Studies on some Soft Computing Techniques: A Case Study for Constrained Handwritten Devnagari Characters and Numerals

Sandhya Arora
Dept. of CSE, Cummins College of Engg for Women, Pune-411052
sandhya.arora02@gmail.com

Abstract :- Classifier plays a crucial role in handwritten text recognition system. The major challenge lies in taking the advantage of multiple classifiers by exploiting the strengths of a classifier and by suppressing its weakness by other classifiers. In this paper, some soft computing techniques like Multilayer perceptrons (MLPs), support vector machines (SVM) and method of minimum edit distance has been considered for classification of handwritten isolated Devnagari characters. After preprocessing the character image, eight types of features are extracted. SVM, MLP and its decision combination techniques are explored on eight types of features. Two approaches of two stage classification have also been applied. In first approach, preliminary grouping of characters has been done using structural properties of Devnagari characters. In second stage, MLP is designed for each group of characters using directional chain code features. In second approach, the character patterns are divided into two sets, certainty set and confusing character set. This division is based on a relative difference measure. After separating characters in two sets, a MLP based classifier is used in the first stage to classify characters of certainty set. And for classifying characters of confusing character set, method of minimum edit distance is applied in the second stage, on detected corners of the sample character using a modified form of Harris corner detector. A study of MLP and SVM has also been done for Handwritten Devnagari Numerals. The accuracy of 90.74% and 95.18% is achieved for characters and numerals respectively.

Keywords:- Feature extraction, minimum edit distance, modified harris corner detection algorithm, neural network, support vector machine, structural features, statistical features.

1. INTRODUCTION

In India, Devnagari is the most popular script and a very few researchers are attracted towards handwritten Devnagari document recognition. Devnagari is the main script used to write more than 16 languages which include Hindi, Marathi, and Nepali. Hindi is the national language of India, and it is the third most popular language in the world after Chinese and English. There are approximately 551 million people all over the world who speak or write in Hindi. Hence any study related to the design of recognition interfaces for handwritten Devnagari character sets would be highly useful to India.

While a large amount of literature is available for recognition of English script, relatively less work has been reported for the recognition of Indian languages. Some of the work reported is by Sinha & Mahabala[13], U. Pal [11,24,25] & Chaudhary [11], Hanmandlu & Murthy [9,10], Kumar & Singh[12], Sethi et.al.[5,8], Bansal [2], Bhattacharya [23]. Main reason for this slow development could be attributed to the complexity in the shapes of Indian scripts, and also the large set of different patterns that exists in these languages, as opposed to English. Most of the work reported above was on printed Devnagari characters. For handwritten Devnagari characters, accuracy reported is not high and dataset used are not large.

The paper is organized as follows. In section 2, peculiarities of Devnagari Script are discussed. Overall approach used, is discussed in section 3. Feature extraction techniques along with the preprocessing techniques are reported in section 4. Section 5, deals with the MLP classifiers design, its combination techniques and SVM used for recognition purpose. Section 6 gives an overview of two hybrid approaches as part of two stage classification schemes. Section 7, discusses the details of handwritten Devnagari numeral recognition using MLP and SVM. Section 8 contains a conclusion and a discussion of the future work.

2. PECULIARITIES OF DEVNAGARI SCRIPT

Devnagari script is different from Roman script in several ways. This script has two-dimensional compositions of symbols: core characters in the middle strip, optional modifiers above and/or below core characters shown in figure 1. Two characters may be in shadow of each other. While line segments (strokes) are the predominant features for English, most of the characters in Devnagari script is formed by curves, holes, and also strokes. In Devnagari scripts, the concept of upper-case, the lower-case characters is absent. However the alphabet itself contains more number of symbols than that of English. The basic set of symbols of Devnagari script consists of 36 consonants (or “vyanjan”) and 13 vowels (or “swar”). A sample set of basic Devnagari characters is given in figure 2.

ISSN 2229-5518

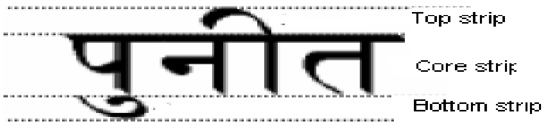


Figure 1: Three Strips of a Devnagari word

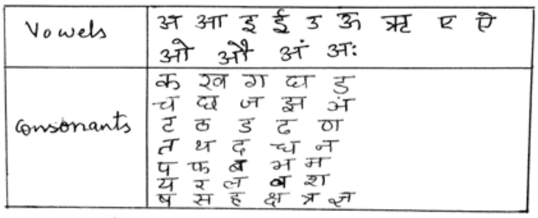
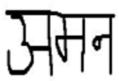


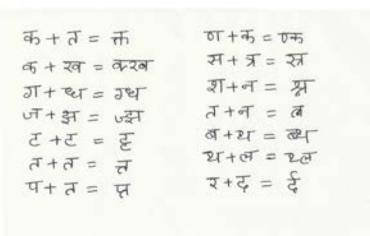
Figure 2: Vowels and Consonants of Devnagari script

An important feature of Devnagari script is the prominent headline or "shirorekha". Almost all the characters in Devnagari script are partially or fully covered by a horizontal line touching the character at its top. All the characters in a word get joined through their headlines, thus forming a common headline for the entire word as shown in figure 3(a). When a vowel following a consonant gets joined with the consonant, the vowel takes a different shape, often called a modifier or an allograph. These modifiers appear either on the top (above headline), or at the bottom, or on the right, or on the left or right extending above the headline covering the top of the joined consonant, which is explained in figure 3(b). When a consonant following another consonant gets joined with it, a compound shape is formed. In forming the compound shape, the first consonant sometimes takes a half form which touches the following consonants, which is described in figure 3(c).



(a)

(b)



(c)

Figure 3: (a) Devnagari word with "shirorekha" (b) Modifiers with their corresponding vowel and a sample character image of "ka" modified with modifier (c) A sample set of Devnagari compound characters

3. THE OVERALL METHODOLOGY

This work deals with the development of an efficient recognizer for constrained, handwritten, isolated, basic (vowels and consonants) characters and numerals using various soft-computing techniques.

Two different sources for data collection have been considered as no standard database is freely available on the web for handwritten Devnagari characters and numerals. First one is the dataset of handwritten Devnagari basic character set prepared by CVPR unit of Indian Statistical Institute (ISI), Kolkata, which contains 4900 samples. The second source is the character samples collected using a preformatted data collection sheet at our own laboratory, from individuals of different age, sex, education, profession, writing instrument, writing surface and also with their varying mood. Total 4130 samples have been collected, consisting of 3430 samples of isolated handwritten Devnagari characters and 700 samples of isolated handwritten Devnagari numerals.

