

Off-line Recognition of Hand-written Bengali Numerals using Morphological Features

Pulak Purkait and Bhabatosh Chanda
ECSU, Indian Statistical Institute, Kolkata, India
{pulak_r, chanda}@isical.ac.in

Abstract

This paper proposes a technique for automatic recognition of Bengali handwritten numerals using multiple feature sets. We discuss about some novel Morphological features and k-curvature feature extraction technique to recognize handwritten scripts. We use different multi-layer perceptron (MLP) classifiers to train this feature spaces and then fuse those classifiers using modified 'Naive'-Bayes combination to increase accuracy of recognition result. The individual feature sets give reasonably high accuracy up-to 96.25% while fused classifier gives accuracy of 97.75%.

Keywords: Bengali numerals, multi-layer perceptron, 'Naive'-Bayes classifier, Morphology, curvature.

1. Introduction

Selection of features in image analysis task is one of the most important factor in achieving high recognition performance. Numerous features were proposed for handwritten as well as printed numeral recognition in last few years. Among these features, projection features, directional features, chain code histogram, crossing and distances features, zoning features and moment features are widely used and have been previously proved to be useful to a large extent. Morphological algorithms are developed in 80's but there are no significant study on the usefulness of morphological features in recognition of handwritten documents. In this paper we elaborate some morphological features, which are deemed to be very efficient in the context of recognizing Bengali numerals.

We propose three sets of morphological features and k-curvature features, which are found good enough for the purpose. Individually MLP classifier can recognize handwritten numeral based on these features quite efficiently. All the classifier are mutually independent and finally, are fused to achieve even higher performance.

In pattern recognition systems fusion of multiple simple classifiers with different feature sets may give more accurate result compared to a single complicated classifier. Classifier fusion usually assumes that all classifiers are trained over the whole feature space, and are considered as competitive classifiers. Here we have modified 'Naive'-Bayes decision while combining the classifiers as it gives much better decision if the classifiers used mutually independent features subspaces.

In the next section we have highlighted some of the past works on recognition of handwritten numerals. In section 3 we discuss the technique used for pre-processing followed by Morphological Feature sets and methods to extract them. Design of simple MLP based classifiers and fusing them using modified 'Naive'-Bayes combination is presented in section 4. Finally, the experimental results and concluding remarks are given in section 5 and 6 respectively.

2. Related Work

Research on Bengali numerals recognition is started in early 90's, even though there are no significant work in mid 90's. Around 2003's and later various approaches have been proposed by the researchers for numeral recognition on Indian scripts.

Sethi and Chatterjee [13] was among the first researchers to develop automatic recognition scheme for hand printed indian scripts. Among later studies, Ramakrishnan [14] considered independent component analysis for feature extraction from handwritten Devanagari numeral images. To take care of variability involved in the writing style of different individuals, Pal and Chaudhuri [10] proposed a robust scheme for the recognition of isolated Bengali off-line handwritten numerals. The scheme is based on features obtained from the concept of water over flow from the reservoir, as well as topological and statistical features.

Dutta and Chaudhuri [5] developed a system for recognition of isolated Bengali alphanumeric handwrit-

ten characters using neural networks. The characters were represented in terms of the primitives and structural constraints. The primitives were characterized on the basis of the significant curvature events like curvature maxima, curvature minima and inflectional points observed in the characters.

Bhattacharya et al. [3] have used a topology adaptive self-organizing neural network to extract the skeletal shape from a numeral pattern represented as a graph. Certain features like loops, junctions, etc. present in the graph are considered to classify a numeral into a smaller group. Finally, multi-layer perception networks are used to classify different numerals uniquely.

The early work in combining decisions of multiple classifiers was done by Bajaj et al. [1] for an Indian script. They represented different numerical object by concatenation of strokes and then calculated strength, density and moment as the feature set. Then they trained different networks and combined them for increasing reliability of the recognition results.

Hanmandlu and Murthy [6] proposed a fuzzy model-based scheme for recognition of handwritten English and Devanagari numerals. This fuzzy sets are derived from the features consisting of normalized distances obtained using the box approach. The membership function is modified by two structural parameters that are estimated by optimizing the entropy subject to the attainment of membership function to unity.

