

Character Recognition using Dynamic Windows

Mithun Biswas

School of Education Technology
Jadavpur University
Kolkata, India

Ranjan Parekh

School of Education Technology
Jadavpur University
Kolkata India

ABSTRACT

This paper proposes a scheme for recognition of English characters based on features derived from partitioning the character image into non-overlapping cells. A dynamic sliding window moves over each cell and pixel counts obtained from the image portion within the boundaries of the window, contribute towards generation of the feature vector. A total of four passes of the window over the image each with a different window size leads to the generation of a 30-element feature vector. A neural network (multi-layered perceptron) is used for classifying the 26 alphabets of the English language. Accuracies obtained are demonstrated to have been improved upon with respect to contemporary works.

General Terms

Character recognition, Image Processing, Computer Vision.

Keywords

Dynamic sliding window, neural network, multi-layered perceptron, feature-vector.

1. INTRODUCTION

Character Recognition is a field of research in pattern recognition, artificial intelligence and machine vision which refers to the mechanical or electronic translation of images of handwritten or printed text into machine-editable text. Character recognition has attracted voluminous research in recent times and received extensive attention in academic and production fields as it allows conversion of a large number of printed materials to be stored within a digital computer in an compact, editable and searchable form. Many algorithms have been proposed by researchers to efficiently recognize printed characters primarily using their shapes, styles, sizes, orientations etc. however because of the difference in writing styles and variety of fonts available today, it still remains a challenging task to do so in a completely error free manner.

This paper proposes a technique for recognizing English alphabets based on a 30-element feature vector derived by using a dynamic sliding window. The organization of the paper is as follows: section 2 provides an overview of related work, section 3 outlines the proposed approach with discussions on overview, feature computation and classification schemes, section 4 provides details of the dataset and experimental results obtained and section 5 provides the overall conclusion and the scope for future research.

2. PREVIOUS WORK

A number of methodologies 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 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 [3] to recognize Bangla handwritten characters. In [4] the authors have decomposed characters into a set of structural shape units and used dynamic time warping based classifiers to identify component shapes in a character. Structural units called strokes have been used in [5] to identify handwritten Bengali characters using a Hidden Markov Model classifier. A histogram of chain code direction of contour points represented using a 64-dimensional feature vector have been utilized in [6] to recognize characters from 6 popular Indian scripts. Wavelet Energy Density Features derived from the DB4 wavelet have been used in [7] to identify numerals 0 to 9 using a 252-element vector. In [8] 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 [9] to identify handwritten numerals. In [10] a 110-element direction code representing structural shape units have been utilized for recognition of handwritten characters. 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. A region growing technique in conjunction with a neural network has been used for letter detection. Hidden Markov Models (HMM) and Dynamic Bayesian Network (DBN) classifiers have been employed in [14] to recognize handwritten Arabic words. In [15] the authors have proposed a novel method to identify script lines in handwritten whiteboard notes by assigning the sample points of the script trajectory using Viterbi Algorithm. A “critical region analysis” technique based on the outputs of the Fisher’s Discriminant, have been used to identify handwritten Chinese characters. In [17] 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. In [18] the authors have proposed twelve directional features based upon gradients of pixels and employed neural networks for classification of handwritten characters.

3. PROPOSED APPROACH

This paper proposes a technique for recognizing the 26 alphabets of English language. Before the recognition process, characters are pre-processed by conversion into binary images (black and white) and normalized into an $N \times N$ matrix. A thinning algorithm is then used to reduce width of line segments to single pixel width. The recognition process involves a kernel window which slides over the character image and pixel values just below the window are summed up at each step. The window is “dynamic” in the sense that (a) its location changes (b) its size changes. The window starts from the top-left corner of the image and is moved from left to right and top to bottom until it covers the entire image area. At each non-overlapping location the sum of the pixel values of the image within the boundaries of the window, are stored in a vector. The size of the window is chosen to be a sub-multiple of the size of the character image. The process is repeated with four different window sizes : N , $N/2$, $N/3$ and $N/4$. All the summation values are finally inserted into a composite feature vector. The proposed approach is summarized as follows :

