Volume 3, Issue 4, April 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Optical Character Recognition using 40-point Feature Extraction and Artificial Neural Network

Sandeep Saha
Murshidabad College of Engg & Tech
Berhampore, West Bengal, India.

Nabarag Paul
Murshidabad College of Engg &Tech.
Berhampore, West Bengal, India.

Sayam Kumar Das
Murshidabad College of Engg & Tech.
Berhampore, West Benga, India.

Sandip Kundu*
Asssistant Professor *Murshidabad College of Engg & Tech.* Berhampore,West *Bengal, India.*

Abstract— We present in this paper a system of English handwriting recognition based on 40-point feature extraction of the character. Basically an off-line handwritten alphabetical character recognition system using multilayer feed forward neural network has been described in our work. Firstly a new method, called, 40-point feature extraction is introduced for extracting the features of the handwritten alphabets. Secondly, we use the data to train the artificial neural network. In the end, we test the artificial neural network and conclude that this method has a good performance at handwritten character recognition. This system will be suitable for converting handwritten documents into structural text form and recognizing handwritten names.

Keywords— Character Recognition, Training, Feature Extraction, Image Processing, ANN.

I. Introduction

Handwriting recognition has been one of the most fascinating and challenging research areas in field of image processing and pattern recognition in the recent years. It contributes immensely to the advancement of an automation process and can improve the interface between man and machine in numerous applications. Several research works have been focusing on new techniques and methods that would reduce the processing time while providing higher recognition accuracy. In general, handwriting recognition is classified into two types as off-line and on-line handwriting recognition methods. In the off-line recognition, the writing is usually captured optically by a scanner and the completed writing is available as an image. But, in the on-line system the two dimensional coordinates of successive points are represented as a function of time. And the orders of strokes made by the writer are also available. The on-line methods have been shown to be superior to their off-line counterparts in recognizing handwritten characters due to the temporal information available with the former. However, in the off-line systems, the neural networks have been successfully used to yield comparably high recognition accuracy levels .Several applications including mail sorting, bank processing, document reading and postal address recognition require off-line handwriting recognition systems. As a result, the off-line handwriting recognition continues to be an active area for research towards exploring the newer techniques that would improve recognition accuracy. In our project we have taken 30 characters for each of English alphabet character starting from A to Z and 10 characters for each of English alphabets for testing of Neural Network to have the accuracy which will make us understand how much accurate we are to make the Artificial Neural Network to recognize each of the English alphabets character perfectly.

II. PREVIOUS WORK

A number of researches have been proposed over the years for character recognition. In [1] the authors have divided each character into a number of predetermined rectangular zones and extracted a 13-element vector comprising of the pixel values in those zones. A neural network classifier has been used to recognize the 26 alphabets of English language. In [2] the authors have proposed twelve directional features based upon gradients of pixels and employed neural networks for classification of handwritten characters. In [3] the authors are concerned with recognizing composite characters in Bengali language formed by joining two or more basic characters, by resizing the characters in a 16 × 16 grid and utilizing a 256 element vector extracted from them by reading the pixel values. Curvelet transforms along with SVM classifiers have been used in [4] to recognize Bangla handwritten characters. In [5] the authors have decomposed characters into a set of structural shape units and used s dynamic time warping based classifiers to identify component shapes in a character. In [6] the authors have used a 392-element feature vector derived from Modified Quadratic Discriminant Function obtained from the gradient image, to identify Bangla compound characters. Fuzzy rule descriptors have been used in [7] to identify handwritten numerals. In [8] a 110-element direction code representing structural shape units have been utilized for recognition of handwritten characters. Wavelet Energy Density Features derived from the DB4 wavelet have been used in [9] to identify numerals 0 to 9 using a 252-element vector. A histogram of chain code direction of contour points represented using a 64-dimensional feature vector have been utilized in [10] to recognized characters from 6 popular Indian scripts. In [11] the authors have used a recursive subdivision of the character image into a number of granularity levels and the coordinates of the points at intersection of each partitioning line is used as the feature vector for recognizing them. In [12] the authors have used a four profile vector (X-profile, Y-profile, diagonal1-profile, diagonal2-profile) to identify Gujarati handwritten numerals using neural network classifiers. In [13] the authors have proposed a method of implicit segmentation of cursive words into their letters without visual cutting and without thinning. In [14] the authors have used convex hull & water reservoir principle to recognize multi-sized and multi-oriented characters of Bangla and Devnagari script, along with Support Vector Machine (SVM) classifiers. Structural units called strokes have been used in [5] to identify handwritten Bengali characters using a Hidden Markov Model classifier.

