Neural Nets for On-line Isolated Handwritten Character Recognition: A Comparative Study

Muhammad Faisal Zafar
Faculty of Computer Science (FSKSM)
Universiti Teknologi Malaysia. 81310
faisal@gmm.fsksm.utm.my

Dzulkifli Mohamad
Faculty of Computer Science (FSKSM)
Universiti Teknologi Malaysia. 81310
dzul@fsksm.utm.my

Razib Othman
Faculty of Computer Science (FSKSM)
Universiti Teknologi Malaysia. 81310
razib@fsksm.utm.my

Abstract- **Handwriting processing is a domain in great expansion which in the present day begins to see several industrial realizations. 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. Handwriting recognition has always been a tough problem because of the handwriting variability, ambiguity and illegibility. This paper describes a simple approach involved in online handwriting recognition. 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, this paper presents 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. This is a writer-independent system based on two neural net (NN) techniques: back propagation neural network (BPN) and counter propagation neural network (CPN). Performances of BPN and CPN are 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, back propagation neural network, extreme coordinates

I. INTRODUCTION

Machine simulation of human writing is one of the most challenging research areas. It has been the subject of intensive research especially in unconstrained handwriting recognition. 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 [1]. Two classes of recognition systems are usually distinguished: online systems 2, 3, 4] for which handwriting data are captured during the writing process, which makes available the information on the ordering of the strokes, and offline systems [5] for which recognition takes place on a static image captured once the writing process is over. Current focus of the market today is on-line handwriting recognition. With the increase in popularity of portable computing devices such as PDAs and handheld computers [6, 7], non-keyboard based methods for data entry are receiving more attention in the research communities and commercial sector. Large number of symbols in some natural languages (e.g., Kanji contains 4,000 commonly used characters) making keyboard entry even a more difficult task [8]. The most promising options are pen-based and voice-based inputs. Digitizing devices like [9] and computing platforms such as the IBM Thinkpad TransNote [10] and Tablet PCs [11], have a pen-based user interface. Such devices, which generate handwritten documents with online or dynamic (temporal) information, require efficient algorithms for processing and retrieving handwritten data [3].

Handwriting recognition has always been a tough problem [12]. Recognition of handwritten characters by computer poses serious problems because of the high variability in the character shapes written by individuals [13]. As people tend to adjust their handwriting style to personal preferences, the resulting variability of handwriting styles often makes reading difficult even for humans. This problem becomes even more complicated when the writer is unknown [12]. Moreover, pairs of characters can be formed which are ambiguous, both for human and machine recognition, for instance U-V, C-G,Q-G, D-O, F-P.

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 Mantas [14]. Character recognition is reviewed in [15, 16, 17, 18, 19] for off-line recognition, and in [20, 21] for on-line recognition.

Numerous techniques for handwriting recognition have been investigated based on four general approaches of pattern recognition, as suggested by [22]: template matching, statistical techniques, structural techniques, and neural networks. Template matching operations determine the degree of similarity between two vectors (groups of pixels, shapes, curvatures, etc) in the feature space. Matching techniques can be grouped into three classes: direct matching [23], deformable templates and elastic matching [24], and relaxation matching [25, 26]. Statistical techniques are concerned with statistical decision functions and a set of optimal criteria, which determine the probability of the observed pattern belonging to a certain class. The statistical scheme is receiving increasing attention in recent years [4]. Statistical techniques use concepts from statistical decision theory to establish decision boundaries between pattern classes [22, 27]. In structural techniques the characters are represented as unions of structural primitives. It is assumed that the character primitives extracted from handwriting are quantifiable, and one can find the relationship among them. Basically, structural methods can be categorized into two classes: grammatical methods [28] and graphical

1-4244-0457-6/06/$20.00 ©2006 IEEE

methods [29]. A Neural Network (NN) is defined as a computing structure consisting of a massively parallel interconnection of adaptive “neural” processors. The main advantages of neural networks lies in the ability to be trained automatically from examples, good performance with noisy data, possible parallel implementation, and efficient tools for learning large databases.

II. SYSTEM OVERVIEW

This paper describes the simple technique involved in online handwriting recognition. This is a writer-independent system based on the statistical 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, this paper presents a simple approach to extract the useful character information. A block diagram of the proposed online recognition system of isolated roman characters is shown in Fig 1.

