Malaysian Journal of Computer Science, Vol. 11 No. 1, June 1998, pp. 58-64

A NEURAL NETWORK BASED CHARACTER RECOGNITION SYSTEM USING DOUBLE BACKPROPAGATION

Joarder Kamruzzaman and S. M. Aziz
Department of Electrical & Electronic Engineering
Bangladesh University of Engineering & Technology
Dhaka-1000, Bangladesh
Tel:880-2-861594
Fax:880-2-863026
email: kjoarder@eebuet.bdmail.net

ABSTRACT

Proposes a neural network based invariant character recognition system using double backpropagation network. The model consists of two parts. The first is a preprocessor which is intended to produce a translation, rotation and scale invariant representation of the input pattern. The second is a neural net classifier. The outputs produced by the preprocessor at the first stage are classified by this neural net classifier trained by a learning algorithm called double backpropagation. The recognition system was tested with ten numeric digits (0~9). The test included rotated, scaled and translated version of exemplar patterns. This simple recognizer with double backpropagation classifier could successfully recognize nearly 97% of the test patterns.

Keywords: ***Neural networks, backpropagation, double backpropagation, character recognition, Rapid Transform***

1.0 INTRODUCTION

Character recognition is one of the applications of pattern recognition which has enormous scientific and practical interest. A lot of scientific efforts have been dedicated to pattern recognition problems and much attention has been paid to develop recognition system that must be able to recognize an object regardless of its position, orientation and size. Numerous methods have been proposed for achieving invariance in pattern recognition. These include statistical as well as structural approaches. Some of these approaches are Fourier descriptor [1]-[3], moment invariants [4]-[6], stochastic models [7]-[8] and decomposition techniques [9].

Recently neural networks have been applied to character recognition as well as speech recognition with performance, in many cases, better than the conventional method [10]. Several neural network based invariant character recognition systems have been proposed. A pattern recognition system using layered neural networks called ADALINE was proposed by Windrow et. al. [11]. The proposed system needs large number of slabs of neurons, each slab being invariant to specific degree of translation, rotation or scaling. This makes the resulting network very large and complex since the network must accommodate all the possible degrees of translation, rotation and scaling. Neocognitron developed by Fukushima [12] is not so effective for rotation invariance. Higher order neural networks are also proposed to achieve invariance [13], but use of higher order increases the number of connections astronomically and makes its implementation for large scale image planes extremely difficult.

This paper presents a character recognition system which uses a preprocessor and an artificial neural network classifier trained by double backpropagation algorithm [14]. The preprocessor extracts the geometrical features of the input pattern in such a way that for any rotated or scaled version of input pattern, the feature values are reduced to cyclically shifted version of those of standard pattern, and then uses Rapid Transform [15], a simple cyclic shift invariant transform, to make the preprocessed outputs invariant to scaling, rotation and translation. The preprocessor is simple, computationally inexpensive and does not increase the overall complexity of the recognition systems. The classifier is trained by a relatively new feedforward learning algorithm called double backpropagation which has been reported to perform better than the popularly used backpropagation algorithm [16]. The performance of the recognition system was tested with ten English digits (0~9) at different degrees of rotation, scaling and translation. Test was also done with classifier of different number of input and hidden units.

Section 2 provides a brief description of different preprocessing steps and double backpropagation learning algorithm. Experimental results and conclusions are presented in section 3 and 4, respectively.

2.0 RECOGNITION SYSTEM

Fig. 1 shows the recognition system which is based on preprocessing the input first by a preprocessor and then classifying the preprocessed outputs by neural net classifier. The system is described below. The preprocessor used in this system is a modification of the one proposed by Ito [17].

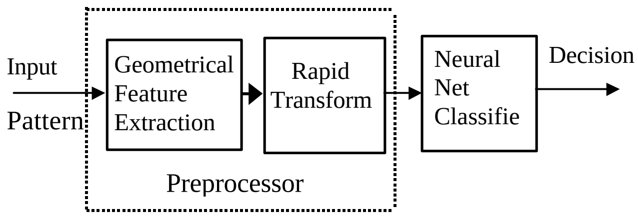

Fig. 1: The recognition system.

