Neural Combination of ANN and HMM for Handwritten Devanagari Numeral Recognition

U. Bhattacharya, S.K. Parui, B. Shaw, K. Bhattacharya

Neural Combination of ANN and HMM for Handwritten Devanagari Numeral Recognition

U. Bhattacharya
CVPR Unit
Indian Statistical Institute, Kolkata
ujjwal@isical.ac.in

S. K. Parui
CVPR Unit
Indian Statistical Institute, Kolkata
swapan@isical.ac.in

B. Shaw
CVPR Unit
Indian Statistical Institute, Kolkata
mrbikashshaw@yahoo.co.in

K. Bhattacharya
B-Tech Final Year
Dept. of CS&E
University of Calcutta
kuntal.bhattacharya@gmail.com

Abstract

In this article, a two-stage classification system for recognition of handwritten Devanagari numerals is presented. A shape feature vector computed from certain directional-view-based strokes of an input character image, has been used by both the HMM and ANN classifiers of the present recognition system. The two sets of posterior probabilities obtained from the outputs of the above two classifiers are combined by using another ANN classifier. Finally, the numeral image is classified according to the maximum score provided by the ANN of the second stage. In the proposed scheme, we achieved 92.83% recognition accuracy on the test set of a recently developed large image database[1] of handwritten isolated numerals of Devanagari, the first and third most popular language and script in India and the world respectively. This recognition result improves the previously reported[2] accuracy of 91.28% on the same data set.

Keywords: Handwritten character recognition, Devanagari numeral recognition, HMM, ANN, Combination of classifiers.

1. Introduction

Handwritten numeral recognition has significant application potentials. The need for such a system has been increasingly felt even in a country like India with the spread of computerization in all sorts of its public / private organizations. However, although significant development has already been achieved on recognition of handwriting in scripts of developed nations, not much work has been reported on Indic scripts. The development of a handwritten character recognition engine for any script is always a challenging problem mainly because of the enormous variability of handwriting styles. The above factors provided the motivation for the proposed research work.

In the present work, we have studied off-line recognition of handwritten Devanagari numerals. Devanagari is the script of a number of Indian languages, including Hindi and Marathi. Hindi is the third most popularly used language in the world after Chinese and English. The earliest available work on recognition of hand printed Devanagari characters is found in [3]. For recognition of handwritten Devanagari numerals, Ramakrishnan et al. [4] used independent component analysis technique for feature extraction from numeral images. Bajaj et al [5] considered a strategy combining decisions of multiple classifiers. In all these three studies, very small sets of samples were considered. In an attempt to develop a bilingual handwritten numeral recognition system, Lehal and Bhatt [6] used a set of global and local features derived from the right and left projection profiles of the numeral images for recognition of handwritten numerals of Devanagari and Roman scripts. There are also some studies on handwritten character recognition of other Indian scripts [7,8].

The development of an efficient system for handwriting recognition, needs a large set of samples with ground truth. Generation of such a data set is always difficult since it is time consuming and labor intensive [9]. Such standard data sets for any Indian script did not exist till in the recent past. However, recently, a large database of handwritten Devanagari numeral images has been developed in the laboratory of the present authors. A few recognition studies have been made on this database and the only result already published can be found in [2]. In the present recognition strategy, a shape vector is fed as the feature to two different classifiers. Finally, a combination of the two classifier outputs significantly improves the previously reported recognition accuracy on this database.

The two classifiers of the first stage of the present recognition system are HMM and ANN (particularly MLP) based. Since there are many uncertainties in handwriting recognition, stochastic modeling is a suitable and popular approach to this problem. An HMM is capable of making use of both the statistical and structural information present in handwritten shapes [10]. On the other hand, neural networks usually perform efficiently in many difficult classification tasks such as handwriting recognition, particularly handwritten isolated character recognition problems. The backpropagation algorithm often trains an MLP classifier to implement highly complex multi-dimensional non-linear mappings on the basis of a large set of training samples.