For designing the suitable recognizer, two major aspects of character recognition problems have been considered i.e. study of different features and study of different classifiers for recognition of handwritten isolated basic characters and numerals of Devnagari script. Before extraction of features, the character images need to be preprocessed. Preprocessing aims to produce noise-free document images which help the OCR systems to operate accurately. Preprocessing steps and different feature extraction methods are discussed in section 4.

Multilayer perceptrons of three layers namely input layer, hidden layer, and output layer with back propagation learning algorithms have been designed for this work. Experiments were carried out with the character set collected from ISI, our own character dataset and the combination of the two. During experimentations on the designed MLP classifiers of eight different feature sets, it has been found that recognition accuracy of some of the classifiers are very good and comparable to each other. Also the character patterns misclassified by those classifiers are not overlapping. This suggested that different classifiers potentially offered complementary information about the pattern to be classified which could be harnessed to improve the performance of the overall classification result. So instead of relying on a single decision making scheme, classifier are combined using majority voting (max voting, min voting), weighted majority schemes, discussed in detail in section 5.

SVMs have demonstrated superior classification accuracies compared to MLP in many experiments. MLP is flexible, but, it is not continuous. The parameters of MLP are generally adjusted by gradient descent. The convergence of MLP training may suffer from local minima of error surface. SVMs are trained by quadratic programming which guarantees finding the global optimum. The performance of SVMs depends on the selection of kernel type and kernel parameters. SVM implementations using RBF kernel with each of the eight features discussed above are tested on isolated handwritten Devnagari characters, which give results comparable with MLP, discussed in section 5.

Two different approaches as two stage classification for handwritten Devnagari characters have also been proposed in the work, discussed in detail in section 6. In first approach structural properties i.e. presence/absence of shirorekha and position of spine of Devnagari characters have been utilized. In first stage of classification, preliminary grouping of characters has been done using these structural properties. In second stage of classification, MLP is designed for each group of characters using directional chain code features. On analyzing the recognition results of MLP classifiers, it has been found that misclassification occurs among similar looking character shapes. So in second approach, the character patterns are divided into two sets, one consisting of character patterns having distinct shapes which can be identified with certainty and another consisting of characters having almost similar shapes or confusing characters. This division is based on a relative difference measure. After separating characters in two sets, a MLP based classifier is used in the first stage to classify characters of certainty set. And for classifying characters of confusing character set, method of minimum edit distance is applied in the second stage, on detected corners of the sample character using a modified form of Harris corner detector.

A two stage scheme for isolated numeral recognition is also presented in section 7. MLP and SVM are applied in two stages for this purpose.

4. FEATURES EXTRACTION

Performance of any OCR system also depends on the choice of appropriate feature set. Here, we have experimented with different types feature sets to test their discriminating power among the character patterns of Devnagari alphabet (only non-compound characters) and numerals. Before extracting features, the character patterns need to be preprocessed.

During preprocessing, for each character image in the Devnagari basic character database, the bounding box of the character pattern is first identified and the image is cropped accordingly. Median filter with window size 3 × 3 pixels is applied on the character image to remove noise. Then the image is binarized, thinned and scaled to a fixed size. Dynamic threshold value is calculated for binarization[22]. Thinning of the binarized character images is necessary for getting their one-pixel wide skeletons. Thinning technique [3] for the binary images successively deletes only those dark pixels along the boundary of the character image whose deletion does not locally disconnect the character image. There are still some redundant pixels in thinned character image. These pixels are removed by applying some specially designed masks[22]. All character images including the images of their skeletons are normalized to size 100 × 100 using affine transformation.

After preprocessing the character image, eight types of topological and structural features namely: shadows, directional chain codes, views, zone based cenroids, intersection points, straight line segment fitting, geometric moments, and longest runs are extracted. The first one is based on calculating shadows or lengths of the projections of character segments on the sides of eight octant dividing triangles on the character image [20]. Total 24 shadow features are calculated. Second feature provides the details of the traversal along the character contour. The direction changes during this traversal are captured using Freeman chain coding procedure [4]. The histogram of chain codes [20] is prepared for 25 zones of character image, so total 200 directional chain code features are calculated. The view based features [18] examine four views (top, bottom, left and right) of the character image i.e. distance between an observer located at any four sides of character image and the visible points on the character contour. We have considered 11 uniformly spaced points on each side of the view and a total of 44 features representing all the views of a character pattern are calculated. Next, 100 zone based centroid features [15], which are reasonably invariant with respect to shape variations caused by various writing styles, are calculated. Coordinate position(x,y)of centroid is calculated for each of 25 zones of character image. Intersection, also referred as a junction is a location which has more than two neighboring pixels in 8-connectivity. Intersections and open end points are calculated for each of 16 zones of character image as 32 set of features [20]. Another type of feature is computed from the straight line fits to the set of foreground pixels or black pixels in each of 16 zones. The slope of each and every straight line is calculated and this slope is used to calculate two features in each of 16 zones [20], so total 32 features calculated for character image. Besides these, 112 features using seven rotation, translation, and scale invariant moments are derived from the second and third order geometric moments [19] for 16 zones of character image. These features are calculated for 16 zones of character image. Finally, 100 longest run features[17] are calculated row-wise, column-wise, and along its major diagonals for 25 zones of character image. The longest-run feature is computed by considering the sum of the lengths of the longest bars that fit consecutive black pixels along each of all the rows, columns, or along its major diagonals of a rectangular region.

All characters have a near straight line part extending horizontally at the top called shirorekha and some characters have a near straight line part extending vertically called spine. Their positions vary in each character. Detecting horizontally and vertically near straight line in handwritten Devnagari characters cannot be done using conventional methods like histogram based method. A differential distance method based technique and priority based search mechanism is proposed for this [21].

5. STUDY OF DIFFERENT CLASSIFIERS

In the present work, we have developed three different MLP based systems for recognition of constrained handwritten non compound Devnagari characters, which are described in detail in the following subsections:

5.1 MLP for classifying handwritten isolated non-compound Devnagari characters

MLPs with 3 layers including one hidden layer are used here. Eight different feature sets, namely:- 32 intersection features, 24 shadow features, 32 straight line fitting based features and 200 directional chain code features, 44 view based features, 112 momentum based features, 100 longest run bar features and 100 zone based centroid features are extracted from the isolated binarized character images. Each MLP based classifier is trained with a single type of above mentioned feature sets separately using standard backpropagation algorithm. 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 have used sigmoid function. Learning rate and momentum term are set to 0.8 and 0.7 respectively. Classification was accomplished by a simple maximum response strategy.

