On-line Handwritten Character Recognition: An Implementation of Counterpropagation Neural Net

Muhammad Faisal Zafar, Dzulkifli Mohamad, and Razib M. Othman

***Abstract*—**On-line handwritten scripts are usually dealt with pen tip traces from pen-down to pen-up positions. Time evaluation of the pen coordinates is also considered along with trajectory information. However, the data obtained needs a lot of preprocessing including filtering, smoothing, slant removing and size normalization before recognition process. Instead of doing such lengthy preprocessing, this paper presents a simple approach to extract the useful character information. This work evaluates the use of the counter- propagation neural network (CPN) and presents feature extraction mechanism in full detail to work with on-line handwriting recognition. The obtained recognition rates were 60% to 94% using the CPN for different sets of character samples. This paper also describes a performance study in which a recognition mechanism with multiple thresholds is evaluated for counter-propagation architecture. The results indicate that the application of multiple thresholds has significant effect on recognition mechanism. The method is applicable for off-line character recognition as well. The technique is tested for upper-case English alphabets for a number of different styles from different peoples.

***Keywords*—**On-line character recognition, character digitization, counter-propagation neural networks, extreme coordinates

I. Introduction

HANDWRITING processing is a domain in great expansion. The interest devoted to this field is not explained only by the exciting challenges involved, but also the huge benefits that a system, designed in the context of a commercial application, could bring [27]. Two classes of recognition systems are usually distinguished: online systems [4, 24, 34] for which handwriting data are captured during the writing process, which makes available the information on the ordering of the strokes, and offline systems [33] for which recognition takes place on a static image captured once the writing process is over. The field of personal computing has begun to make a transition from the desktop to handheld devices, thereby requiring input paradigms that are more suited for single hand entry than a keyboard. Online handwriting recognition allows for such input modalities. On-line handwritten scripts are usually dealt with pen tip traces from pen-down to pen-up positions.

Muhammad Faisal Zafar is doctoral candidate in the Faculty of Computer Science nd Information System at the Universiti Teknologi Malaysia (phone: +60 7 5532339 Fax: +607 5565044; e-mail: faisal@ gmm.fsksm.utm.my).

Dzulkifli Mohamad is Associate Professor at Faculty of Computer Science nd Information System at the Universiti Teknologi Malaysia (e-mail: dzul@fsksm.utm.my).

Razib M. Othman He is currently working toward the Ph.D. at the Software Engineering Department, Universiti Teknologi Malaysia (e-mail: razib@fsksm.utm.my).

There is extensive work in the field of handwriting recognition, and a number of reviews exist. General methodologies in pattern recognition and image analysis are presented in [25]. Character recognition is reviewed in [1, 6, 11, 19, 32] for off-line recognition, and in [28, 29] for on-line recognition. Most of the researchers have chosen numeric characters for their experiment [2, 3, 10, 12, 15, 16]. So, some maturity can be observed for isolated digit recognition. However, when we talk about the recognition of alphabetic characters, the problem becomes more complicated. The most obvious difference is the number of classes that can be up to 52, depending if uppercase (A–Z) and lowercase (a–z) characters are distinguished from each other. Consequently, there is a larger number of ambiguous alphabetic characters other than numerals. Character recognition is further complicated by other differences such as multiple patterns to represent a single character, cursive representation of letters, and the number of disconnected and multi–stroke characters [18]. Few researches have addressed this complicated subject. In fact, it can be said that character recognition still an open problem [6].

Neural Nets (NN) and Hidden Markov Models (HMM) are the popular, amongst the techniques which have been investigated for handwriting recognition. It has been observed that NNs in general obtained best results than HMMs, when a similar feature set is applied [17]. The most widely studied and used neural network is the Multi-Layer Perceptron (MLP) [5]. Such an architecture trained with back-propagation [20] is among the most popular and versatile forms of neural network classifiers and is also among the most frequently used traditional classifiers for handwriting recognition. See [37] for a review. Other architectures include Convolutional Network (CN) [21], Self-Organized Maps (SOM)[38], Radial Basis Function (RBF) [5], Space Displacement Neural Network (SDNN) [26], Time Delay Neural Network (TDNN)[22], Quantum Neural Network (QNN) [39], and Hopfield Neural Network (HNN) [23].