- The characters are scanned and each character image is resized to a standard dimension of 48 by 48 pixels.
- The image is converted to binary form by choosing an appropriate threshold.
- Using a thinning algorithm the lines in the image are reduced to a single pixel width.
- A 12×12 pixel window is superimposed on the image at the top left corner and all pixel values of the image within the boundaries of the window are summed up.
- The window is then shifted towards the right to the next non-overlapping position and again the 12×12 pixel values of the image within the window, is counted.
- This process is repeated 16 times as the window moves from left to right and top to down until all the pixels of the image have been counted.
- The sum of pixel values at 16 different window positions gives rise to a 16-element vector V_{16} for characterizing each character image. This constitutes the first pass.
- In the second pass, the window is resized to 16×16 and the process is repeated starting from the top-left corner.
- The window is shifted to 9 different positions over the image area and sum of pixel values at these positions gives rise to a 9-element vector V_9 .
- In the third pass, the window is resized to 24×24 and shifted to 4 different positions over the image, giving rise to a 4-element vector V_4 .
- In the fourth pass, the window is resized to 48×48 and a single position over the image gives rise to a single element vector V_1 .
- All these values are subsequently concatenated to generate a 30-element vector which is used as the feature vector for character recognition i.e. $V = V_{16} + V_9 + V_4 + V_1$.

The feature vector generation process is illustrated in Fig. 1.



Fig 1: Feature vector generation using dynamic windows

Classification of the characters is done by a collection of features obtained from 25 different instances of the character obtained during a training phase. The features are then fed to a neural network classifier using feed-forward back-propagation architecture (MLP : multi-layer perceptron) to compute class probability of 10 instances of test images for each character. The procedure is repeated for all the 26 characters of the English alphabet and the overall accuracy of the recognition system is determined.

4. EXPERIMENTATIONS & RESULTS

4.1 Dataset

The dataset consists of 910 images of lower-case English alphabets of various appearances divided into training and testing sets. The training set consists of 25 different instances of each of the 26 English alphabets, a total of 650 images. The training set is indicated by legends AT, BT, CT, ..., ZT. 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 AS, BS, CS, ..., ZS. Samples of the characters used are shown below.



Fig 2: Samples of characters in the dataset

4.2 Training Phase

The training phase consists of computing the 30-element feature vectors from each of the 650 images of the training set, using the sliding window algorithm described in section 3. The feature plots for the training set, is shown below. The legend ‘T’ denotes the Training set. Fig. 3 indicates the variation of the mean values of the first 16 elements of the feature vector over all the 25 instances of each character, shown for the first 12 characters.

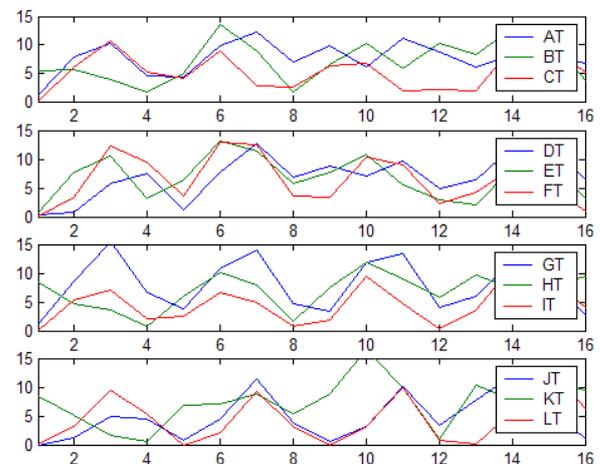


Fig 3: Mean values of elements 1 – 16 of feature vector of training set

Fig. 4 indicates the variation in mean values of the next 9 elements of the feature vector (i.e. elements 17-25) over all the 25 instances of each character, shown for the first 12 characters.

