A New Quad Tree Based Feature Set for Recognition of Handwritten *Bangla* Numerals

Abhinaba Roy, Navonil Mazumder, Nibaran Das*, Ram Sarkar, Subhadip Basu, Mita Nasipuri

Computer Science and Engineering Department,
Jadavpur University

*Corresponding author, email:nibaran@gmail.com

***Abstract*— Recognition of handwritten Bangla numerals has always been an open problem for researchers. Selection of appropriate preprocessing and feature extraction techniques to achieve maximum recognition accuracy is a challenging problem. In this paper, a new Quad Tree based feature set is introduced for the recognition of handwritten Bangla numeral dataset developed here. On experimentation with the database of 4200 image samples using Support Vector Machine (SVM), the technique yields an average recognition rate of 93.338% evaluated after three-fold cross validation of results. The result is compared with recognition rate obtained from previously established standard dataset using the same feature set.**

Keywords: Bangla numerals; Preprocessing; Gradient Feature; Quad Tree Structure; Classification.

I. INTRODUCTION

Handwritten character recognition, an almost fifty year old research problem is still very much pertinent due to the enormous variations in writing styles among different writers. The task becomes more complex over the variations present in the glyphs of different scripts. Among different scripts, researches on *Roman* [1] [2] [3] [4], *Chinese* [5], *Japanese* [6] scripts have always been in focus for the last few decades. In recent times, Indic scripts such as *Bangla*, *Devanagari*, *Tamil*, and *Telugu etc* have been receiving a broad level of attentions among different pattern recognition research communities. Among different Indic scripts, *Bangla* is the second most popular script in Indian sub-continent. Various recognition methods have already been developed for recognition of handwritten *Bangla* numerals. Compares to the number of works, the availability of data set for handwritten Bangla numerals is very few. This motivates us to develop a new dataset of *Bangla* numerals and recognize them using a new set of Quad tree based gradient features.

II. PREVIOUS WORK

Among different recognition strategies for *Bangla* numerals, Liu et al. [7] set a benchmark result on ISI-Bangla dataset using gradient direction histogram feature. They applied Linear Normalization, Moment Normalization, Bi-moment Normalization for pre-processing in order to set benchmark results. Sobel operator was used in their work for the calculation of x and y component of local gradient vector at each pixel. The gradient vector was decomposed into discrete directions using parallelogram decomposition for getting the direction histograms. Recognition accuracies of 99.20 %, 98.90 %, 99.40 % was reported using SVM, MLP and CFPC classifier respectively. In the paper [8], Bhattacharya et al. proposed a majority voting approach based on multi resolution wavelet analysis for recognition of *Bangla* numerals. They obtained a recognition accuracy of 98.23% on the same data set. Das et al. [9] developed a GA-SVM based region sampling technique for recognition of handwritten Bangla numerals. They achieved a recognition accuracy of 97.70% on the dataset entitled CMATERdb 3.1.1 [10]. Pal et al. [11] reported 16 direction gradient histogram feature based recognition technique on handwritten numeral recognition of six Indian scripts, including *Bangla*. They used quadratic classifier for the recognition purpose and obtained a recognition accuracy of 98.99% on 5-fold cross validation. All the authors of the papers [7], [8], [9], [11], [12] used some image partitioning strategies for extraction of local features apart from the global features from the images.

III. PRESENT WORK

A. Dataset Preparation

In this present work, a new dataset of *Bangla* numerals, is developed. The dataset has been prepared at the Center for Microprocessor Applications for Training Education and Research (CMATER) laboratory, Jadavpur University, taking data samples from people of different age-group and sex and educational qualification, and hence effort has been made to reach a possible wide range of people. This in turn affects in understanding and measuring the impact of variations in writing styles on the recognition process. All these accumulated data sample sheets are optically scanned with the resolution of 300 dpi using a HP F380 flatbed scanner. From these data sheets, individual data images were extracted using a histogram based segmentation technique. A group of randomly selected samples from the dataset are shown in TABLE I.

B. Randomization of data samples

We have collected the data from 200 individuals. An individual may write more than one sample but not greater than four samples per character class. As consecutive data samples of an individual are written consecutively, it might put an adverse biasness in the final result. In order to remove that possibility, the entire dataset is randomized.

