Handwritten Devnagari Numeral Recognition using SVM & ANN

Sandhya Arora1. Debotosh Bhattacharjee2, Mita Nasipuri2, M. Kundu2 and D. K. Basu3 and L.Malik4

[1] Dept. of CSE & IT, Meghnad Saha Institute of Technology
kolkata, 700150,India
[1]*sandhyabhagat@yahoo.com*
[2]Department of Computer Science and Engineering, Jadavpur University
Kolkata, 700032,India
[3]AICTE Emeritus Fellow, Department of Computer Science and Engineering, Jadavpur University
Kolkata, 700032,India
[4]Dept. of Computer science. G.H. Raisoni college of Engineering
Nagpur, India

Abstract: This paper proposes a system for recognizing offline Handwritten Devnagari numerals using support vector machine and artificial neural networks. The proposed system classifies numeral, in two stages. Various preprocessing operations are performed on the digitized image to enhance the quality of the image. It involves image acquisition and numeral image extraction, binarization, scaling, thinning, smoothing and noise removal. Feature extraction where some statistical and structural features, such as - shadow based features, zone based directional features, zone based centroid features and view based features, are extracted after preprocessing. Finally, the classification phase takes place in two stages. In first stage, numerals are classified using MLP. Unrecognized numerals of first stage, are then classified in second stage by SVM using one-against-all technique to classify 10 handwritten devnagari numeral shapes. The proposed system has been tested on 18300 data samples. The system has achieved nearly 93.15% recognition rate.

1. Introduction

In Optical Character Recognition [OCR], a character/numeral which has to recognize can be machine printed or handwritten. There is extensive work in the field of handwriting recognition, and a number of reviews exist. Handwritten numeral recognition is an exigent task due to the restricted shape variant, unusual script style & different kind of noise that breaks the strokes in number or changes their topology. Recognize of is gaining wider importance today and is one of the benchmark problem in document analysis. As handwriting varies when person write a same character twice, one can expect enormous dissimilarity among people. These are the reason that made researchers to find techniques that will improve the knack of computers to characterize and recognize handwritten numerals.

Recognizing handwritten numerals is an important area of research because of its various application potentials. Automating bank cheque processing, postal mail sorting, job application form sorting, automatic scoring of tests containing multiple choice questions and other applications where numeral recognition is necessary. Some research has been done on the recognition of Roman, Arabic and Chinese numerals which is excellently reviewed in [1]. Le Cun et al [2] have developed an algorithm for identifying Arabic numerals with a high recognition rate.Few works is available for Devnagari numeral recognition using Neural Networks but none for SVM. The first research report on handwritten Devnagari characters and numerals was published in 1977 [3] but not much research work has been done after that. Hanmandlu and Murthy [4] proposed a Fuzzy model based recognition of handwritten Hindi numerals and they obtained 92.67%accuracy. Bajaj et al [5] employed three different kinds of features namely, density features, moment features and descriptive component features for classification of Devnagari Numerals. They proposed multi-classifier connectionist architecture for increasing the recognition reliability and they obtained 89.6% accuracy. Bhattacharaya et al. [6] proposed a Multi-Layer Perceptron (MLP) neural network based classification approach for the recognition of Devnagari handwritten numerals and obtained 91.28% recognition accuracy. An excellent survey of the area is given in [7].

In this paper, a system for off-line recognition of handwritten Devnagari numerals is proposed using ANN and SVM classifiers. There have been several attempts for OCR of Indian printed characters but very few of these are for recognition of handwritten numerals, and none using SVM. We applied ANN and SVM at different stages of classification. In first stage, the numeral image is preprocessed (section 3) and four features namely: shadow based features, zone based directional features, zone based centriod features and view based features are extracted (section 4). These features are then fed to MLP's (section 5), designed separately for all four features for recognition. Results of four MLP's are combined using weighted majority scheme. Numerals not classified by MLP, in first stage are classified using SVM (section 5), in second stage. The section 6 provides discussion regarding results and conclusion is summarized in section 7.

