International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:4, No:7, 2010

Recognition of Isolated Handwritten Latin Characters using One Continuous Route of Freeman Chain Code Representation and Feedforward Neural Network Classifier

Dewi Nasien, Siti S. Yuhaniz, Habibollah Haron

***Abstract*—**In a handwriting recognition problem, characters can be represented using chain codes. The main problem in representing characters using chain code is optimizing the length of the chain code. This paper proposes to use randomized algorithm to minimize the length of Freeman Chain Codes (FCC) generated from isolated handwritten characters. Feedforward neural network is used in the classification stage to recognize the image characters. Our test results show that by applying the proposed model, we reached a relatively high accuracy for the problem of isolated handwritten when tested on NIST database.

***Keywords*—**Handwriting Recognition, Freeman Chain Code and Feedforward Backpropagation Neural Networks.

I. Introduction

CHARACTER recognition is the subset of handwriting recognition. Character can be in the format printed or handwritten. Recognition of handwritten characters by computer is a serious problem. The problem and difficulties of handwriting recognition task can be classified into four categories which are nature of the handwriting signals, handwriting styles, writer dependency and vocabulary sizes [1].

A. Nature of the Handwriting Signals

Handwritten can be defined into two components, namely off-line and on-line recognition. Off-line recognition is a system that accepts it is input from a digital scanner using image processing algorithm. On the other hand, on-line recognition system that accepts input data from on-line input devices and then [2] computed the relationships between points to extract the features in real time.

B. Handwriting Styles

According to [1] there are five main types of handwriting styles as shown in the Fig. 1 such as:

1) *Boxed discrete characters*
   In form processing usually user provided a form and requested to write each isolated letter in a box rectangles. For type "boxed discrete characters", do not need segmentation because already isolated character. Handwritting character recognition (HCR) that to play in character extraction from box rectangles and finally recognized.
2) *Spaced discrete characters*
   The "spaced discrete characters" where the position of characters separately written. Each character doesn't have to link with it is neighboring characters. Segmentation algorithm is needed to retrieve each isolated characters.
3) *Run-on discretely written characters*
   For this case, each character is written one after another. Consequently, it is to link between neighboring characters. These caused by writing speed.
4) *Pure cursive handwriting*
   In "pure cursive handwriting" each character has to be regularity link its neighboring characters. It is nicely link and diacritic mark which are i, j dot and t, f bar are wrote after main part of word.
5) *Unconstraint handwriting*
   This is most frequently found in our daily life. It is combined of space discrete, run-on discrete and cursive handwriting.

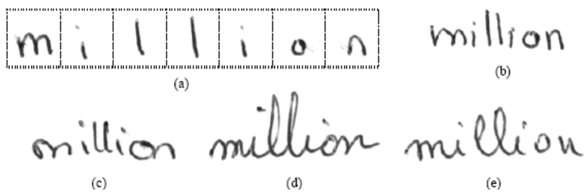

Fig. 1 The differences of handwriting styles: (a) box discrete characters, (b) spaced discrete characters, (c) run-on discretely written characters, (d) pure cursive handwriting and (e) unconstraint handwriting architecture of recognition system [1].

C. Writer Dependency

Handwriting styles are different of every individuals. In recognition of a character is to easy to recognized from single person than many persons. Three kinds of criteria level of difficulties by [1]:

1) *Mono-sciptor*

Dewi Nasien is with Soft Computing Research Group, Department of Modeling and Industrial Computing, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor Bahru, Malaysia (phone: +60-163-095-888; fax: +60-755-650-44; e-mail: dwien82@gmail.com).
Habibollah Haron is with Soft Computing Research Group, Department of Modeling and Industrial Computing, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor Bahru, Malaysia (e-mail: habib@utm.my).

In mono-scriptor, the same writer is involved where the accuracy is higher because similar writer. Handwriting recognition system is trained and tested on the same scriptor.

2) *Multi-scriptor*
For middle problem is multi-scriptor is to deal with the different person of handwriting. The training and testing in this case are come from the same group of scriptors.