We have considered two different datasets for experimentation (section 3). From ISI dataset, 3430 samples were used for training and 1470 samples for testing. From our own created dataset 2401 samples were used for training and 1029 samples for testing. For 3-fold cross validation of results the whole dataset is divided into three parts. In first fold, first two parts are used for training and third part is used for testing. In second fold, first and third part is used for training and second part is used for testing. In fold three, second and third part is used for training and first part is used for testing. The recognition accuracies of eight different MLP based classifiers designed for eight types of feature sets are evaluated and results obtained after 3-fold cross validation are shown in table 1 for ISI dataset and for our own dataset. Number of neurons in input layer is decided by number of features. Number of neurons in hidden layer is not fixed; we have experimented on the values between 20-70 to get optimal results. These optimal number hidden layer neurons are mentioned in table 1. Number of character classes decides number of neurons in output layer, which is 49 for the present work.

Table 1: Recognition accuracies of the MLP based classifiers for each feature type on ISI dataset and on our dataset

Feature set	No. of Input layer Neurons	No. of Hidden Layer Neurons	No. of Output Layer Neurons	Average recognition accuracy after 3-fold cross validation on ISI dataset	Average recognition accuracy after 3-fold cross validation on our dataset
Straight Line Fitting based	32	40	49	Top 1 54.83%	Top 1 65.43%
				Top 5 67.77%	Top 5 74.43%
Intersection based	32	20	49	Top 1 56.71%	Top 1 66.91%
				Top 5 70.47%	Top 5 76.12%
Shadow based	24	30	49	Top 1 60.59%	Top 1 72.56%
				Top 5 83.47%	Top 5 83.87%
Directional chain code based	200	70	49	Top 1 67.90%	Top 1 79.90%
				Top 5 87.32%	Top 5 90.81%
View based	44	30	49	Top 1 60.07%	Top 1 73.87%
				Top 5 73.86%	Top 5 85.78%
Momentum based	112	60	49	Top 1 61.45%	Top 1 72.45%
				Top 5 72.37%	Top 5 83.14%
Longest-run bar based	100	50	49	Top 1 60.45%	Top 1 71.66%
				Top 5 71.10%	Top 5 82.65%
Zone centroid based	100	50	49	Top 1 63.12%	Top 1 73.52%
				Top 5 76.79%	Top 5 86.45%

5.2 Decision combination of MLP based classifiers for handwritten isolated non compound Devnagai characters

It is observed from the above MLP experiment, that the sets of character patterns misclassified by those classifiers do not overlap. This finding leads to the possibility of improving the recognition performance by combining the decisions of different MLP based classifiers, designed here, which provides complimentary information about the character patterns. So instead of relying on a single decision making scheme we can combine classifiers to get higher accuracy. Various classifier combination schemes [14] are found in the literature, we have experimented with Min, Max and weighted majority technique [17,20]. In min and max decision combination scheme, choice has to be made between the "consensus decision" strategy and the "decision delivered by the most competent classifier". A simple enhancement to the simple majority systems can be made if the decisions of each classifier are multiplied by a weight to reflect the individual confidences of these decisions. The higher the confidence, the higher is the value of ω. And D_k is the competence of each classifier/expert.

$$\omega_k = \frac{D_k}{\text{------- -----------}}$$

$$\sum_{k=1}^{8} D_k$$

Outputs from several classifiers are combined to produce a more accurate result. We have eight MLP based classifiers (discussed in section 5.1), and their results are shown in table 1. The competence of k^{th} classifier D_k (k=1,2,…8) is taken as its recognition accuracy. The values of **ω1, ω2, ω3, ω4, ω5, ω6, ω7, ω8** are 0.1135, 0.1174, 0.1254, 0.1365, 0.1244, 0.1273, 0.1249 and 0.1306 for ISI data set and 0.1135, 0.1161, 0.1259, 0.1388, 0.1282, 0.1257, 0.1243 and 0.1275 for our data set respectively. These values are calculated using top1 results of eight different classifiers discussed in table 1. Similarly we calculate the values for D_k and ω_k considering top5 results for both ISI and our datasets. Results on combined (ISI and our own collected dataset) dataset is given in table 2.

Table 2 : Results of decision combination on ISI dataset, our dataset, and combined dataset for isolated handwritten Devnagari characters

5.3 SVM for classifying handwritten Devnagari characters

SVM [1, 27] is defined as a set of related supervised learning methods used for classification and regression. It is a useful data classification technique for 2-class problems. For multi-class classification, One-Against-One and the One-Against-All classification schemes can be used. Both schemes have their own advantages and disadvantages. Considering the better accuracy for large number of classes and reduced training time, the one-against-one classification scheme for SVMs have been used here.

We used LIBSVM software for SVM classification [17]. LIBSVM is simple, easy-to-use, and efficient software for SVM classification and regression. A classification task usually involves with training and testing data which consist of some data instances. Each instance in the training set contains one target value and several attributes. The goal of SVM is to produce a model which predicts target value of data instances in the testing set which are represented only with the attributes. Our implementation of multiclass support vector machine [6] consists of a *learning module* and a *classification module*. The classification module can be used to apply the learned model to new examples. The format of training and testing data file is:

<label> <index1>:<value1> <index2>:<value2> ...

Each line contains an instance and is ended by a '\n' character. For classification, <label> is an integer indicating the class label (multi-class is supported). The pair <index>:<value> gives a feature (attribute) value: <index> is an integer starting from 1 and <value> is a real number. Labels in the testing file are only used to calculate accuracy or errors.

We experimented with SVM that use the RBF kernel, on the all characters of dataset discussed above because this kernel nonlinearly maps samples into a higher dimensional space so it, unlike the linear kernel, can handle the case when the relation between class labels and attributes is nonlinear. The second reason is the number of hyperparameters which influences the complexity of model selection. The polynomial kernel has more hyperparameters than the RBF kernel. Finally, the RBF kernel has fewer numerical difficulties. One key point is $0 < K_{ab} \leq 1$ in contrast to polynomial kernels of which kernel values may go to infinity ($\gamma a\ b + r > 1$) or zero ($\gamma a\ b + r < 1$) while the degree is large. Moreover, the sigmoid kernel is not valid (i.e. not the inner product of two vectors) under some parameters. So the choice of kernel type is RBF kernel, mainly because of their localized and finite response across the entire range. We chose the value of γ as 0.1 for the below mentioned RBF kernel function:

Classifier Combination Technique	On ISI dataset		On our dataset		On combined dataset	
	Training set	Test set	Training set	Test set	Training set	Test set
Min Voting	76.75% (top1)	62.92% (top1)	91.08% (top1)	78.49% (top1)	83.92% (top1)	70.71% (top1)
	91.23% (top5)	72.45% (top5)	99.28% (top5)	94.79% (top5)	99.01% (top5)	89.76% (top5)
Max Voting	84.12% (top1)	74.65% (top1)	98.24% (top1)	93.93% (top1)	91.18% (top1)	84.29% (top1)
	97.68% (top5)	92.36% (top5)	100% (top5)	100% (top5)	99.82% (top5)	97.52% (top5)
Weighted majority Voting	82.15% (top1)	70.38% (top1)	97.94% (top1)	90.44% (top1)	90.05% (top1)	80.41% (top1)
	93.31% (top5)	90.74% (top5)	99.51% (top5)	99.08% (top5)	96.41% (top5)	94.91% (top5)

$$K(a,b) = exp(-\gamma \,||a-b||^2)$$

The results for the experiments are given in Table 3.