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. 2, and after sensing the character pixels in grid boxes, the character is digitized in a binary string. This binary string is applied at the input of a counter propagation neural network for training and recognition. Grid size of 14x8 (i.e. 14 rows and 8 columns) were used in the experiments.

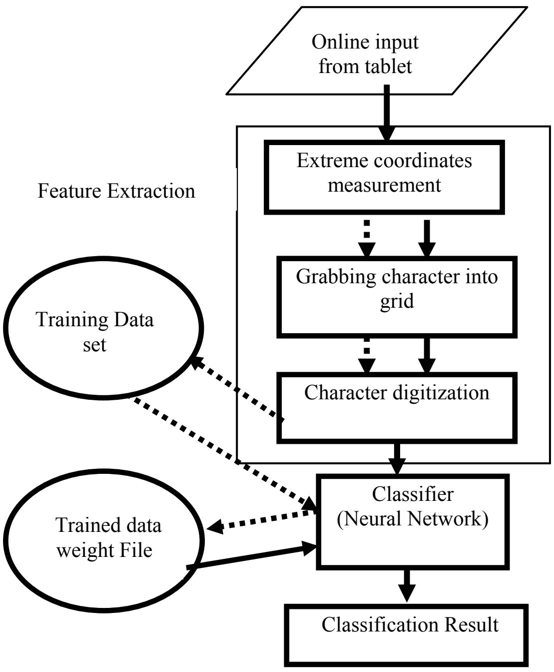

Fig. 1. Block diagram of the system. The flow of data during training is shown by the dashed line arrows, while the data flow during recognition is shown by solid line arrows.

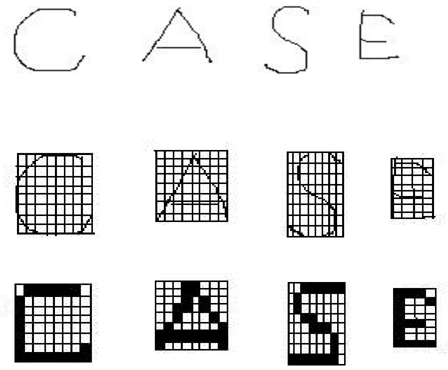

Fig 2- Steps in Feature Extraction

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 2.

IV. 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 A. K. Jain [3] 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. Learning / Training

For classification purpose, two neural networks techniques: back propagation neural networks (BPN) and counter propagation neural networks (CPN) have been used.

In the BPN, sigmoid PEs were used in the hidden and the output layer. Twenty-six output layer processing elements (PEs) corresponded to Twenty-six English alphabets to be recognized. For example, a 'high' output value on the second PE in the output layer and 'low' on the others would mean that the network classifies the input as a 'B'. As another example, a network output vector of [0.00 0.01 0.16 0.02 0.09 0.96 0.13 0.00 --- 0.15 0.04] would be translated as the model classifying the input character as 'F'.

The GDR was preferred over the exact steepest descent algorithm because of the large difference in convergence time between the two. The authors experimented with the number of hidden layer PEs in search of better convergence behaviour. The selection of *learning rate* and *momentum coefficient* (proportionality constant for sliding in the direction of negative gradient) was also more of an art than a science; their values were kept between 0 and 1 with typically a gradual decrease in magnitude as training progressed.

Experiments were started with a strict convergence criterion: training was stopped only when the network classified all the training samples correctly. While checking the network's performance during training, an output layer PE's output value of ≥ 0.9 was translated as 'high'. Thus, for this criterion, an output vector [0.00 0.03 0.06 0.12 0.09 0.16 0.03 0.91 0.17 --- 0.15 0.14] for character 'H' in the training sample set would be termed as proper classification; a 0.85 instead of 0.91 in the output vector would render the training input as not properly recognizable till that stage in the training process.

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 ([30], 238-258). This technique is intuitively attractive if the k vectors lie close to one another in the n-dimensional 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. Table 1 shows the summary of different parameter's values used for BPN during training. Training was stopped, with 1 sample (out of 286) and 5 samples (out of 572), 116 samples (out of 858), 295 samples (out of 1144), 429 samples (out of 1430) and 345 samples (out of 1716), remained unclassified after 15316, 51000,7129980, 880880, 1801800 and 3517800 training presentations for 11 samples/character model, 22 samples/character model, 33 samples/character model. 44 samples/character model, 55 samples/character model and 66 samples/character model respectively (see Table 1).