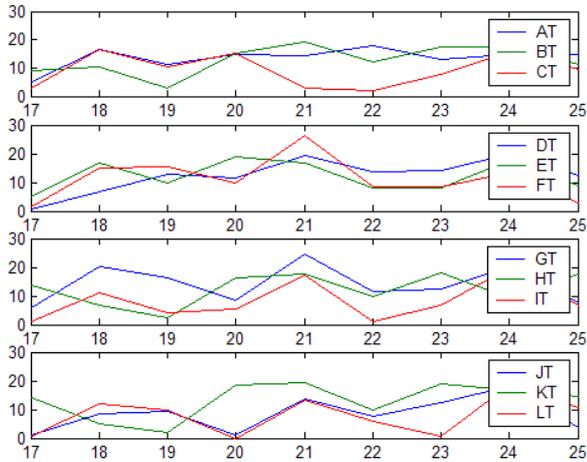


Fig 4: Mean values of elements 17 – 25 of feature vector of training set

Fig. 5 indicates the variation in mean values of the next 4 elements of the feature vector (i.e. elements 26-29) over all the 25 instances of each character, shown for the first 12 characters.

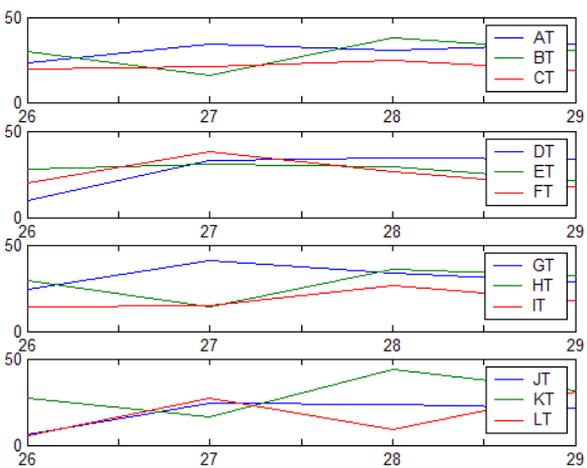


Fig 5: Mean values of elements 26 – 29 of feature vector of training set

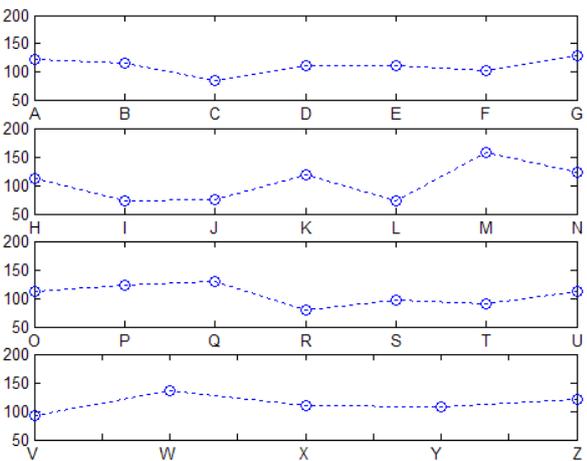


Fig 6: Mean values of element 30 of feature vector of training set

Fig. 6 indicates the variation in mean values of the last element of the feature vector (i.e. element 30) over all the 25 instances of each character, shown for all the 26 characters.

4.3 Testing Phase

The testing phase consists of computing the 30-element feature vectors from each of the 260 images of the testing set, using the sliding window algorithm described in section 3. The feature plots for the testing set, is shown below. The legend ‘S’ denotes the Testing set. Fig. 7 indicates the variation of the mean values of the first 16 elements of the feature vector over all the 10 instances of each character, shown for the first 12 characters.