Few attempts have been found in the literature in which counter-propagation (CPN) architecture has been used for the recognition of handwritten characters. Ahmed et al., [2] made an attempt but only for digit recognition. The main objective of this work is the implementation of the CPN for the recognition of online upper case English alphabets and to evaluate its performance. Although this study deals with a limited number of 26 upper case character classes, there is a space to extend this work for all alphanumeric characters including lower case characters. This work also suggests a simple approach to extract the useful character information. The performance CPN architecture is evaluated based on extracted character information by applying this simple feature extraction approach.

The recognition system applies a global decision module which decides either to accept the recognition result or reject it. In classification step, a pattern is considered ambiguous if it cannot be reliably assigned to a class, whereas a pattern assigned low confidence for all hypothesized classes can be treated as an outlier. Three different criteria (thresholds) of decision making have been applied.

This paper is organized as follows. The Section 2 gives an overview of the proposed system. Section 3 describes the feature extraction steps from the handwritten character. Section 4 introduces CPN architecture. In Section 5 the experimental results are provided with some analyses and discussions. Section 6 presents the concluding remarks and future work.

II. System Overview

This section describes the simple technique involved in our proposed online handwriting recognition system. This is a writer-independent system based on the neural net method. Conventionally, the data obtained needs a lot of preprocessing including filtering, smoothing, slant removing and size normalization before recognition process. Instead of doing such lengthy preprocessing, we present a simple approach to extract the useful character information. The whole process requires no preprocessing and size normalization. The method is applicable for off-line character recognition as well. A block diagram of the proposed online recognition system of isolated roman characters is shown in Fig 1. The flow of data during training is shown by the dashed line arrows, while the data flow during recognition is shown by solid line arrows. The input to the system is a sequence of handwritten character patterns. After receiving input from tablet the extreme coordinates i.e. left, right, top, and bottom are calculated. Then character is captured in a grid as shown in Fig. 3 and sensing the character pixels in grid boxes, the character is digitized in a binary string. This binary string is applied at the input of CPN for training and recognition. Grid size of 14x8(i.e. 14 rows and 8 columns) were used in the experiments.

A. Data Acqusition

Tablet SummaSketch III has been used to take the samples from different subjects. Upper case alphabets characters have been used. Each subject was asked to write on tablet board (writing area). No restriction was imposed on the content or style of writing; the only exception was the stipulation on the isolation of characters. The writers consisted of university students (from different countries), professors, and employees in private companies. The simulation of each written character could be seen on computer screen as white digital ink with black background. Thus one can make use of black and white colours for some useful processing of written characters.

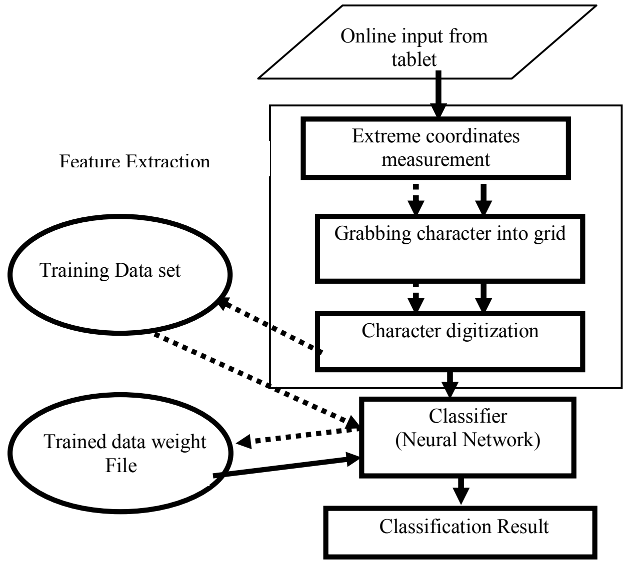

Fig. 1 Block diagram of the system.

B. Character Detection

As stated earlier that character is written with white digital ink, so the algorithm for character detection is quite simple. It searches from left to right for white pixels starting from left-top corner of the area specified for writing. A trace of a white pixel is the indication of the presence of a character.