III. OPTICAL CHARACTER RECOGNITION (OCR)

The Optical Character Recognition is a process of automatic recognition of different characters from a document image and also provides a full alphanumeric recognition of printed or handwritten characters, text, numerals, letters and symbols into a computer process able format such as ASCII. At electronic speed by simply scanning the form. The process involves clear and unambiguous recognition, analysis and understanding of the document content. OCR is an area of pattern recognition and processing of handwritten character is motivated largely by desire to improve man and machine communication. OCR of Indian scripts is in preliminary stage. Much of the research work has been done for developing OCR systems in Roman scripts. Compared to this; extensive research and development activities are required for developing OCR systems for Indian scripts. OCR involves photo scanning of the text, which converts the paper document into an image, and then translation of the text image into character codes such as ASCII. Cheque reading, postcode recognition, form processing, signature verification are the application of OCR.

IV. BRIEF OF PROBLEM TO BE SOLVED

Recognition of handwritten English character is a process which loads a handwritten English character image, preprocesses the image, extracts proper image features, classify the characters based on the extracted image features and the known features are stored in the Matlab library, and recognizes the image according to the degree of similarity between the loaded image and the image models.

V. TASKS INVOLVED

In this section we will accomplish the following tasks. Data acquisition, preprocessing and segmentation, feature extraction and classification.

A. *Data Acquisition*

Images are collected from different handwritten fonts. It can also be obtained by using a scanner. Write some characters on a white, thick paper with a black signature pen and make black and white show a striking contrast gradient. The image of the handwritten characters is shown in Fig.1below:

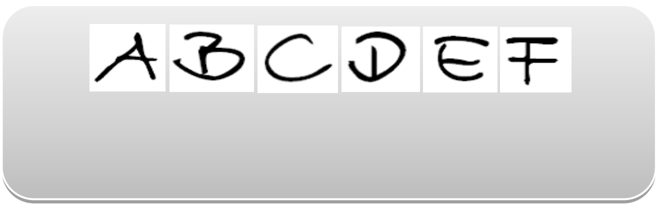

Fig.1.The images of data set.

B. *Preprocessing And Segmentation*

The image preprocessing is accomplished in three steps to reduce useless data and keep valuable information

1. Image cropping 2. Gray the image and morphing 3. Binarization on the image

B.1**.** *Image cropping*

Here the captured image size is so high i.e. high resolution. So, the size of the input image must be reduced. The reduction is done so carefully that the aspect ratio remains same.

B.2. *Gray the image*

In this process the image is converted to two dimensional images from three dimensional images and the matrix containing single character is changed from 64*64*3 to 64*64.The elements of the matrix covers from 0 to 255 now. The image after graying is shown in Fig2.

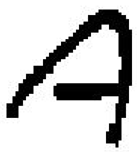

Fig 2:The image after gray

With n = Inf, thins objects to lines. It removes pixels so that an object without holes shrinks to a minimally connected stroke, and an object with holes shrinks to a connected ring halfway between each hole and the outer boundary.The "morphing" Is such a feature that can be used in matlab to thin the images that are acquired for feature extraction. This feature tracks out the **skeleton** of the images for better performance.The lining of each feature point is manually corrected by **S= bwmorph (I,'thin', Inf)** command which makes the picture skeleton lining and the corners of the pictures perfect for better result.