2. Challenges in Handwritten Devnagari Numeral Recognition

Devnagari is the most popular script in India. Hindi is written in Devnagari script. Nepali, Sanskrit and Marathi are also written in Devnagari script. Further, Hindi is the national language of India and Hindi is the third most popular language in the world. According to a recent survey English is being used by 125.3 million people, Bengali is used by 91.1 million people, Telugu and Marathai is being used by 85 million and 84.2 million people in India. Thus, work on recognition of handwritten Devnagari numerals is of considerable practical importance. Because of the writing styles of different individuals, numerals can have different shapes. Handwritten Devnagari numeral recognition is an exigent task due to the restricted shape variant, unusual script style & different kind of noise that breaks the strokes. As a result recognition of handwritten numerals becomes a difficult task. Some sample handwritten Devnagari numerals are shown in Figure 1 to give some idea of the vast disparity in writing styles for different characters. Figure 2 shows some of the handwritten samples (phone numbers) written by four different writers.

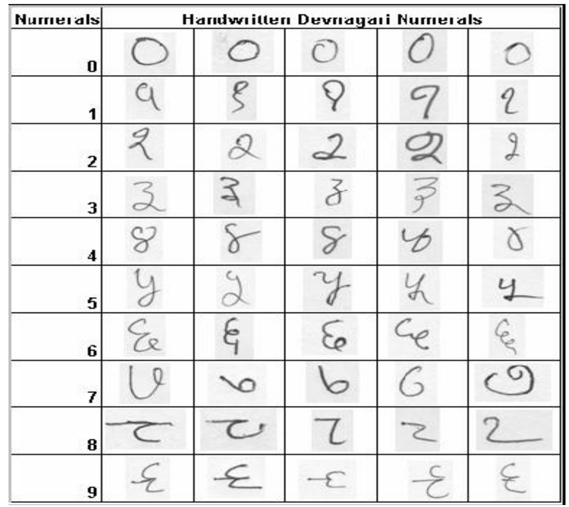


Figure 1. Examples of Handwritten Devnagari Numerals

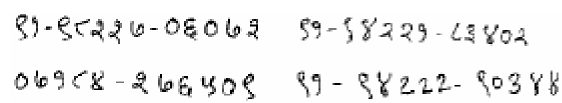


Figure 2. Numeral Samples

2. Process overview

The character recognition system is usually validated by running them on independent test sets, on which the systems have not been trained. For these tests to be conclusive, the set should include a fairly large number of samples to reflect the variety of writing styles that are found in real-life applications. The task of the recognition of handwritten numerals has been broken down into the following steps:-

(i) binarization of sample image;
(ii) thinning of the image;
(iii) smoothing;
(iv) normalization of the image to a standard size;
(v) feature extraction;
(vi) recognition.

To enable recognition, steps (i)–(iv) are applied on a training set of all 10 numerals as part of the pre-processing. While performing feature extraction, simultaneously the Knowledge Base of reference features is created, discussed in section 4.

3. Preprocessing

The preprocessing steps remove any noise, distortions in the input character and convert the character in a form processed and recognizable by the system. The preprocessing steps are performed is for improving the quality of images for ensuring better quality in the subsequent processing of image. In the scanning process, some distortion in images may be introduced due to pen quality, light hand handwriting, poor quality of the paper on which the numerals are written etc. The preprocessing steps performed in this work are consists of the following:-

3.1. Binarization

Frequently, binarization is carried out before the character recognition phase. Ideally an input character should have two tones, i.e., black and white pixels (commonly represented by 1 and 0, respectively). Image binarization converts an image of up to 256 gray levels into a two-tone image. The goal of this step is to identify a threshold value dynamically that would help to distinguish between image pixels that belong to text and those that belong to the background. The threshold value identified would be completely dependent on the nature and the properties of the documents, the contrast level, the difference of contrast between the foreground and background pixels and their concentration in the document. The methodology would be applied to gray-level image (range of pixel values 0 to 255).