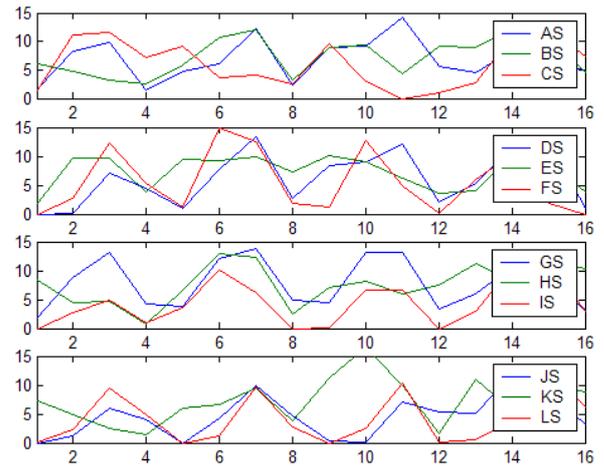


Fig 7: Mean values of elements 1 – 16 of feature vector of testing set

Fig. 8 indicates the variation in mean values of the next 9 elements of the feature vector (i.e. elements 17-25) over all the 10 instances of each character, shown for the first 12 characters.

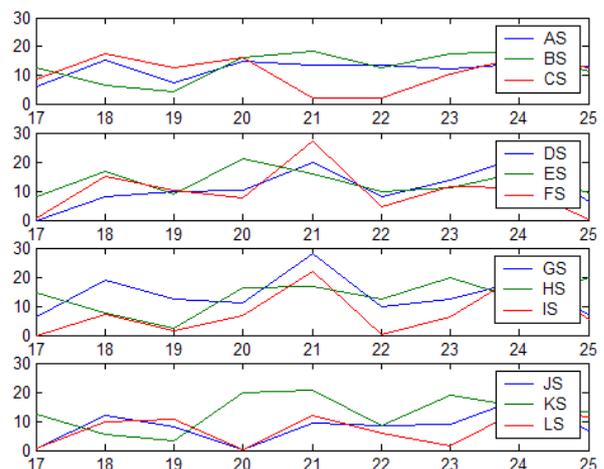


Fig 8: Mean values of elements 17 – 25 of feature vector of testing set

Fig. 9 indicates the variation in mean values of the next 4 elements of the feature vector (i.e. elements 26-29) over all the 10 instances of each character, shown for the first 12 characters.

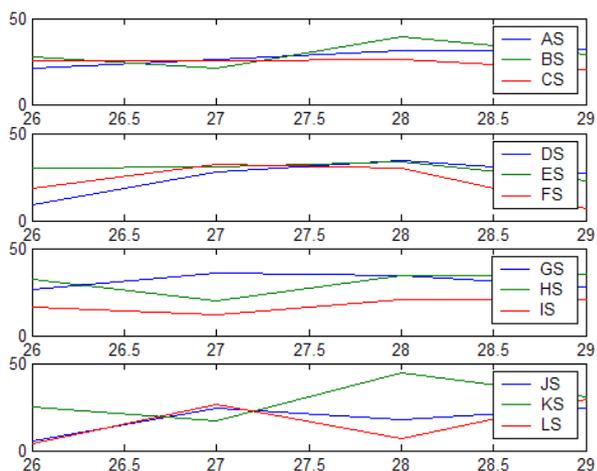


Fig 9: Mean values of elements 26 – 29 of feature vector of testing set

Fig. 10 indicates the variation in mean values of the last element of the feature vector (i.e. element 30) over all the 10 instances of each character, shown for all the 26 characters.