3) *Omni-scriptor / Scriptor Independent*
Another highest problem is the omni-scriptor is to deal with handwriting styles that it has never seen during the design or training. As a result, it is difficult by system in training and testing. For trained is to taken from available database from one group of scriptors and similarly for the testing on fresh database from entirely different group of scriptors. Hence, usually the problem for real commercial from handwriting recognition.

C. Vocabulary Size

The differences categories based on size of vocabulary was proposed by Steinherz [3]. They are assumed three categories which are small and specific, limited but dynamic and large for the detail as shown in the Table I.

TABLE I
THE VOCABULARY SIZES DEPEND ON THE CATEGORY

Category	**Vocabulary**	**Application**
Small and specific	<100	Legal amount recognition
Limited but dynamic	<1000	Address recognition
Large	>1000	General applications

HCR is the ability of a computer to receive and interpret intelligible handwritten input then analyzed to many automated process system. Generally, HCR can be divided into three steps namely pre-processing, feature extraction and classification (recognition). This paper will describes the all the steps in HCR.

Heuristic is used on image extraction phase of handwritten character recognition system. Specifically, it proposed randomized algorithm to generate the FCC of handwritten binary image character. The image is represented by binary number of 0s and 1s. This study only considers the 1-number and intends to generate a sequence of chain which connect all of the 1-numbers. As in Fig. 2 shows a binary image character of a "B" handwritten character. FCC is 8-direction neighborhood is used as shown in Fig. 3. It starts from 1 (different from common FCC is uses 0) and move counterclockwise. 0 is not used for distinguish direction or non direction (value is zero) of chain code.

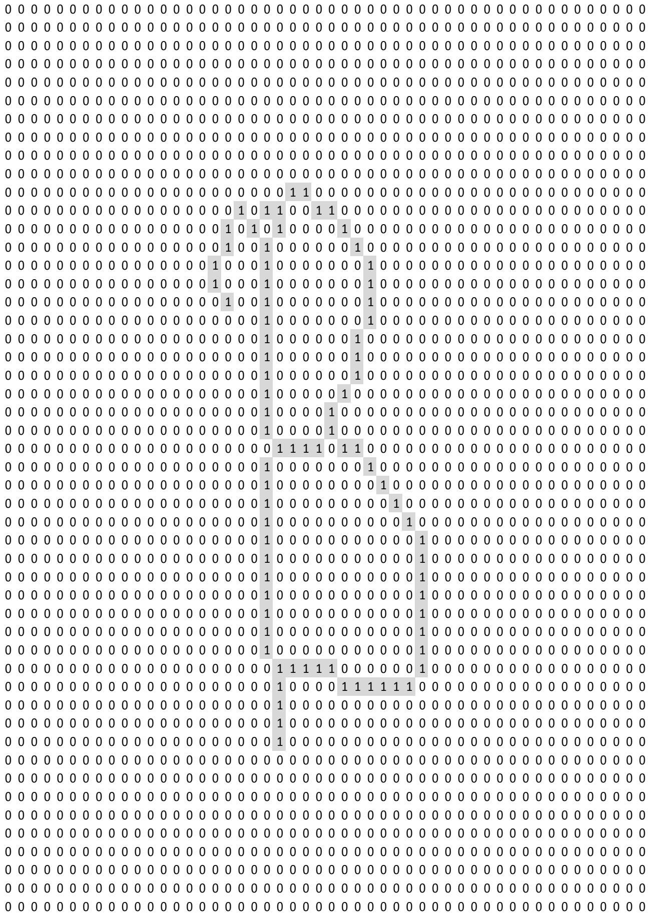

Fig. 2 A binary of "B" handwritten character

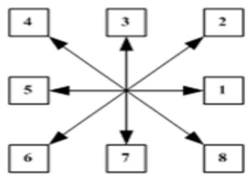

Fig. 3 Freeman chain codes is 8-neighborhood

Basically, the technique in representation during the feature extraction are classified to their processing approaches namely [4]: (1) one dimensional function for shape representation, (2) polygonal approximation, (3) spatial approximation, (4) moments, (5) scale space approaches and (6) shape transform domains as shown in Fig. 4 and Table II shows the properties of shape representation.