TABLE 1

DETAILS OF DIFFERENT PARAMETER VALUES USED FOR BPN DURING TRAINING PHASE

Samples / Character	Total no. of characters	Iterations	Sum of Squared Error (SSE)	Learning rate	Momentum Parameter	Hidden Elements	# of untrained Characters
5 Each	130	8680	0.145210	0.9	0.5	30	0
11 Each	286	15316	0.166071	0.9	0.5	30	1 (0.3%)
22 Each	572	51000	2.515676	2.0	0.3	15	5 (0.8%)
33 Each	858	7129980	58.005	0.999	0.5	25	116 (13%)
44 Each	1144	880880	147.54	1.5	0.9	30	295 (26%)
55 Each	1430	1801800	214.54	2.0	0.5	35	429 (30%)
66 Each	1716	3517800	172.54	0.9999	0.5555	40	345 (20%)

V. 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 models in practical online recognition systems. The quality of an online handwriting recognizer is related to its ability to translate drawn characters irrespective of writing styles.

For the developed ***BPN** model*, the debate on a valid high PE output in the output layer was resolved by evaluating the performance for different decision making criteria. Model was tested with high thresholds of 0.9 and 0.5, using the PE with the highest value above the threshold for input classification. Another criterion used in translating the BPN model's outputs was to eliminate the concept of threshold and simply use the highest value. Note that the first criteria will always have the possibility of a **recognition failure**: a network decision of not attributing any logo to the input image. The last criteria will eliminate this somewhat desirable feature in the decision making process.

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. ($\mathbf{W_i} \cdot \mathbf{I} = |\mathbf{W_i}|\,|\mathbf{I}| \cos\theta_i = \cos\theta_i$) [30]. 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 2 and Table 3 present the statistics for BPN and CPN respectively. **CRs**, **FRs**, and **RFs** are abbreviation for Correct Recognitions, False Recognitions, and Recognition Failures respectively.

TABLE 2

PERFORMANCE OF BPN MODELS WITH THREE DIFFERENT CRITERIA OF CLASSIFICATION

Samples/ Character	'Threshold': NONE			'Threshold': 0.5			'Threshold': 0.9		
	CRs	***FRs***	***RFs***	***CRs***	***FRs***	***RFs***	***CRs***	***FRs***	***RFs***
5 Each	70%	30%	0%	65%	10%	25%	60%	10%	30%
11 Each	73%	27%	0%	71%	5%	24%	65%	5%	30%
22 Each	75%	25%	0%	81%	4%	15%	83%	6%	11%
33 Each	77%	23%	0%	71%	7%	22%	67%	2%	31%
44 Each	65%	35%	0%	60%	4%	36%	51%	6%	43%
55 Each	71%	29%	0%	62%	4%	34%	53%	4%	43%
66 Each	83%	17%	0%	79%	6%	15%	79%	2%	19%

TABLE 3

PERFORMANCE OF CPN MODELS WITH THREE DIFFERENT CRITERIA OF CLASSIFICATION

Samples/ Character	'Threshold': NONE			'Threshold': 0.5			'Threshold': 0.9		
	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%

VI. PERFORMANCE ANALYSIS

For developed BPN models, it was observed that learning became more difficult and, even after long time of training, models were unable to fully learn the training sets. The gradual increase in the percentage of untrained samples can be seen in Table 1. An increase of hidden PEs was also helpful towards the convergence when sample/character were increased. Sometimes, initially a big value of learning rate showed a rapid learning even with less number of hidden PEs but most of the time a value < 1 appeared suitable for learning rate. Generally, the recognition performance of BPN models improved with increase in samples/character.

It is important to note that the sigmoid functions in the output layer behave as 'smoothed' bipolar switches; the inputs to these bipolar switches are values of the decision functions. These decision functions or decision surfaces have positive value for a PE's output greater than 0.5 and negative values for PE outputs of less than 0.5. The evaluation of weights during training can be thought of as development of such decision functions. Poor performance of a trained neural network may imply improper decision functions which are good enough for the training samples but not appropriate for other inputs. The recognition rate without any threshold (NONE) was highest (up to 83%) but at the cost of more false recognition. This recognition rate gradually decreases by applying tough thresholds (0.5 and 0.9) but this makes the system more reliable by tempting less false recognitions. However, overall the false recognitions were much less than recognition failure (RFs), after applying thresholds, which is a plus point. More RFs are due to a large number of untrained samples. This number can be reduced by experimenting more suitable combinations of hidden PEs and learning rates. It will ultimately improve the recognition rate.