The two sets of posterior probabilities obtained from the outputs of the two classifiers of the first stage are integrated by another MLP classifier at the second stage of the proposed scheme. The recognition accuracy obtained by such a combination is significantly higher than the same provided by each of the two individual classifiers. Such classifier combination approach [11] has recently become very common in the handwritten character recognition community.

In HMM based character recognition approaches, the states are usually defined as pre-determined entities. However, a novelty in the HMM proposed here is that a data-driven or adaptive approach has been taken to the determination of its states. The shapes of the strokes present in the database of handwritten Devanagari numeral images are studied and their statistical distribution is modeled as a multivariate mixture distribution. Each component in the mixture is a state of the HMM. The proposed HMM is robust in the sense that it is independent of several aspects of input such as thickness, size etc.

2. Extraction of Features

2.1. ISI database of handwritten Devanagari numerals

In the present work, we have used a recently developed database of isolated handwritten Devanagari numerals [1]. This consists of 22535 samples collected from real-life documents like postal mail pieces, job application forms etc. These documents were scanned at 300 dpi using a HP flatbed scanner and stored as gray-level images with 1 byte per pixel. A few samples from this database are shown in Fig.1.

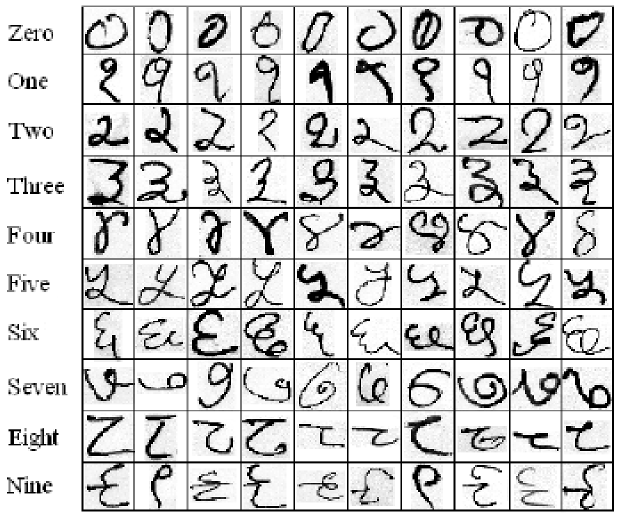

Figure 1. Samples from ISI database of handwritten Devanagari numerals.

The above database is exclusively divided into training and test sets. The distribution of samples in these training and test sets over 10 digit classes are given in Table 1.

2.2. Preprocessing

For cleaning of possible noise in the input image, it is first binarized by Otsu's thresolding technique followed by its smoothing using median filter of window size 5. No size normalization is done at the image level since it is taken care of during feature extraction. A sample image from the present database and the same after binarization and smoothing are shown respectively in Figs.2(*a*), 2(*b*) and 2(*c*).

Table 1. Distribution of samples in the database

Digits	Training Set	Test Set	Total
0	1842	369	2211
1	1890	378	2268
2	1890	378	2268
3	1881	377	2258
4	1875	375	2250
5	1888	378	2266
6	1868	374	2242
7	1868	378	2246
8	1886	377	2263
9	1885	378	2263
Total	**18773**	**3762**	**22535**

(*a*) (*b*) (*c*)

Figure 2. (*a*) An input numeral image, (*b*) image obtained after binarization of image in (*a*), (*c*) image obtained after smoothing of binarized image in (*b*).

2.3. Extraction of strokes

Let *A* be the input numeral image after its preprocessing. We extract vertical and horizontal strokes that are present in *A*. Such a stroke will be represented as a digital curve which is one-pixel thick and in which all the pixels except two have exactly two 8-neighbours, the other two pixels being the end pixels. In order to get the digital curves representing the vertical and horizontal strokes, two directional view based binary images from *A* are created as follows.