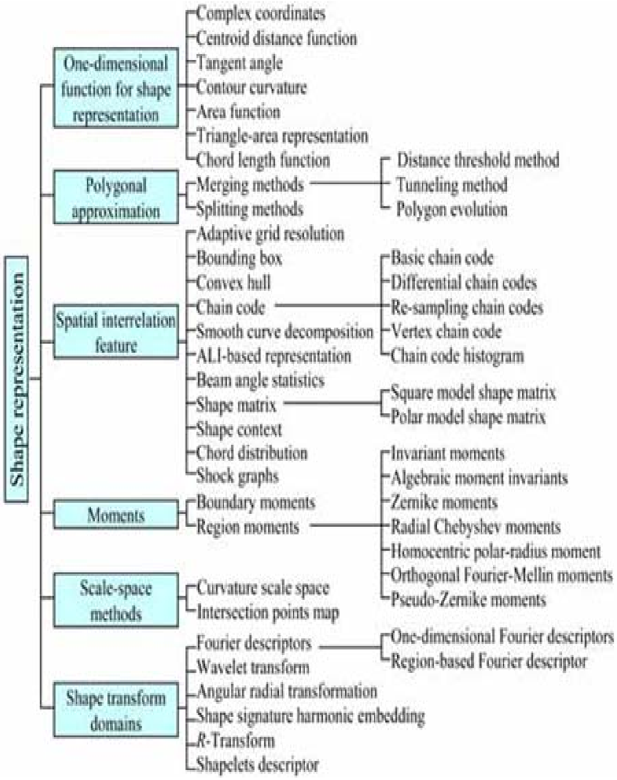

Fig. 4 The overview of shape description [4]

FCC is selected in the representation of a character image. In a handwritten character recognition, it is often to find several branches and this makes difficult to decide where it should go. Moreover, a revisit to the previous visited node is often needed to visit all of the nodes. These difficulties motivate us to use heuristic method. Heuristic methods are excellent in finding approximate solutions but have acceptable time and space complexity. Time complexity refers to running time of the program and space complexity refers to the quantity of computer memory. The characteristics of heuristic [5] are: easy to implement, easy to design, often fast although there is no guaranteed upper bound on the expected run time and often producing good results although there is no guarantee that the solution is optimal or close to optimal. In this paper a heuristic methods are proposed which is randomized algorithm.

This paper is organized as follows. Section 2 presents the methodology. In section 3, problem data and parameter setting are presented. Section 4 shows experimental results and discussion. Finally, this paper is concluded in section 5.

II. METHODOLOGY

This section describes the fundamental concept and method is used in this paper. This section covers the pre-processing, feature extraction and classification.

A. Pre-processing

Pre-processing stage involves all of the operations to produce a clean character image, so that it can be used directly and efficiently by the feature extraction stage.

TABLE II
THE PROPERTIES OF SHAPE FEATURE EXTRACTION APPROACHES [4]

