot f'_{k}(\text{net}_{pk}^o), \quad \text{if } (Y - O) < \text{zero} \quad (5b)$$

The error function defined in Optical Backpropagation ([20]) 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) has been adopted to replace the traditional square error criterion used in Optical backpropagation. The Equation of the cubic error is given as:

$$\delta_{pk}^o = -3(Y_{pk} - O_{pk})^2 \cdot f'_{k}(\text{net}_{pk}^o) \quad (6)$$

The cubic error in Equation (2.4) was manipulated mathematically in order to maximize the error of each output unit which will be transmitted backward from the output layer to each unit in the intermediate layer. These are as shown in Equations (7a) and (7b) below:

$$\text{Modified } \delta_{pk}^o = 3((1 + e^t) \cdot f'_{k}(\text{net}_{pk}^o)) \quad \text{If } (Y_{pk} - O_{pk})^2 \geq 0 \quad (7a)$$

$$\text{Modified } \delta_{pk}^o = -3((1 + e^t) \cdot f'_{k}(\text{net}_{pk}^o)) \quad \text{If } (Y_{pk} - O_{pk})^2 < 0 \quad (7b)$$

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 is used to keep the training process going in the same general direction. In the modified Optical Backpropagation network, momentum was introduced. Hence, the weight update for the output unit is:

$$W_{kj}^o(t+1) = W_{kj}^o(t) + \mu W_{kj}^o(t) + (\eta \cdot \text{Modified } \delta_{pk}^o \cdot i_{pj}) \quad (8)$$

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:

- (i) Error signal function
- (ii) Application area

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 Equations (7a) and (7b)
7. Calculate the error term for the hidden units, through applying Modified δ_{pk}^o also

$$\text{Modified } \delta_{pj}^h = f_j^h(\text{net}_{pj}^h) \cdot (\sum_{k=1}^M \text{Modified } \delta_{pk}^o \cdot W_{kj}^o) \quad (9)$$

8. Update weights on the output layer.

$$W_{kj}^o(t+1) = W_{kj}^o(t) + \mu W_{kj}^o(t) + (\eta \cdot \text{Modified } \delta_{pk}^o \cdot i_{pj}) \quad (10)$$

9. Update weights on the hidden layer.

$$W_{ji}^h(t+1) = W_{ji}^h(t) + (\eta \cdot \text{Modified } \delta_{pj}^h \cdot X_i) \quad (11)$$

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 in OBP is stopped when the cubes of the differences between the actual and target values summed over units and all patterns are acceptably small.

The Hybrid Neural Network Algorithm

This research work employed hybrid of modified Counterpropagation and modified Optical Backpropagation neural networks for the training and classification of the input pattern. The training algorithm involves the following two stages:

Stage A: Performs the training of the weights from the input nodes to the Kohonen hidden node.

Step 1: Weight adjustment between the input layer and the hidden layer

The weight adjustment procedure for the hidden layer weights is same as that of the conventional CPN. It follows the unsupervised methodology to obtain the stabilized weights. This process is repeated for a suitable number of iterations and the stabilized set of weights are obtained. After convergence, the weights between the Kohonen hidden layer and the output layer are calculated.

Step 2: Weight adjustment between the hidden layer and the output layer

The weight adjustment procedure employed in this work is significantly different from the conventional CPN. The weights are calculated in the reverse direction without any iterative procedures. Normally, the weights are calculated based on the criteria of minimizing the error. But in this work, a minimum error value is specified initially and the weights are estimated based on the error value. The detailed steps of the modified algorithm are given below.

Step 1: The stabilized weight values are obtained when the error value (target output) is equal to zero (or) a predefined minimum value. The error value used for convergence in this work is 0.1. The following procedure uses this concept for weight matrices calculation.

Step 2: Supply the target vectors t_1 to the output layer neurons

Step 3: Since $(t_1 - y_1) = 0.1$ for convergence

$$(12)$$

The output of the output layer neurons is set equal to the target values as:

$$y_1 = t_1 - 0.1 \quad (13)$$

Step 4: Once the output value is calculated, the sum of the weighted input signals. Thus without any training methodology, the weight values are estimated. This technique accounts for higher convergence rate since one set of weights are estimated directly.

STAGE B: Performs the training of the weights from the second hidden node to the output nodes.

1. Calculate the net-input values from the Kohonen to the second hidden layer units.
2. Calculate the outputs from the second hidden layer.

3. Calculate the net-input values to the output layer units
4. Calculate the outputs from the output units
5. Calculate the error term for the output units, using Equations (7a) and (7b)
6. Calculate the error term for the hidden units, through applying modified δ^o_{pk} as in Equation (9)
7. Update weights on the output layer using (10)
8. Update weights on the hidden layer using Equation (11)

Repeat steps from step 1 to step 8 until the error ($Y_{pk} - O_{pk}$) is acceptably small for each of the training vector pair. The proposed algorithm as classical BP is stopped when the cubes of the differences between the actual and target values summed over units and all patterns are acceptably small.

The description of notation used in the training procedure is as given below:

- X_{pi} : net input to the i th input unit
 net^h_{pj} : net input to the j th hidden unit
 W^h_{ji} : weight on the connection from the i th input unit to j th hidden unit
 i_{pj} : net input to the j th hidden unit
 net^o_{pk} : net input to the k th output unit
 W^o_{kj} : weight on the connection from the j th hidden unit to k th output unit
 O_{pk} : actual output for the k th output unit

IV. 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.

4.1 Software Implementation

All the algorithms have been implemented using C# programming language and RUN under Windows7 operating system on Pentium (R) 2.00GB RAM, 1.83GH Processor and Pentium (R) 4.00GB RAM, 2.13GH Processor respectively.