C. Calculating the Number of Rows

The algorithm searches for the presence and absence of white pixels going from top to bottom. The continuous absence of white pixels (or presence of black pixels) could be a gap between two rows. To make sure whether it is a gap, algorithm searches from left to right against every black pixel, if there is no trace of white pixel for the entire row, the gap is confirmed. In this way, all the horizontal gaps, in the image, are traced out and from this, number of rows are calculated. Fig. 2 shows a sequential algorithm for finding the number of rows in the document.

i. *Find the text boundary of the whole image by scanning from top to bottom for upper border Y1, left to right for left border X1, right to left for right borderX2 and bottom to top for lower borderY2.*

ii. *Scan the binary image from Y1 towards Y2,*

iii. *If there is a black pixel, then scan from X1 to X2 for that particular row to detect any white pixel.*

iv. *If no white pixel found, there is a row gap.*

v. *Repeat steps ii - iii for the whole image to find total number of row gaps.*

Fig. 2 Sequential algorithm for finding the number of rows

D. Character Boundary Calculation

After detecting the character, the next step is to calculate its boundary. The algorithm checks from left to right and top to bottom for left, top, right and bottom boundaries of the character. While going from left to right, the first white pixel is the left boundary and last white pixel is the right boundary of the character. Similarly from top to bottom, first white pixel is the top boundary and last white pixel is the bottom boundary of the character. If there is a vertical gap between two portion of a same character, e.g., ‘ H’ then the algorithm also check from top to bottom for that particular area. The presence of white pixel will eliminate the doubt of a true gap. Similar checks are employed for horizontal gaps between two portion of a same character like ‘S’ and ‘F’. In this way boundaries of the characters in a row are calculated and stored in a data base. After calculating the total number of characters in a row, the individual width and height of each character is measured.

III. Feature Extraction

In the proposed online handwriting recognition system, feature extraction consists of three steps: extreme coordinates measurement, grabbing character into grid, and character digitization. The handwritten character is captured by its extreme coordinates from left /right and top/bottom and is subdivided into a rectangular grid of specific rows and columns. The algorithm automatically adjusts the size of grid and its constituents according to the dimensions of the character. Then it searches the presence of character pixels in every box of the grid. The boxes found with character pixels are considered “on” and the rest are marked “off”. A binary string of each character is formed locating the “on” and “off” boxes (named as character digitization) and presented to the neural network input for training and recognition purposes. The total number of grid boxes represented the number of binary inputs. A 14x8 grid thus resulted in 112 inputs to the recognition model. An equivalent statement would be that a 14x8 grid provided a 112 dimensional input feature vector. The developed software contains a display of this phenomenon by filling up the intersected squares. The effect has been produced in Fig. 3.

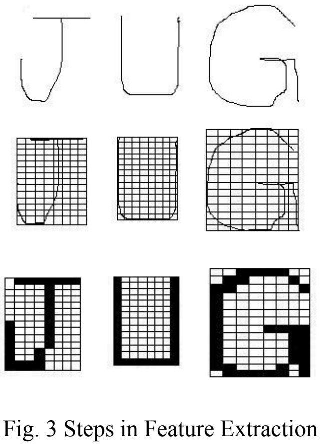

Fig. 3 Steps in Feature Extraction

IV. Neural Net Approach

Neural network classifiers exhibit powerful discriminative properties and they have been used in handwriting recognition particularly with digits, isolated characters, and words in small vocabularies(Alessandro et al., 2002). In this research work, Counterpropagation Neural network (CPN) have been implemented and its performances has been evaluated. This section presents a brief introduction of the architecture.

A. Counterpropagation Neural Network

Hecht-Nielsen [13] proposed CPN as an alternate function approximator which can be developed on the available input-output data. The first counter-propagation network consisted of a bi-directional mapping between the input and output layers. In essence, while data is presented to the input layer to generate a classification pattern on the output layer, the output layer in turn would accept an additional input vector and generate an output classification on the network's input layer. The network got its name from this counter-posing flow of information through its structure. Most developers use a uni-flow variant of this formal representation of counter-propagation. In other words, there is only one feed-forward path from input layer to output layer. The forward-only counterpropagation network architecture, consists of three slabs: an input layer (layer 1) containing n fan out units that multiplex the input signals $x_1, x_2, \ldots, x_n$, (and m units that supply the correct output signal values $y_1, y_2, \ldots, y_m$ to the output layer), a middle layer (layer 2 or Kohonen layer) with N processing elements that have output signals $z_I, z_2, \ldots, z_N$, and a final layer (layer 3) within processing elements having output signals $y_1', y_2', \ldots, y_m'$. The outputs of layer 3 represent approximations to the components $y_1, y_2, \ldots, y_m$ of y = f(x) [14].