In recent work Bhattacharya and Chaudhuri [2] proposed a multi-resolution wavelet analysis method for feature extraction and a complicated multilevel classifier for recognition of Devanagari and bengali numerals. They used chain code histogram as feature on each of wavelet sub-bands in different resolutions, multi-layer perceptron (MLP) neural networks are trained for classification on each resolution, and then combined the classification results by majority voting technique.

3. Recognition Scheme

3.1. Pre-processing

Usually unconstrained handwritten numerals when scanned to a fixed resolution, produces images of different size, aspect ratio as well as variable stroke thickness. Paper and ink quality along with A/D converter may introduce non-negligible noise. So our first task is to resize all the image into a fixed size with fixed average thickness. There are lots of pre processing technique available in the literature. We used morphological filtering technique to get smooth binary image.

Input to our proposed recognition system are gray-level images of isolated handwritten numerals. The

numerals represented by the images may be of different sizes having different stroke thickness and limited variation in orientation. So first we resize images to a fixed size of 60×60 and then binarize the image using Otsu's [9] method minimizing inter class difference. Followed by morphological opening and closing with a small size structuring element to remove background noise and also to make the image more compact (removing holes inside the numeral object). Since each numeral object has ribbon like structure, we need to make average thickness fixed for all objects. A morphological skeleton and pruning are followed by a morphological dilation with fixed size structuring element. Here we have chosen disk as structuring element and radius of the disk as the average thickness of the numeral. Some features are extracted from the skeleton of the numeral images and some obtained from the dilated output image. Small rotation of the numerals with respect to image frame is ignored to reduce computational cost.

	original	binarize	resize	skeleton	dilation
Zero					
one					
two					
three					
four					
five					
six					
seven					
eight					
nine					
	(a)	(b)	(c)	(d)	(e)

Figure 1. The (a)first column is the original numeral image, (b) after binarize (c)applying some morphological filtering algorithm and resizing (d) skeletonization (d) Dilation of skeleton image

3.2. Feature Extraction

Morphological Features : In this paper we propose some new morphological features and a set of k-curvature feature. The feature extraction is done as follows :

1. Apply morphological opening on the pre-processed numeral image with line structuring element in four directions horizontal, vertical, and two diagonal directions to get four images. Each open image represents mass content of the numerals in four directions. Let $[S^1, S^2, S^3, S^4]$ be four linear structuring element in four directions. Here length of the structuring element S^i is twice the average thickness of the numeral. Let X be the preprocessed image, and X_i be the image obtain from the morphological opening of X by the structuring element S^i .i.e. $X_i = X \circ S^i, i = 1, \dots, 4$, where \circ denotes morphological opening. Then divide each of the images X_i into 36 blocks, each of size 10×10 . Calculate mass ($m_i^k, k = 1, \dots, 36$) of portion of the numeral inside each blocks. Normalization is done by dividing each value by the maximum value, i.e. normalized value $nm_i^k = m_i^k / \max_{k=1, \dots, 36}(m_i^k), i = 1, \dots, 4; j = 1, \dots, 36$, to get (36×4) 144 dimension feature vector $X = [nm_1, nm_2, nm_3, nm_4]^T$ where $nm_i = [nm_i^1, nm_i^1, \dots, nm_i^{36}], i = 1, \dots, 4$.

2. Second morphological feature set is obtained from the preprocessed image by the first applying morphological closing operation in four directions and then apply same procedure as before. Here close images are obtained as $X_i = X \bullet S^i, i = 1, \dots, 4$, where \bullet denotes morphological closing operation and $\{S^i, i = i, \dots, 4\}$ are the same structuring elements. Then dividing each close image X_i into 36 blocks of size 10×10 and histogram of each block is calculated to get 144 dimension feature vector as before.

In the following figure we have shown different numerals and corresponding results of directional opening and the closing. Figure 2.(a) shows result of directional opening of numeral images and figure 2.(b) shows that for the directional closing.