2.1 Preprocessor

The proposal of preprocessing is to create an intermediate representation of the input pattern which later serves as inputs to the classifier. To achieve high recognition rate, the preprocessed outputs must remain unchanged or little changed even if the input pattern is rotated, scaled or translated before attempting classification. The pre-processor consists of two blocks whose functions are described below.

2.1.1 Geometric Feature Extraction

Each pattern is represented by an NxN binary pixel where an '1' represents an *on*-pixel and an '0' represents an *off*-pixel. This block computes the center of gravity of the image by arranging the *x* and *y* co-ordinates of the *on*-pixels. Co-ordinates of all the *on*-pixels are then shifted with respect to this center. The center of gravity $(\bar{x}, \bar{y})$ of the pattern is calculated as

$$\bar{x} = \sum_{i=1}^{N} \sum_{j=1}^{N} f(x_i, y_j).x_i \Big/ \sum_{i=1}^{N} \sum_{j=1}^{N} f(x_i, y_j),$$

$$\bar{y} = \sum_{i=1}^{N} \sum_{j=1}^{N} f(x_i, y_j).y_j \Big/ \sum_{i=1}^{N} \sum_{j=1}^{N} f(x_i, y_j),$$

where the function $f(x_i,y_j)$ of the pixel (x_i,y_j) is either 1 or 0. The radial distance (r_{ij},θ_{ij}) of each *on*-pixel with respect to the center of gravity $(\bar{x}, \bar{y})$ is calculated. The computed radial distance is then normalized by the average radial distance of all *on*-pixels. The computation is as follows:

$$r_{ij}(x_i, y_j) = \sqrt{(x_i - \bar{x})^2 + (y_j - \bar{y})^2},$$

$$\theta_{ij}(x_i, y_j) = \cos^{-1} \frac{(x_i - \bar{x})}{r_{ij}(x_i, y_j)},$$

$$r'_{ij} = \frac{r_{ij}(x_i, y_j)}{r_{av}},$$

$$r_{av} = \sum_{i=1}^{N} \sum_{j=1}^{N} r_{ij}(x_i, y_j) \Big/ \sum_{i=1}^{N} \sum_{j=1}^{N} f(x_i, y_j),$$

where $r_{ij}(x_i,y_j)$ is the radial distance of the pixel (x_i,y_j) from $(\bar{x}, \bar{y})$ and r'_{ij} is the normalized value of $r_{ij}(x_i,y_j)$ by the average radial distance r_{av}. $\theta_{ij}(x_i,y_j)$ is the angle the pixel (x_i,y_j) makes with the *x*-axis considering $(\bar{x}, \bar{y})$ as the origin. For all i and j, the pair (r_{ij}, θ_{ij}) is described on the r-θ plane according to the increasing order of θ_{ij}.

The feature values (r_{ij},θ_{ij}) are translation invariant. The outputs of Geometric Feature Extraction block (r_{ij},θ_{ij}) are both translation and scale invariant, or at least almost invariant in case of deformation due to scaling. The most important property of the output of this block is that, any rotation of the pattern by γ degree will cause (r_{ij},θ_{ij}) to be cyclically shifted by nearly γ degree along the θ-axis. Thus, to make the system rotational invariant, a cyclic shift invariant transform is needed. The outputs of the present block go through the Rapid Transform (RT) in the next stage and this makes the preprocessed outputs nearly invariant to translation, scaling and rotation.

The condition of using the Rapid Transform is that the number of inputs L must be of the form $L=2^M$. So (r_{ij}, θ_{ij}) are grouped into 2^M groups which are then used as inputs of RT. All the r_{ij}'s within the angel-slot of $\Delta\theta$ $(\Delta\theta=2\pi/L)$ are summed up to compute X_l, the l-th input to RT, in the following manner.

$$X_l = \frac{1}{n_l} \sum_{\Delta\theta . l \le \theta_{ij} \le \Delta\theta . (l+1)} r'_{ij} \Big|_{r'_{ij}, \theta_{ij}}, \qquad 0 \le l \le L-1$$

Where n_l is the number of (r_{ij}, θ_{ij}), i.e., *on*-pixels within the l-th interval. The reason of dividing by n_l is that, when scaled, a pattern may become thick or thin causing increment or decrement of *on*-pixels within that interval.