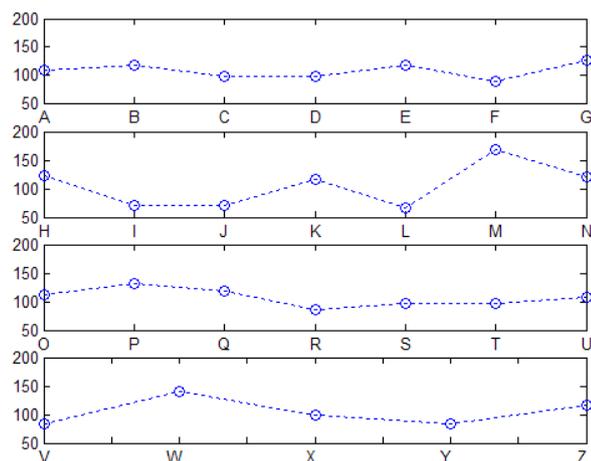


Fig 10: Mean values of element 30 of feature vector of testing set

4.4 CLASSIFICATION

Classification is done using a neural network (NN) (MLP : multi-layer perceptron). The MLP consists of 30 inputs for feeding in the 30-element feature vector for each character, and 26 outputs for discriminating between the characters. The input is fed in groups of 25 for instances of each character. The activation transfer functions are of log-sigmoid type. The best overall accuracy of 83.84% was achieved with 180 units in the hidden layer. Table 1 below reports accuracy rates obtained.

Table 1. Percentage Recognition Accuracies

A	B	C	D	E	F	G
80	80	80	100	100	90	40
H	I	J	K	L	M	N
50	50	90	90	80	100	90
O	P	Q	R	S	T	U
100	100	90	90	70	90	60
V	W	X	Y	Z	Overall	
70	100	90	100	100	83.84	

The MSE (mean square error) obtained after 150000 epochs was around 0.007. The NN convergence plot is shown in Fig. 11.

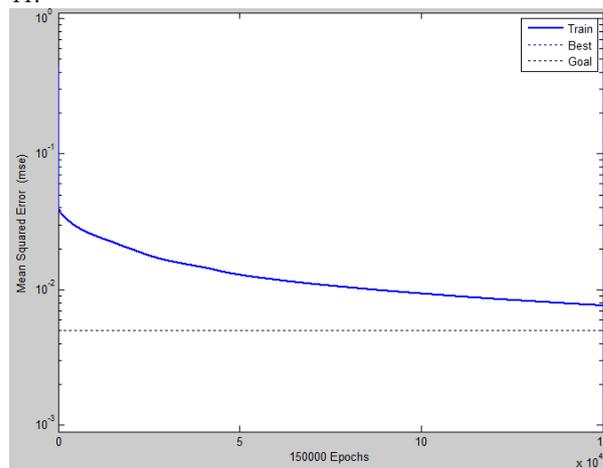


Fig 11: NN convergence plot

The NN output plots are shown in Fig. 12 depicting the recognition probabilities of the characters of the testing set.

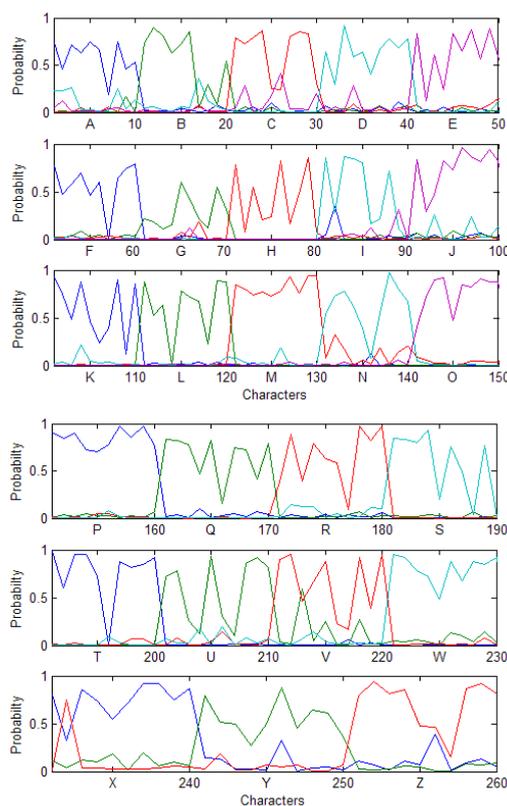


Fig 12: NN output plots

The testing set instances were ordered as follows : 1-10 (A), 11-20 (B), 21-30 (C), 31-40 (D), 41-50 (E), 51-60 (F), 61-70 (G), 71-80 (H), 81-90 (I), 91-100 (J), 101-110 (K), 111-120 (L), 121-130 (M), 131-140 (N), 141-150 (O), 151-160 (P), 161-170 (Q), 171-180 (R), 181-190 (S), 191-200 (T), 201-210 (U), 211-220 (V), 221-230 (W), 231-240 (X), 241-250 (Y), 251-260 (Z).