3. Another morphological feature is obtained from the pre-processed skeleton image \hat{X} . Skeleton image is eroded by the diagonal structuring element shown in figure 3(a), to get the number of co-occurrences of skeletal pixels in diagonal direction. Other directional co-occurrences are obtained

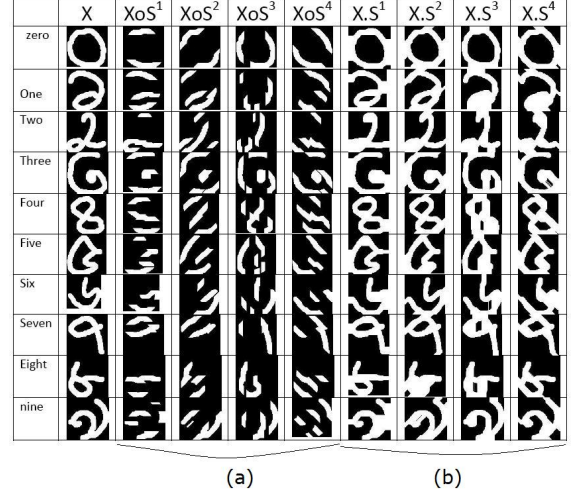


Figure 2. Directional morphological (a) Opening and (b)closing images in four directions

by using morphological erosion of the skeleton image by the structuring elements shown in figure 3(b)-3(d).



Figure 3. Four structuring element (a)-(d) with center as the first element used to find the images containing the portion of skeleton in the respective direction.

Here structuring elements are $S^1 = \{(0, 0), (1, 1)\}$, $S^2 = \{(1, 0), (0, 1)\}$, $S^3 = \{(0, 0), (1, 0)\}$, and $S^4 = \{(0, 0), (0, 1)\}$. Each image $\hat{X}_i = X \ominus S^i, i = 1, \dots, 4$ contains the pixels at which a particular directional co-occurrences on skeleton occur, where \ominus is the morphological erosion operation. Then we divide each image \hat{X}_i in 10×10 blocks and make histogram of the directional co-occurrences ($s_i^j, j = 1, \dots, 36$) of portion of skeleton of each block. Then the histogram frequency is normalized by dividing maximum value, i.e. normalized value $ns_i^j = s_i^j / \max_{j=1, \dots, 36}(s_i^j), i = 1, \dots, 4; j = 1, \dots, 36$, to get (4×36) 144 dimension feature vector $X = [ns_1, ns_2, ns_3, ns_4]^T$ where $ns_i = [ns_i^1, ns_i^1, \dots, ns_i^{36}], i = 1, \dots, 4$.

K-Curvature Feature : The slope of a chain code at any point is a multiple of 45° , and the slope of a crack code is a multiple of 90° . In order to measure more continuous range of slope, we use some type of smoothing over, say, k such chain code to define k -curvature [12].

Here left and right k -slopes at a point P on a curve are defined as the slopes of the line joining P to the points k steps away along the curve on each side of P , and k -curvature of P as the difference between its left and right k -slopes. In this definition it is assumed that the curve from P to the fixed k -steps away can be approximated by straight line segment. This assumption is better satisfied if k is small. On the other hand, in the case of small k , the slope may be influenced by small perturbation due to noise. So value of k is selected compromising these two conflicting characteristics of slope measure.

We calculate k -slopes at each point of the morphological skeleton of the numeral object as the angle with respect to horizontal axis, and hence k -curvature as acute angle between left and right segment at P . Usually the value of k -curvature lies between 0° to 180° , we quantize the interval $[0^\circ, 180^\circ]$ into six unequal bins as follows : $0^\circ \leq \theta < 90^\circ$, $90^\circ \leq \theta < 120^\circ$, $120^\circ \leq \theta < 140^\circ$, $140^\circ \leq \theta < 160^\circ$ and $160^\circ \leq \theta \leq 180^\circ$. We take $k = 4$ (or sometimes $k = 5$), and observe that the value of k -curvature in most of the pixels lies between 90° and 180° , with a bias towards 180° . That is why we take finer bias in the interval after 90° at the time of quantization. We make histogram on the basis of this distribution. Now there are four possibility which are shown in the figure 4(a)-(d). The histograms of k -curvature on each of them are nearly identical.

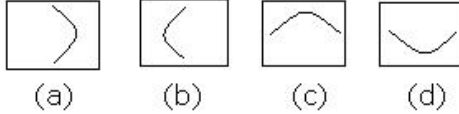


Figure 4. Four possibilities (a)-(d) where histogram of k -curvature are identical.