Shape representation			Recon-struc-ture	Invariance				Resistance			Computational complexity
				Translation	Rotation	Scale	Affine transform	Noise	Occultation	Non-rigid deformation	
Shape signatures	Complex coordinates		Yes	Bad	Bad	Bad	Bad	Average	Good	Bad	Low
	Central distance		No	Good	Good	Good	Bad	Average	Good	Bad	Low
	Tangent angle		No	Good	Good	Good	Bad	Bad	Good	Average	Low
	Curvature function		No	Good	Good	Good	Bad	Bad	Good	Average	Low
	Area function		No	Good	Good	Good	Good	Good	Good	Bad	Low
	Triangle-area representation		No	Good	Good	Good	Good	Good	Average	Average	Low
	Chord length function		No	Good	Good	Good	Bad	Bad	Bad	Bad	Low
Polygonal approximation	Merging	Distance threshold	No	Good	Good	Good	Bad	Good	Bad	Bad	Average
		Tunneling	No	Good	Good	Good	Bad	Good	Bad	Bad	Average
		Polygon evolution	No	Good	Good	Good	Bad	Good	Bad	Bad	Average
	Splitting		No	Good	Good	Good	Bad	Good	Bad	Bad	Average
Space interrelation Feature	Adaptive grid resolution		Yes	Good	Good	Good	Bad	Good	Good	Bad	Low
	Bounding box		Yes	Good	Good	Good	Average	Good	Good	Average	Average
	Convex hull		No	Good	Good	Good	Good	Average	Bad	Bad	High
	Chain code	Basic chain code	Yes	Good	Bad	Bad	Bad	Bad	Good	Bad	Low
		Vertex chain code	Yes	Good	Bad	Bad	Bad	Bad	Good	Bad	Low
		Statistic chain code	No	Good	Bad	Bad	Bad	Bad	Bad	Bad	Low
	Smooth curve decomposition		No	Good	Good	Good	Bad	Good	Good	Average	Average
	ALI-based representation		No	Good	Good	Good	Average	Good	Average	Bad	Average
	Beam angle statistics		No	Good	Good	Good	Bad	Good	Bad	Bad	Low
	Shape matrix	Square model	Yes	Good	Good	Good	Bad	Bad	Good	Bad	Average
		Polar model	Yes	Good	Good	Good	Bad	Bad	Good	Bad	Low
	Shape context		No	Good	Good	Good	Bad	Bad	Average	Average	Average
	Chord distribution		No	Good	Good	Good	Bad	Good	Bad	Bad	Low
	Shock graphs		Yes	Good	Good	Good	Good	Good	Good	Good	High
Moments	Boundary moments		No	Good	Good	Good	Bad	Average	Bad	Bad	Low
	Region moments	Invariant moments	No	Good	Good	Good	Bad	Bad	Bad	Bad	Average
		Algebraic Moment	No	Good	Good	Good	Good	Average	Bad	Bad	Average
		Zernike Moments	No	Good	Good	Good	Bad	Good	Average	Average	High
		Radial Chebyshev Moments	No	Good	Good	Good	Bad	Good	Average	Average	High
Scale-space methods	Curvature scale space		No	Good	Good	Good	Average	Good	Good	Average	Average
	Intersection points map		No	Good	Good	Good	Average	Good	Good	Bad	Average
Shape transform domains	Fourier descriptors	1-D Fourier descriptor	No	Good	Good	Good	Bad	Bad	Bad	Bad	Average
		Region-based Fourier descriptor	No	Good	Good	Good	Good	Good	Average	Average	High
	Wavelet transform		No	Good	Good	Good	Good	Average	Average	Bad	Average
	Angular radial transformation		No	Good	Good	Good	Bad	Good	Bad	Bad	High
	Signature harmonic embedding		No	Good	Good	Good	Average	Good	Average	Bad	High
	$\Re$ -Transform		No	Good	Good	Good	Bad	Good	Average	Average	High
	Shapelets descriptor		No	Good	Good	Good	Bad	Good	Bad	Bad	High

Thinning is an important pre-processing step in OCR. The purpose of thinning [6] is to delete redundant information and at the same time retain the characteristic features of the image. Thinning is applied to find a skeleton of a character. Skeleton is an output of thinning process. This pre-processing stage only involves of a thinning process.

B. Feature Extraction (Specific Extraction Methods)

The main objective of feature extraction is to remove redundancy from data. Chain code is one of the representation techniques that is useful for image processing, shape analysis and pattern recognition fields. FCC is selected as the technique to represent numbers and Latin characters.

This paper proposes to use randomized algorithm to generate the FCC. The pseudo-code of randomized algorithm is depicted in Table III and the description of randomized algorithm is shown in Fig. 5. As in Fig. 6 shows the architecture of recognition system. The procedure of the randomized algorithm is as following: start from the first node, which is node-method and end-node-method. Node-method is to find the first character for every aspects of boundary such as left upper, left lower, right upper and right lower. End-node-method is to find the first character based on the end position of a character.

In this randomized algorithm, if the number of visited node less than the number of nodes, there would be three kinds of characteristics, which are unvisited, visited and taboo neighbours. Unvisited neighbours are nodes that never went through the route searching. Visited neighbours indicate the nodes that have went through the route searching. Taboo neighbours are used to keep track of the visited search space and revisited node with one step after current node.