Normalization by n_l can tackle this thicking/thinning problem.

2.1.2 Rapid Transform

Rapid Transform (RT) [15] has the property of circular shift and reflection invariance of the input data sequence. RT is attractive for its computational simplicity. RT is not an orthogonal transform and has no inverse transform. The computation of RT is as follows:

$$X_{2l}^{(R)} = \left| X_{l}^{(R-1)} + X_{l+L/2}^{(R-1)} \right|,$$

$$X_{2l}^{(R)} = \left| X_{l}^{(R-1)} - X_{l+L/2}^{(R-1)} \right|, \quad (l = 0,1,\ldots, L/2-1)$$

where L is the number of input data, R is the transformation step, ($1 \le R \le M$), $M = \log_2 L$), M is the number of required steps, and X_l is the l-th component of the transformed data.

The preprocessed outputs will be invariant to translation, scaling and rotation in an ideal case, i.e., on the assumption that scaling and rotation do not introduce noise in the pattern. However, in practical cases, a pattern is always deformed when scaled and/or rotated introducing some amount of noise. The preprocessed outputs will therefore, be somewhat changed on scaling or rotation. A neural network which has the desired capability of classifying patterns even with some amount of noise present is used as classifier in this system with the expectation that it will be able to deal with the deviation caused by scaling and rotation. A detailed description of the classifier is given in the following section.

2.2 Neural Network Classifier

At the final stage, a neural network classifier is used for classification. A relatively new multilayer feedforward learning algorithm called double backpropagation [14] is used to train the classifier. Compared to the standard backpropagation [16], double backpropagation has been reported to have improved generalization capability [14], [18]. In this paper, both types of networks were trained as classifier and a comparison is presented. The outputs of the preprocessor were given as inputs and locally represented target vectors were taken as the outputs of the network. In the following lines, double backpropagation algorithm is briefly discussed.

In double backpropagation, the energy term to be minimized is of the form

$$E = E_f + E_b$$

$$= E_f + \left\{ \frac{1}{2}\left(\frac{\partial E_f}{\partial x_1}\right)^2 + \frac{1}{2}\left(\frac{\partial E_f}{\partial x_2}\right)^2 + \cdots + \frac{1}{2}\left(\frac{\partial E_f}{\partial x_l}\right)^2 \right\},$$

where x_i refers to the i-th (of total l) input component, E_b is called the backward energy and E_f is the energy function used in standard backpropagation and defined as

$$E_f = \frac{1}{2} \sum_{j=1}^{m} \left(d_j - y_j\right)^2,$$

where d_j and y_j are the j-the component of the desired and actual output, respectively.

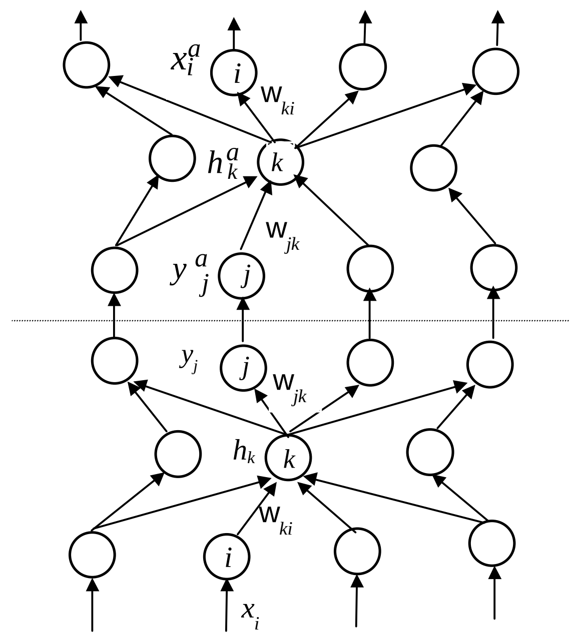

Fig. 2: A double backpropagation network. The network above the dotted line is the appended network. Not all the weights are shown.

Fig. 2 shows a double backpropagation network. The network below the dashed line is the original backpropagation network and the one above the dotted line is the appended network [14]. The learning algorithm is implemented in the following steps.