Let *E* be a binary image consisting of object pixels in *A* whose right or east 4-neighbour is a background pixel, that is, where the pen movement is upward or downward. In other words, the object pixels of *A* that are visible from the east form the image *E* as shown in Fig.3(*a*). Similarly, *S*, shown in Fig.3(*b*), is defined as the binary image consisting of object pixels in *A* whose bottom or south 4-neighbour is a background pixel, that is, where the pen movement is side-wise. The connected components in E are called vertical strokes while the connected components in S are called horizontal strokes. Vertical strokes consisting of pixels less than 20% of the height and horizontal strokes consisting of pixels less than 20% of the width of the input image are not considered for further processing. The final strokes (from among those shown in Fig.3) which are analyzed for extraction of shape features in the next stage, are shown in Fig.4.

(*a*) (*b*)

Figure 3. (*a*) E image consisting of vertical strokes obtained from the image in Fig.2(*c*), (*b*) S image consisting of horizontal strokes obtained from the image in Fig.2(*c*).

(*a*) (*b*)

Figure 4. (*a*) Final E image after removal of smaller vertical strokes from the image in Fig.3(*a*). (*b*) Final S image after removal of smaller horizontal strokes from the image in Fig.3(*b*).

2.4. Computation of shape feature

From each stroke in final E and S images, 6 scalar features are extracted. These features indicate the shape and position of a digital curve with respect to the numeral image. A curve C in E is traced from bottom upward. Suppose the bottom most and the top most pixel positions in C are P_0 and P_5 respectively. The 4 points $P_1, \ldots, P_4$ on C are found such that the curve distances between P_{i-1} and P_i (i=1, … , 5) are equal, using the algorithm in [12]. Let α_i, i=1, … , 5 be the angles that the lines $\overrightarrow{P_{i-1}P_i}$ make with the horizontal axis. Since the stroke here is vertical, $45^0 \leq \alpha_i \leq 135^0$. α_i (i=1,… ,5) are features that are invariant under scaling and represent the shape of C. The position feature (along the horizontal axis) of C is given by $\overline{X}$ which is the x-coordinate of the centre of gravity of the pixel positions in C. $\overline{X}$ is used to arrange the strokes present in an image from left to right. This representation of an image as a string of strokes will be useful in the HMM described later. The shape vector (SV) of the vertical stroke C is defined as $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5)$, which does not depend on the position and size of the stroke.

The features extracted from a horizontal stroke C in S are similar. Here C is traced from west to east. Suppose $Q_0, \ldots, Q_5$ are equidistant points on C such that Q_0 is the west most and Q_5 is the east most pixel on C. These points are found in the same way as P_i. Let β_i be the angles that the lines $\overrightarrow{Q_{i-1}Q_i}$ make with the x-axis. Since the stroke is horizontal, $-45^0 \leq \beta_i \leq 45^0$. β_i (i=1,… ,5), like α_i, are invariant under scaling. The shape vector (SV) of a horizontal stroke C is defined as $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5)$. $\overline{X}$ is defined in the same way.

3. Recognition Scheme

Several classification approaches can be considered based on the feature vector described in the previous section. However, hidden Markov models (HMM) and multilayer perceptrons (MLP) are the two most popular classification tools used in handwriting recognition tasks. Quantities computed at the output of both of these two classifiers provide an estimate of Bayesian a posteriori probabilities. So, the results of these two classifiers may naturally be combined to improve the recognition performance.

In the present scheme, the shape feature as described in Section 2.4 is further customized for feeding it to an HMM and an MLP classifiers. The results obtained as outputs of these two classifiers are combined using a second MLP. Final classification result is obtained based on the maximum score produced at the output layer of the second MLP. A block diagram of the proposed scheme is shown in Fig.5.