4.5 ANALYSIS

One of objectives and contributions of this work is to improve upon the accuracy obtained using the 13-element vector reported in [1]. The feature plots corresponding to the work in [1] applied to the first 12 characters of the current dataset is shown below. Fig. 13 shows the mean values of the 13-element feature vector applied to the current training set.

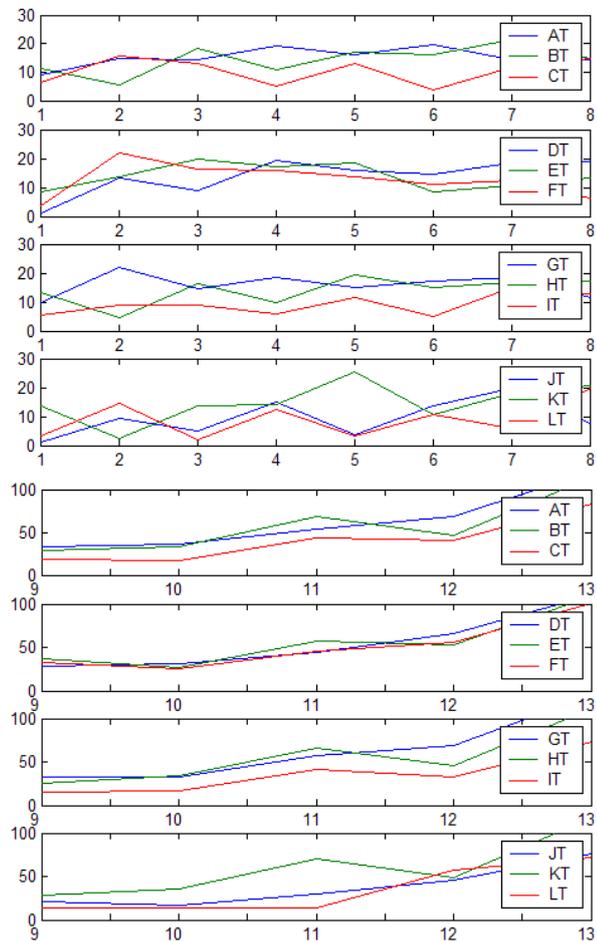


Fig 13: Mean values of elements 1 – 13 of feature vector of article [1] applied to current training set

Fig. 14 shows the mean values of the 13-element feature vector applied to the current testing set. The training and testing vectors are fed to a 13-180-26 neural net in groups of 25 instances for training and in groups of 10 instances for testing and the best accuracy obtained was 58.07%. The accuracy rates for individual characters are depicted in Table 2.