1) The input to the network propagates forward through the original network, produces hidden unit's output h_k and output unit output y_j, and weight change is calculated as follows:

$$\Delta\omega_{jk} = \eta\delta_j h_k,$$

$$\Delta\omega_{ki} = \eta\delta_k x_i,$$

$$\delta_j = \left(d_j - y_j\right) f'\left(net_j\right),$$

$$\delta_k = \left(\sum_{j=1}^{m} \delta_j \omega_{jk}\right) f'\left(net_k\right),$$

$$h_k = f\left(\mathrm{net}_k\right) = f\left(\sum_{i=1}^{m} \omega_{ki}\, x_i + \theta_k\right),$$

$$\mathrm{y}_i = f\left(\mathrm{net}_j\right) = f\left(\sum_{k=1}^{q} \omega_{jk}\, h_k + \theta_j\right),$$

where '*i*', '*k*', '*j*' represents an input, hidden and output unit respectively, *l*,*q*,*m* are the respective total numbers, and q's are the bias values.

2) Signal is propagated forward through the appended network and produces outputs y_j^a, h_k^a and x_i^a at unit '*j*', '*k*' and '*i*' respectively as follows:

$$y_j^a = \left(y_j - d_j\right) f'\left(net_j\right),$$

$$h_k^a = \left(\sum_{j=1}^{m} y_j^a\, \omega_{jk}\right) f'\left(net_k\right),$$

$$x_i^a = \left(\sum_{k=1}^{q} h_k^a\, \omega_{ki}\right).$$

Backward energy function is backpropagated from the top of the appended network down through the appended network. Weight change in the appended network is given by

$$\Delta \mathrm{w}_{ki} = -\mathrm{h}\, x_i^a\, h_k^a,$$

$$\Delta \mathrm{w}_{jk} = \mathrm{h}\left(\sum_{l=1}^{l} x_i^a\, \mathrm{w}_{ki}^a\right) f'\left(net_k\right) \cdot y_j^a.$$

Thus, in full double backpropagation, weight change is first calculated for one backward pass through the original network and then for another backward pass through both the appended and original network. In one training cycle, total change in each weight is the summation of these two changes.

3.0 SIMULATION RESULTS AND DISCUSSION

Based on the system described in the preceding section, simulation was carried out to classify and recognize ten English digit (0~9). Each digit consists of 64x64 binary image. One exemplar pattern per category is considered. Exemplary patterns are shown in Fig. 3. Each pattern goes through the preprocessing step and produces outputs which are then fed to neural network classifier. Here, 9 different architectures with different number of hidden (5,7,9) and input (16,32,64) units have been studied to investigate the effect of network size on overall performance. At the output layer, 10 units are used to represent 10 categories.

690 different patterns are used in classification phase. The test set for each category consists of 10 translated, 18 rotated, 15 scaled, and 36 both scaled and rotated patterns. The rotated test patterns were formed by rotating the exemplar from 5° to 90° at 5° interval. Similarly, test patterns were formed by scaling the pattern by factor 0.5 (50% reduction) to 2.0 (200% enlargement) at 0.1 interval. Test patterns both scaled and rotational are formed first by scaling by factor 0.8 and 1.5 and then by rotating from 5°-90°. A sample of test patterns are shown in Fig. 4.

Fig. 5 shows the outputs of the feature extraction stage for exemplar pattern '5' and its 90° rotated version. It shows that the output is cyclically shifted by 90° when the pattern is rotated. Since RT is cyclic shift invariant,