So we need to distinguish among the histogram of curve segments shown in above figure. The first two i.e. (a)-(b) and the rest two i.e. (c)-(d) can be separated by the difference of the row number and column number of the end points of the curve segments. Then they may be further distinguished by checking the location of mid-point with respect to the straight line joining the end points of the curve segments. Then we divide the image into 12×12 size blocks and calculate the normalized histogram having (5×4) 20 bins for each block to get $(25 \times 5 \times 4)$ 500 dimension at feature vector.

After calculating feature vectors we use Principle Component Analysis (PCA) to each of the feature vectors to reduce dimension equal to 75.

4. Classifier Design

We use feed-forward Multi-layer perceptron model (MLP) for the classifier to train the feature sets. We have taken only one hidden layer consists of 30 nodes and sigmoid function as activation function. Individual classifier are taken to train one feature set at a time.

For fusion of classifiers, we have used a set of four MLP classifiers and fused the classifiers using ‘Naive’-Bayes combination.

‘Naive’-Bayes combination (NB) : This scheme assumes that the classifiers are mutually independent. This is why Ludmila [8] use the name naive. Xu et al. [15] call it Bayes combination. They apply Naive-Bayes combination for crisp label at the output of the classifiers at the time of fusion. We modify this for soft label vector for output of the classifiers.

Let $x \in \mathfrak{R}^n$ be a n -dimensional feature vector and $\{1, 2, \dots, c\}$ be the label set of c classes. Every mapping $[D : \mathfrak{R}^n \rightarrow [0, 1]^c - \{\mathbf{0}\}]$, where $\{\mathbf{0}\} = [0, 0, \dots, 0]^T$ is the origin of \mathfrak{R}^c , is treated as a classifier. We call the output of D a ‘class label’ and denote it by $\mu_D(x) = [\mu_D^1(x), \dots, \mu_D^c(x)]^T$, $\mu_D^i \in [0, 1]$. Given the feature vector x , the components $\mu_D^i(x)$ may be regarded as (estimates of) the posterior probability for the i^{th} classes, i.e. $\mu_D^i = P(i|x)$ for $i = 1, \dots, 4; j = 1, \dots, c$. The decision of D is hardened so that a crisp class label in $\{1, 2, \dots, c\}$ is assigned to a feature vector x of the training set. This is typically done by the *maximum membership rule* :

$$D(x) = k \iff \mu_{D(x)}^k = \max_{i=1, \dots, c} \{\mu_{D(x)}^i\} \quad (1)$$

Let $\{D_1, \dots, D_L\}$ be the set of L classifiers. We denote the output of the i^{th} classifier as $D_i(x) = [d_{i,1}(x), d_{i,2}(x), \dots, d_{i,c}(x)]^T$, where $d_{i,j}(x)$ is the degree of ‘support’ given by the classifier D_i to the hypothesis that x comes j . For each classifier D_j , a $c \times c$ confusion matrix CM_j is calculated by applying D_j to the training data set. The $(k, s)^{th}$ entry of this matrix, $cm_{k,s}^j$ is the number of elements of the data set whose true class label is k , but is assigned to class s by the classifier D_j .

By $cm_{.,s}^j$ we denote the total number of objects labeled by D_j into class s . This is calculated as the sum of the s^{th} column of CM_j using these values, a $c \times c$ label matrix LM_j is computed, whose $(k, s)^{th}$ entry $lm_{k,s}^j$ is an estimate of the probability that the true label is k given that D_j assigns crisp class label s .

$$lm_{k,s}^j = \hat{P}(k|D_j(x) = s) = \frac{cm_{k,s}^j}{cm_{.,s}^j} \quad (2)$$

For every feature vector $x \in \mathfrak{R}^n$ of test set, we take output of D_j as a soft label vector $D_j(x)$ to make classifier more flexible to take final decisions instead of crisp label class what Ludmila [8] and Xu et al. [15] have taken in their work. Here the vector $D_j(x)$ associated with the class s is a soft label vector $[\hat{P}(1|D_j(x) = s), \dots, \hat{P}(c|D_j(x) = s)]^T$, which is the s^{th} column of the label matrix LM_j . Let $D_1(x), \dots, D_L(x)$ be the soft class labels assigned to x by L classifiers D_1, \dots, D_L , respectively obtained by normalizing the output of the classifiers. Since the output of classifier are independent of each other, so by independence assumption for the feature sets, the estimate of the probability that the true class label is i , (which is the i^{th} component of the final label vector) is calculated by

$$\mu_{\hat{D}(x)}^i = \prod_{j=1}^L \left(\sum_{s=1}^c d_{j,s}(x) \hat{P}(i|D_j(x) = s) \right), \quad i = 1, 2, \dots, c \quad (3)$$

i.e.