1. Take the threshold to be 128 (midway between 0 and 255).
2. Take all the pixels with grayscale values above 128 as background and all those with values below 128 as foreground.
3. Find the mean grayscale value of background pixels as bMean and that of all fore ground pixels as fMean.
4. Find the average of bMean and fMean and make this the new threshold.
5. Go back to step 2 and continue this process of refining the threshold till the change of the threshold from one iteration to next becomes less than 2% of the range of 0 to 255.

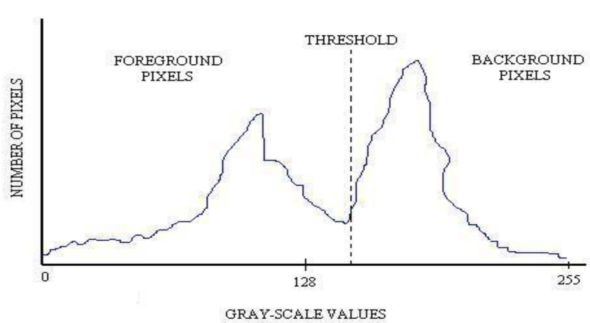


Figure 3. Finding the threshold

3.2. Thinning and smoothing

A two-tone digitized image is defined by a matrix A, whose element $a_{i,j}$ is either 1 if character is present or 0 otherwise. Iterative transformations are applied on A to obtain the thinned character. We have used a thinning algorithm of Ref. [8,17], by which a point at a binary pattern consists of successive deletion of dark points (i.e. changing them to white points) along the edges of the pattern until it is thinned to a line. This results some redundant pixels. To remove this redundancy we have applied certain masks [17], which gives one pixel-wide skeleton.

3.3 Size Normalization

Normalization is thus the process of equating the size of all numerals in order to extract features on the same footing. To achieve this, we use standard Affine Transformation to perform a linear mapping from 2D coordinates to other 2D coordinates that preserves the “straightness” and ‘parallelness” of lines. Affine transformation can be constructed using sequences of translations, scales, flips, rotations and shears. Image is scaled in 100x100 pixel resolution.

4. Feature Extraction

In the following, we discuss the extracted features for classifiers. These features are used in MLP and SVM classifiers discussed in section 5.

4.1 Shadow Features

Shadow is basically the length of the projection on the sides. For computing shadow features [10, 11] on scaled binary image, the rectangular boundary enclosing the numeral image is divided into eight octants. For each octant shadows or projections of numeral image segment on three sides of the octant dividing triangles are computed so, a total of 24 shadow features are obtained. Each of these features is divided by the length of the corresponding side of the triangle to get a normalized value.

4.2 Zone based directional features of Character Contour

Chain code provides the points in relative position to one another, independent of the coordinate system. In this methodology, we first find the contour points of the scaled image, and then direction chain coding of connecting neighboring contour pixels, and the outline coding are captured [13]. If cf denote a contour chain using Freeman codes dk, such that, Cf = d1 d2 d3……dk…dn

Where dk € {0,1,2,3,4,5,6,7} , and n is the length of chain.

We divide the contour image in 5 × 5 blocks. In each of these blocks, the frequency of the direction code is computed and a histogram of chain code is prepared for each block. Thus for 5 × 5 blocks we get 5 × 5 × 8 = 200 features for recognition.

4.3 View based features

This method is based on the fact, that for correct character-recognition a human usually needs only partial information about it – its shape and contour. This feature extraction method [11], examines four "views" of character. The view is a set of points that plot one of four projections of the object (top, bottom, left and right). In the considered examples, eleven uniformly distributed characteristic points are taken for each view. These quantities are normalized so that their values are in the range <0, 1>. Now, from 44 obtained values the feature vector is created to describe the given numeral, and which is the base for further analysis and classification.

4.4 Zone based Centroid Features

For extracting the feature, the zone-based hybrid approach is proposed. The most important aspect of the handwriting recognition scheme is the selection of a good feature set, which is reasonably invariant with respect to shape variations caused by various writing styles. The major advantage of this approach stems from its robustness to small variations ease of implementation. The zone-based feature extraction method gives good results even when certain preprocessing steps like filtering; smoothing and slant removing are not considered.