The underlying principle for CPN is simple: for a given independent variable vector **I** not present in the available data set, find the independent variable vector in the data set closest to **I**. The criterion of closeness in the n-dimensional Euclidean space can either be distance based (minimum Euclidean distance) or can be angle based (minimum angle between vectors of normalized lengths). If $\mathbf{X_k}$ is the vector found closest in the data set , then the value of $f(\mathbf{I})$ can be approximated as the dependent variable value corresponding to $\mathbf{X_k}$. This technique runs into problems when the data set becomes very large [8]. During training, these "correct" values are supplied to the units of the final layer from, the input units of layer 1.

During training the network is exposed to examples of the mapping f. After each $\mathbf{x}_k$ is selected, $\mathbf{y}_k = f(\mathbf{x}_k)$ is determined and both $\mathbf{x}_k$ and $\mathbf{y}_k$ are input to the network. An important component of training in the CPN is reduction of the data set into a respective data set of lesser, specified size. This is achieved and the estimates of the dependent variable values corresponding to the new (and reduced in number) independent variable vectors can also be calculated [14]. 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 [2].

For the network to operate properly, the input vector must be normalized. This means that for every combination of input values, the total "length" of the input vector must add up to one. Normalization of the inputs is necessary to ensure that the Kohonen layer finds the correct class for the problem. Without normalization, larger input vectors, bias many of the Kohonen processing elements such that weaker value input sets cannot be properly classified. Because of the competitive nature of the this layer, the larger value input vectors overpower the smaller vectors [36]. A three layered CPN implements the principles discussed above. For a CPN in its final form, each PE in the hidden layer represents an independent variable entry in the (reduced) lookup table; the weights to one such PE in the hidden layer from all the PEs in the input layer represent the components of the corresponding independent variable vector. The hidden layer PE whose incoming weights are closest to an input vector "wins" the competition and provides an output value of +1; all other hidden layer PEs supply zero outputs. The weights from the hidden layer PEs to the output layer PEs represent the dependent variable values. A CPN with one linear PE in the output layer thus behaves as estimating one function. With multiple PEs in the output layer, the CPN becomes an estimator of more than one functions.

V. EXPERIMENTS

A. Data Set and Model Parameters

The data used in this work was collected using tablet SummaSketch III . It has an electric pen with sensing writing board. An interface was developed to get the data from tablet. Anoop and Jain [4] pointed out that the actual device for data collection is not important as long as it can generate a temporal sequence of x and y positions of the pen tip. However, the writing styles of people may vary considerably on different writing surfaces and the script classifier may require training on different surfaces.

Upper case English alphabets were considered in case study. In the data set, the total number of handwritten characters is about 2000 characters, collected from 40 subjects. Experiments were examined with grid size of 14x8. Every developed model was tested on characters drawn by individuals who did not participate in the sample collection for data set. Each subject was asked to write on tablet board (writing area). No restriction was imposed on the content or style of writing; the only exception was the stipulation on the shape of 'I'. The grid based character digitization proved improper for characters with negligible width. The shape for handwritten 'I' was thus standardized with horizontal lines at the top and the base. The writers consisted of university students (from different countries), professors, and employees in private companies.

B. Training

In the CPN model, the look-up table grows with increase in training samples. Instead of using Kohonen's learning algorithm for reducing the size of the look-up table, a much simpler technique was employed. Since there were k samples for a character in a particular model, why not reduce the k vectors to one vector by taking the average of the sample vectors? Each component of the resultant averaged vector was average of the corresponding components of the k vectors. This somewhat simplistic approach is mentioned by Freeman & Skapura in their discussion of the CPN ([9], pp 238-258). This technique is intuitively attractive if the k vectors lie close to one another in the n-dimentional Euclidean space (where n = no. of extracted image features). The under laying assumption would be that the clusters of input vector samples corresponding to different characters do not overlap. The performance of such models discussed in the next section indicates that the above assumption was reasonable.

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 both models with gradually increasing the number of samples/character.

C. Recognition Performance

As mentioned earlier, models were evaluated on samples taken from individuals who did not participate in the initial process of setting up 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.

For developed ***CPN model***, closeness was evaluated by measuring the angel between the normalized input and weight vectors. If $\mathbf{I}$ is the normalized input vector and $\mathbf{W_i}$ is the normalized weight vector from the input layer to the i^{th} hidden layer PE, then the cosine of the angle between the two can be found by evaluating the dot product. ($\mathbf{W_i} \,.\, \mathbf{I} = |\,\mathbf{W_i}\,|\,|\,\mathbf{I}\,|\, \mathrm{Cos}\, \theta_i = \mathrm{Cos}\, \theta_i$) [9]. All the angles between each of the feature vectors of the unknown character and their closest corresponding feature vectors in the reference character are summed and missing or extra feature points are penalized. Identification is then a matter of finding the character in the look up table that is within a certain threshold angle of the unknown character. Table 1 present the statistics for CPN. **CRs**, **FRs**, and **RFs** are abbreviation for Correct Recognitions, False Recognitions, and Recognition Failures respectively.

