Computer Engineering and Intelligent Systems
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol 3, No.7, 2012
www.iiste.org

Implementation of a Modified Counterpropagation Neural Network Model in Online Handwritten Character Recognition System

*Fenwa O.D., Emuoyibofarhe J. O., Olabiyisi S.O., Ajala F. A., Falohun A. S.
Department of Computer Science and Engineering,
Ladoke Akintola University of Technology, P.M.B 4000, Ogbomoso, Nigeria.
Tel: +2348163706286 E-mail: odfenwa@lautech.edu.ng

Abstract

Artificial neural networks are one of the widely used automated techniques. Though they yield high accuracy, most of the neural networks are computationally heavy due to their iterative nature. Hence, there is a significant requirement for a neural classifier which is computationally efficient and highly accurate. To this effect, a modified Counter Propagation Neural Network (CPN) is employed in this work which proves to be faster than the conventional CPN. In the modified CPN model, there was no need of training parameters because it is not an iterative method like backpropagation architecture which took a long time for learning. This paper implemented a modified Counterpropagation neural network for recognition of online uppercase (A-Z), lowercase (a-z) English alphabets and digits (0-9). The system is tested for different handwritten character samples and better recognition accuracies of 65% to 96% were obtained compared to related work in literature.

Keywords: Artificial Neural Network, Counterpropagation Neural Network, Character Recognition, Feature Extraction.

1. Introduction

As technology advances, computer systems become an invaluable asset of humans. It enhances communication and increases efficiency. Computer systems are greatly influencing the lives of human beings and their usage 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).

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 et al., 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). Neural network classifiers exhibit powerful discriminative properties and features such as ability to be trained automatically from examples, good performance with noisy data, parallel implementation and efficient tools for learning large databases. Hence, it has been popularly and extensively used in handwriting recognition (Jude, Vijila and Anitha, 2010).

2. Research Methodology

In this paper, four stages of development of the proposed character recognition system which include; data acquisition, character processing which consists feature extraction and character digitization, training and classification using modified Counterpropagation neural network model and testing are presented as shown in Figure 2.1. Experiments were performed with 6200 handwriting character samples (English uppercase, lowercase alphabet 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 Correct Recognition (CR), False Recognition (FR) and Recognition Failure (RF). The results of the proposed system showed better recognition performance compared with similar work in literature.

Character Acquisition and Feature Extraction

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 data (character information) such as stroke number and pressure of the stroke 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 (62 x 2 x 50) were collected from 50 subjects as each individual was requested to write each of the characters twice (this was 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. Samples of characters collected were as shown in Figure 2.2

Proposed Feature Extraction Method

For extracting the feature, the modified hybrid zone based feature extraction is used. The major advantage of this approach stems from its robustness to small variation, ease of implementation and provides good recognition rate. Zone based feature extraction method provides good result even when certain pre-processing steps like filtering, smoothing and slant removing are not considered.

The Zoning Algorithm

In this paper, a hybrid of modified Image Centroid and Zone-based (ICZ) and Zone Centroid and Zone-based (ZCZ) distance metric feature extraction algorithm proposed by Fenwa, Omidiora and Fakolujo (2012). Modifications of the two algorithms were in terms of:

(i) Number of zones being used

(ii) Measurement of the distances from both the Image Centroid and Zone Centroid

(iii) The area of application.

Few zones were adopted in their work but emphasis was being laid on how to effectively measure the pixel densities in each zone. However, pixels pass through each of the zones at varied distances that was why an average of five distances was measured for each of the zones at an angle of 20^0.