Definition 1. Centroid Feature Vector: The numeral image centroid is computed and the numeral image (100x100) is divided into 25 equal zones (20x20). Zone centroid is computed. This procedure is sequentially repeated for the entire zone present in the numeral image (50 features). There could be some zones having empty foreground pixels. Hence feature value of such zone in the feature vector is zero. Distance of zone centroids with image centroid is also calculated.

By the definition of centroid, the centroid of the image can be calculated as follows:

$$\begin{cases} x_c = \dfrac{\sum_{(x,y)\in p} xI(x,y)}{\sum_{(x,y)\in p} I(x,y)} \\ y_c = \dfrac{\sum_{(x,y)\in p} yI(x,y)}{\sum_{(x,y)\in p} I(x,y)} \end{cases}$$

where xc, yc are called as the coordinates of X-axis and Y-axis of numeral image p. A feature θ = (θ1, θ2, · · · , θ 100) is formed.

Input: Scaled and Thinned, Binarized Handwritten Numeral Image
Output: Extracted Features for Classification

Method Begins
Step1: Calculate the numeral image centroid xc, yc

Step2: Divide the image into 25 equal zones.

Step3: For each zone calculate the zone centroid xi,yi where i=1,2…25(50 features)

Step4: Calculate the distance between zone centroid and image centroid as xc-xi and yc-yi (50 features)

Step5: Finally 100 features are extracted for classification and numeral recognition.

5. Evaluated classifiers

We classified numeral images in two stages. In first stage classification, above discussed features are fed to MLP's, designed separately for all four features for recognition. Results of four MLP's are combined using weighted majority scheme[12]. Numerals not classified by MLP, in first stage are classified using SVM, in second stage.

5.1 Neural Network

We used the same MLP with 3 layers including one hidden layer for four different feature sets consisting of 24 shadow features, 100 zone based centroid features, 200 zone based directional features and 44 view based features. The experimental results obtained while using these features for recognition of handwritten Devnagari numerals is presented in the next section. At this stage all numerals are non-compound, single numeral so no segmentation is required.

The classifier is trained with standard Backpropagation [9]. It minimizes the sum of squared errors for the training samples by conducting a gradient descent search in the weight space. As activation function we used sigmoid function. Learning rate and momentum term are set to 0.8 and 0.7 respectively. As activation function we used the sigmoid function. Numbers of neurons in input layer of MLPs are 24, 44, 200 or 100, for shadow features, view based features, zone based directional features and zone based centroid features respectively. Number of neurons in Hidden layer is not fixed, we experimented on the values between 20-70 to get optimal result and finally it was set to 30, 40, 30 and 70 for shadow features, zone based centroid features, view based features and zone based directional features respectively. The output layer contained one node for each class., so the number of neurons in output layer is 10.

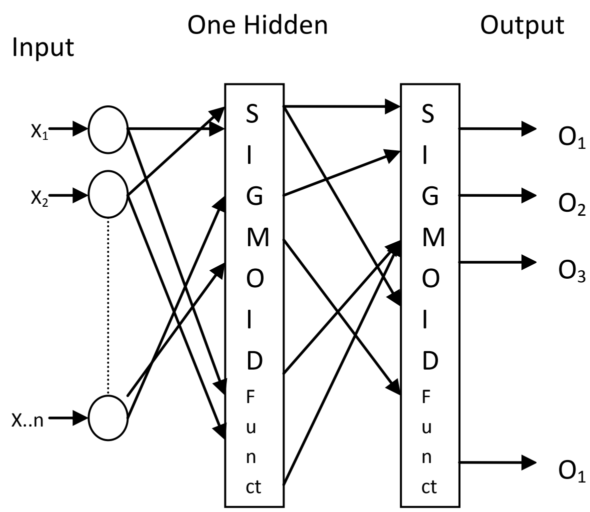


Figure 4. Block diagram of MLP

Outputs from several classifiers can be combined to produce a more accurate result. We have four similar Neural networks classifiers as discussed above, which are trained on 24 shadow features, 200 zone based directional based features, 44 view based features and 100 zone centroid based features respectively. The outputs are confidences associated with each class. As these outputs cannot be compared directly, we used an aggregation function for combining the results of all four classifiers. Our strategy is based on weighted majority voting scheme [12].