TABLE I
PERFORMANCE OF CPN MODELS WITH THREE DIFFERENT CRITERIA OF CLASSIFICATION

Samples/ Character	**'Threshold': NONE**			**'Threshold': 0.5**			**'Threshold': 0.75**		
	CRs	***FRs***	***RFs***	***CRs***	***FRs***	***RFs***	***CRs***	***FRs***	***RFs***
5 Each	80%	20%	0%	60%	40%	0%	70%	7%	23%
11 Each	83%	17%	0%	79%	21%	0%	72%	6%	22%
22 Each	88%	12%	0%	76%	23%	1%	80%	6%	14%
33 Each	92%	8%	0%	84%	15%	1%	83%	4%	13%
44 Each	93%	7%	0%	82%	17%	1%	76%	8%	16%
55 Each	87%	13%	0%	88%	8%	4%	86%	3%	11%
66 Each	94%	6%	0%	93%	6%	1%	92%	1%	7%

D. *Performance Analysis*

For developed CPN models, there was no need of training parameters nor it is an iterative method like back-propagation architecture which took a long time for learning. A general trend of increase in performance with increase in samples/character has also been observed in this case. The difference in recognition rates with and without a threshold for input classification is understandable (Table I). Though threshold reduces the correct recognitions but at the same time it prevents the system to go for more false recognitions. False Recognition (FRs) is another important factor in any recognition system, lower the false recognition rate, more reliable the system [7, 35]. Instead of FRs, system goes for recognition failure (RFs) which is less dangerous than FRs. On the other hand, performance of the system increases without threshold but at cost of more FRs. Fig. 4 presents a graphical overview of CPN performances with three different decision criteria of Recognition.

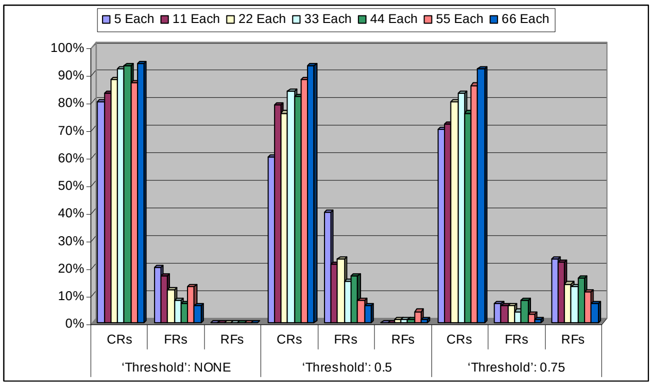

Fig 7 Graphical Presentation of CPN performances with three different criteria of Recognition

VI. Conclusion

From the results it can be concluded that CPN is a good promise in terms of recognition capability which has not been put on trial in the field of handwriting recognition.. More over CPN is more economical than convergence of other NN architectures e.g. back-propagation where the training time can take long time. The experiments provided the authors an opportunity to explore this pattern recognition methodology; the exercise provided a theoretical base for further investigations and impetus for development work in this discipline. The obtained results motivate the continuity of the system development considering a preprocessing mechanism including normalization and slant removal. Other future work might involve some new feature extraction approaches.

References

[1] Alessandro Vinciarelli (2002). A survey on off-line Cursive Word Recognition. Pattern Recognition, 35: 1433.

[2] Ahmed, S.M.,et.al, (Nov.1995). Experiments in character recognition to develop tools for an optical character recognition system, IEEE Inc. 1st National Multi Topic Conf. proc. NUST, Rawalpindi, Pakistan, 61-67.

[3] Alexandre Lemieux, Christian Gagne and Marc Parizeau. (2002). Genetical Engineering of Handwriting Representations. Proc. of the International Workshop on Frontiers in Handwriting Recognition (IWFHR), Niagara-On-The-Lake, August 6-8.

[4] Anoop M. Namboodiri, Anil K. Jain. (2004). Online Handwritten Script Recognition. IEEE Trans. PAMI. 26(1): 124-130.