The features that become the input to the classification stage are the chain codes that have been converted to 69 features (8 directions of chain code x 8 routes of FCC = 64 features + 5 extra features = 69 features). Sixty four features are created from the generated FCC and five extra features are from the calculated values of ratio-upper, ratio-right, ratio-height-weight, ratio-height and the number of tittles. For the detail about five extra features can depicted below:

- The ratio-upper is calculated from firstly by cropping the image and then defining the centre of the image character. After that, the total number of upper character is divided with the total number of character. For “B” character based on in “Fig. 2” ratio-upper = 43/81. The formula of ratio-upper as shown in Equation 1 below:

Ratio-upper = number of position of upper character / number of total character (1)

- The ratio-right is calculated similarly to the ratio-upper. After cropping the image and defining the centre of the image character finally, calculate the ratio-right. For “B” character based on in Fig. 2 ratio-right = 34/81. The formula of ratio-right as shown in Equation 2 below:

Ratio-right = number of position of right character / totalnumber of character (2)

- The ratio-height-weight is calculated similarly to the ratio-upper. After cropping the image and defining the centre of the image character finally, calculate the ratio-height-weight. For “B” character based on in “Fig. 2” ratio-height-weight = 31/17. The formula of ratio-height-weight as shown in Equation 3 below:

Ratio-height-weight = number of position of height character / number of position of weight character (3)

- The ratio height is calculated from firstly by cropping the image and then calculated the ratio height. For “B” character based on in Fig. 2 ratio-height = 31/50. The formula of ratio-height as shown in Equation 4 below:

Ratio-height = number of height of character/ number of height of image (number of pixel image) (4)

- The number of titles (or dots) for lower-case *i* and *j* are calculated as 1 and the rest of characters are calculated as 0.

TABLE III
THE PSEUDO-CODE OF RANDOMIZED ALGORITHM

```
Initialize Data
while Termination Not Met do
   Select First Node Randomly
   {Node-Method, End-Node-Method}
           while Number of Visited Node < Number of Node do
                   if there are Unvisited neighbours
                         Select One Node Randomly
                   elseif there are Visited neighbours
                         Select One Node Randomly
                   elseif there are Taboo neighbours
                         Select One Node Randomly
                   endif
           endwhile
endwhile
Display Solutions
```

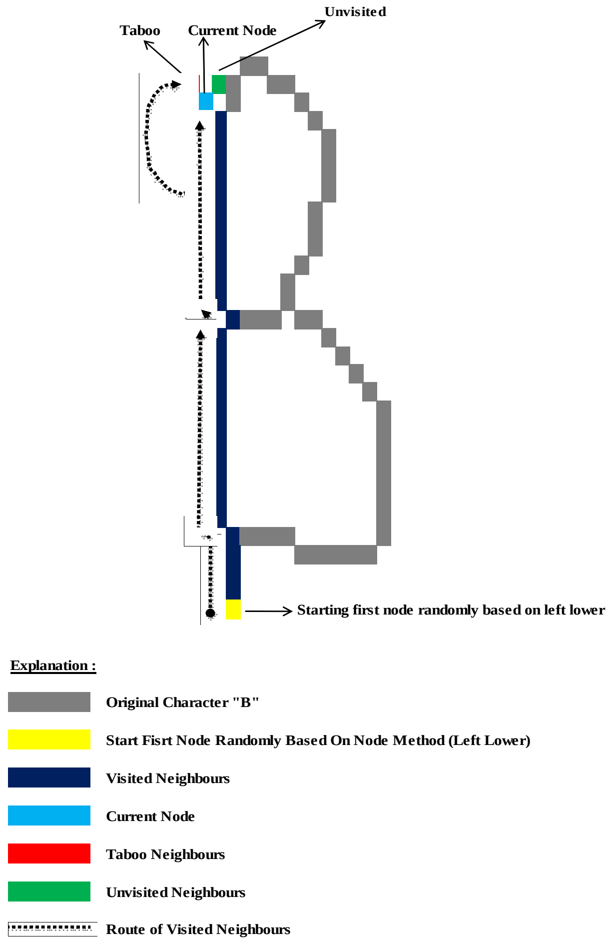

Fig. 5 Description of randomized algorithm

A. Classification (Neural Network Classifier)

Neural network classifier is used to classify data. Artificial Neural networks (ANN) is a computational model naturally performing a parallel processing of information [7]. The processing ability of ANN is similar to the brain that it can learn, recognize and remember through imitating and realizing abstract thinking and associative memory [8] as shown in Table IV. The biological of NN as shown in Fig. 7.