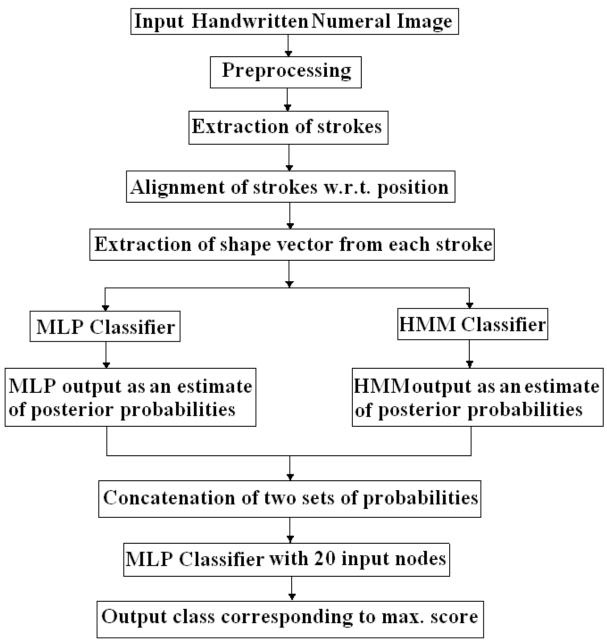

Figure 5. Block diagram of the proposed recognition scheme.

3.1. HMM classifier

The HMM used in the present work is non-homogeneous. Let us denote this HMM by $\boldsymbol{\gamma} = (\pi, A, B)$, where $\pi = \{\pi_i\}$ is the initial state distribution, $A(t) = \{a_{ij}(t)\}$ are the state transition probability distributions and $B = \{b_i\}$ is the distribution of observation symbols. $a_{ij}(t)$ is the probability of occurrence of the j-th state at time $t+1$ given the occurrence of the i-th state at time t. $b_i(O_t)$ is the observation symbol probability distribution for state i where O_t is the observation at time t.

In the present paper, we have considered 10 different HMMs denoted by $\boldsymbol{\gamma}_j$, $j = 1, 2, \ldots, 10$. For an input numeral pattern X of unknown class, we first compute the feature vector, i.e., the observation sequence $\mathbf{O} = O_1, \cdots, O_T$. Each O_i, $i = 1, 2, \ldots, 5$ is a shape vector, as described in Section 2.4, of a horizontal/vertical stroke after their arrangement in a left to right fashion determined by their $\overline{X}$ values. T is the total number of strokes in the image after ignoring smaller ones. The probability $P(\mathbf{O}/\boldsymbol{\gamma}_j)$ is computed for each model $\boldsymbol{\gamma}_j$. For a given $\boldsymbol{\gamma}$, $P(\mathbf{O}/\boldsymbol{\gamma})$ is computed using the well known forward and backward algorithms [13].

It may be noted that the states of the present HMM are certain shape primitives (more specifically, individual 5-dimensional Gaussian distributions in the space of shape vectors) that are found below using EM algorithm.

3.1.1. Obtaining state space of the HMM

Here we assume that the shape vectors $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$ (each $\boldsymbol{\theta}$ is either $\boldsymbol{\alpha}$ or $\boldsymbol{\beta}$ depending on whether the underlying stroke is vertical or horizontal) follow a multivariate Gaussian mixture distribution. In other words, $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$ has a distribution $f(\boldsymbol{\theta})$ which is a mixture of K 5-dimensional Gaussian distributions, namely, $f(\boldsymbol{\theta}) = \sum_{k=1}^{K} P_k\, f_k(\boldsymbol{\theta})$, where $f_k(\boldsymbol{\theta}) =$

$$\exp\{-0.5(\boldsymbol{\theta} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu}_k)\} / \{(2\pi)^{5/2} |\boldsymbol{\Sigma}_k|^{1/2}\}$$

and P_k is the prior probability of the k-th component of the mixture distribution. The unknown parameters of the mixture distribution, namely, P_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ $(k = 1, \ldots, K)$ are estimated using the EM (Expectation Maximization) algorithm [14] that maximizes the log likelihood of the whole set of observed samples (shape vectors) $\{\boldsymbol{\theta}_i, i = 1, 2, \cdots, n\}$ coming from the distribution given by $f(\boldsymbol{\theta})$. This whole set of observed samples is obtained by pooling all the horizontal and vertical strokes that have been obtained from all the E and S images corresponding to all of the training samples of each numeral in our handwritten numeral database. The EM algorithm is employed separately for 10 numerals. Let K_j, $j = 1, 2, \cdots, 10$ be their K-values.

The state space of the HMM $\boldsymbol{\gamma}_j$ consists of K_j states characterized by K_j 5-dimensional Gaussian probability distributions. These K_j distributions are called <u>shape primitives</u> for the numeral class j.