TABLE I. Sample *Bangla* data

Original	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
০					
১					
২					
৩					
৪					
৫					
৬					
৭					
৮					
৯					

Figure 1. Some normalized image sample

C. Pre Processing

1) Normalization

The images of the dataset are cropped and normalized into 32x32 pixels automatically using "Irfan view 4.27" software.

2) Skeletonization

Experimentation is done on the dataset using two different techniques, one with those normalized data and the other after applying skeletonization, a common pre-processing technique on the same data.

A fast and simple method for thinning originally proposed by Zhang- Suen [13] is used here. The method works in two sub-iterations. In each step, points of the region that can be deleted are identified. Here, a threshold value is used for marking all background pixels as '1' and all foreground pixels as '0'.

The method is described in details.

- 1st sub-iteration :

  A point p_1 is marked for deletion if all the 4 conditions are true

  1. $2 <= B(p_1) <= 6$;
  2. $A(p_1) = 1$;
  3. $p_2 \cdot p_4 \cdot p_6 = 0$;
  4. $p_4 \cdot p_6 \cdot p_8 = 0$;

- Delete marked points.
- 2nd sub-iteration :

  Same as the 1st sub iteration but 3rd and 4th conditions are only different.

  1. $2 <= B(p_1) <= 6$;
  2. $A(p_1) = 1$;
  3. $p_2 \cdot p_4 \cdot p_8 = 0$;
  4. $p_2 \cdot p_6 \cdot p_8 = 0$;

- Marked points are deleted.
- When at any sub-iteration, there is no point to delete, and then the skeletonization is complete.

Where,

A (p_1) = number of 0 to 1 transition in the order $p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_2$.

$B(p_1) = p_2 + p_3 + p_4 + p_5 + p_6 + p_7 + p_8 + p_9$.

Here, first sub-iteration removes south or east boundary points or north-west corner points whereas the second sub-iteration removes north or west boundary points or south-east corner points.

3) Binarization

The image samples are binarized in this step for the extraction of different feature values. We have used Otsu's method [14] to binarize the image samples. Here after binarization foreground images are represented as '1'and background images are represented as '0'.

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

Figure 2. $A(p_1)=2$

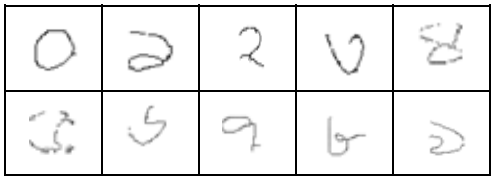


Figure 3. Few Skeletonized samples

IV. FEATURE EXTRACTION

A. Quad Tree Structure

From a data structural point of view, quad tree is a tree which has upto four children except the leaf node. The data structure is most useful in representaing a two dimensional area sub divided into four regions. In this present work we have used a modified version of quad tree structure to divide a pattern into four sub patterns. And that has been achieved by partitioning the area by a horizontal and vertical line passing through its Centre of Gravity (CG) of foreground pixels (black pixels). If the depth of a quad tree is L, then the total number of leaf nodes in that tree structure is 4^L. The coordinates of the CG (C_x, C_y) is defined as

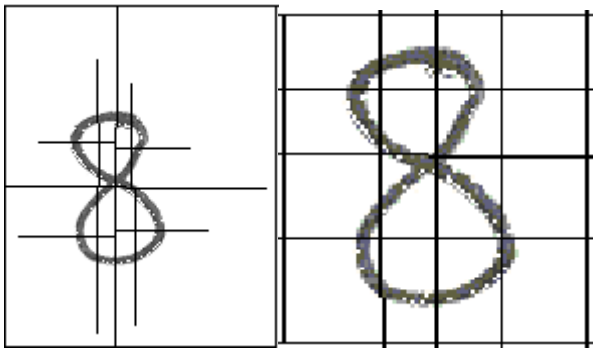
$$C_x = \frac{1}{mn}\sum_{mn} x.f(x,y)$$

$$C_y = \frac{1}{mn}\sum_{mn} y.f(x,y)$$

Where,

$$f(x,y) = \begin{cases} 0, & \text{for all background pixels} \\ 1, & \text{for all foreground pixels} \end{cases}$$

Here, (x, y) are the coordinates of the concerned pixels in a sub-image of size mXn. The difference between simple equal partitioning and CG based partitioning to produce a quad tree structure is shown in Figure 4. In most of the cases, equal partitioning may produce a less informative sub-image, whereas, quad tree based sub-division may be much more informative.