Hybrid of the Modified Zoning Feature Extraction Algorithms

The following were the algorithms to show the working procedure of the modified hybrid zoning feature extraction methods:

Modified Algorithm1: Image Centroid and Zone-based (ICZ) distance metric feature extraction algorithm (Fenwa, Omidiora and Fakolujo, 2012)

Input: Pre-processed character images

Output: Features for classification and recognition

Method Begins

Step 1: Divide the input image in to 25 equal zones as shown in Figure 2.3

Step 2: Compute the input image centroid as shown in Figure 2.4 using the formula:

$$\text{Centre of gravity in the horizontal direction (x–axis)} = \sum_{n=0}^{n-1} x_i \text{, where n = width} \tag{2.1}$$

$$\text{Centre of gravity in the vertical direction (y–axis)} = \sum_{m=0}^{m-1} y_i \text{, where m = height} \tag{2.2}$$

Step 3: Compute the distance between the image centroid to each pixel present in the zone as shown in Figure 2.4

Step 4: Repeat step 3 for the entire pixel present in the zone (five points in this case):

$$d = d_1 + d_2 + d_3 + d_4 + d_5 \tag{2.3}$$

Step 5: Compute average distance between these points as:

Average Image Centroid Distance $D_I = d/5$ (2.4)

where d = total distance between the image centroid to the pixel measured at an of angle 20^0

Step 6: Repeat this procedure sequentially for the entire zone (25 zones).

$$\text{Total Distance (P)} = D_{I1} + D_{I2} + D_{I3} + \ldots + D_{Im} \tag{2.5}$$

$$\text{Total Average Distance} \quad j = \sum_{z=0}^{m-1} \frac{D_Z}{m} \tag{2.6}$$

where m = 25 (total number of zones)

Step 7: Finally, 25 such features was obtained for classification and recognition.

Method Ends.

Modified Algorithm2: Zone Centroid and Zone-based (ZCZ) distance metric feature extraction algorithm (Fenwa, Omidiora and Fakolujo, 2012)

Input: Pre-processed character image

Output: Features for classification and recognition

Method Begins

Step 1: Divide the input image in to 25 equal zones as shown in Figure 2.3

Step 2: Compute the zone centroid for the entire pixel present in the zone as shown in Figure 2.5 using the formula:

$$\text{Centre of gravity in the horizontal direction (x–axis)} = \sum_{n=0}^{n-1} x_i \text{, where n = width} \quad (2.7)$$

$$\text{Centre of gravity in the vertical direction (y–axis)} = \sum_{m=0}^{m-1} y_i \text{, where m = height} \quad (2.8)$$

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

Step 4: Repeat step 3 for the pixel present in a zone (5 points in this case).

$$\text{Total Distance} \quad D = D_1 + D_2 + D_3 + D_4 + D_5 \quad (2.9)$$

Step 5: Compute average distance between these points as:

$$\text{Average distance } D_Z = D/5 \quad (2.10)$$

where D = distance between the zone centroid measured at angle 20^0 to the pixel in the zone.

Step 6: Repeat this procedure sequentially for the entire 25 zones.

$$\text{Total Distance (Q)} = D_{Z1} + D_{Z2} + D_{Z3} + .. + D_Zm \quad (2.11)$$

$$\text{Total Average Distance} \quad k = \sum_{z=0}^{m-1} \frac{D_Z}{m} \quad (2.12)$$

where m = 25 (total number of zones)

Step 7: Finally, 25 such features was obtained for classification and recognition.

Method Ends

The Hybrid Zoning Algorithm: Hybrid of Modified ICZ and Modified ZCZ (Fenwa, Omidiora and Fakolujo, 2012)

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

Digitization

Digitization of an image into a binary matrix of specified dimensions makes the input image invariant of its actual dimensions. Hence an image of whatever size gets transformed into a binary matrix of fixed pre-determined dimensions. This establishes uniformity in the dimensions of the input and stored patterns as they move through the recognition system. It is the conversion of the features selected in the canvas into binary digit which is then fed into the input layer of the neural network**.** The fact that all Neural Networks take numeric input and produces alphanumeric output inspired the idea that the data gathered for the Neural Network be represented numerically. The data were carefully scaled to enable the network to learn. If a white area is found in input character matrix then it will mark the corresponding pattern vector element as 0 (zero) and for black area it will mark as 1 (one). A method known as one-of-N encoding, which employs the use of numeric variables to represent a single nominal variable (e.g. character ‘A’ which is represented with the bits (000000011110000000..111000000000000111) was employed in the data representation for the network.