[5] Bishop M. (1995). Neural Networks for Pattern Recognition. Oxford Univ. Press, Oxford-U.K.

[6] Bortolozzi, F., Britto Jr,. A., Oliveira, L. S. and Morita, M., (2005). Recent Advances in Handwriting Recognition. In Umapada Pal et al editors, Document Analysis, 1-31.

[7] Dzulkifli Mohamad, and Zafar, M. F., (2004a). Recognition of Complex Patterns Using a Novel Feature Vector by Backpropagation Neural-Network. Pattern Recognition And Image Analysis. 14(3): 479-487.

[8] Dzulkifli Mohamad, and Zafar, M. F., (2004b). Comparative Study of Two Novel Feature Vectors For Complex Image Matching Using Counterpropagation Neural Network. Journal of Information Technology, vol.16, No.1, FSKSM, UTM.

[9] Freeman, J. A., and Skapura, D. M., (1991). Neural Networks: Algrithms, Applications and Programming Tecniques, Addison-Wesley Publishing Company.

[10] Gader P. D., Forester B., Ganzberger M., A. Billies, B. Mitchell, M. Whalen, T. Youcum (1991). Recognition of handwritten digits using template and model matching. Pattern Recognition, 5(24):421-431.

[11] Govindan, V. K.. and Shivaprasad, A. P. (1990), Character Recognition-a Review, Pattern Recognition, 23: 671-683.

[12] Hebert Jean-Francois, Parizeau Marc and Nadia Ghazali. (1998). A new fuzzy geometric representation for on-line isolated character recognition. Proc. of the 14th International Conference on Pattern Recognition, Brisbane: 1121-1123.

[13] Hecht-Nielsen, R. (1987). Counterpropagation networks. Proc. Of IEEE First Int'l Conference on Neural Networks. 1987 II: 19-32.

[14] Hecht-Nielsen Robert.(1990). Neurocomputing. Addison-Wesley Publishing Company.

[15] Hiroto Mitoma, Seiichi Uchida, and Hiroaki Sakoe. (2005). Online character recognition based on elastic matching and quadratic discrimination. Proceedings of 8th International Conference on Document Analysis and Recognition (ICDAR 2005, Seoul, Korea).1(2):36—40.

[16] Iqbal, A., Zafar, M. Faisal., (1998). Pc Based Optical Objective Test Marking System Using Neural Networks. International Workshop on Recent Advances in Computer Vision proc. SZABIST, Karachi, Pakistan,: 41-50.

[17] Kapp, M.N., Freitas, C. and Sabourin, R. (2004).Handwritten Brazilian Month Recognition: An Analysis of Two NN Architectures and a Rejection Mechanism. 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR-9), Tokyo, Japan,

[18] Koerich L. (2002). Large Vocabulary Off-Line Handwritten Word Recognition. PhD thesis, École de Technologie Supérieure, Montreal-Canada.

[19] Koerich A. L., Sabourin R., Suen C. Y. (2003). Large vocabulary off-line handwriting recognition:A survey. Pattern Anal Applic, 6: 97–121.

[20] LeCun Y., Bottou L., Orr G. B., Muller K. R. (1998a). Eficient backprop. In Orr G. and K. Miller, editors, Neural Networks: Tricks of the Trade. Springer.

[21] LeCun Y., Bottou L., Bengio Y., and Haffner P. (1998b). Gradient-Based Learning Applied to Document Recognition. Proc. IEEE, 86(11): 2278-2324.

[22] Lethelier, Leroux M., and Gilloux M. (1995). An automatic reading system for handwritten numeral amounts on french checks. In Proc. 3th International Conference on Document Analysis and Recognition, Montreal-Canada, pp. pages 92-97.

[23] Ling, M. Lizaraga, N. Gomes, and A. Koerich (1997). A

prototype for brazilian bankcheck recognition. In S.Impedovo et al, editor, International Journal of Pattern Recognition and Artificial Intelligence,World Scientific, pp. 549-569.

[24]Liu Cheng-Lin, Stefan Jaeger, and Masaki Nakagawa (2004). Online Recognition of Chinese Characters:The State-of-the-Art. IEEE Trans. on Pattern Analysis and Machine Intelligence, 26(2), 198-203.

[25]Mantas, J. (1986), An Overview of Character Recognition Methodologies, Pattern Recognition, 19 (1986) 425-430.