5.2 Support Vector Machines

The objective of any machine capable of learning is to achieve good generalization performance, given a finite amount of training data, by striking a balance between the goodness of fit attained on a given training dataset and the ability of the machine to achieve error-free recognition on other datasets. With this concept as the basis, support vector machines have proved to achieve good generalization performance with no prior knowledge of the data.

Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. They belong to a family of generalized linear classifiers. A special property of SVMs is that they simultaneously minimize the empirical classification error and maximize the geometric margin; hence they are also known as maximum margin classifiers. Viewing the input data as two sets of vectors in an n dimensional space, an SVM will construct a separating hyper plane in that space, one which maximizes the "margin" between the two data sets. To calculate the margin, construct two parallel hyper planes, one on each side of the separating one, which are "pushed up against" the two data sets. Intuitively, a good separation is achieved by the hyper plane that has the largest distance to the neighboring data points of both classes. The hope is that, the larger the margin or distance between these parallel hyper planes, the better the generalization error of the classifier will be. Consider training data set {(x1,c1),(x2.c2)…(xn,cn)} where the ci is either 1 or −1, indicating the class to which the point belongs. Each is a p-dimensional real vector. We want to give the maximal-margin hyper plane which divides the points having ci = 1 from those having ci = − 1. Any hyper plane can be written as the set of points satisfying Maximum-margin hyper plane and margins for a SVM trained with samples from two classes. Samples on the margin are called the support vectors. Linear classifier is shown on the Figure 5.

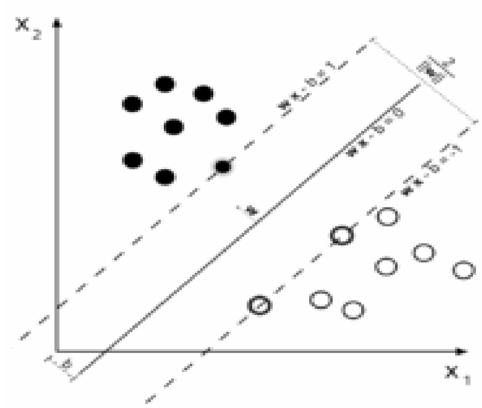


Figure 5. SVM with Linear Classifier

The points x lies on the hyper plane satisfy w.x + b=0, the vector w is a normal to the hyper plane. |b|/ ||w|| is a perpendicular distance from hyper plane to the origin as shown in Figure 9 and ||w|| is the Euclidean norm of w. The parameter b determines the offset of the hyper plane from the origin along the normal vector. Choose the w and b to maximize the margin, or distance between the parallel hyper planes that are as far apart as possible while still separating the data. These hyper planes can be described by the equations

$$X_i \cdot W + b \geq +1 \quad \text{for } C_i = +1$$

$$X_i \cdot W + b \leq -1 \quad \text{for } C_i = -1$$

Note that if the training data are linearly separable, select the two hyper planes of the margin in a way that there are no points between them and then try to maximize their distance. By using geometry, we find the distance between these two hyper planes is 2/|w|, so we want to minimize |w|. The optimal separating hyper plane can be determined without any computations in the higher dimensional feature space by using kernel functions in the input space. We used linear kernel K(x, y) = x.y, because they are simple and can be computed quickly. There is no kernel parameter choices needed to create a linear SVM, but it is necessary to choose a value for the soft margin in advance.