Table 3: Recognition accuracy of SVM on ISI dataset and our dataset

Feature set	Feature Vector Dimension	Classification Accuracy on ISI dataset	Classification Accuracy on our dataset
Straight Line Fitting based	32	60.12%	65.12%
Intersection based	32	65.34%	72.14%
Shadow based	24	67.25%	77.45%
Directional chain code based	200	73.83%	84.92%

ISSN 2229-5518

View based	44	62.02%	70.47%
Momentum based	112	63.18%	69.23%
Longest-run bar based	100	63.38%	70.38%
Zone centroid based	100	64.34%	73.38%

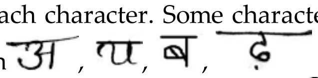
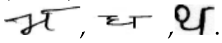
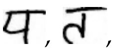
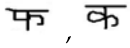
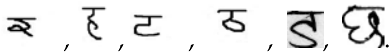
6. TWO STAGE CLASSIFICATION SCHEMES FOR HANDWRITTEN ISOLATED NON COMPOUND CHARACTERS OF DEVNAGARI SCRIPT

It is observed from experiment that accuracy can be further improved if the recognition can be done in two stages. We propose here, two approaches of two stage classification for handwritten isolated non compound Devnagari characters.

First approach (discussed in section 6.1) utilizes some structural properties like presence/absence of shirorekha, position of spine, of handwritten Devnagari characters. A preliminary grouping of Devnagari characters has been done in first stage using these structural properties. For second stage classification, a three layer MLP is designed using directional chain code features for each group of characters.

In second approach (discussed in section 6.2), handwritten Devnagari character set is divided in two sets i.e. characters recognized with certainty and group of confusing characters. This division is based on the relative differences of top three output values obtained after combination of the decisions of 8 MLPs using weighted majority voting technique. A 3-layer MLP designed using eight features is used to classify characters of certainty set in first stage of classification. Method of minimum edit distance is used for classifying confusing character set as second stage classification, on detected corners of the sample character image using a modified form of Harris corner detector.

6.1 Two stage classification approach for isolated handwritten Devnagari characters using some structural properties in first stage and MLP in second stage

Devnagari characters have a very special property i.e. all characters have a near straight line horizontally and vertically called shirorekha and spine respectively. Their position varies in each character. Some characters have a full shirorekha as in  and some have partial shirorekha as in . Some characters have spine at the end of the character as , , some characters have spine at middle of the character as in  and some characters does not have spine at all as in .

Preliminary grouping of characters has been done using these above mentioned structural properties of Devnagari characters i.e. presence/absence of shirorekha and position of spine[21], in first stage of classification. In second stage of classification, MLP is designed for each group of characters using directional chain code features. We used only this feature for classification, because this feature gives the highest recognition accuracy when experimented on MLP compared to other features.

6.1.1 Preliminary classification

The presence and position of shirorekha and spine divides character set of Devanagri into different classes, so the entire character map for Devanagari characters is grouped according to the following criteria:-

Shirorekha Continuity

Some characters contain a shirorekha throughout, while the others contain a partial shirorekha. Thus two groups can be obtained by this method namely partial shirorekha group and total shirorekha group.

Spine Location

Another important aspect of the character is its "spine". Characters can be divided into three groups i)End Spine (The spine is the rightmost part of the character), ii)Mid Spine (The spine exists in between, i.e. there are some parts of the character on either side of the spine), iii)No Spine (There are no spines in these characters). By taking an intersection of the above two properties, the entire character map can be divided into small groups. This eases the task of recognition. Thus the groups formed are i) partial shirorekha and end spine group, ii) total shirorekha and end spine group, iii) total shirorekha and mid spine group, iv) total shirorekha and no spine group. No characters are found in partial shirorekha and mid or no spine group for Devnagari alphabets. Characters of each group are shown in figure 5.

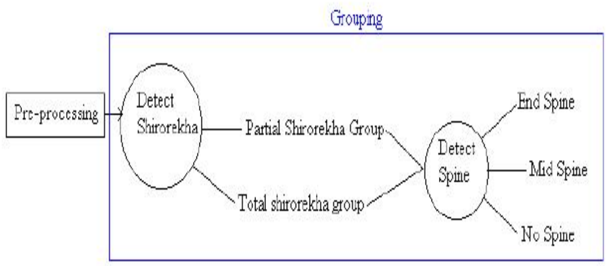


Figure 4: Preliminary classification

Figure 5: Devnagari character groups using structural properties

6.1.2 MLP design for second stage classification

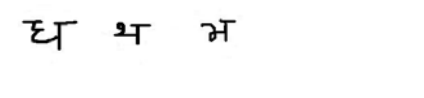
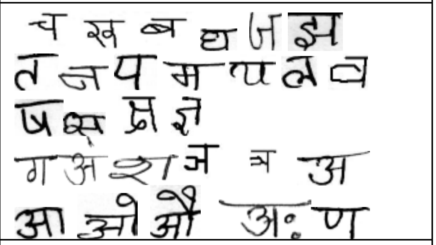
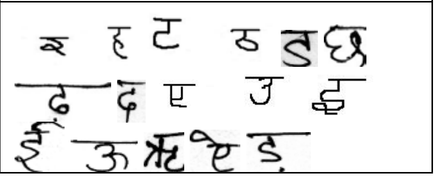
For each group of characters, a separate MLP with three layers including one hidden layer is used. Feature set used here to represent each character image consists of 200 directional chain code features [20]. MLPs are trained using standard backpropagation algorithm. It minimizes the sum of squared errors for the training samples by conducting a gradient descent search in weight space. As activation function we have used sigmoid function. Learning rate and momentum term are set to 0.7 and 0.95 respectively. Number of neurons in input layer of MLP is 200 for directional chain code feature. Number of neurons in hidden layer was set to 70 after experimentation. The number of neurons in output layer is different for each group, which were formed after preliminary classification as discussed above (in figure 5). Partial shirorekha and end spine group have 3 characters, so output layer of MLP for this group contains 3 neurons, total shirorekha and end spine group have 28 characters so output layer of MLP contains 28 neurons for this group, for total shirorekha and mid spine, total shirorekha and no spine groups have 2, 16 characters respectively so output layer of MLP for these groups contains 2, 16 neurons respectively. Classification is accomplished by a simple maximum response strategy. Classification results are shown in table 4.