3. Related Work

Few attempts have been found in the literature in which conventional Counterpropagation Neural Network (CPN) architecture has been used for the recognition of handwritten characters. Ahmed (1995) made an attempt but only for digit recognition. Muhammad et al., (2005) implemented 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. This paper has focused on implementing a modified Counterpropagation neural network for recognition of online uppercase (A-Z), lowercase (a-z) English alphabets and digits (0-9).

Counterpropagation Neural Networks

An important component of training in the conventional CPN is reduction of the data set into a respective data set of lesser, specified size. This is achieved by the estimates of the dependent variable values corresponding to the new (and reduced in number) independent variable vectors can also be calculated (Hecht-Nielsen, 1990). Thus CPN actually operates as a closest-match lookup table and training a CPN is an attempt to appropriately reduce the size of the lookup table (Ahmed, 1995). Regarding the training process of the counter-propagation network, it can be described as a two-stage procedure: in the first stage the process updates the weights of the synapses between the input and the

Kohonen layer, while in the second stage the weights of the synapses between the Kohonen and the Grossberg layer are updated.

This work adopted the modified CPN proposed by Jude, et al, 2010 for abnormal brain tumour classification. Jude, Vijila and Anitha (2010) described Counterpropagation neural network as a hybrid neural network employing both supervised and unsupervised training methodologies. It consists of three layers: the input layer, the competition layer and the output layer. Given the input training set ($\bar{X}_i\ \bar{Y}_i$) i = 1, 2......N, where $\bar{X}_i$ = (x_{i1}, x_{i2}, ...x_{in}) and $\bar{Y}_i$ = (y_{i1}, y_{i2}, ...y_{in}), the configuration of the network is as follows: number of neurons in the input layer = n, number of neurons in the competition layer = N. The architecture of the network type used is as shown in Figure 2.6.

The learning algorithm proceeds in two steps. In the first step, each component of the training instance $\bar{X}_i$ is presented to the input layer. Let Uij be the arbitrary initial weights vector assigned to the links connecting input node i with the competition node j. The transfer function of the competition layer is defined by the Euclidean distance d_j between the weight vector $\bar{U}_i$ and the input vector $\bar{X}_k$ as:

$$d_j = //\bar{U}_j - \bar{U}_k// = 1/2(\textstyle\sum_i (Uij - Xki)^2)^{1/2} \quad j = 1,2, ...N \qquad (2.13)$$

For each X_k, each node in the competition layer competes with the other nodes, and the node with the shortest Euclidean distance wins. The output of the winning node is set to 1 and the rest to 0, thus, the output of the jth node in the competition layer is

$$Z_j = 1.0 \quad \text{if } d_j < d_i \qquad (2.14)$$

$$Z_j = 0.0 \quad \text{otherwise} \qquad (2.15)$$

The weight updation between the input layer and the competition layer is given by

$$U_{ij}^{t+1} = U_{ij}^{t} + \alpha(x_{ji} - U_{ij}^{t}) \qquad (2.16)$$

where t is the iteration number and α, the learning coefficient is such that $0<\alpha \le 0.8$, as suggested by Heicht- Neilsen in 1990. After the weight vectors U*ij* have stabilized, as the second step, the output layer begins learning the desired output.

The weight adjustments for the output layer are given by:

$$V_{ji}^{t+1} = V_{ji}^{t} + \beta\,(y_{ki} - V_{ji}^{t})Z_j \qquad (2.17)$$

where the learning coefficient lies in the range $0 < \beta \le 1.0$

A stabilized set of weights are obtained by training the neural network with the input samples. After training, the network was tested with unknown data set. The major drawback of the CPN is the high convergence time period which leads to computational complexity. A suitable modification is made in the conventional CPN to improve the convergence rate besides yielding sufficient classification accuracy.