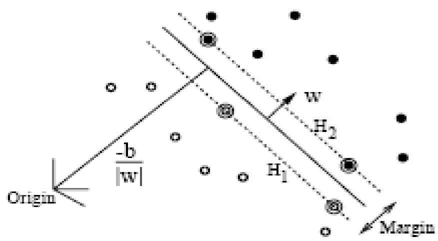


Figure 6. Linear separating hyper planes for the separable case

6. Performance Evaluation

The experiments of character recognition reported in the literature vary in many factors such as the sample data, pre-processing technique, feature representation, classifier structure and learning algorithm. Only a few works have applied different classification/learning methods for classification. We tested performance of Handwritten Devnagari Numerals using ANN and SVM applied at different stages. At first stage, numeral images are classified using four MLP's designed using four features (section 4). Results of different features using MLP's are given in Table 1. Results of four MLP's are combined using weighted majority scheme, illustrated in Table 2. Numerals not classified by MLP in first stage, are fed to SVM, in second stage for classification, which gives 96.29% accuracy. Results of both classifiers are combined to get higher accuracy. The overall accuracy achieved is 93.15%, by combining the 92.91% accuracy of MLP and 96.29% accuracy of SVM.

The experiment of Devnagari numerals dataset contains 18300 handwritten samples, 12810 samples in train dataset and 5490 samples for test results. The detailed recognition results of individual numerals are given in Table 4 and Table 5. In Table 4, results of MLP on individual numeral are given. Numerals not classified by MLP are fed to SVM. Table 5, gives the results of individual numerals classified by SVM in second stage, i.e. numerals not classified by MLP in first stage. Detail comparisons of results are given in Table 3.

Table 1. Results of ANN for different features

MLP	Input layer Neuron	Hidden Layer Neuron	Output Layer Neuron	Result
Shadow based features	24	30	10	85.19%
Zone based Centroid features	100	40	10	80.87%
View Based Features	44	30	10	86.87%
Zone based directional Feature	200	70	10	90.92%

Table 2: Accuracy of ANN in first stage and SVM in second stage

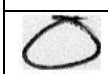
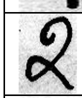
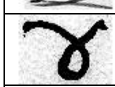
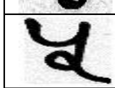
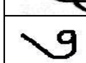
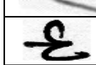
Classifier	**Accuracy**
ANN using weighted majority scheme	92.91%(top 1) 97.31% (top 2) 99.66% (top 3) 100% (top 4)
SVM	96.29%
Combined ANN (top 1) and SVM	93.15%

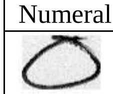
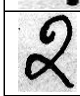
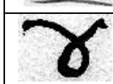
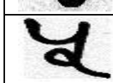
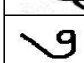
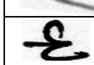
Table 3: Comparison of Results

Sl.No.	**Method Proposed by**	**Technique**	**Data set**	**Accuracy**
1	Hanmandlu and Ramana Murthy[4]	Fuzzy model based recognition using exponential membership function fitted to fuzzy sets derived from features consisting of normalized distances obtained using the Box approach.	350	95%
2	Bajaj al. [5]	density features, moment features and descriptive component features with multi-classifier connectionist architecture	2460	89.6%
3	Bhattacharaya et al. [6]	Multi-Layer Perceptron (MLP) neural network based classification approach on shape feature vector computed from certain directional-view-	22535	91.28%

		based strokes of numeral image		
4	C. V. Lakshmi, et al. [15]	Edge directions histograms and splines along with PCA	9800	94.25%
5.	R. J. Ramteke, S. C. Mehrotra[16]	central invariant moments as features with Gaussian Distribution Function for classification	2000	92%
6.	Our proposed method	Shadow, zone based directional,view based, zone based centroid features with MLP neural network and SVM based classification approach in two stages	18300	93.15%

Table 4: Individual numeral accuracy using ANN in first stage Table 5: Accuracy using SVM in second stage

Numeral	Accuracy
	96.96%
	96.96%
	72.72%
	87.87%
	96.96%
	87.87%
	96.96%
	93.93%
	87.87%
	93.93%

Numeral	Accuracy
	100%
	100%
	66.66%
	100%
	100%
	100%
	100%
	100%
	100%
	100%

7. Conclusion

The result obtained for recognition of Devnagari numerals show that reliable classification is possible using SVMs. We applied SVMs on different feature data namely Shadow based, Chain code Histogram, View based features and zone centroid based features. The SVM-based method described here for offline Devnagari numerals can be easily extended to other Indian scripts numerals also.