(a) (b)

Figure 4. Subdivision of an image sample based on
(a) Quad tree based subdivision
(b) Normal subdivision

B. 8 Directional Gradient Features

Gradient features can be calculated after applying different operators such as Krish mask, Robert's filter. Krish mask calculates the gradient in four directions where as Robert's mask (2×2 neighbourhood) calculates the x and y components of the local gradient vector. Among these, Robert's filter has been successfully used in [11]. After applying Robert's mask, the strength of the gradient values is defined as

$$f(x,y) = \sqrt{\Delta u^2 + \Delta v^2}$$

$$\Delta u = g(x+1, y+1) - g(x,y)$$

$$\Delta v = g(x+1, y) - g(x, y+1)$$

Here, g(x, y) = intensity at pixel (x, y). The direction is defined as,

$$\theta(x,y) = \frac{\Delta v}{\Delta u}$$

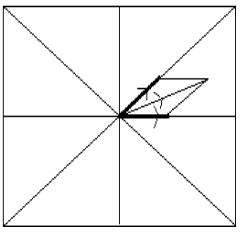


Figure 5. Parallelogram decomposition of gradient vector in 8 directions

Now the gradient vector is decomposed into two neighboring directions of the 8 directional map (Figure 5). This is continued for each pixel in the image. Sub images at the 2nd depth of the CG based quad tree structure (as discussed in the previous section) are created, as the image is sub divided into 16 sub-images. In every sub-image a chain-code count of the directions with components of the feature vector is added and then an average is taken into count. Thus from each subdivision 8 features are extracted and in total

128(i.e.16X8=128) feature values are obtained from an image.

C. *Longest Run Feature*

Within a rectangular image region of a character, longest run features [12] are computed in four directions; row wise, column wise and along the directions of two major diagonals. The row wise 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 of the region.

In fitting a bar with a number of consecutive black pixels within a rectangular region, the bar may extend beyond the boundary of the region if the chain of black pixels is continued there. The three other longest-run features within the rectangle are computed in the same way. Each of the longest run feature values is to be normalized by dividing it with the product of the height (h) and the width (w) of the entire image.

Here, at every quad-tree node 4 longest run features are extracted. Quad tree of depth 3 is used (Figure 4(a)), a root node, and 4 nodes at level 1 and 16 nodes at level 2. So in total 84 features (4 longest run features from each node, hence (16+4+1)X4=84) are extracted from an image.

V. SUPPORT VECTOR MACHINE

Support vector machine (SVM) [15], a well-known dichotomizer, has been used successfully in pattern recognition for last few years. Although SVM was initially developed for two-class classification problems by constructing an optimal hyper plane between the positive and negative datasets, it can be used for multi-class problems too.

For the SVM, an open source software LibSVM [16] tool is used. In general, a classification task usually involves a dataset which is used for training and testing purpose. Each data in the training set contains one "target value" (class labels) and several "attributes" (features). The goal of SVM is to produce a model which predicts target value of data in the testing set which are given only the attributes.Given a training set of instance-label pairs (x_i, y_i); i = 1,...,*l* where $x_i \in R^n$ and $y_i \in \{-1,1\}^l$, SVM require the solution of the following optimization problem:

$$\min_{w,b,\xi} \quad \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i$$

$$\text{subject to} \qquad y_i (w^T \Phi(x_i) + b) \geq 1 - \xi_i ,$$

Training vector x_i is mapped into a higher dimensional space using the function $\Phi(\ )$. Then SVM finds a suitable hyper plane in that higher dimension, which linearly separates the region with maximal margin. C (> 0) is the penalty parameter of the error term. Furthermore, $k(x_i, x_j) \equiv \Phi(w^T).\Phi(w)$ is called the kernel function. Here, RBF kernel is used and the corresponding expression is given below:

$$k(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2), \gamma > 0$$

Where, γ is the kernel parameter. The rationale behind the choice of RBF kernel is due to its ability to perform better for handwritten character recognition [17]. The value of gamma is chosen empirically during experimentation.