[26]Matan O., Burges J. C., LeCun Y., Denker J. S. (1992). Multi-digit recognition using a space displacement neural network. In J. E. Moody, S. J. Hanson, and 165 R. L. Lippmann, editors, Advances in Neural Information Processing Systems, volume 4, Morgan Kaufmann, pp. 488-495.

[27]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.

[28]Nouboud, F., and Plamondon,(1990). On-Line Recognition of Handprinted Chara.cters: Survey and Beta Tests, Pattern Recognition, 23: 1031-1044.

[29]Plamondon Rejean, and Sargur N. Srihari, (2000). On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey. 1EEE Transactions on PAMI. 22(1): 63-84.

[30]Rumelhart, D. E., Hinton, G. E., and Williams, R. J., (1986). Learning internal representations by error propagation. in Rumelhart, D. E. And McClelland, J. L. [Ed], Parallel Distributed Processing: Explorations in the Microstructure of Cognition, 1, 318-362, MIT Press, Cambrige.

[31]Shridhar M. and Badreldin A. (1986). Recognition of isolated and simply connected handwritten numerals. Pattern Recognition, 19(1):1-12.

[32]Suen, C. Y., Berthod, M., and Mori, S. (1980), Automatic Recognition of Ha.nd printed Character-the State of the Art, Proceedings of IEEE. 68 (1980) 469-487.

[33]Steinherz T., Rivlin E., and Intrator N. (1999). Offline Cursive Script Word Recognition—A Survey. Int’l J. Document Analysis and Recognition, vol. 2, 90-110.

[34]Tappert C.C., Suen C.Y., Wakahara T.(1990). The state of the art in on-line handwriting recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence, 12(8), 787-808.

[35]Zafar, M. F., and Dzulkifli Mohamad. (2005). Comparison of Two Different Proposed Feature Vectors For Classification of Complex Image. Jurnal Teknologi, Universiti Teknologi Malaysia, 42(D): 65-82.

[36]Zafar, M. F., and Dzulkifli Mohamad. (2002). A Comparison Of Two Neural Network Techniques On Feature Based Complex Image Recognition. 2nd World Engineering Congress, WEC2002, Kuching, Sarawak, Malaysia.

[37]Zhang G. P. (2000). Neural networks for classification: a survey. IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews, 30(4):451-462.

[38]Zhang, Fu M., Yan H., and Fabri M. A. (1999). Handwritten digit recognition by adaptativesubspace self organizing map. IEEE Trans. on Neural Networks, 10:939-945.

[39]Zhou J (1999). Recognition and Verification of Unconstrained Handwritten Numeral. PhD thesis, Concordia University, Montreal-Canada.

Muhammad Faisal Zafar received the BSc degree in mathematics and physics in 1990 from Govt, College Faisalabad, Pakistan, MSc degree in physics in 1993 from Govt. College Lahore, Pakistan, , the Postgraduate Diploma in computer training in 1995 from Computer Training Centre (CTC) Islamabad, Pakistan. He has been with Informatics Complex (ICCC), Islamabad, Pakistan, since 1995 as a scientific / senior scientific officer, where he worked on pattern recognition tasks. He was offered fellowship from Universiti Teknologi Malaysia (UTM), in 2001 and he completed his MSc degree in computer science in 2003 from UTM. Currently he is a PhD (computer science) student at Faculty of Computer Science and Information System (FSKSM), UTM, Malaysia. His research interests include pattern recognition, neural networks, machine print and handwriting recognition.

Dzulkifli Mohamad received the BSc degree in computer science and statistics from the National University of Malaysia in 1978, the Postgraduate Diploma in computing science from University of Glasgow in 1981, and the MSc and PhD degrees in computer science from the University of Technology, Malaysia in 1991 and 1997. He has been with the University of Technology, Malaysia since 1978, where he is currently an associate professor of computer science. His research interests include image processing, pattern recognition, medical imaging, information theory, and artificial intelligence.

Razib M. Othman received B.Sc. in computer science in 1999 and M.Sc. in computer science in 2003 both from the Universiti Teknologi Malaysia. He is currently working toward the Ph.D. at Universiti Teknologi Malaysia in the field of computational biology. Since 1999 he has been working as a lecturer at the Software Engineering Depart-ment, Universiti Teknologi Malaysia and researcher at the Artificial Intelligence and Bioinformatics Laboratory in the same university. His main research interests are genetic algorithms, software agents, and semantic web.