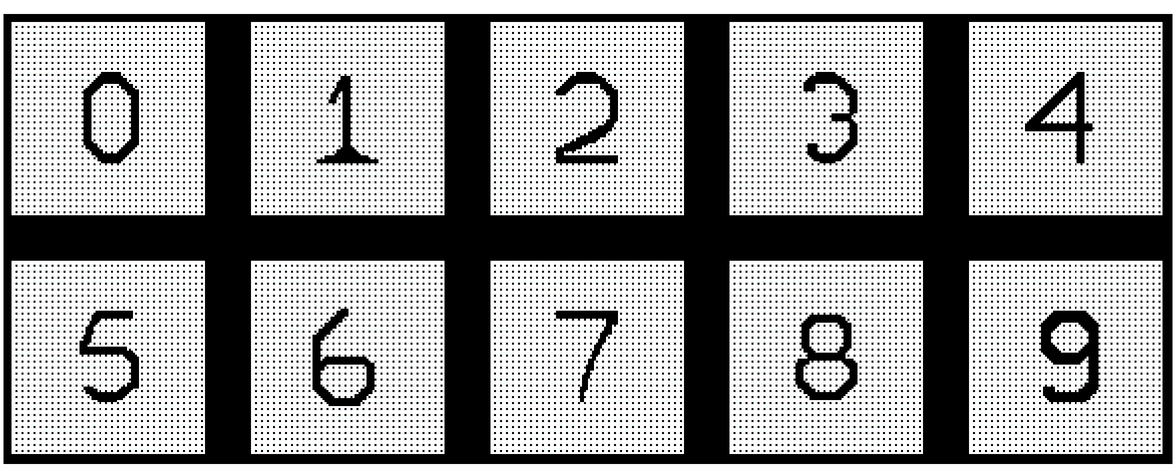

Fig. 3: Ten English digits (0~9) used in simulation

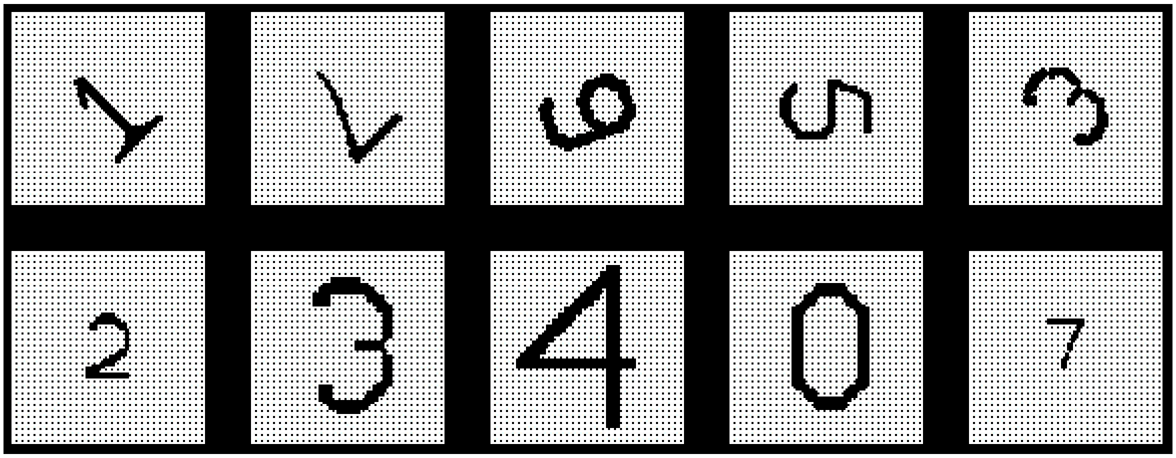

Fig. 4: Examples of some of test patterns used to test recognition ability of the system