Table 4: Recognition accuracy for isolated handwritten Devnagari characters using structural properties for preliminary classification and MLP for fine classification

Character group	Recognition accuracy	Character group	Recognition accuracy
Partial shirorekha and End spine group	95%	Total shirorekha and Mid spine group	98%
Total shirorekha and End spine group	89.12%	Total shirorekha and No spine group	90%

6.2 Two stage classification for handwritten isolated non compound Devnagari characters using MLP in first stage and method of minimum edit distance in second stage

The overall recognition accuracy of our system on the combined test data (ISI data set and our own dataset) obtained with combination of decisions of the MLPs using weighted majority voting scheme (discussed in section 5.2) is 80.41% as top1 choice result and 94.91% when top5 choices of the recognition result is considered with zero percent confusion. There are more chances of false acceptance of characters in case of min voting and max voting classifier combination technique compared to weighted majority voting combination technique, so in our proposed two stage scheme, results of weighted majority voting combination technique is used.

Character group	Some example data samples
Partial shirorekha and End spine	
Total shirorekha and End spine	
Total shirorekha and Mid spine	
Total shirorekha and No spine	

In this proposed two stage classification approach, the character set is divided into two categories, the first one is the characters recognized with certainty and the other one is the groups of confusing characters. Both MLP based classifier and minimum edit distance based method have been tried for classifying the characters both in the certainty set and confusing set. After experimenting on both classifier for both character sets, MLP based classifier combined using weighted majority voting scheme have been used for certainty character set classification in stage one of the present work. In second stage, minimum edit distance classifier is used for confusing characters. Advantage of this two stage approach is that with small number of characters in each group better classification is achieved.

6.2.1 Confused character set identification using relative difference measure

From experiment it was noticed that mainly the error occurred because of the presence of similar shaped characters. These similar looking characters have a high chance to be misclassified among themselves. Some examples of such similar looking handwritten Devnagari non compound characters are listed in groups in figure 6. Each such group is called a confusing character group. Confusing character samples are identified on the basis of relative differences of top three output values obtained after combination of the decisions of 8 MLPs using weighted majority voting technique.

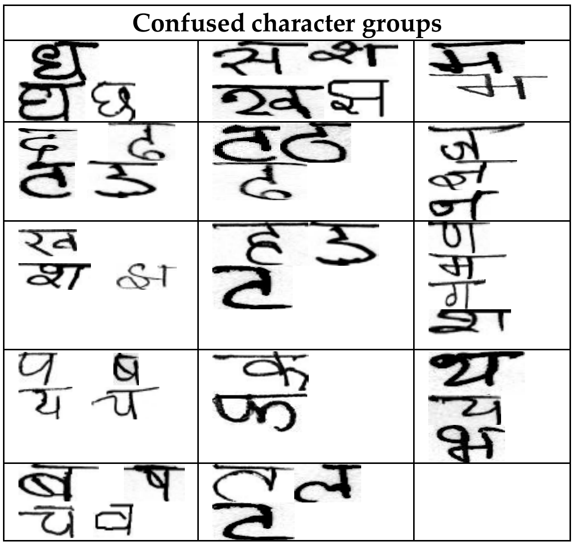


Figure 6: Some examples of confused character groups

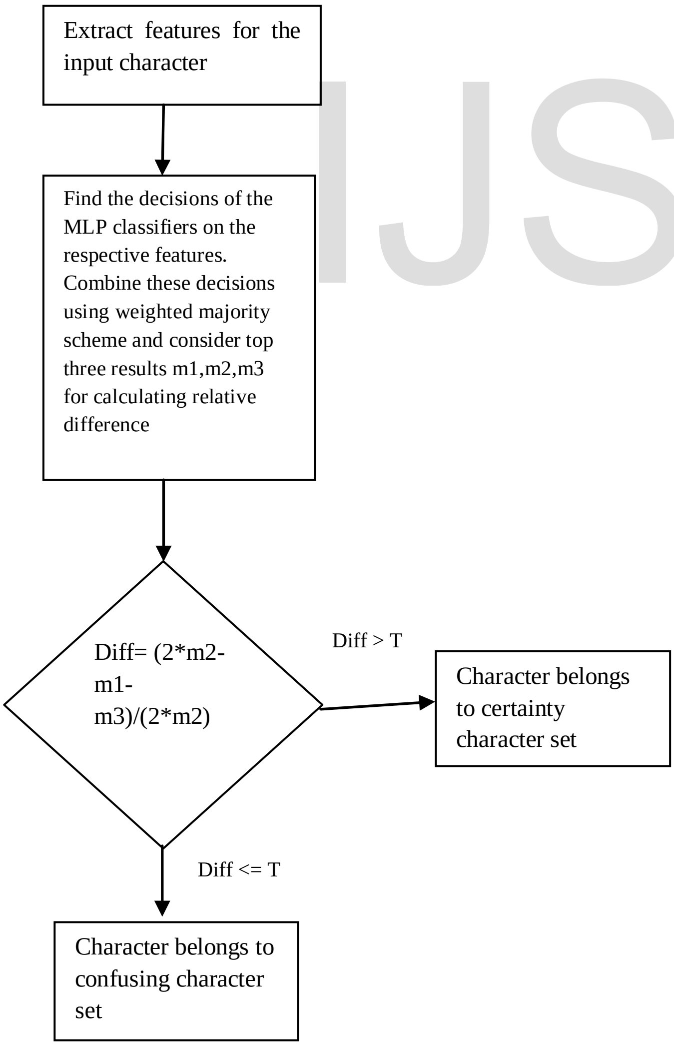


Figure 7: Block diagram for identification of confusing character set and certainty character set

Relative difference measure is computed as:- Diff = (2* max2-max1-max3) / (2*max2)

where max1, max2, max3 are top three output values of combined MLP classifier using weighted majority scheme. If this relative difference is greater than some threshold value, we infer that the top choice determines the class of the sample character with certainty. On the other hand, if the relative difference measure is less than or equal to the threshold value, we infer that the sample character belongs to a group of confusing character identified by top three choices. Considering the threshold relative difference value T=0.1, 57.3% characters were separated in characters with certainty set which have distinct shapes and rest 42.7% characters were in confusing character set which are of very similar shapes. For classifying the characters both in certainty set and confusing set, both MLP based classifier and minimum edit distance based method have been tried.