To determine the optimum value of each K, we use the Bayesian information criterion (BIC) [15] which is defined as $BIC(K) \equiv -2L + m \log(n)$, for a Gaussian mixture model with K components, L is the log likelihood value, m is the number of independent parameters to be estimated, n is the number of observations. For several K values, the $BIC(K)$ values are computed. The first local minimum indicates the optimum K value. The optimum K value is found for each of the 10 classes. Thus, the states here are not determined a priori but are constructed adaptively on the basis of the training set of handwritten numeral sample images.

3.1.2. Estimation of HMM parameters

In each proposed HMM, the observation symbol probability distribution $b_i(O_t)$ is, in fact, the Gaussian distribution $f_i(\theta) = N(\mu_i, \Sigma_i)$. The parameters produced by EM algorithm are $P_1, \cdots, P_K, \mu_1, \cdots, \mu_K, \Sigma_1, \cdots, \Sigma_K$. For each stroke O_t in an observation sequence, compute $h_i(O_t) = \dfrac{p_i b_i(O_t)}{\sum_{j=1}^{N} p_j b_j(O_t)}$ and O_t is assigned to state k where $k = \arg\max_{1 \le i \le N} \{h_i(O_t)\}$. This assignment to respective states is done for all L observation sequences (L is the number of training images). From these L state sequences, the estimates of the initial probabilities π_i and the transition probabilities $a_{ij}(t)$ are obtained. These HMM parameter estimates are fine-tuned using re-estimation by Baum-Welch forward-backward algorithm [13].

3.2. MLP classifiers

Use of ANN in handwritten character recognition tasks has become popular because the ANN tools (specially, the MLP classifiers) perform efficiently when input data are affected by noise and distortions. Also, the parallel architecture of a network model and its adaptive learning capability are added advantages.

To prepare the feature vector from a numeral image to be fed to an MLP classifier, we have considered the frequency distributions of the number of horizontal and vertical strokes present in the images of the training set. It is seen that in about 99.66% of the images, the number of horizontal strokes is less than 7 and the number of vertical strokes is less than 5. Consequently, we have considered only the first 6 horizontal strokes and the first

4 vertical strokes from the left (found using the $\overline{X}$ values). The training and the actual classification by the MLP are based on the information that these 10 strokes contain about a numeral image pattern. The feature vector has 50 components of which the first 30 are obtained by concatenating the shape vectors of the first 6 horizontal strokes and the last 20 are obtained by concatenating the shape vectors of the first 4 vertical strokes. If the number of horizontal and/or vertical strokes obtained from an input numeral image falls short, the respective positions of the above feature vector are filled with 150, an impossible value in the present context.

3.2.1. MLP architecture

In the first stage of the present recognition scheme, we used a 5-component shape vector representing each of the 10 strokes extracted from an input numeral image as described above. Thus, the MLP classifier of the first stage consists of 50 nodes at the input layer. On the other hand, in the second stage of our recognition scheme, an MLP classifier combines two sets of posterior probabilities for 10 numeral classes. So, this latter MLP consists of 20 nodes in the input layer.

As far as the hidden layer size is concerned, there exist several methods [16] for its optimal choice. However, in the present implementation, we performed simulation studies with different number of hidden nodes and the best recognition results among those choices have been shown in Section 4. Finally, both the MLP classifiers of the proposed recognition scheme, has 10 nodes in the respective output layers corresponding to the 10 numeral classes.

3.2.2. Criterion for termination of training

Overtraining is a common problem of the backpropagation algorithm used for MLP training. Here we have used a validation set [17] of samples for determining the optimal amount of training of the MLP classifiers. We have randomly chosen 250 samples from each class of the training set described in Table 1 and taken them out to form the validation set. Thus, we have used a training set of 16,273 samples and a validation set of 2500 samples. Usually, during initial training period, the system error on both the training and validation sets decreases monotonically. However, after a certain amount of training, the error on the validation set starts increasing although the same on the training set keeps reducing. This indicates the time when overtraining of the MLP classifier starts and its generalization capability starts reducing. We stop training of an MLP classifier when for the first time it is found that system error on the validation set increases for three consecutive sweeps. The connection weight values before this error starts increasing are considered final.