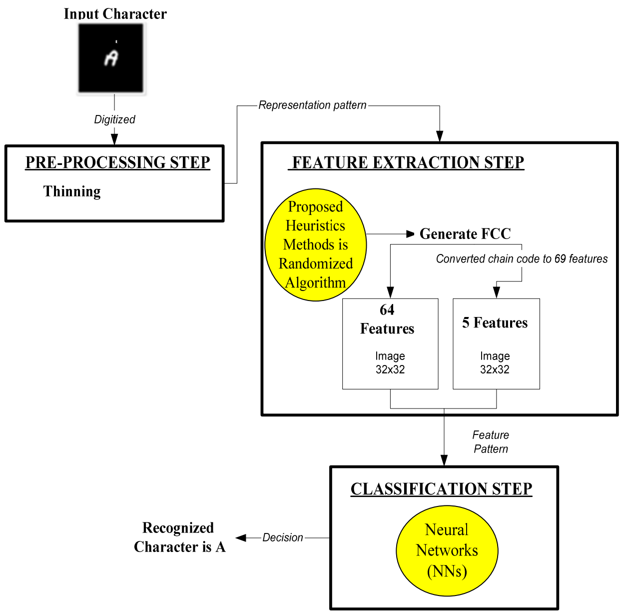

Fig. 6 The architecture of recognition system

TABLE IV
THE SIMILARITY BIOLOGICAL NEURAL NETWORK AND ANN

Biological Neural Network	**ANN**
Soma	Neuron
Dendrite	Input
Axon	Output
Synapse	Weight

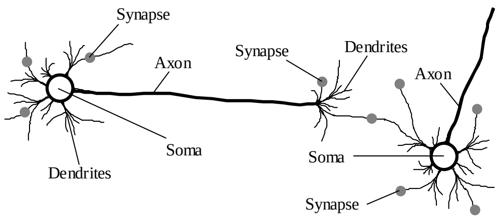

Fig. 7 The biological of NN [7]

ANN can be classified into feedforward or recurrent depend on their connectivity. An ANN is a feedforward network if an arbitrary input vector is propagated forward through the network and caused an activation vector to be produced in the input layer. On the other hand, recurrent network if the output vector is propagated backward to the previous layer. For this experiment feedforward backpropagation network is used.

Multilayer perceptron consists of an input layer, hidden layer (at least one layer) and an output layer as shown in Fig. 8.

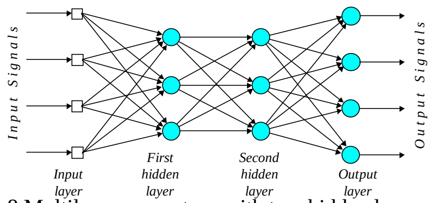

Fig. 8 Multilayer perceptron with two hidden layers [7]

In a backpropagation network two processes involved are input signal and error signal as shown in Fig. 9. Input signal presented to input layer and continued from layer to layer until produced the output layer. On the contrary, error signals is caused the different from the desired output then error is calculated after that propagated backwards from the output layer to the input layer.

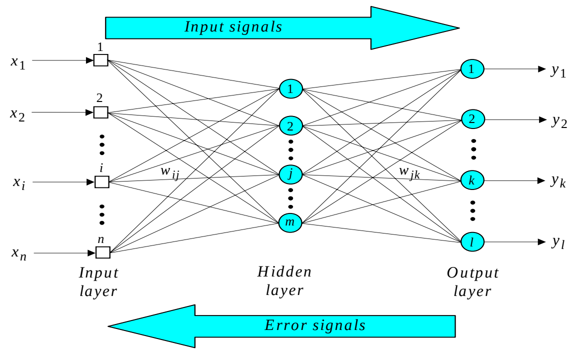

Fig. 9 Three-layer back-propagation neural network [7]

III. PROBLEM DATA & PARAMETER SETTING

We evaluate the performance of heuristic method from randomized algorithm. The scope area is isolated handwritten on upper-case characters (A-Z), lower-case characters (a-z), letter (upper-case+lower-case) characters, digit (0-9) and all characters (A-Z, a-z, 0-9). The database is used NIST (National Institute of Standards and Technology) which is hsf-4 only. The original data of hsf-4 for upper-case (12,000), lower-case (11,941), letter (23,941), digit (58,646) and all characters (82,587).