6.2.2 Classification using method of minimum edit distance

For classifying characters, minimum edit distance based approach has been applied here. For extracting features, we detected corners in character image using modified form of Harris corner detector (discussed in section 6.2.2.2). After detecting the corners, the character image, is divided into 16 segments, and in each segment number of corner points are counted for each character to construct the feature set. The distance among these characters are computed using method of minimum edit distance (discussed in section 6.2.2.3), by utilizing this feature set [16].

6.2.2.1 Harris corner detection algorithm

Corner detector is of interest as they assign a measure of cornerness to all pixels in an image. The brute force method of comparing every pixel in the two images is computationally prohibitive. Intuitively one can relate two images by matching only locations in the image that are in some way represent some information about the pattern shapes. Such points are referred to as interest points and are located using an interest point detector. Many corner detection algorithms are available like Moravec, Harris/Plessey etc [7]. We chose Harris/Plessey corner detector with some modification. This corner detector is computationally demanding, but directly addresses many of the limitations of the other corner detectors. Algorithm for detecting corners using Harris corner detector in all confused characters is as follows:-

A1	A2 B1	A3 B2	B3
A4	A5 B4	A6 B5	B6
A7	A8 B7	A9 B8	B9

1. For each pixel (x, y) in the image calculate the autocorrelation matrix M as

B1	B2	B3
A1 B4	A2 B5	A3 B6
A4 B7	A5 B8	A6 B9
A7	A8	A9

$$M = \begin{bmatrix} P & R \\ R & Q \end{bmatrix}$$

Where $P = \left(\frac{\partial I}{\partial x}\right)^2 \otimes w$ $Q = \left(\frac{\partial I}{\partial y}\right)^2 \otimes w$ $R = \left(\frac{\partial I}{\partial x}\frac{\partial I}{\partial y}\right)^2 \otimes w$

$\otimes$ is the convolution operator , w is the Gaussian window of size 5 and $\left(\frac{\partial I}{\partial x}\right), \left(\frac{\partial I}{\partial y}\right), \left(\frac{\partial I}{\partial x}\frac{\partial I}{\partial y}\right)$ are horizontal, vertical and diagonal intensity gradients respectively.

2. Construct the cornerness map by calculating the cornerness measure C(x, y) for each pixel (x, y):
   $C(x,y) = \det(M)-k(\mathrm{trace}(M))^2$
   $\det(M) = \lambda_1 \lambda_2 = PQ-R^2$
   $\mathrm{trace}(M) = \lambda_1 + \lambda_2 = P + Q$
   k = constant , λ_1 and λ_2 are eigenvalues of M
3. Threshold the cornerness map by setting all C(x, y) below a threshold T to zero.
4. Perform non-maximal suppression to find local maxima.

The basic idea behind detecting corners of an image in this algorithm is to estimate the measurement of local autocorrelation so that intensity variation can be measured in different directions for that purpose. Intensity variation calculation for Harris operator is approximated using the gradients of the intensities along horizontal, vertical and diagonal directions. Thus in step 1, parameters P, Q, and R can be defined as:

P= Weighted horizontal intensity gradient Q= Weighted vertical intensity gradient R= Weighted diagonal intensity gradient

Next the intensity gradient is convolved with the Gaussian window. Gaussian window is a circular window that puts more weight on measurement made closer to the centre of the window. This is desirable so that the Euclidean distance from the centre pixel to the edge is same in all directions. This improves the estimate of the local intensity variation.

Horizontal intensity gradient is calculated by convolving with the window, shown in figure 8(a). Similarly vertical and diagonal intensity gradients are computed by convolving with the windows, shown in figures 8(b) and 8(c) respectively. Finally a gaussian window of size 5, shown in figure 8(d) is used to convolve with the intensity gradient maps to generate the corresponding weighted versions P, Q and R.

(a) (b)

W1 .004	W2 .015	W3 .026	W4 .015	W5 .004
W6 .015	W7 .059	W8 .095	W9 .059	W10 .015
W11 .026	W12 .095	W13 .15	W14 .095	W15 .026
W16 .015	W17 .059	W18 .095	W19 .059	W20 .015
W21 .004	W22 .015	W23 .026	W24 .015	W25 .004

	B1	B2	B3
A1	A2 B4	A3 B5	B6
A4	A5 B7	A6 B8	B9
A7	A8	A9	

(c)

(d)

Figure 8: a)Horizontal intensity gradient window b) Vertical intensity gradient window c) Diagonal upward intensity gradient window d) Gaussian Window of size 5

6.2.2.2 Modification to Harris corner detection algorithm

This algorithm was modified because it just considers weighted intensity gradients in three directions i.e. in horizontal, vertical and diagonal upward direction, which sometimes fail to detect corners properly for Devnagari characters. In the present work, one intensity gradient was considered i.e. weighted diagonal intensity gradient in downward direction S to supplement the previous three weighted gradients P, Q and R. Now the weighted diagonal intensity gradient in downward direction S is calculated by convolving with the specified window in figure 9 and then with the gaussian window of size 5.

Figure 9: Diagonal downward intensity gradient window

So the autocorrelation matrix M in step 1 is modified as

$$M = \begin{bmatrix} P & R+S \\ R+S & Q \end{bmatrix}$$

Steps 2, 3 and 4 are performed as specified above. All non-zero points remaining in the cornerness map are corners. We used k=0.04 in our algorithms.

6.2.2.3 Method of minimum edit distance

For classifying characters of similar shapes discussed above, minimum edit distance approach has been applied [26]. Corners in character image were detected using modified form of Harris corner detector. After detecting the corners, the character image, is divided into 16 segments, and in each segment number of corner points are counted. So for each character a 16 element feature set is computed, where each element represent the number of corners in the respective segment. This feature set is utilized for calculating the distance among the characters using the method of minimum edit distance. Distance is a measure of similarity between two strings which is referred as source string 's' and target string 't'. The distance is the number of deletions, insertions or substitutions required to transform 's' into 't'. The basic idea behind calculating the distance Dist(i,j) between two corner string s1[1…i] and s2[1….j] is as follows:-

A two-dimensional matrix, Dist[0..|s1|,0..|s2|] is used to hold the edit distance values:

Dist[i,j] = Dist(s1[1..i], s2[1..j])

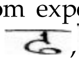
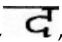
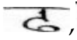
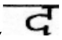
Dist[0,0] = 0

Dist[i,0] = i, i=1..|s1|

Dist[0,j] = j, j=1..|s2|

$$\text{Dist}(i, j) = \text{minimum}\begin{pmatrix} \text{Dist}(i-1, j)+1, \\ \text{Dist}(i, j-1)+1, \\ \text{Dist}(i-1, j-1)+t(i, j) \end{pmatrix}$$

$$t(i, j) = \begin{cases} 0 & \text{if } s1(i) = s2(j) \\ 1 & \text{if } s1(i) != s2(j) \end{cases}$$

From experiment, i.e. character  is being confused for , ,  and . We calculated the string of corner of size 16, for each confused character image and distance of  is calculated with , ,  and  which is 9, 0, 6 and 8 respectively. Character is said to be classified as the character for which it gives minimum distance i.e. as .