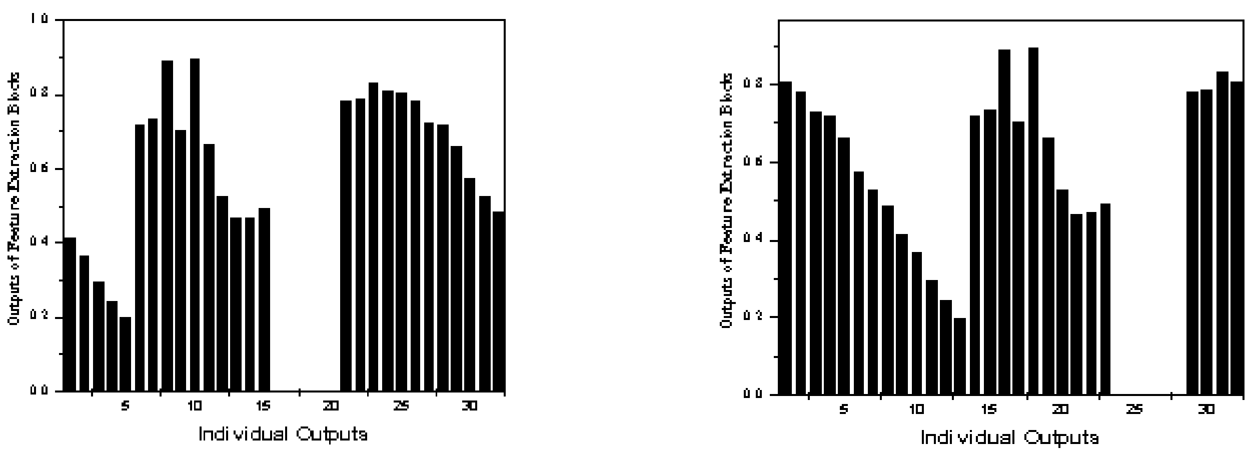

Fig. 5: (a) Outputs of Feature Extraction Block for exemplar pattern '5'. (b) those for 90° rotated version of '5'

outputs of the preprocessor remain the same in both cases. This substantiates the efficacy of the preprocessor. However, rotation of any exemplar pattern by an angle other than 90° deforms the pattern and causes deviation in the preprocessed output. This deviation is expected to be tackled by the classifier.

Recognition ability of double backpropagation network and standard backpropagation was tested on the test patterns mentioned earlier. Upon presenting a test pattern to the system, the test pattern was recognized to belong to the category represented by the output unit of maximum value. Results are summarized in Table 1. Each network was trained 10 times and results presented in the table are the average of 10 trials. In the case of the translated pattern, recognition ability is always 100% and is not shown in the table. A 32-9-10 double BP network (32 input units, 9 hidden units, 10 output units) shows an overall recognition ability of nearly 97% on the test patterns.

Table 1: Percentage of correct recognition by BP and double BP network

Learning Algorithm	Network size	Rotation	Scaling	Rotation & Scaling
BP	16-9-10	94.55	90.00	86.14
	32-9-10	95.17	95.16	89.44
	64-9-10	99.56	88.06	85.55
Double BP	16-9-10	96.94	94.53	91.69
	32-9-10	98.83	97.06	94.59
	64-9-10	99.61	93.93	92.39

The number of inputs to the classifier actually depends on the number of inputs and outputs of Rapid Transform. For example, when the number of inputs and outputs of RT is 64, input size of classifier is 64. This can be interpreted as extracting the feature of input pattern within a window of

5.625° angle (360°/64=5.625°) looking through the center of gravity of the pattern. Similarly, 11.25° and 22.5° angle for 32 and 16 inputs respectively. Results reveal that varying the number of input size does vary the recognition performance. Increasing the input size improves the recognition rate for rotation but does not guarantee improvement for scaling [18]. A narrower window gives finer details in feature extraction and consequently gives better recognition for rotation. But thicking/thinning due to scaling, specially for size reduction, seems not to work well in narrower window. Since a single network is used for classification, we must choose a network that gives good results in all cases. A network of a given size might perform best for rotation and another for scaling. In this experiment, a 32-9-10 network yielded best recognition rate. Again double backpropagation network shows better performance than standard backpropagation. However, the best network size is problem specific and has to be determined by trial.

4.0 CONCLUSION

In this paper, a simple invariant pattern recognition system is proposed. The preprocessor needs simple computational steps which make it computationally inexpensive and easier to implement. When the neural network is trained by double backpropagation algorithm, the system shows better recognition performance. However, the system needs to be tested with real-world data, e.g., handwritten characters and other similar applications.

REFERENCES

[1] C. T. Zahn and R. Z. Roskies, "Fourier Description for Plane Closed Curves", *IEEE Trans. Computers.*, Vol. C-21, 1972, pp. 269-281.

[2] E Persoon and K. S. Fu, "Shape Discrimination using Fourier Descriptors", *IEEE Trans. Syst. Man Cybern.*, Vol. 7, 1977, pp. 170-179.

[3] S. Wang, P. Chen and W. Lin, "Invariant Pattern Recognition by Moment Fourier Descriptor", *Pattern Recognition*, Vol. 27, No. 12, 1994, pp. 1735-1742.

[4] S.A. Dudani et. al., "Aircraft Identification by Moment Invariants", *IEEE Trans. Computers*, vol. C-26, 1977, pp. 39-45.