From the experiment, not all image characters in the NIST hsf4 database can be used in the experiment. For instance, NIST database has 12,000 upper-case letters; however, only 11,729 samples can be used due to the very poor quality samples and sometimes broken parts as shown in Table V.

TABLE V
THE DATA SETS ORIGINAL HSF4 IN NIST AND DATA SETS AFTER GENERATE FEATURES OF NN

No	**Kinds of data sets**	**Original data (HSF4 in NIST)**	**Data sets after generate Features by NN**
1	Upper-case	11,941	11,729
2	Lower-case	12,000	11,864
3	Letter (Upper+Lower)	23,941	23,593
4	Digit	58,646	57,586
5	All (Letter+Digit)	82,587	81,179

The specification of hardware and software are shown below:

a) Desktop Hawlett-Packard (HP) Model a6665
  - Memory (RAM) 3GB
  - Processor is Intel ® Core ™ 2 Quad CPU Q8200 @ 2.33 GHz

b) Matlab R2008a (version 7.6.0)

Parameter values used in the proposed NN are as follow:

a) Input layer (number of class data) = 69
b) Hidden layer (S1) = 69
c) Output layer (S2) = 69
d) Net.performfFcn = 'sse' (Sum-squared error performance function)
e) Net.trainParam.goal = 0.1 (Mean sum-squared error goal)
f) Net.trainParam.show =20 (Frequency of progress displays (in epoch))
g) Net.trainParam.epochs = 5,000 (Maximum number of epochs to train)
h) Net.trainParam.mc = 0.95 (Momentum constant)
i) Net.trainParam.lr = 0.1 (Learning rate)
j) Cross validation number = 5
k) Limit data =0
l) Number of Data Tested = 5,000

IV. Experimental Result and Discussion

The experiment only concentrates in hsf-4 of NIST databases. Table IV shows the result of five datasets using the feedforward backpropagation network. Figure 10 and Figure 11 shows the computation time and accuracy for each data sets based on Table VI.

TABLE VI
THE RESULT OF THREE DATASETS USING NN

Data Sets	Samples	Accuracy (%)	Computation Time (seconds)
Upper-case	11,729	98.92	2,579.23
Lower-case	11,864	98.54	766.46
Letter (Upper+Lower)	23,593	98.01	1,111.84
Digit	57,586	98.77	6,659.64
All (Letter + Digit)	81,179	97.48	22,965.40

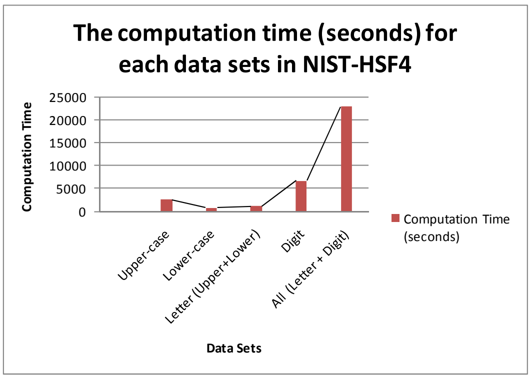

Fig. 10 The computation time (seconds) for each data sets in NIST–HSF4

From the results depicted in Table VI, it can be seen that the proposed method have successfully recognised the handwriting characters from NIST-HSF4 database with high accuracy of more than 98% for upper-case, lower-case, letter and digit and more than 97% for all data set.

V.Conclusion

This paper proposes a model for recognition of handwritten Latin characters. The proposed model starts with pre-processing. Thinning algorithm is used in the pre-processing stage that produced a skeleton of a character. The second step is feature extraction. Randomized algorithm has been applied to generate FCC from handwritten characters and minimized the length of the FCC.