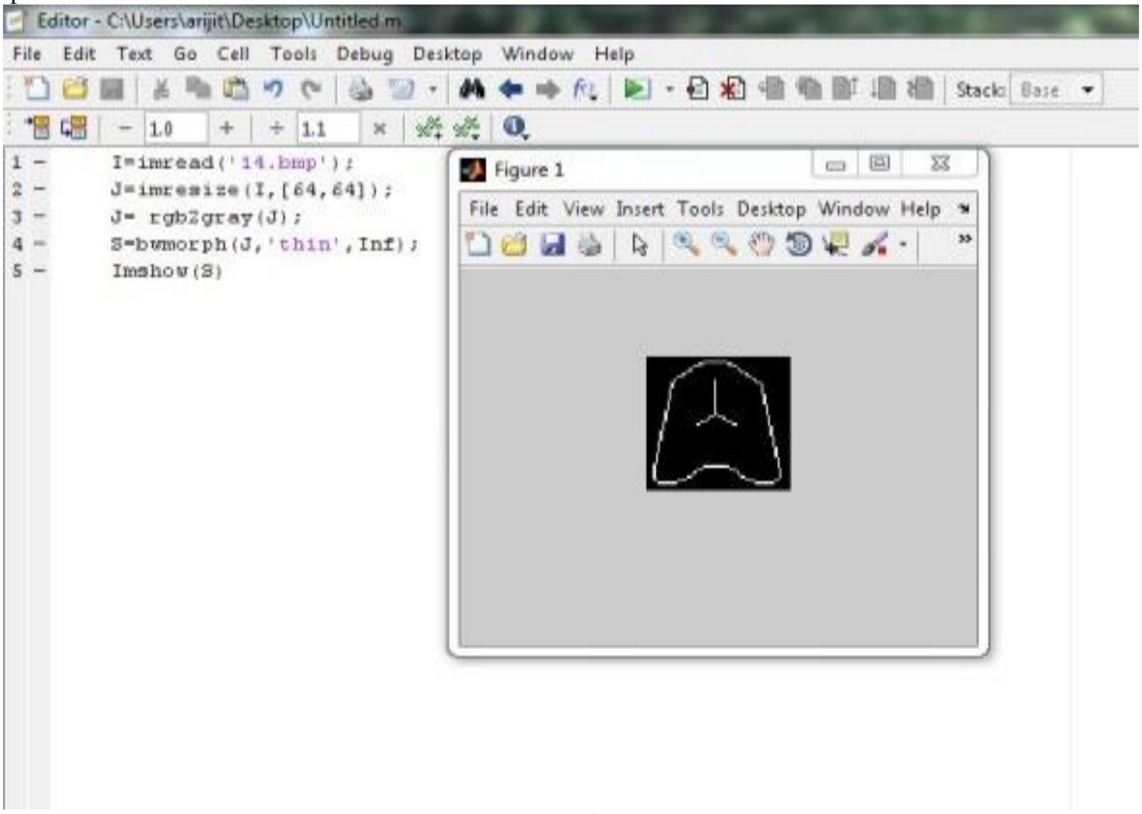

Fig3. Screenshot of morphing

B.3. *Binarization Of Image*

The matrix which we have got after gray scale conversion is complicated to further calculation because the elements in the matrix cover from o to 255.Therefore we make a processing of binarization on the images. The images originally from '0' to '255' are replaced by '0' or '1'.

B.4 *Feature Extraction Procedure*

B.4.1 *Step one*

Firstly we divide the whole image zone averagely into 16 zones with the corresponding mark as shown by figure4.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Fig4.first 16 zone of the image

B.4.2 *Step two*

Secondly, the entire image is divided diagonally i.e. starting from the left side of the top of the image towards the right side of the bottom of image. Zone 17 consists of zone1, zone2, zone5 and zone6; zone18 consists of zone1, zone2, zone3, zone5, zone6,zone7,zone9, zone10 and zone11. And then we take the entire image.

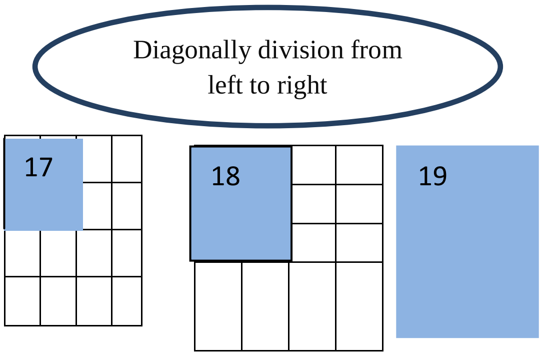

Fig5.zone17-19 of the image

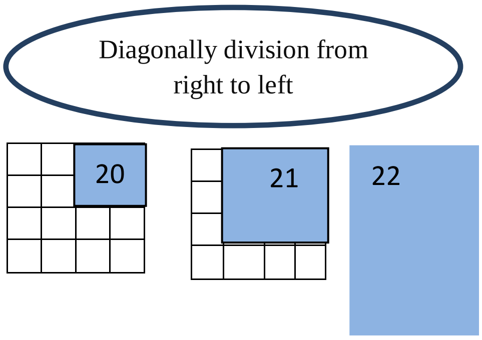

Fig6.zone20-22 of the image

B.4.2 *Step three*

Likewise, the entire image is again divided diagonally from another side i.e. starting from the right side of the top of the image towards the left side of the bottom of image. Zone 20 consists of zone3, zone4, zone7 and zone8; zone21 consists of zone2, zone3, zone4, zone6, zone7, zone8, zone10, zone11 and zone12. And then we take the entire image in figure5.

B.4.4 *Step four*

Similarly the entire image is again divided from the left side of the bottom of the image towards the right side of the top of image. Zone 23 consists of zone9, zone10, zone13andzone14; zone24 consists of zone5, zone6zone7, zone9, zone10, zone11, zone13, zone14 and zone15.Zone 25 consists of the whole image in figure6.