VI. EXPERIMENTAL RESULTS

Writing styles of different human beings may affect recognition process drastically. Keeping that fact in mind the present dataset has been prepared, consisting of 4200 images of handwritten data samples. For the preparation of training set 2/3rd of the data samples are selected first. Rest 1/3rd data samples are used as test set for the present work. Three such pairs of the training and the test sets are formed with the original dataset for the cross validation of results. The extracted features are used for classification purpose using SVM classifier. The recognition performance is compared with another quad tree based feature set [12] used by Basu et al. for recognition of Handwritten numerals of different Indic scripts. Results of experiments using the two different feature extraction methods are shown in TABLE II and TABLE III respectively.

Fold #1: First 280 (140 + 140) samples of a class are used as training set and last 140 samples of a class are used as test set.

Fold #2: First 140 samples of a class are used as test set and last 280(140 + 140) samples of a class are used as training set.

Fold #3: First 140 and last 140 samples (140 + 140=280)of a class are used as training set and middle 140 samples of a class are used as test set.

TABLE II. Recognition performance of Quad tree based 8 Directional Gradient features

	Fold#1	**Fold#2**	**Fold#2**	**Average**
Skeletonized image samples	89.931	90.017	**90.237**	90.062
Normal image samples	93.469	**93.574**	92.971	93.338

TABLE III. Recognition performance using Quad tree based Longest Run features

	Fold#1	Fold#2	Fold#2	Average
Skeletonized image samples	92.235	**92.541**	92.324	92.366
Normal image samples	94.684	**95.081**	94.99	94.918

TABLE IV. Recognition performance on present and ISI database Using two quad tree based feature sets

Data Set	Quad-tree based Longest Run Feature	Quad-tree based Gradient Feature
Present database	95.081	93.574
ISIBangla Dataset	97.35	96.975

TABLE V. Correctly classified sample using quad tree based 8 directional feature

sample	
Correct class	০ ১ ৩ ৫ ৭

We have evaluated the performance on the standard ISIBangla numeral dataset [18]. The recognition rate obtained from the database using two different feature sets are shown in TABLE IV.

It can be observed from TABLE II and TABLE III that features extracted from the normal dataset without using skeletonization provides better recognition accuracy than that obtained after applying the skeletonization on the same data samples. This can be attributed to the fact that after normalization, when skeletonization is done, some data became broken. The broken data hardly represent the original shapes of that class. Thus the recognition accuracy becomes lower. In TABLE V and TABLE VII few correctly classified samples using the Quad tree based 8 directional gradient Feature set are shown. In TABLE VI and TABLE VII some misclassified data are shown. Although great measure has been taken in order to maintain the standard way of writing numerals, due to variety of handwriting styles some of the data

TABLE VI. Misclassified data using Quad tree based 8 Direction Gradient Feature Set

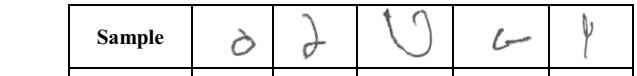
Sample	
Classified as	০ ১ ৩ ৮ ৭
Original class	১ ৮ ৬ ৩ ১

TABLE VII. Correctly classified data using Longest Run Feature

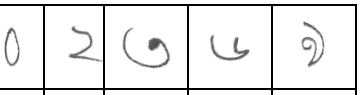
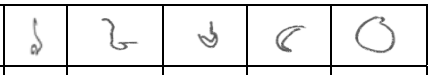
Sample	
Correct class	০ ২ ৩ ৬ ৯

TABLE VIII. Misclassified data using Longest Run Feature

sample	
Classified as	১ ৬ ০ ৪ ৫
Original class	৯ ৮ ৬ ৫ ০

samples in the dataset has come out as ambiguous. But again, that is the case with almost all handwritten datasets. From the TABLE IV we can be observed that ISI dataset provides better result than that of the newly introduced dataset. This clearly indicates the lack of universality within the databases in terms of writing styles, and variations among handwritten data samples.