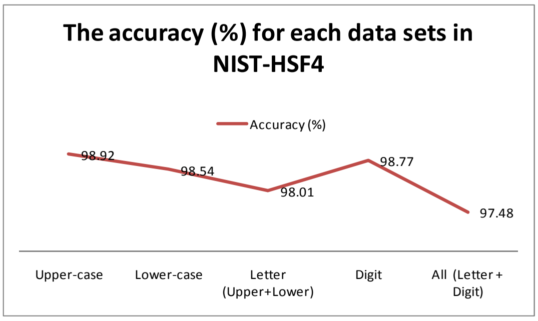

Fig. 11 The accuracy (%)for each data sets in NIST-HSF4

From the experiment, not all image characters in the NIST HSF-4 database can be used in the experiment due to the very poor quality samples and sometimes broken parts, which made recognition.

Our test results based on experiment and proposed method shows we get the high accuracy for Latin and numbers handwritten characters. The efficiency of FCC in the representation can be seen by the numbers of image characters that can be recognized.

Acknowledgment

The authors honorably appreciate Ministry of Science, Technology and Innovation (MOSTI) for the eScienceFund grant with VOT number 79369 and Research Management Center (RMC), Universiti Technologi Malaysia (UTM) for the support in making this projects success.

References

[1] Tay, Y.H., *Offline Handwriting Recognition using Artificial Neural Network and Hidden Markov Model*. PhD Thesis. Universiti Teknologi Malaysia, 2002.
[2] Al-Rashaideh, H., *Preprocessing Phase for Arabic Word Handwritten Recognition*. Information Transmission in Computer Network. 2006.
[3] Steinherz, T., Rivlin, E., Intrator, N., *Off-Line Cursive Script Word Recognition - A Survey*. International Journal on Document Analysis and Recognition. 1999.
[4] Mingqiang, Y., Kidiyo, K., Joseph, R., *A Survey of Shape Feature Extraction Techniques*. Book Pattern Recognition (Edited by: Peng-Yeng Yin). November 2008. IN-TECH.
[5] Wegener, I., *Towards a Theory of Randomized Search Heuristics*. Lecturer Notes in Computer Science. Springer. 2003.
[6] Ravichandran, A., *An Algorithm for Thinning Noisy Images*. International Conference on Acoustics Speech and Signal Processing, 1990.
[7] Alba, E., Chicana, J.F., *Training Neural network with GA Hybrid Algorithm*. Proc of GECCO'04. Seattle, Washington. 2004.
[8] Chen, T-P, Tian.J-X., *The Research of Artificial Neural Network as a Tool of Adjustment Prediction in Eco-City Construction*. Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005.

Dewi Nasien was born in Pekanbaru, Indonesia, June 30, 1982. She received her degree in the field of Information and Technology at Universitas Islam Negeri SUSQA Pekanbaru, Indonesia in 2005. She obtained a M.Sc in Computer Science at Universiti Teknologi Malaysia, Johor Bahru, Malaysia in 2008.

Currently, she is concluding a Ph.D at the Department modeling and industrial computing, Universiti Teknologi Malaysia, Johor Bahru, Malaysia. Her research interest are in the field of pattern recognition and soft computing.

Siti Sophiayati Yuhaniz was born in Selangor, Malaysia in 1980. She received her Bachelor degree in the field of computer science at Universiti Teknologi Malaysia, then obtained her PhD in the field of on-board satellite image processing at University of Surrey, UK in 2008.

She is one of the faculty members of Faculty of Computer Science and Information System, UTM, under the Modeling and Industrial Computing Department. She began lecturing in July 2008, and a member of Soft Computing Research Group (SCRG) which is under the K-Economy Research Alliance, Universiti Teknologi Malaysia. Her research interests are in the field of soft computing, pattern recognition, and remote sensing.

Habibollah Haron was born in Malacca, Malaysia in 1966. He received his degree in Computer Science from Universiti Teknologi Malaysia in 1989, then Master degree in 1995 from University of Sussex, UK. He obtained his PhD in the area of Computer Aided Geometric Design (CAGD) in the year of 2004 at Universiti Teknologi Malaysia.

He has been with Universiti Teknologi Malaysia since 1991 as a faculty member, and was appointed as Associate Professor in 2008. He was also the Head of Department for Modeling and Industrial Computing Department, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia before currently attached with Imam Mohammad Ibn Saud University, Riyadh for his sabbatical work. Some of his research areas are computer aided design and soft computing.