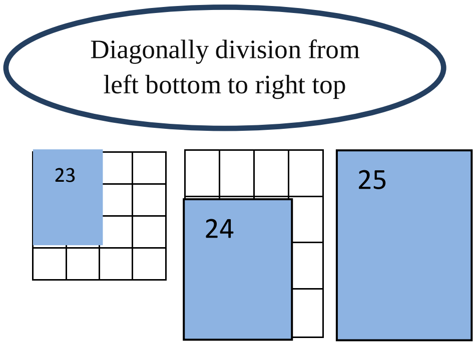

Fig7.zone23-25 of the image

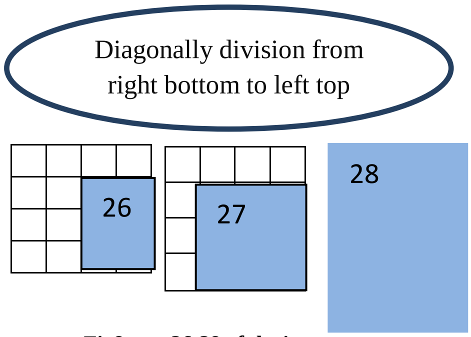

Fig8.zone26-28 of the image

B.4.5 *Step five*

Again the entire image is divided diagonally from the right side of the bottom of the image towards the left of the top of the image. Zone26 consists of zone 11, zone12, zone15 and zone16. Zone27 consists of zone6, zone7, zone8, zone10, zone11, zone12, zone14, zone15 and zone16.Zone 28 consists of the entire image in figure7.

B.4.6 *Step six*

Zone 29 is made by taking the innermost 4 cells which make a square themselves and one cell is made as if the whole 64*64 image has been divided into 64 cells [15] each of size 8*8.Zone 30 is done by taking 16 innermost cells which make a square themselves and zone 31 is made by taking 32 innermost cells which make a square as well.Zone32 comprises of the whole image in figure8.

B.4.7 *Step seven*

The zone33-40 has been taken as shown in the figure9 below.

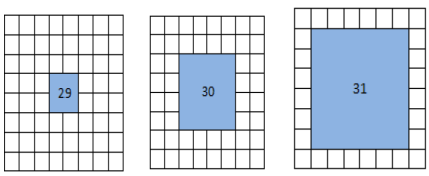

Fig9.zone29-31 of the image

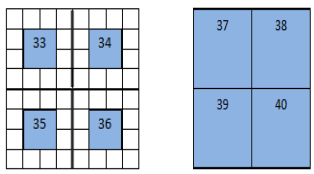

Fig10.zone33-40 of the image

VI. EXPERIMENTATION AND RESULTS

1. *Data Set*

The dataset consists of 1040 images of upper-case English alphabets of various appearances divided into training and testing sets. The training set consists of 30 different instances of each of the 26 English alphabets, a total of 780 images. The training set is indicated by legends ATR, BTR, CTR… ZTR. The testing set consists of 10 different instances of each of the 26 alphabets, a total of 260 images. The testing set is indicated by legends ATS, BTS, CTS… ZTS.

2. *Training phase*

The training phase consists of computing the 40-element feature vectors from each of the 780 images of the training set, using the dynamic window method. The feature plots for the training set, is shown below. The legend 'TR' denotes the Training set. Fig. 7 indicates the variation of the mean values of the first 14 elements of the feature vector over all the 40 instances of each character, shown for the first 9 characters [16], here x-axis refers zones and y-axis refers corresponding zone values