3.3. Combination of classifiers

Often, in handwriting recognition tasks, recognition results can be improved by combining multiple classifiers [11]. There exist various schemes in the literature [18] for classifier combination. In [19], majority voting was used for combination of multiple MLP classifiers to recognize handwritten Bangla (the second most popular Indic script) numerals. In the present work, we have used an MLP classifier for combining outputs from an MLP and an HMM classifiers.

In the first stage of the present recognition task, both the individual classifiers produces outputs at measurement level. Thus, the outputs from the HMM and the MLP classifiers may be considered as two feature vectors. From this point of view, an MLP classifier seems to be suitable to recognize the input numeral using the concatenated output vectors of the HMM and the MLP classifiers of the first stage. Such an MLP based combination of the MLP and HMM classifier outputs has provided improved recognition accuracy in the present problem.

4. Experimental Results

The training, validation and test databases of handwritten Devanagari numerals, used in the present simulation, consists of respectively 16273, 2500 and 3762 sample images of all the 10 classes. From the training sample images, 52274 horizontal and 35867 vertical strokes have been extracted as described in Section 2.3. The distributions of these two types of strokes over 10 numeral classes of the training set are shown in Table 2. The parameters of the HMM corresponding to a numeral class are estimated using shape vectors computed from its set of strokes as described in Sections 3.1.1 and 3.1.2.

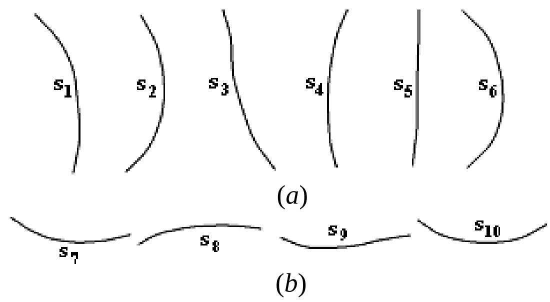

Figure 6. Shape primitives of the numeral class '7' (*a*) for vertical strokes and (*b*) for horizontal strokes.

For example, for numeral class "7", the '*K*' value is found to be 10. The shape primitives corresponding to the 10 mean vectors $\boldsymbol{\mu}_k$ are shown in Fig.6.

Correct recognition rates of the HMM classifier on the training and test sets are respectively 90.29% and 87.69%.

Recognition rate of the MLP classifier (100 hidden nodes) of the first stage are 94.15% and 90.46% on training and test sets respectively. Final recognition accuracies obtained by the MLP classifier (15 hidden nodes) of the second stage are respectively 95.64% and 92.83%.

Table 2. Distribution of horizontal and vertical strokes in the training set

	No. of Horizontal Strokes	No. of Vertical Strokes	'K' Value
०	3762	3395	15
१	3280	3961	13
२	4710	2185	12
३	6298	5772	17
४	6286	4097	17
५	4997	3818	13
६	6529	3740	17
७	4304	3717	10
८	5060	2903	10
९	7048	2279	12
Total	52274	35867	136

Table 3. Final confusion matrix on the test set

	0	1	2	3	4	5	6	7	8	9	Correct Recog(%)
०	348	3	0	0	8	0	1	7	0	2	94.30
१	5	354	3	0	6	0	5	3	0	2	93.65
२	7	6	346	5	3	1	0	4	0	6	91.53
३	1	2	8	354	4	3	5	0	0	0	93.89
४	8	0	1	1	360	2	3	0	0	0	96.00
५	0	4	9	3	5	347	2	3	4	1	91.79
६	0	4	1	5	9	2	351	0	1	1	93.85
७	9	0	6	3	7	5	8	330	3	7	87.30
८	0	5	0	4	6	2	3	0	350	7	92.83
९	2	3	4	0	2	6	5	0	4	352	93.12

5. Conclusions

In the present work, we extracted horizontal and vertical strokes from handwritten numeral images and computed a shape feature vector for each such stroke. These feature vectors were used along with HMM and MLP classifiers for recognition of handwritten Devanagari numerals. Combination of two sets of posterior probabilities obtained as outputs of these classifiers provided improved recognition performance. In future, we plan to use both the size and positional information of the strokes as additional features for further improvement in recognition results.