Modified CPN Training Algorithm (Jude, Vijila, and Anitha, 2010)

The training algorithm involves the following two phases:

(i) 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. Equations (2.13) to (2.17) summarises this procedure. This process is repeated for a suitable number of iterations and the stabilized set of weights are obtained. After convergence, the weights between the hidden layer and the output layer are calculated.

(ii) 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. In conventional CPN, 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 CPN 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 following procedure uses this concept for weight matrices calculation.

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

Step 3: Since $(t_l - y_1) = R$ for convergence (2.18)

where t_l is the target value (target output), y_1 is the network output and R is the minimum error value (threshold value). The output of the output layer neurons is set equal to the target values as:

$$y_1 = t_l - R \quad (2.19)$$

Step 4: Once the output value is calculated, the sum of the weighted input signals (y_in_1) can be estimated. Since the sigmoid activation function is used, the following equation yields the value for y_in_1

$$y\_in_1 = \ln \left[\frac{y_1}{/1-y_{-j}}\right] \quad (2.20)$$

Step 5: Based on the values of y_in_1, the weight matrix v_{j1} is calculated using the following expression.

$$y\_in_1 = \Sigma\, h_j . v_{j1} \quad (2.21)$$

where h_j is the output value of the hidden layer and this value is obtained at the completion of phase 1.

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.

.

4. Results and Discussion

The modified CPN model did not require training parameters because it is not an iterative method like back-propagation architecture which took a long time for learning. Rather, a minimum error value is specified initially and the weights are estimated based on the specified error value and this is what accounts for higher convergence rate of the model since one set of weights are estimated directly.

Seven different data sets: 5 samples/character, 11 samples/character, 22 samples/character 33 samples/character, 44 samples/character, 55 samples/character, and 66 samples/character were being experimented to evaluate the performance of the modified CPN model with gradually increasing the number of samples/character. The system was evaluated on samples taken from individuals who did not participate in the initial process of data acquisition for the training data set. This was done keeping in view the eventual aim of using the model in practical online recognition system. The quality of an online handwriting recognizer is related to its ability to translate drawn characters irrespective of writing styles.

A general trend of increase in performance with increase in samples/character was observed in the experiment. False Recognition (FRs) is an important factor in any recognition system, the lower the false recognition rate, the more reliable the system becomes. The developed system was able to obtain lower False Recognition (FR), lower Recognition Failure (RF) and higher Correct Recognition (CR) compared with related work in literature. Table 2.1 showed the experimental results of both Conventional CPN model (Mohammad, et al 2005) and the modified CPN model. Bar chart representation of the results is as shown in Figure 2.7

5. Conclusion

In conclusion, the result of this work showed better recognition performances in terms of accuracy and speed compared with existing work in literature. Future work can explore the necessity to integrate into the learning algorithm, an optimization algorithm to further enhance the performance of the system.

6. References

Ahmed, S.M. (1995): ''Experiments in character recognition to develop tools for an optical character recognition system'', IEEE Inc. First National Multi Topic Conference Proceedings, NUST, Rawalpindi, Pakistan, 61-67.

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

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

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.

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.

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.