	A1	A2	A3
B1	B2 A4	B3 A5	A6
B4	B5 A7	B6 A8	A9
B7	B8	B9	

6.2.3 MLP design

Three layer MLPs including one hidden layer is used for classification. 8 MLPs are designed as discussed in section 5.1 using eight different feature sets. Learning rate and momentum term are set to 0.7 and 0.95 respectively. Detail results of each MLP is shown in table 1 of section 5.1. Results of MLPs are combined using weighted majority voting technique (section 5.2).

6.2.4 Recognition results

As already mentioned, using the relative difference measure, the handwritten isolated non compound Devnagari characters were separated in certainty set and confusing character set. For the characters in the certainty set, MLP based classifier and minimum edit distance based classifier were tried. Recognition accuracy of 97.27% was achieved with MLP classifier, whereas that for minimum edit distance based classifier was 72.27%. So the obvious choice of classifier for classifying the characters in certainty character set is MLP based classifier.

For the characters in the confusing character set, again MLP and minimum edit distance based classifiers were experimented. A recognition accuracy of 68.16% was achieved for MLP classifier and minimum edit distance based classifier correctly recognizes 82% characters. So for confusing character set classification, the obvious choice was of minimum edit distance based classifier.

The combined accuracy of MLP based classifier for characters in certainty set and minimum edit distance based classsifier for characters in confusing set is found to be 90.74% as shown in table 5.

ISSN 2229-5518

Table 5: Classification results of MLP based classifier for certainty set and method of minimum edit distance based classifier for confusing set for isolated handwritten Devnagari characters

Character set	% characters in test dataset	Classification method	Accuracy
Certainty set	57.3%	MLP	97.27%
		Minimum edit distance	72.27%
Confusing set	42.7%	MLP	68.16%
		Minimum edit distance	82%
Combined accuracy of MLP for certainty set and minimum edit distance method for confusing set	90.74%		

7. RECOGNITION OF ISOLATED HANDWRITTEN DEVNAGARI NUMERAL USING MLP AND SVM

MLP and SVM are used for handwritten isolated Devnagari numeral recognition. Method of minimum edit distance is not used as there is less confusion among the shapes of the numerals and corner detection is not relevant for numeral shapes. MLP using eight different types of features and SVM is applied for handwritten Devnagari numeral classification [15].

7.1 Handwritten Devnagari numeral classification

The recognition of handwritten Devnagari numerals has been done in two stages. In the first stage, numeral images are classified using eight MLPs designed using eight different types of features (section 4) namely: shadow based, zone based centroid, view based, directional chain code, straight line fitting, intersection, momentum and longest-run bar features. MLPs are trained using standard backpropagation algorithm. It minimizes the sum of squared errors for the training samples by conducting a gradient descent search in weight space. Sigmoid function is used as activation function. Learning rate and momentum term are set to 0.7 and 0.95 respectively. Results of these different features using MLP's are given in table 6.

Results of all MLPs are combined using weighted majority scheme. The values of ω_1, ω_2, ω_3, ω_4, ω_5, ω_6, ω_7, ω_8 are 0.133, 0.123, 0.130, 0.139, 0.104, 0.125, 0.124, 0.122. These values are calculated using top1 choice results of eight different

When applied MLP based classifier combined using weighted majority voting technique on whole dataset, 90.19% accuracy was achieved. SVM using RBF kernel was also experimented for combined features including directional chain code, view based and shadow features on whole dataset, and accuracy of 89.27% was achieved. Results of both classifiers are comparable so to get higher recognition accuracy, MLP based classifier and SVM are applied in two stages.

Relative difference was calculated using top three results obtained after combining the eight MLPs using weighted majority voting technique. For 61.3% numerals, threshold relative difference value (discussed in section 6.2.1) T was > 0.1 and for rest 38.7% numerals value of T was $\leq$ 0.1. If the threshold relative difference value T > 0.1, numeral was classified using MLP, designed using features discussed in section 4 and combined using weighted majority voting technique. The top choice result produced by MLP determines the class of the numeral at first stage. Accuracy of 92.91% was achieved using MLP in stage one. If the value of T $\leq$ 0.1, SVM was used to classify the numeral. SVM using RBF kernel (discussed in section 5.3) on combined directional chain code feature, view based and shadow features were used to classify numeral in second stage. Accuracy of 98.79% was achieved for SVM in second stage. Results of both classifiers are combined to get higher accuracy. In table 7, results of combined MLP (using weighted majority scheme) and SVM is given. The overall accuracy achieved is 95.18%, by combining the 92.91% accuracy of MLP and 98.79% accuracy of SVM.

The experiment of Devnagari numerals dataset contains 18300 handwritten samples, 12810 samples in training dataset and 5490 samples for testing results.

Table 6: Results of MLP for different features on isolated handwritten Devnagari numerals

Table 7: Accuracy of MLP in first stage and SVM in second stage for isolated handwritten Devnagari numerals

Classifier	Accuracy
MLP(Combined using weighted majority scheme)	92.91% (top1) 100% (top5)
SVM	98.79%
Combined MLP and SVM	95.18%

Feature set	Number of Input layer Neuron	Number of Hidden Layer Neurons	Number of Output Layer Neuron	Result
Shadow based	24	30	10	(Top 1) 86.19%
				(Top 5) 96.56%
Zone centroid based Intersection based	100	40	10	(Top 1) 80.87%
				(Top 5) 89.13%
View based	44	30	10	(Top 1) 85.87%
				(Top 5) 97.12%
Directional chain code based	200	70	10	(Top 1) 90.92%
				(Top 5) 99.34%
Straight Line Fitting based	32	40	10	(Top 1) 73.45%
				(Top 5) 84.33%
Intersection based	32	20	10	(Top 1) 81.91%
				(Top 5) 89.12%
Momentum based	112	60	10	(Top 1) 81.45%
				(Top 5) 89.94%
Longest-run bar based	100	50	10	(Top 1) 80.66%
				(Top 5) 90.65%

8. CONCLUSION

Some soft computing techniques using structural and statistical feature sets for recognition of constrained, handwritten, isolated, basic Devnagari characters and numerals have been implemented. The work may be extended to include modifiers, and compound characters which are the integral part of Devnagari script. More robust features can be incorporated to handle skew and rotation invariance and some other classification methods can be studied to improve the recognition accuracy. The work just deals with offline isolated characters and numerals. Document recognition requires many more issues to be dealt with. Online character recognition can also be worked on in future.