References

[1] R. Plamondon and S. N. Srihari, ‘‘On-line and off-line handwritten character recognition: A comprehensive survey’’, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 22, pp 62-84, 2000.

[2] Y. Le Cun, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel and D. Henderson, ‘‘Handwritten digit recognition with a back-propagation network’’, Advances in neural information processing systems, 1990, pp 396 to 404.

[3] I.K. Sethi and B. Chatterjee, ‘‘Machine Recognition of constrained Hand printed Devnagari’’, Pattern Recognition, Vol. 9, pp. 69-75, 1977.

[4] M. Hanmandlu and O.V. Ramana Murthy, ‘‘Fuzzy Model Based Recognition of Handwritten Numerals’’, Pattern Recognition, 40 (1840-1854), 2006.

[5] Reena Bajaj, Lipika Dey, and S. Chaudhury, ‘‘Devnagari numeral recognition by combining decision of multiple connectionist classifiers’’, Sadhana, Vol.27, part. 1, pp.-59-72, 2002.

[6] U. Bhattacharya, B. B. Chaudhuri, R. Ghosh and M. Ghosh, ‘‘On Recognition of Handwritten Devnagari Numerals’’, In Proc. of the Workshop on Learning Algorithms for Pattern Recognition (in conjunction with the 18th Australian Joint Conference on Artificial Intelligence), Sydney, pp.1-7, 2005.

[7] U. Pal and B.B. Chaudhuri, ‘‘Indian script character recognition: A Survey’’, Pattern Recognition, Vol. 37,pp. 1887-1899, 2004.

[8] M. Tellache, M. A. Sid-Ahmed, B. Abaza . Thinning algorithms for Arabic OCR. IEEE Pac Rim 1993. 248-251

[9] J. Hertz, A. Krogh, R.G. Palmer, ”An Introduction to neural Computation”, Addison-Wesley (1991)

[10] S. Basu, N.Das, R. Sarkar, M. Kundu, M. Nasipuri, D.K. Basu, “Handwritten Bangla alphabet recognition using MLP based classifier”, NCCPB, Bangladesh, 2005

[11] S. Arora, D. Bhattacharjee, M. Nasipuri , D.K. Basu , M.Kundu , “Study of Different Features on Handwritten Devnagari Character”,ICETET09, Nagpur ,India2009.

[12] S. Arora, D. Bhattacharjee, M. Nasipuri , D.K. Basu , M.Kundu ”Combining Multiple Feature Extraction Techniques for Handwritten Devnagari Character Recognition”, 2008 IEEE Region 10 Colloquium and

the Third ICIIS, IIT Kharagpur, INDIA December 8-10, 2008.

[13] S. Arora , D. Bhattacharjee, M. Nasipuri , D.K. Basu , M.Kundu," Application of Statistical Features in HandwrittenDevnagari Character Recognition ", International Journal of Recent Trends in Engineering, Vol 2, No. 2, November 2009,pp40-42

[14] S. Arora , D. Bhattacharjee, M. Nasipuri , D.K. Basu , M.Kundu ,"Recognition of Non-Compound Handwritten Devnagari Characters using a Combination of MLP and Minimum Edit Distance" , International Journal of Computer science and security vol 4 issue 1 :2010

[15] C. Vasantha Lakshmi, Ritu Jain, C.Patvardhan, "Handwritten Devnagari Numerals Recognition with higher accuracy", International Conference on Computational Intelligence and Multimedia Applications 2007

[16] R. J. Ramteke, S. C. Mehrotra,"Recognition of Handwritten Devnagari Numerals", International Journal of Computer Processing of Oriental Languages, *2008*

[17] S. Arora, D. Bhattacharjee, M. Nasipuri , L.Malik,"A Novel Approach for Handwritten Devanagari Character Recognition" in IEEE – International Conference on Signal And Image Processing, Hubli, Karnataka, Dec 7-9, 2006