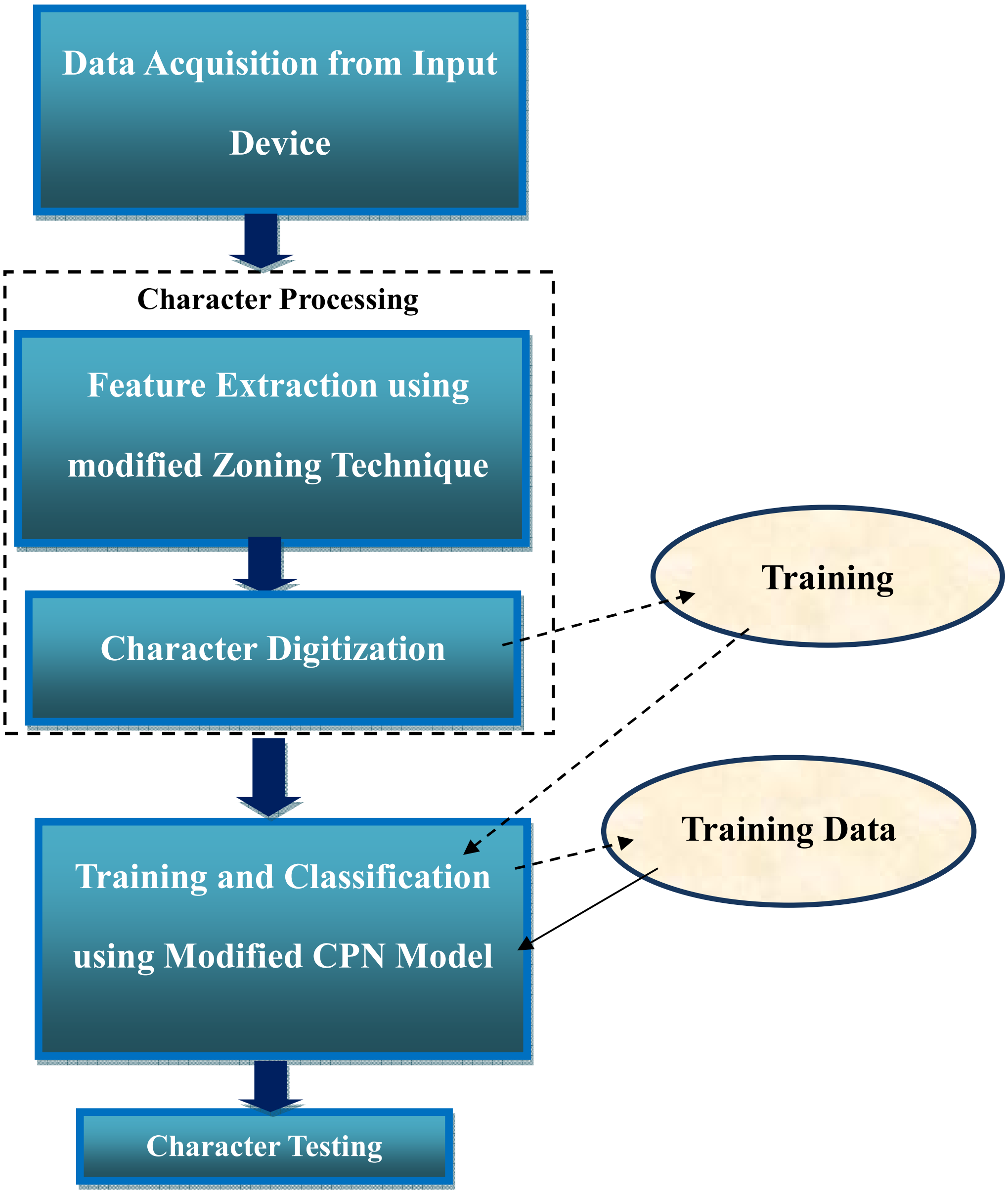

Figure 2.1: Block Diagram of the proposed Character Recognition System

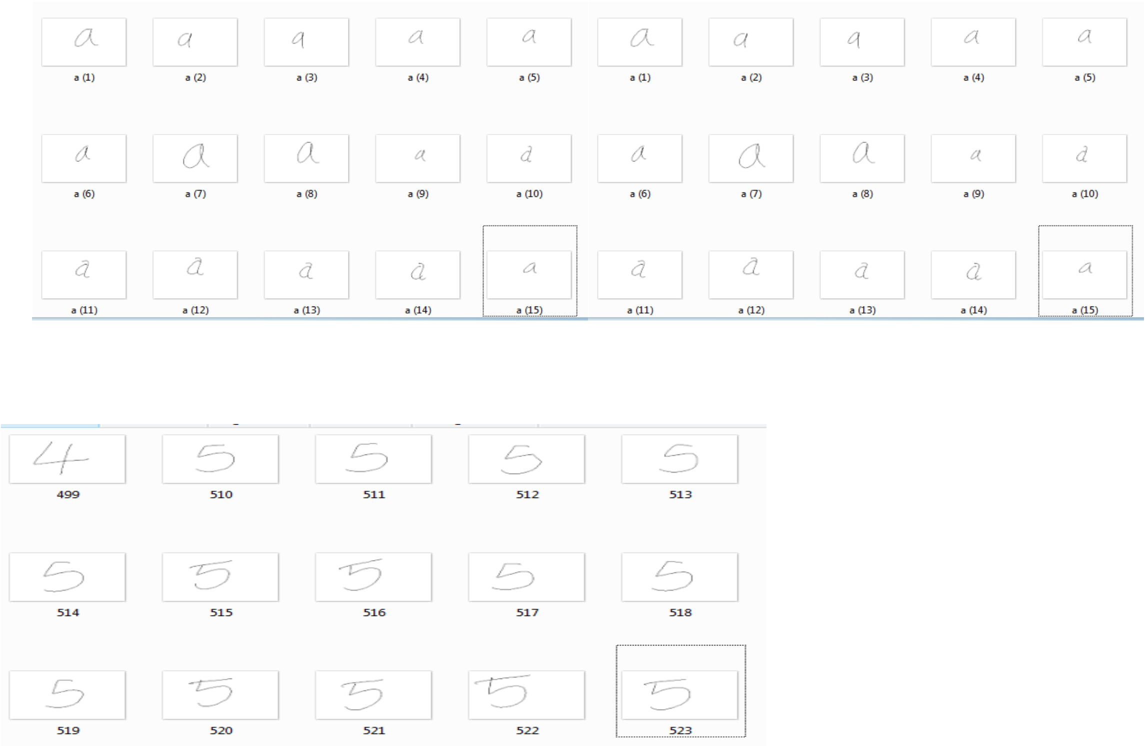

Figure 2.2: Sample handwritten characters

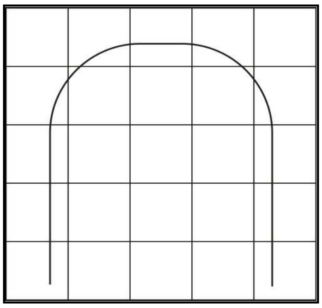

Figure 2.3: Character ‘n’ in 5 by 5 (25 equal zones)