$$\mu_{\hat{D}(x)}^i = \prod_{j=1}^L \left(\sum_{s=1}^c lm_{i,s}^j d_{j,s}(x) \right), \quad i = 1, 2, \dots, c \quad (4)$$

Then we finally assign to the test objects the class labels for which $\hat{D}(x)$ has maximum value, where $\mu_{\hat{D}(x)}^i$ is the probability that the object x belongs to class i as suggested by the output of the fused classifier $\hat{D}(x)$. i.e. class label is given by

$$s = \max_{i=1, \dots, c} \{ \mu_{\hat{D}(x)}^i \}$$

Figure-5 shows a schematic block diagram of 'Naive'-Bayes fusion technique. Classifier D_j generate an output $D_j(X)$ when a feature vector X corresponding to a test image is given as an input. LM_j 's are the label matrices obtained from the training images using different classifiers as discussed in previous paragraph. Multiplying D_j by LM_j we get a column vector for each classifier. The numeral guess of the fused classifier will be the index of the point-wise product (element-wise product) of all the column vectors.

5. Experimental Results :

We have implemented our scheme using MATLAB and tested on a public domain data-set. In the present work we use Bengali numerical database developed by

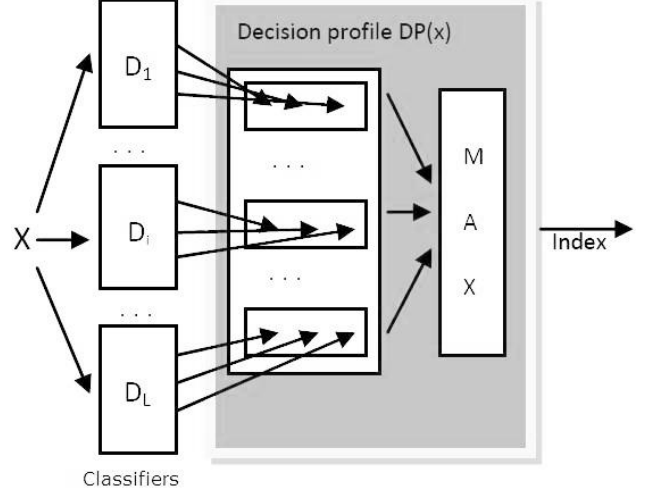


Figure 5. Block diagram of modified 'Naive'-Bayes Fusion technique.

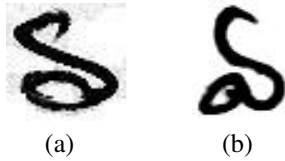
Bhattacharya and Chaudhuri [2]. This database contains numerals of different size with different thickness of numeral objects. It consists of 19,392 training and 4,000 test samples among them 2,000 training samples taken for validation set (200 for each class). Here for Bengali numerals we have 10 classes ($c = 10$) and 4 first level classifier ($L = 4$) to train 4 different feature sets. As the optimization of parameters is one of the important tasks for any classifier, we have done extensive experiment and set the following parameters for the entire investigation as : number of nodes on the hidden layer is 30, number of iteration is 20, learning rate equal to 0.1 and momentum factor equal to 0.05. Details of recognition rate of Morphological and k-curvature feature sets are depicted in table-1.

As shown in the table-1 with each individual, we have got nearly same performance. Although morphological directional opening feature sets giving the best performance (96.25%) among the four feature sets. The diversity of feature set demands fusion of classifier. Taking these points into account, we have applied a modified 'Naive'-Bayes combination and got a recognition rate of 97.75% which is almost equal to the best existing result (98.20%) on this database obtained from a complicated multistage classifier [2].

Also, it may be noted that performance of the system is worst for 'one'(ek) and 'nine'(noy) among others. The system has confused between these two numerals as their shape are very close as depicted by an example shown in the figure-6.