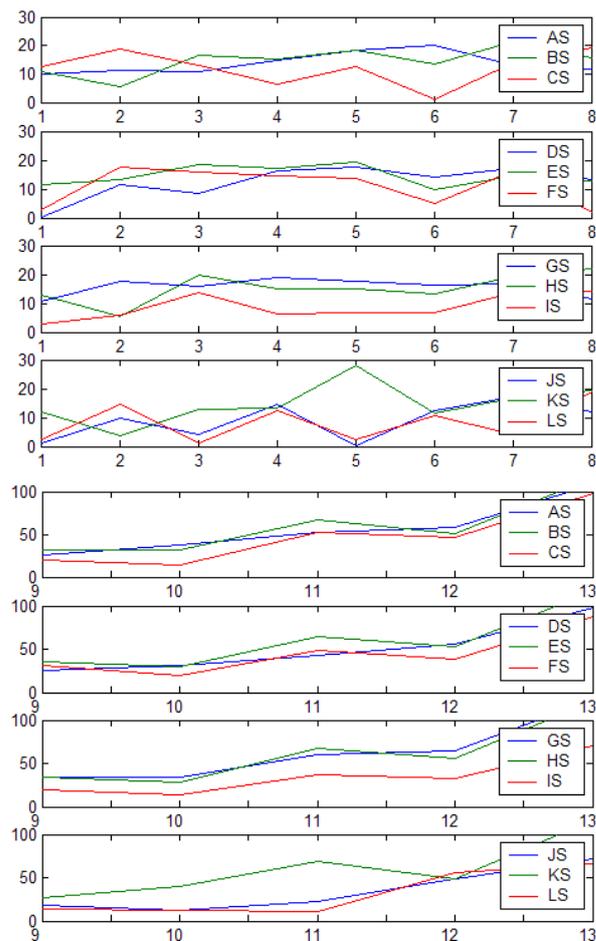


Fig 14: Mean values of elements 1 – 13 of feature set of article [1] applied to current testing set

Table 2. Percentage Recognition Accuracies using 13-element feature vector of article [1]

A	B	C	D	E	F	G
10	70	70	90	60	100	60
H	I	J	K	L	M	N
30	30	90	80	70	80	30
O	P	Q	R	S	T	U
10	100	70	70	0	90	20
V	W	X	Y	Z	Overall	
80	70	0	50	80	58.07	

The NN output plots are shown in Fig. 15 depicting the recognition probabilities of the characters of the testing set. The results indicate a substantial improvement achieved in the current work compared to that reported in article [1] tested on the same dataset.

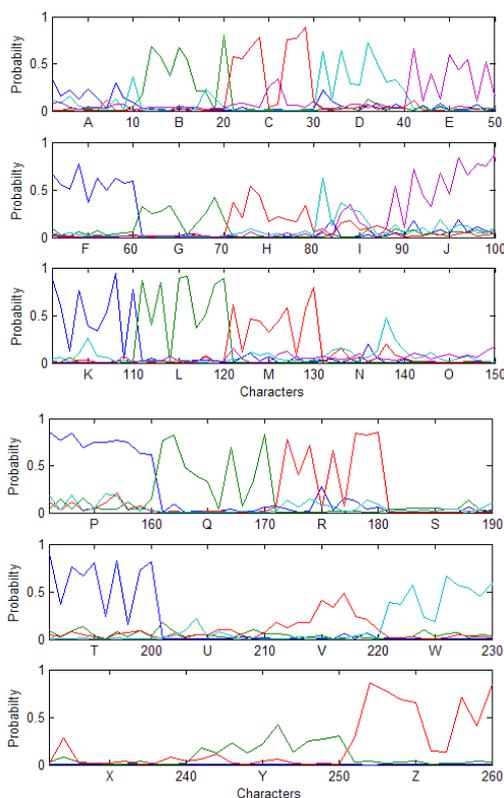


Fig 15: NN output plots for 13-element vector of [1]

5. CONCLUSIONS & FUTURE SCOPE

This paper outlines a scheme for English character recognition based on dynamic window or sliders using ANN. The accuracies are acceptable as compared to contemporary works. One of the main objectives of this work is to improve upon the results reported in [1] using a 13-element feature vector. The 30-element feature vector used here has been able to substantially improve upon the overall accuracy from 58% using the methodology in [1] to 83% for the current approach, when experimented using the same dataset. Other papers have reported higher accuracies e.g. 96.5% in [2], 95.5% in [3], 97.33% in [4], 87.7% in [5] etc. on their respective datasets, however for a meaningful comparison between the techniques all of them need to be tested using the same dataset. Such comparison work is currently in progress. Efforts are also being made to improve upon the current accuracies obtained here by using different coefficients of the sliding window as well as incorporating features obtained using shape recognition techniques like Moment Invariants model and Centroid-Radii model into the feature vector.