4.2 Result Discussion

The results of the developed system were shown in table 4.1, 4.2 and 4.3

Table4.1: Results of variation of image size and Epoch Values

Image Sizes	COMOB	PSO-Based COMOB
5 by 7	750	600
10 by 14	1005	780
20 by 28	1220	1010

It was shown in table 4.1 that the more the dimensional input vector (character matrix size), the less the recognition performance due to introduction of more noise as the image size increases. 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 more 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 lower the recognition accuracy, although, the rate of change was small. The recognition time also increases as the image size increases due to increment in vector space to be processed by the network. From the above table, it is evident that the PSO-based MCPN is superior over the other classifiers in terms of convergence rate.

Table 4.2: The training Time (in minutes) of the three classifiers under different datasets

Dataset	COMOB Training Time	PSO-Based COMOB Training Time
1,200	0.21	0.13
2,480	3.01	2.32
3,720	4.15	3.62
4,960	9.60	6.45
6,200	15.98	12.11

From table 4.2 the training time of PSO-Based COMOB model is smaller when compared with ordinary COMOB model due to its ability to achieve dimensionality reduction and removal of irrelevant features of character images.

Table 4.3: Recognition Accuracies of COMOB and PSO-Based COMOB Classifiers

Dataset	COMOB	PSO-Based COMOB
1,200	83	85
2,480	86	89
3,720	89	91
4,960	91	96
6,200	95	99

Table 4.3 showed the effect of variation of database size and image size on recognition rates. Increase in database size had a positive proportionality relation to the recognition rates due to the fact that network was able to attribute the test character to larger character samples in the vector space. However, rate of increment in recognition rate with respect to database size was considerably small. PSO-Based COMOB has better recognition accuracy than ordinary COMOB model

V. 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 hybrid of Counterpropagation and modified Optical Backpropagation Neural Network model. Experimental results suggest better improvement in the classification accuracy for the PSO-based COMOB over ordinary COMOB classifier. However, an increase in the convergence rate is also achieved by the PSO-based 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 work 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.

REFERENCES

- [1] Anita, P. And Dayashankar S. (2010): ‘‘Handwritten English Character Recognition using Neural Network’’, *International Journal of Computer Science and Communication*, 1(2): 141-144.
- [2] Fenwa, O. D., Omidiora, E. O. and Fakolujo O. A. (2012): ‘‘Development of a Feature Extraction Technique for Online Character Recognition System, *Journal of Innovative Systems Design and Engineering*, International Institute of Science, Technology and Education, Vol. 3, No.3, pp. 10-23.
- [3] Fenwa O.D., Omidiora E.O. Fakolujo O.A. And Ganiyu R.A. (2012): ‘‘Development of a Writer-Independent Online Handwritten Character Recognition System using Modified Hybrid Neural Network Model’’, *Journal of Computer Engineering and Intelligent System (JCEIS)*, 3(4): 89-105. New York, USA,
- [4] Olusayo D. Fenwa, Funmilola A. Ajala and Alice O. Oke (2014): ‘‘A PSO-Based Modified Counterpropagation Neural Network Model for Online Handwritten Character Recognition System’’, *International Journal of Emerging Technology and Advanced Engineering*, 4(6): 768-776
- [5] Anoop, M. N. and Anil K.J. (2004): ‘‘Online Handwritten Script Recognition’’, *IEEE Trans. PAMI*, 26(1): 124-130.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.

- [10] 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.
- [11] Gupta, K., Rao, S.V., and Viswanath (2007): ‘‘Speeding up Online Character Recognition’’, Proceedings of Image and Vision Computing Newzealand, Hamilton: 41-45.
- [12] Shanthi, N., and Duraiwamy, K. (2007): ‘‘Performance Comparison of Different Image Sizes for Recognizing Unconstrained Handwritten Tamil Characters Using SVM’’, Journal of Science, 3(9): 760-764.
- [13] Manuel, J., Fonseca, and Joaquim, A.J., (2001): ‘‘Experimental Evolution of an Online Scribble Recognizer’’, Pattern Recognition Letters, 22(12): 1311-1319.
- [14] Rejean, P. and Sargur, S.N. (2000): ‘‘On-line and Off-line Recognition: A Comprehensive Survey’’, IEEE Transaction on Pattern Analysis and Machine Intelligence, 22(1): 63-84.
- [15] Kittler, J. and Roli, F. (2000): ‘‘1st International Workshop on Multiple Classifier Systems’’, Cagliari, Italy.
- [16] Kittler, J. and Roli, F. (2001): ‘‘2nd International Workshop on Multiple Classifier Systems’’, Cagliari, Italy.
- [17] Simon G. and Horst B. (2004): ‘‘Feature Selection Algorithms for the Generalization of Multiple Classifier Systems and their Application to Handwritten Word Recognition’’, Pattern Recognition Letters, 25(11): 1323-1336.
- [18] Marc, P., Alexandre, L., and Christian, G. (2001): ‘‘Character Recognition Experiment using Unipen Data’’, Parizeau and al., Proceeding of ICDAR, Seattle: 10-13.
- [19] 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.
- [20] Otair, M.A. and Salameh, W.A. (2005): ‘‘ Speeding up Backpropagation Neural Network’’ Proceedings Of the 2005 Informing Science and IT Education Joint Conference 2: 167-173.