Table 1. Maximum recognition rate obtained using different feature sets with MLP classifier.

Numerals	Directional opening	Directional closing	Directional erosion	k-curvature	Fusion of classifiers
0	99.5%	97.5%	98.75%	96.25%	99.5%
1	91.75%	92.5%	88.25%	91.75%	93%
2	97.75%	96.5%	97.25%	95.75%	98.75%
3	96.25%	90.5%	93.75%	92.75%	98%
4	99.25%	98.5%	99.25%	97%	99.75%
5	95.25%	96.25%	93.25%	96.5%	97.5%
6	95%	96.75%	96%	95%	96.75%
7	98.5%	99%	99.25%	98.75%	99.75%
8	98.5%	99%	99.5%	98.5%	99.5%
9	90.25%	92.25%	91.75%	90%	94.5%
average	96.25%	95.87%	95.75%	95.25%	97.75%

**Figure 6. (a) 'one'(ek) having same shape as (b)'nine'(noy).**

6. Conclusion

Here we have studied some morphological feature and k-curvature feature sets and their usefulness in recognition of ISI Bengali numeral data-set. We have explored how morphological features could be useful for handwritten numeral recognition. We also have modified existing 'Naive'-Bayes (NB) combination from crisp label classification to soft label classification technique for further fusion of classifiers. The results are very satisfactory.

Acknowledgement : We would like to thank Dr. Ujjwal Bhattacharya of *Indian Statistical Institute* for providing ISI Bengali numeral data-set used in this work.

References

- [1] R. Bajaj, L. Dey, and S. Chaudhury. Devnagari numeral recognition by combining decision of multiple connectionist classifiers. *Sadhana*, 27(Part 1):59–72, 2002.
- [2] U. Bhattacharya and B. B. Chaudhuri. Handwritten numeral databases of indian scripts and multistage recognition of mixed numerals. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(3):444–457, 2009.
- [3] U. Bhattacharya, T. K. Das, A. Datta, S. K. Parui, and B. B. Chaudhuri. Recognition of handprinted bangla

numerals using neural network models. *Springer Berlin / Heidelberg*, 2275:139–161, 2002.

- [4] B. Chanda and D. Majumdar. *Digital Image Processing And Analysis*. Prentice' Hall of India Private Limited, New Delhi, 2005.
- [5] A. Dutta and S. Chaudhury. Bengali alpha-numeric character-recognition using curvature features. *Pattern Recognition*, 26(12):1757–1770, 1993.
- [6] M. Hanmandlu and O. V. R. Murthy. Fuzzy model based recognition of handwritten numerals. *Some Fine Journal*, (40):1840–1854, 2007.
- [7] W. Lu, Y. Lu, Y. Pengfei, and S. Pengfei. Handwritten bangla numeral recognition system and its application to postal automation. *Pattern Recognition*., 40(1):99–107, 2007.
- [8] Ludmila and I. Kuncheva. Decision templates for multiple classifier fusion: An experimental comparison. *Pattern Recognition*, 34(2):299–314, 2001.
- [9] N. Otsu. A threshold selection method from gray-level histogram. *IEEE Trans. Systems Man, and Cybernetics*, 9:62–66, 1979.
- [10] U. Pal, A. Belaid, and B. B. Chaudhuri. A system for bangla handwritten numeral recognition. *IETE Journal of Research*, (3):444–457, 2006.
- [11] U. Pal and B. B. Chaudhuri. Indian script character recognition: a survey. *Pattern Recognition*, 37(9):1887–1899, 2004.
- [12] A. Rosenfeld and A. C. Kak. *Digital Picture Processing*, volume 2. Academic Press, 24/28 Oval Road, London NW1 7DX, 1982.
- [13] I. K. Sethi and B. Chatterjee. Machine recognition of constrained handprinted devnagari. *Pattern Recognition*, 9:69–76, 1977.
- [14] S. H. Srinivasan, K. R. Ramakrishnan, and S. Bhagavathy. The independent components of characters are 'strokes'. In *ICDAR '99: Proceedings of the Fifth International Conference on Document Analysis and Recognition*, page 414, 1999.
- [15] L. Xu, A. Krzyzak, and C. Y. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Trans. on Systems, Man and Cybernetics*, 22(3):418–435, 1992.