REFERENCES

1. B.E. Boser, I. Guyon, V. Vapnik, "A training algorithm for optimal margin classifier", Proceeding of fifth annual conference on Computational Learning Theory (COLT-92),USA, pp 144-152, 1992
2. Bansal V, Sinha R. M. K., "*Integrating Knowledge Resources in Devnagri. Text recognition system*", IEEE Transaction on System, Man & Cybernatics Part A: Systems & Humans. V30 n 4 july 2000.p 500-505
3. E.R. Davies, A.P. plummer, "Thinning algorithms: A critique and a new methodology", Pattern recognition, vol 14, pp 53-63, 1981
4. Freeman, H., "On the Encoding of Arbitrary Geometric Configurations, IRE Trans. on Electronics Computer or TeleCommunication vol 10, No. 2, pp 260-268, June 1961
5. I.K. Sethi and B. Chatterjee, "*Machine Recognition of constrained Hand printed Devnagari*", Pattern Recognition, Vol. 9, pp. 69-75, 1977.
6. J.Weston,C.Watkins, "Support vector machines for multi-class pattern recognition", Proceedings of the Seventh European Symposium On Artificial neural networks, pp 219-224,1999
7. K.Saeed, M. Tabedzki, M. Adamski, "A New Approach for Object-Feature Extract and Recognition", 9th international conference on advanced computer systems , pp 389-397, Miedzyzdroje 2002
8. K. Sethi and B. Chattarjee, "*Machine recognition of Handprinted Devnagri Numerals*". Journal of Instituions of Electronics & Telecommunication Engineers, India Vol 22, pp 532-535, 1976.
9. M. Hanmandlu and O.V. Ramana Murthy, "*Fuzzy Model Based Recognition of Handwritten Hindi Numerals*", InProc. International Conference on Cognition and Recognition, 2005, pp. 490-496.
10. M. Hanmandlu, O.V. Ramana Murthy, Vamsi Krishna Madasu, "*Fuzzy Model based recognition of Handwritten Hindi characters*", IEEE Computer society, Digital Image Computing Techniques and Applications , 2007
11. Pal U., B.B. Chaudhuri 1997 "*Printed Devnagari Script OCR System*", Vivek, vol 10, Pgs. 12-24
12. S. Kumar and C. Singh, "*A Study of Zernike Moments and its use in Devnagari Handwritten Character Recognition*", *In Proc. International Conference on Cognition and Recognition*, 2005, pp. 514-520.
13. Sinha R.K., Mahabala 1979 "*Machine Recognition of Devnagari Script*", IEEE Trans. System Man Cyber Pgs 435-441

14. S. Tulyakov, S. Jaeger, V. Govindaraju, D. Doermann, "Review of classifier combination methods", Machine Learning in Document Analysis and Recognition, in book of Studies in computational intelligence, Springer, vol 90, 361-389, 2008
15. Sandhya Arora, Debotosh Bhattacharjee, Mita Nasipuri, M. Kundu, D.K. Basu, L. Malik, "Handwritten Devnagari Numeral Recognition using SVM & ANN", International Journal of Computer Science and Emerging Technologies(IJCSET), ISSN(Online) 2044-6004: Vol. 1, Issue 2, August 2010
16. Sandhya Arora, Debotosh Bhattacharjee, Mita 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(IJCSS), ISSN(Online): 1985-1553, Vol. 4, Issue 1, March 2010
17. Sandhya Arora, Debotosh Bhattacharjee, Mita Nasipuri, L. Malik, M. Kundu, D.K. Basu, "Performance Comparison of SVM and ANN for Handwritten Devnagari Character Recognition", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, ISSN (Online): 1694-0784 ISSN (Print): 1694-0814, May 2010
18. S. Arora, D. Bhattacharjee, M. Nasipuri, D.K. Basu, M. Kundu, L. Malik, "Study of different Features on Handwritten Devnagari Characters", in IEEE-International Conference on Emerging Trends in Engineering and Technology(ICETET-09), G.H. Raisoni college of Engineering, Nagpur, 16th-18th Dec 2009
19. S. Arora, D. Bhattacharjee, M. Nasipuri, D.K. Basu, M. Kundu, "Application of Statistical Features in Handwritten Devnagari Character Recognition", International Journal of Recent Trends in Engineering[ISSN 1797-9617], IJRTE, Vol. 2, No.2, Nov 2009
20. S. Arora, D. Bhattacharjee, M. Nasipuri, D.K. Basu, M. Kundu, "Combining Multiple Features Extraction Techniques for Handwritten Devnagari Character Recognition", in IEEE-Third International conference on Industrial and Information Systems(ICIIS08), IIT, Kharagpur, Dec 8th-10th 2008
21. S. Arora, D. Bhattacharjee, M. Nasipuri, L. Malik, "A Two Stage Classification Approach for Handwritten Devnagari characters", in IEEE International Conference on Computational Intelligence and Multimedia Applications(ICCIMA 07), Sivkasi, Tamil Nadu, India, 13th-15th Dec 2007
22. S. Arora, L.Malik, D. Bhattacharjee, M. Nasipuri, "A Novel Approach for Handwritten Devanagari Character Recognition", in IEEE–International Conference on Signal And Image Processing, Hubli, Karnataka, Dec 7th-9th, 2006
23. 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.
24. U. Pal, N. Sharma, T. Wakabayashi and F. Kimura, *"Off-Line Handwritten Character Recognition of Devnagari Script"*, In Proc. 9th International Conference on Document Analysis and Recognition, 2007, pp. 496-500.
25. U. Pal, T. Wakabayashi, N. Sharma and F. Kimura, *"Handwritten Numeral Recognition of Six Popular Indian Scripts"*, In Proc. 9th International Conference on Document Analysis and Recognition, 2007, pp. 749-753.
26. V. I. Levenshtein, , "Binary codes capable of correcting deletions, insertions and reversals", Soviet Physics-Doklady 10, pp 707-710, 1966
27. Yu H., Kim S.,"SVM Tutorial: Classification, Regression, ranking", Handbook of Natural Computing, Springer, 2010