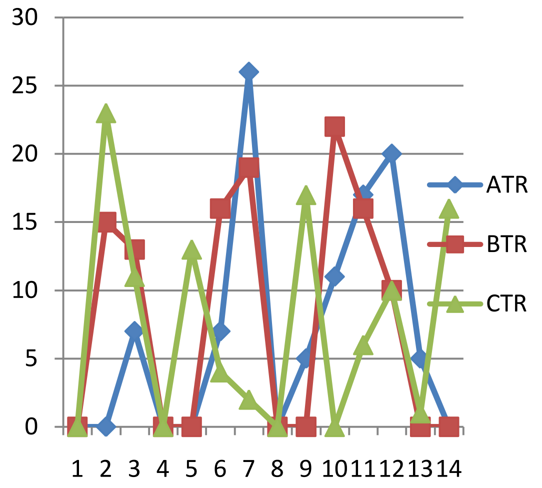

Fig11.Mean values of first 14 element of feature training vector

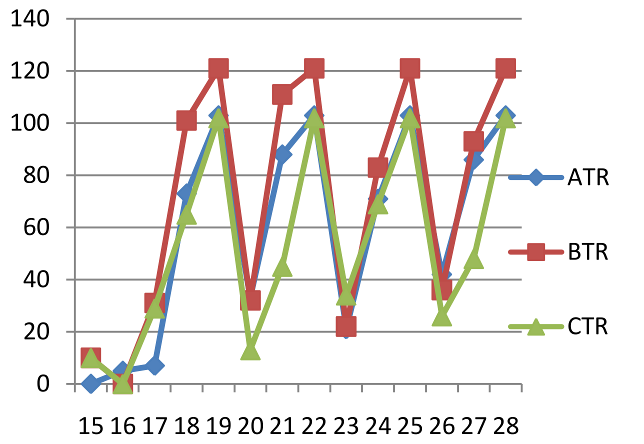

Fig12.Mean values of element 15-28 of feature training vector

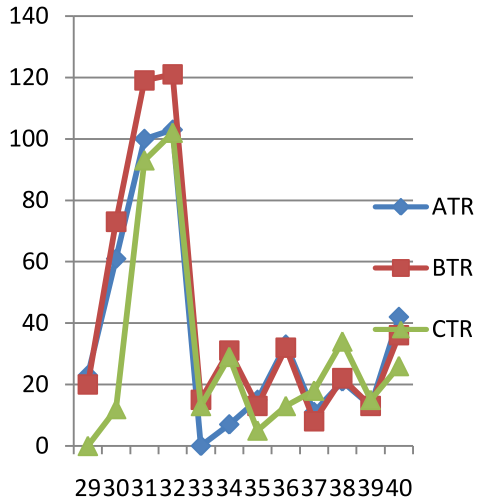

Fig13.Mean values of first element 29-40 of feature training vector

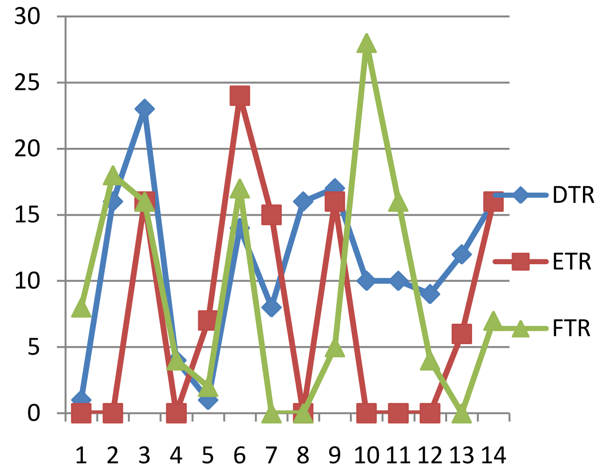

Fig14.Mean values of first 14 element of feature training vector

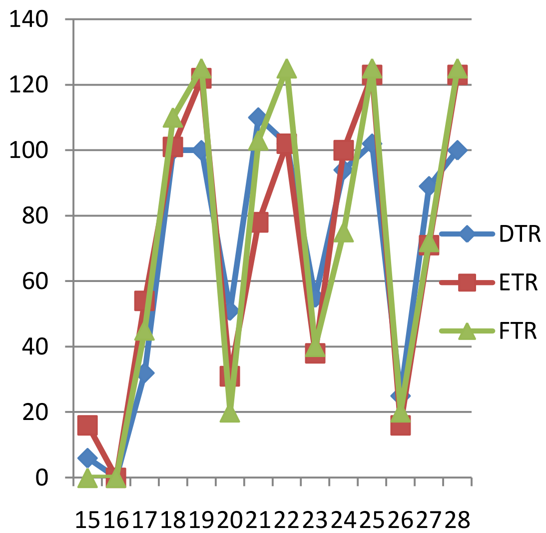

Fig15.Mean values of first element 15-28 of feature training vector

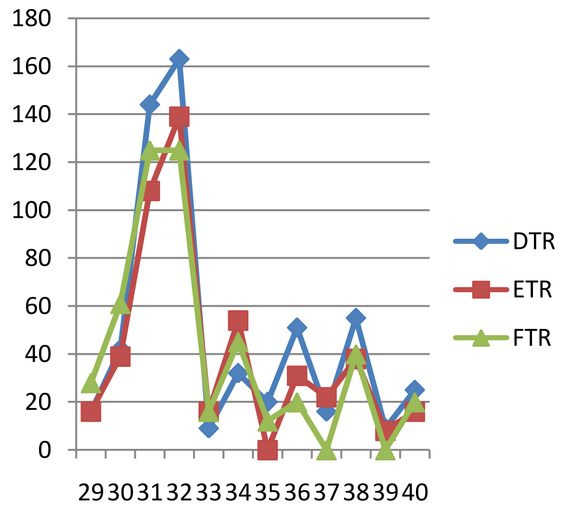

Fig16.Mean values of first element 29-40 of feature training vector

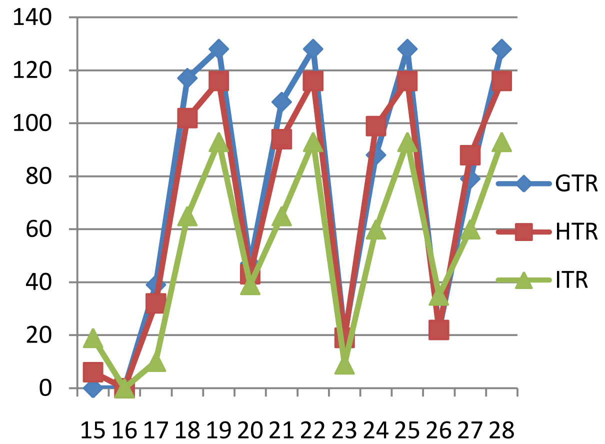

Fig17. Mean values of first 14 element of feature vector of training set