Dataset preparation and maintenance are also of equal importance and has to be carried out properly and meticulously in order to increase the chances of recognition accuracy for a model. Introduction of new and better preprocessing techniques on the newly introduced dataset may boost the recognition accuracy in such a way that it can reach up to a benchmark. Even script specific stronger feature set can also improve the recognition rate. It is useful for applications related to OCR of handwritten Bangla Digit and can also be extended to include OCR of handwritten characters of Bangla alphabet.

REFERENCES

[1] R.M. Bozinovic and S.N. Srihari, "Off-line Cursive Script Word Recognition," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 11, pp. 68-83, 1989.

[2] C.-L. Liu, *et al.*, "Handwritten digit recognition: benchmarking of state-of-the-art techniques," *Pattern Recognition,* vol. 36, pp. 2271-2285, 2003.

[3] Y.S. Huang , C.Y. Suen, “A method of combining multiple experts for the recognition of unconstrained handwritten numerals,” IEEE Trans. PAMI, vol. 17, pp. 90-94,1995.

[4] Y. Xu and G. Nagy, “Prototype Extraction and Adaptive OCR,” IEEE Trans. PAMI, vol. 21, pp. 1280-1296, 1999.

[5] P.K. Wong and C. Chan, “Off-line Handwritten Chinese Character Recognition as a Compound Bays Decision Problem,” IEEE Trans. PAMI, vol. 20, pp. 1016-1023, 1998.

[6] Hiromichi Fujisawa, “Forty years of research in character and docu- ment recognition - an industrial perspective,” Pattern Recognition, vol 41,no. 8, pp. 2435-2446, 2008.

[7] C.-L. Liu and C. Y. Suen, "A new benchmark on the recognition of handwritten Bangla and Farsi numeral characters," Pattern Recognition, vol. 42, pp. 3287-3295, 2009.

[8] U. Bhattacharya and B. B. Chaudhuri, "Handwritten Numeral Databases of Indian Scripts and Multistage Recognition of Mixed Numerals," IEEE Transactions on PAMI, vol. 31, pp. 444-457, 2009.

[9] N. Das, R. Sarkar, S. Basu, M. Kundu, M. Nasipuri and D. K. Basu, "A Genetic Algorithm based region sampling for selection of local features in handwritten digit recognition application," Applied Soft Computing 2011.

[10] (2011, 22nd July). CMATERdb 3.1.1: Handwritten Bangla numeral database. Available: http://code.google.com/p/cmaterdb/

[11] U. Pal, T. Wakabayashi, N. Sharma and F. Kimura, "Handwritten Numeral Recognition of Six Popular Indian Scripts," ICDAR 2007, pp. 749-753.

[12] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri and D. Kumar Basu, "A novel framework for automatic sorting of postal documents with multi- script address blocks," Pattern Recognition, vol. 43, pp. 3507-3521, 2010.

[13] T. Y. Zhang , C. Y. Suen, “A fast parallel algorithm for thinning digital patterns,” Communications of the ACM, vol.27, no.3, pp. 236-239, 1984.

[14] N. Otsu, “A Threshold selection method from grey level histogram,” IEEE Trans on SMC, Vol.9, pp.62-66, 1979.

[15] V. Vapnik, “An overview of statistical learning theory,” IEEE Trans. on Neural Network, vol.10, pp. 989-999, 1999.

[16] C.-C. Chang and C.-J. Lin, "LIBSVM : a library for support vector machines," ACM Transaction of Intelligent System and Technology, vol. 2, pp. 1- 27, 2011.

[17] N. Das, B. Mondal, S. Basu, R. Sarkar, M. Kundu , M. Nasipuri, “An SVM-MLP Classifier Combination Scheme for Recognition of Handwritten Bangla Digits,” ACVIT- 2009, pp. 615-623.

[18] Indian Statistical Institute (ISI), Kolkata, India, http://www.isical.ac.in/ujjwal/download/database.html