[5] S. S. Reddi, "Radial and Angular Moment Invariants for Image Identification", *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 3, 1981, pp. 240-242.

[6] L. Gupta and M. D. Srinath, "Contour Sequence Moments for the Classification of Closed Planer Shapes", *Pattern Recognition*, Vol. 20, 1987, pp. 267-272.

[7] R. L. Kashyap and R. Chellappa, "Stochastic Models for Close Boundary Analysis: Representation and Reconstruction", *IEEE Trans. Information Theory*, Vol. IT-27, No. 5, 1981, pp. 627-637.

[8] M. Das, M. J. Paulik and N. K. Loh, "A Bivariate Autoregressive Modeling Technique for Analysis and Classification of Planer Shapes", *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 12, 1990, pp. 97-103.

[9] T. Pavlidis, *Structural Pattern Recognition*, Springer-Verlag, New York, 1977.

[10] I. Guyan et. al., "Design of a Neural Network Character Recognizer for a Touch Terminal", *Pattern Recognition*, Vol. 24, No. 2, 1991, pp. 105-119.

[11] B. Widrow, R. Winter and R. A. Bayter., "Layered Neural Nets for Pattern Recognition", *IEEE Trans. Acoustics, Speech and Signal Process.*, Vol. 36, No. 7, 1988, pp. 1109-1118.

[12] K. Fukushima, S. Miyake and T. Ito, "Neocognitron: A Neural Network Model for a Mechanism of Visual Pattern Recognition", *IEEE Trans. Syst., Man, Cyber.*, Vol. 13, No. 5, 1983, pp. 826-834.

[13] L. Spirokovsha and M. B. Reid., "Connectivity Strategies for Higher-Order Neural Networks", in *Proc. Intern. Joint Conf. on Neural Networks, IJCNN, 1990, San Diego*, Vol. 1, pp. 21-26.

[14] H. Drucker and Y.L. Cun, "Improving Generation Performance Using Double Backpropagation", *IEEE Trans. on Neural Networks*, Vol. 3, No. 6, 1992.

[15] H. Reitboeck and T. P. Brody, "A Transformation With Invariance Under Cyclic Permutation for Applications in Pattern Recognition", *Inf. & Control.*, Vol. 15, No. 2, 1969, pp. 130-154.

[16] D.E. Rumelhart et. al. and the PDP Research group, *Parallel Distributed Processing*, Vol. 1, MIT Press, 1986.

[17] K. Ito, M. Hamamoto, J. Kamruzzaman and Y. Kumagai, "Invariant Object Recognition by Artificial Neural Network using Fahlman and Lebeire's Learning Algorithm," *IEICE Trans. Fundamentals*, Vol. E76-A, No. 7, 1993, pp. 1267-1272.

[18] J. Kamruzzaman, Y. Kumagai and S. M. Aziz, "Character Recognition by Double Backpropagation Neural Network" *in Proc. TENCON'97, Australia*, 1997, pp. 411-414.

BIOGRAPHY

Joarder Kamruzzaman received the B.Sc. and M.Sc. degrees in Electronic Engineering from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh in 1986 and 1988, respectively, and the Dr. Eng. Degree in Information Systems Engineering from Muroran Institute of Technology, Muroran, Japan in 1993. He is currently an Associate Professor in the Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology. He has authored and co-authored more than 25 publications in science and technical journals, and international conference proceedings. His research interest includes neural networks, pattern recognition, image processing, and related applications.

S. M. Aziz is a Professor of Electrical and Electronic Engineering at Bangladesh University of Engineering and Technology (BUET). He received B.Sc. and M.Sc. degrees in Electrical and Electronic Engineering from the same university in 1984 and 1986 respectively. He obtained his Ph.D. degree in Electronic Engineering in 1993 from the University of Kent, UK and specialized in VLSI design and testability. In 1996, Dr. Aziz was a visiting scholar at the University of Texas at Austin and worked at Crystal Semiconductor Corporation in advanced VLSI Circuit Design. His research interests include design for testability, VHDL based design, computer arithmetic and architectures, neural networks etc. Dr. Aziz is a member of *IEEE Computer Society*, *Circuits and Systems Society*, and *Solid State Circuits Society*.