REFERENCES

[1] C. Zhong, Y. Ding, J. Fu. 2010. 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), 414-417.
 [2] A. R. Md. Forkan, S. Saha, Md. M. Rahman, Md. A. Sattar. 2007. Recognition of Conjective Bangla Characters by Artificial Neural Network. In Proceedings of International Conference on Information and Communication Technology (ICICT), 96-99.
 [3] B.B. Chaudhuri and A. Majumdar. 2007. 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).

[4] A. Bandyopadhyay, B. Chakraborty. 2009. Development of Online Handwriting Recognition System : A Case Study with Handwritten Bangla Character. In Proceedings of World Congress on Nature & Biologically Inspired Computing (NaBIC), 514-519.
 [5] S. K. Parui, K. Guin, U. Bhattacharya, B. B. Chaudhuri, 2008. Online Handwritten Bangla Character Recognition Using HMM. In Proceedings of 19th International Conference on Pattern Recognition (ICPR).
 [6] U. Pal, T. Wakabayashi, N. Sharma and F. Kimura. 2007. Handwritten Numeral Recognition of Six Popular Indian Scripts, In Proceedings 9th International Conference on Document Analysis and Recognition (ICDAR), 749-753.
 [7] M. Li, C. Wang, R. Dai. 2008. Unconstrained Handwritten Character Recognition Based on WEDF and Multilayer Neural Network. In Proceedings of the 7th World Congress on Intelligent Control and Automation, 1143-1148.
 [8] U. Pal, T. Wakabayashi and F. Kimura. 2007. Handwritten Bangla Compound Character Recognition using Gradient Feature. In Proceedings of 10th International Conference on Information Technology (ICIT), 208-213.
 [9] Md. M. Hoque, Md. M. Islam, Md. M. Ali. 2006, An Efficient Fuzzy Method for Bangla Handwritten Numerals Recognition. In Proceedings of 4th International Conference on Electrical and Computer Engineering (ICECE), 197-200.
 [10] U. Bhattacharya, B. K. Gupta and S. K. Parui. 2007. Direction Code Based Features for Recognition of Online Handwritten Characters of Bangla In Proceedings of Ninth International Conference on Document Analysis and Recognition (ICDAR).
 [11] G. Vamvakas, B. Gatos, S. J. Perantonis. 2010. Handwritten character recognition through two-stage foreground sub-sampling. Pattern Recognition, 2807-2816.
 [12] A. A. Desai. 2010. Gujarati handwritten numeral optical character reorganization through neural network. Pattern Recognition, 2582-2589.
 [13] K. Saeed, M. Albakoor. 2009. Region growing based segmentation algorithm for typewritten and handwritten text recognition. Applied Soft Computing, 608-617.
 [14] J. H. Al Khateeb, O. Pauplin, J. Ren, J. Jiang. 2011. Performance of hidden Markov model and dynamic Bayesian network classifiers on handwritten Arabic word recognition. Knowledge-Based Systems, 680-688.
 [15] J. Schenk, J. Lenz, G. Rigoll. 2009. Novel script line identification method for script normalization and feature extraction in on-line handwritten whiteboard note recognition. Pattern Recognition, 3383-3393.
 [16] K. C. Leung, C.H. Leung. 2010. Recognition of handwritten Chinese characters by critical region analysis. Pattern Recognition, 949-961.
 [17] U. Pal, P. P. Roy, N. Tripathy, J. Lladós. 2010. Multi-oriented Bangla and Devnagari text recognition. Pattern Recognition, 4124-4136.
 [18] D. Singh, S. K. Singh, M. Dutta. 2010. Handwritten Character Recognition using Twelve Directional Feature Input and Neural Network. International Journal of Computer Application, 82-85.