For developed CPN models, there was no need of training parameters nor it is an iterative method like BPN 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 3). 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 [31]. 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. Figure 3 presents a graphical overview of CPN and BPN performances with three different decision criteria of Recognition. CPN performance results are more attractive than those of BPN. More over CPN is more economical than convergence of BPN where the training time can take long time. On the other hand, it has been observed that BPN has more perfection in recognition than CPN.

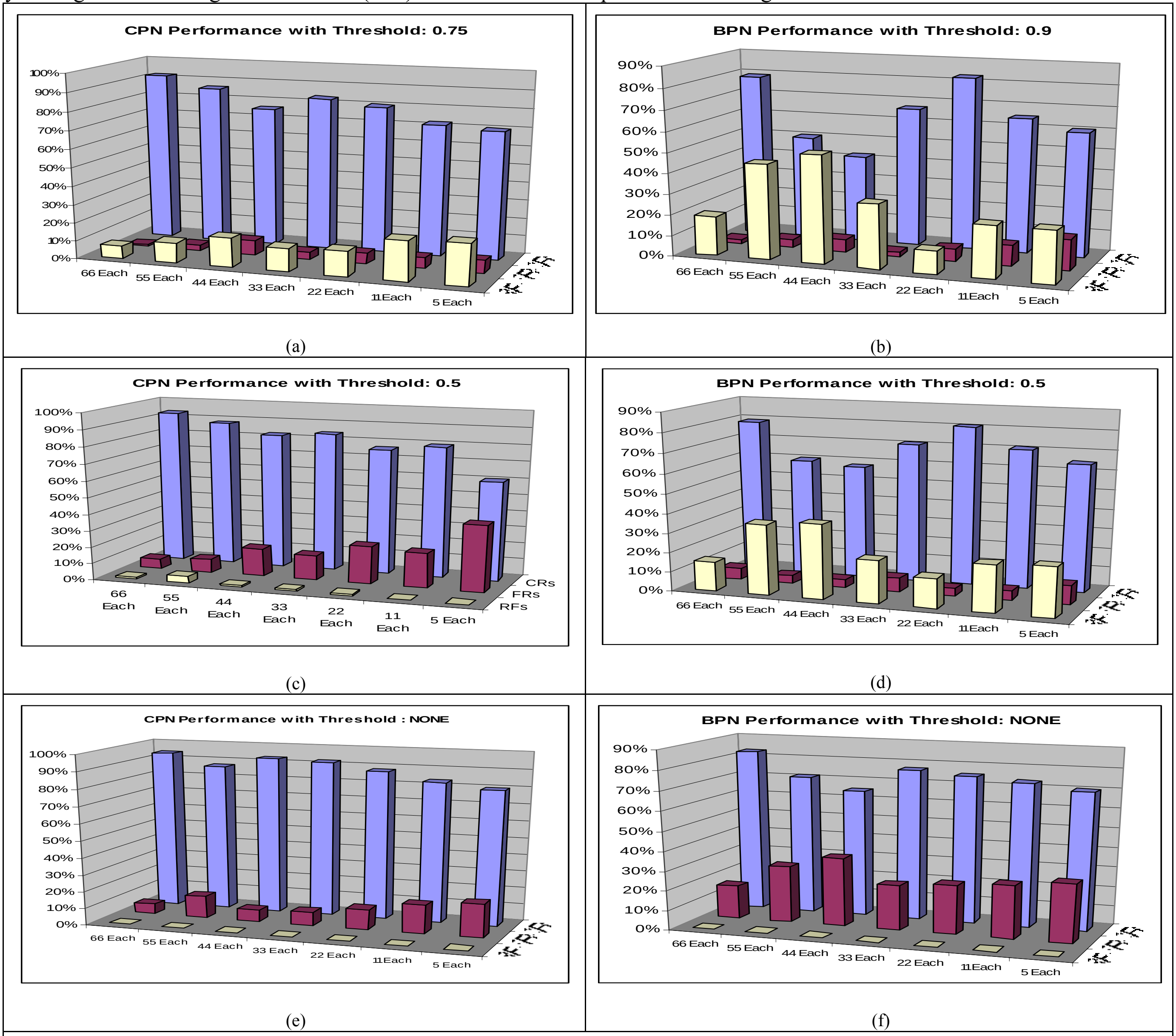

Fig. 3 : (a), (c) and (e) are graphical presentations of BPN performances with three different criteria of Recognition 0.9, 0.5 and 0.0 respectively. (b), (d) and (f) are the corresponding graphical Presentation of BPN performances with criteria of Recognition 0.75, 0.5 and 0.0.

VII. Conclusion

An elementary online handwriting recognition prototype for isolated upper case English characters has been developed using a very simple approach without an application of preprocessing process. The system is writer-independent based on neural network approach. For training and recognition purposed CPN and BPN have been used and recognition rates of 94% and 83% have been achieved respectively. These recognition rates are still worth considering and highly desirable in pattern recognition. The preliminary results are quite encouraging. The experiments provided the authors an opportunity to explore the two pattern recognition methodologies; the exercise provided a theoretical base for further investigations and impetus for development work in this discipline. CPN has shown better performance than that of BPN.

REFRENCES

[1] Morita M., Sabourin R., Bortolozzi F., Suen C. Y. (2003). A Recognition and Verification Strategy for Handwritten Word Recognition. *ICDAR'03)*, Edinburgh, Scotland, August 3-6: 482-486.

[2] 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.

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

[4] Cheng-Lin Liu, 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.

[5] 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.

[6] Evan Koblentz (May 2005). The Evolution Of The PDA: 1975-1995. *Computer Collector Newsletter* , version 0.993.

[7] Pen Computing Magazine: PenWindows, (2005). http://www.pencomputing.com/PenWindows/index.html,

[8] Scott D. Connell. (2000).Online Handwriting Recognition Using Multiple Pattern Class Models. *PhD Thesis, Dept. of Computer Science and Engineering, Michigan State University* USA.