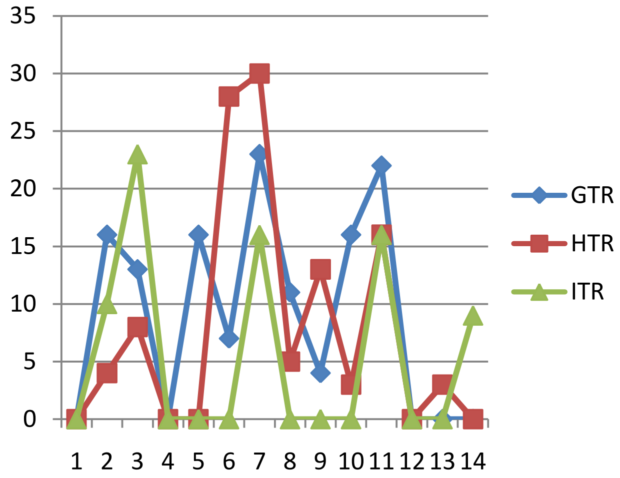

Fig18 Mean values of elements 15-28 of feature vector of training set

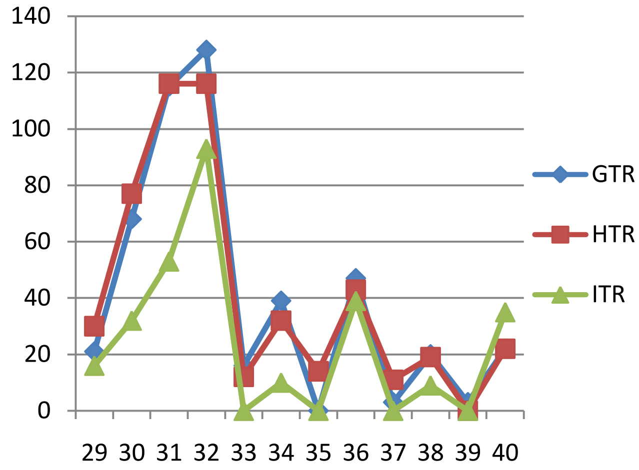

Fig19. Mean values of elements 29-40 of feature vector of training set

3. *Testing Phase*

The testing phase consists of computing the 40-element feature vectors from each of the 260 images of the testing set, using the dynamic window method. The feature plots for the testing set, is shown below. The legend 'TS' denotes the

Testing set. Fig. 10 indicates the variation of the mean values of the first 14 elements of the feature vector over all the 40 instances of each character, shown for the first 6 characters here x-axis refers zones and y-axis refers corresponding zone values