References

[1] U. Bhattacharya and B. B. Chaudhuri, "Databases for research on recognition of handwritten characters of Indian scripts", *Proc. of the 8th ICDAR*, Seoul, 2005, pp. 789-793.

[2] U. Bhattacharya, B. B. Chaudhuri, R. Ghosh and M. Ghosh, "On Recognition of Handwritten Devanagari Numerals", *Proc. of the Workshop on Learning Algorithms for Pattern Recognition* held in conjunction with the 18th AI, Sydney, 2005, Australia.

[3] I.K. Sethi and B. Chatterjee, "Machine recognition of constrained handprinted Devanagari". *Pattern Recognition*, Vol. 9, 1977, pp.69-75.

[4] K.R. Ramakrishnan, S.H. Srinivasan and S. Bhagavathy, "The independent components of characters are 'Strokes'", *Proc. of the 5th ICDAR*, 1999, pp. 414-417.

[5] R. Bajaj, L. Dey and S. Chaudhuri, "Devanagari numeral recognition by combining decision of multiple connectionist classifiers". *Sadhana*, Vol. 27, Part 1, 2002, pp. 59 – 72.

[6] G.S. Lehal and Nivedan Bhatt, "A recognition system for Devnagri and English handwritten numerals", *Advances in Multimodal Interfaces– ICMI 2001*, T. Tan, Y. Shi and W. Gao (Editors), LNCS, Vol. 1948, 2000, pp. 442-449.

[7] U. Bhattacharya, T.K. Das, A. Datta, S.K. Parui and B. B. Chaudhuri, "A hybrid scheme for handprinted numeral recognition based on a self-organizing network and MLP classifiers", *IJPRAI*, Vol. 16(7), 2002, pp. 845-864.

[8] M.B. Sukhaswami, P. Seetharamulu and A.K. Pujari, "Recognition of Telugu characters using neural networks", *Int. J. Neural Syst.*, Vol. 6, 1995, pp.317-357.

[9] Huanfeng Ma, David S. Doermann, "Adaptive Hindi OCR using generalized Hausdorff image comparison". *ACM Trans. Asian Lang. Inf. Process*. Vol.2, 2003, pp. 193-218.

[10] H. Park and S. Lee, "Off-line recognition of large-set handwritten characters with multiple hidden Markov models", *Pattern Recognition*, Vol.29, 1996, pp. 231-244.

[11] L. Lam, Y. S. Huang, C. Y. Suen, "Combination of Multiple Classifier Decisions for Optical Character Recognition", in *Handbook of Character Recognition and Document Image Analysis*, Eds. H. Bunke and P. S. P. Wang, World Scientific Publ. Comp., 1997. pp. 79-101.

[12] S K Parui and D Dutta Majumder, "Shape similarity measures for open curves", *Pattern Recognition Letters*, Vol.1(3), 1983, pp. 129-134.

[13] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proc. Of the IEEE*, 77(2), 1989, pp. 257–285.

[14] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, San Diego, 2nd Ed., 1990.

[15] J. Bernardo and A. Smith, *Bayesian Theory*. John Wiley & Sons, 1994.

[16] Y. Hirose, K. Yamashita, and S. Hijiya, Back-propagation algorithm which varies the number of hidden units, *Neural Networks*, 4, 1991, pp. 61-66.

[17] M. H. Hassoun, *Fundamentals of Artificial Neural Networks*, Cambridge: The MIT Press, 1995, pp. 226.

[18] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their application to handwritten numeral recognition," *IEEE Trans. On SMC*, Vol. 22, 1992, pp. 418 -435.

[19] U. Bhattacharya, B.B. Chaudhuri, "A Majority Voting Scheme for Multiresolution Recognition of Handprinted Numerals", *Proc. 7th ICDAR*, Vol. I, 2003, pp. 16-20.