[9] Smart Technologies Inc. Homepage, (2005). http://www.smarttech.com/

[10]IBM ThinkPad TransNote, (2005). http://www-132.ibm.com/content/search/transnote.html

[11]Windows XP Tablet PC Edition Homepage, (2005). http://www.microsoft.com/windowsxp/tabletpc/default.asp

[12]Robert Kazimierz Powalka, "*An algorithm toolbox for on-line cursive script recognition*", PhD thesis The Nottingham Trent University May 1995.

[13]G. Sagar and S. Barman, 2001, **"**Isolated Handwritten Character Recognition" proc. of Intelligent Processing and Manufacturing of Materials I P M M – 2001**,** 3rd International Conference July 29 - August 3, 2001Vancouver, British Columbia, Canada.

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

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

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

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

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

[19]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.

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

[21]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.

[22]Jain K, Duin R. P. W., Mao J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4-37.

[23]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.

[24]Dimauro G., Impedovo S., Pirlo G., Salzo A. (1997). Automatic bankcheck processing: A new engineered system. In S. Impedovo et al, editor, *International Journal of Pattern Recognition and Artificial Intelligence*, World Scientific, pp. 467-503.

[25]Xie S. L., and Suk M. (1988). On machine recognition of hand-printed Chinese character by feature relaxation. *Pattern Recognition*, 21(1):1-7.

[26]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.

[27]Vuurpijl L, and Schomaker L. (2000). Two-stage character classification: A combined approach of clustering and support vector classifiers. *Proc 7th International Workshop on Frontiers in Handwriting Recognition*, Amsterdam, Netherlands, 423–432.

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

[29]Kim H. Y. and Kim J. h. (1998). Handwritten Korean character recognition based on hierarchical randon graph modeling. In *Proc. 6th International Workshop on Frontiers of Handwriting Recognition*, Taegon-Korea, August, pp. 557-586.

[30]Freeman, J. A., and Skapura, D. M., (1991). Neural Networks: Algorithms, Applications and Programming Techniques, *Addison-Wesley* Publishing Company.

[31]Dzulkifli Mohamad, 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, Universiti Teknologi Malaysia,16(1).