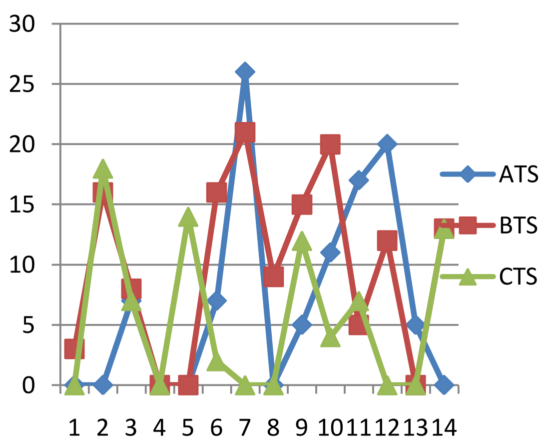

Fig20.Mean values of first 14 element of feature vector of testing set

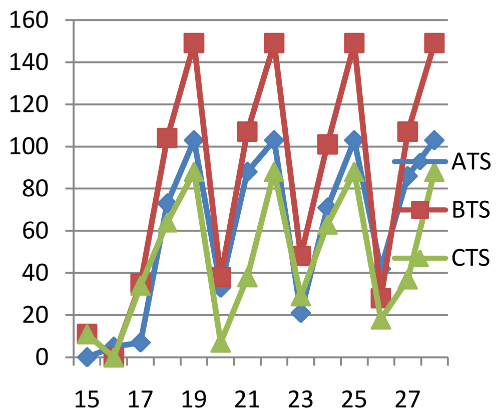

Fig21. Mean values of element 15-28 of feature vector of testing set

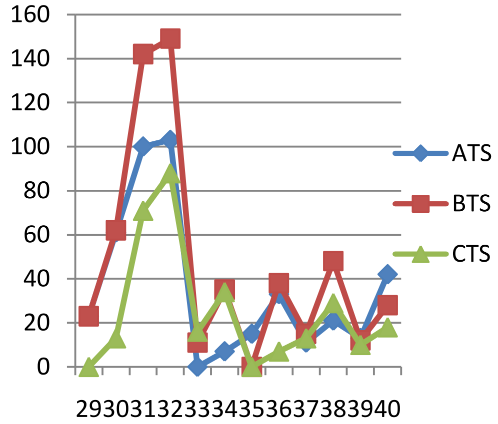

Fig22. Mean values of element 29-40 of feature vector of testing set

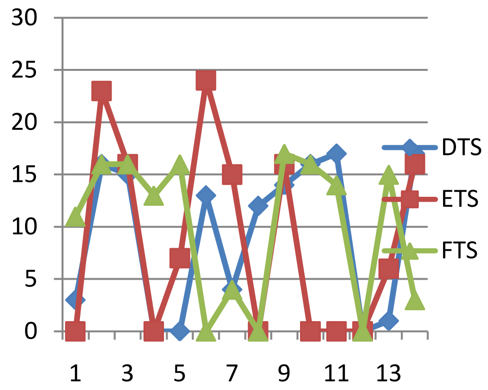

Fig23.Mean values of first 14 element of feature vector of testing set

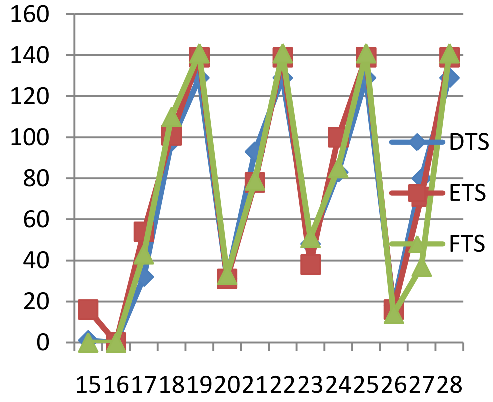

Fig24. Mean values of element 15-28 of feature vector of testing set

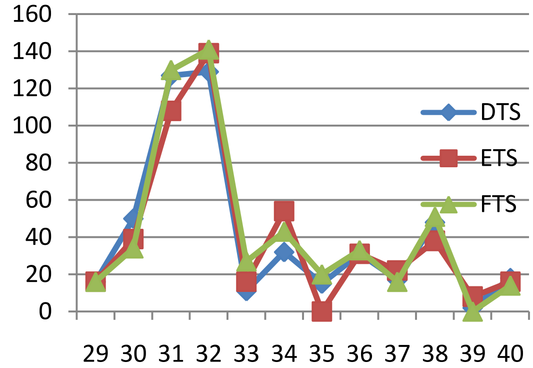

Fig25. Mean values of element 15-28 of feature of testing set

4. *Classification*

Classification is done using a neural network (NN) (MLP: multi-layer perceptron) [17]. The MLP consists of 40 inputs for feeding in the 40-element feature vector for each character, and 26 outputs for discriminating between the characters. The activation transfer functions are of log-sigmoid type. The best overall accuracy of 83.84% was achieved with 260 units in the hidden layer. Table 1 below reports accuracy rates obtained.

TABLE1.
PERCENTAGE RECOGNITION ACCURACIES

A	B	C	D	E	F	G
90	90	80	100	20	80	80
H	I	J	K	L	M	N
90	70	70	90	100	60	90
O	P	Q	R	S	T	U
100	80	80	90	80	80	70
V	W	X	Y	Result		
90	100	100	100	83.84%		

The MSE (mean square error) obtained after 100000 epochs was around 0.0[illegible] convergence plot is shown in Figure 26.

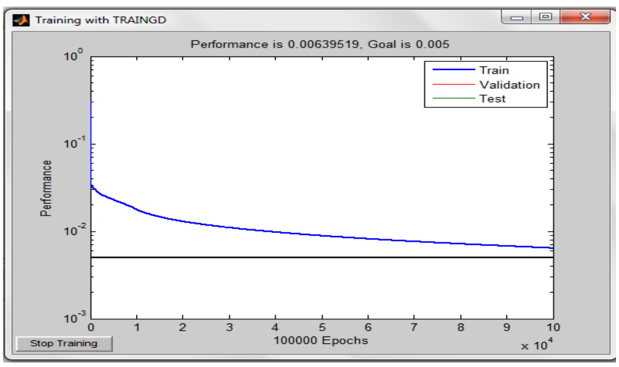

Fig26. Screenshot of NN plot

VII. Conclusion

The main objective of the project is to determine characters from any given text of A-Z. An Artificial Neural Network has been created and trained to diagnose a single character. 30 set of each character has been used to train the Neural Network. 40 point feature extraction forms the basic underlying part of recognizing each character during testing. The different attributes and character morphology of single alphabets are highlighted by the 40 point feature extraction technique and stored in the Matlab created Neural Network. When testing is done the features extracted from the tested character are simultaneously matched with those previously stored in the neural network. The maximum percentage of matching of the features extracted from the training and the testing characters give the resultant alphabet as output in a graph.