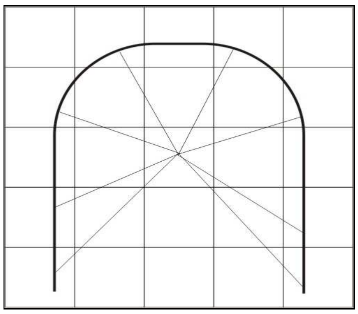

Figure 2.4: Image Centroid of character ‘‘n’’ in zoning

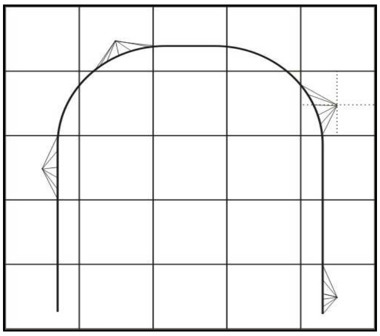

Figure 2.5: Zone Centroid of character 'n' in zoning

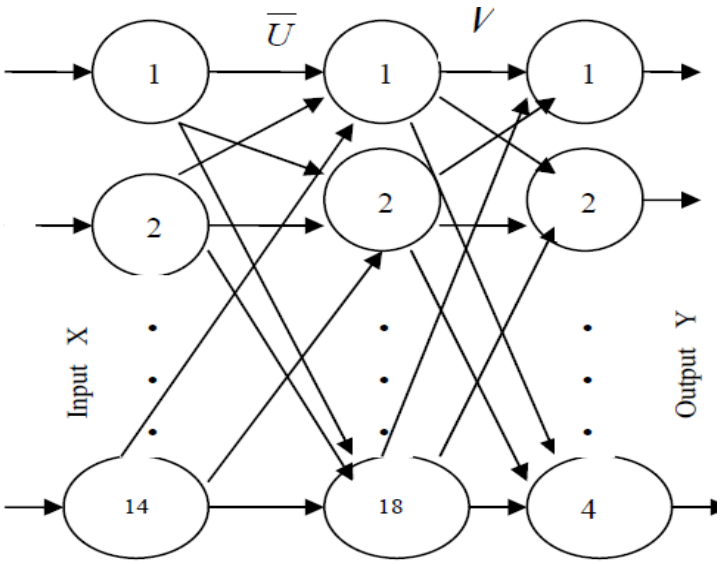

Figure 2.6: Topology of CPN (Jude, Vijila, and Anitha, 2010)

Table 2.1: Experimental results of conventional CPN model and modified CPN model

	Conventional Counter propagation (Mohammad, et al 2005)			Modified Counterpropagation		
Character Samples	CR (%)	FR (%)	RF (%)	CR (%)	FR (%)	RF (%)
5 Each	60	40	0	65	10	0
11 Each	79	12	0	80	5	0
22 Each	76	23	1	79	4	0
33 Each	84	15	1	86	7	1
44 each	82	17	1	84	4	1
55 each	88	8	4	90	4	2
66 each	93	6	1	96	6	1

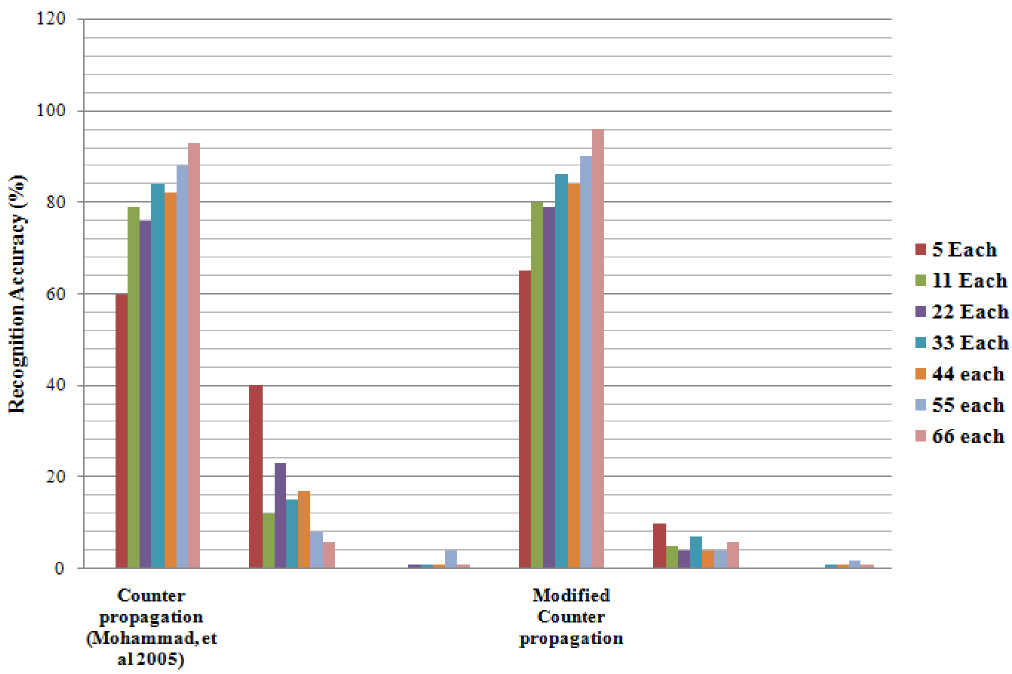

Figure 2.7: Performance comparison of conventional CPN and modified CPN of Table 2.1