Acknowledgments

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. Our thanks go to all the colleagues and friends who have contributed their valuable time and efforts to help us to develop this paper in an organized way.

Reference

[1] C. Zhong, Y. Ding, J. Fu., "Handwritten Character Recognition Based on 13-point feature of Skeleton and Self-Organizing Competition Network", In Proceedings of 10th International Conference on Intelligent Computation Technology and Automation (ICICTA), pp. 414-417, 2010.

[2] D. Singh, S. K. Singh, M. Dutta , "Handwritten Character Recognition using Twelve Directional Feature Input and Neural Network". International Journal of Computer Application, 2010, pp. 82 – 85.

[3] A. R. Md. Forkan, S. Saha, Md. M. Rahman, Md. A. Sattar, "Recognition of Conjuctive Bangla Characters by Artificial Neural Network", In Proceedings of International Conference on Information and Communication Technology (ICICT), 2007, pp. 96-99.

[4] B.B. Chaudhuri and A. Majumdar, "Curvelet–based Multi SVM Recognizer for Offline Handwritten Bangla: A Major Indian Script", In Proceedings of Ninth International Conference on Document Analysis and Recognition (ICDAR), 2007.

[5] A. Bandyopadhyay, B. Chakraborty, "Development of Online Handwriting Recognition System: A Case Study with Handwritten Bangla Character", In Proceedings of World Congress on Nature & Biologically Inspired Computing (NaBIC), 2009, pp.514-519.

[6] U. Pal, T. Wakabayashi and F. Kimura, "Handwritten Bangla Compound Character Recognition using Gradient Feature", In Proceedings of 10th International Conference on Information Technology (ICIT), 2007, pp.208-213.

[7] Md. M. Hoque, Md. M. Islam, Md. M. Ali, "An Efficient Fuzzy Method for Bangla Handwritten Numerals Recognition", In Proceedings of 4th International Conference on Electrical and Computer Engineering (ICECE), 2006,pp.197-200.

[8] U. Bhattacharya, B. K. Gupta and S. K. Parui, "Direction Code Based Features for Recognition of Online Handwritten Characters of Bangla", In Proceedings of Ninth International Conference on Document Analysis and Recognition (ICDAR), 2007.

[9] M. Li, C. Wang, R. Dai, "Unconstrained Handwritten Character Recognition Based on WEDF and Multilayer Neural Network", In Proceedings of the 7th World Congress on Intelligent Control and Automation, 2008, pp. 1143-1148.

[10] U. Pal, T. Wakabayashi, N. Sharma and F. Kimura, "Handwritten Numeral Recognition of Six Popular Indian Scripts", In Proceedings of 9th International Conference on Document Analysis and Recognition (ICDAR),2007,pp. 749 -753.

[11] G. Vamvakas, B. Gatos, S. J. Perantonis, "Handwritten character recognition through two-stage foreground sub-sampling", Pattern Recognition, 2010, pp. 2807-2816.

[12] A. A. Desai, "Gujarati handwritten numeral optical character reorganization through neural network", Pattern Recognition, 2010, pp. 2582–2589.

[13] K. Saeed, M. Albakoor, "Region growing based segmentation algorithm for typewritten and handwritten text recognition", Applied Soft Computing, 2009, pp. 608 – 617.

[14] U. Pal, P. P. Roy, N. Tripathy, J. Llados , "Multi-oriented Bangla and Devnagari text recognition", Pattern Recognition, 2010, pp. 4124–4136.

[15] B.V.Dhandra, Gururaj Mukarambi, Mallikarjun Hangarge "Handwritten Kannada Vowels and English Character Recognition System", International Journal of Image Processing and Vision Sciences (IJIPVS) Computer Applications (IJCA), Volume-1, Issue-1, pp.12-17, 2012.

[16] Mithun Biswas, Ranjan Parekh "Character Recognition using Dynamic Windows", Jadavpur University, Kolkata, India. International Journal of Computer Applications (0975 – 8887), Volume 41– No.15, pp.47-52, March 2012.

[17] Rakesh Kumar Mondal, N R Manna, "Handwritten English Character Recognition using Row-wise Segmentation (RST)", International Journal of Computer Applications® (IJCA),pp.5-9,2011.

