Grouping of Handwritten Bangla Basic Characters, Numerals and Vowel Modifiers for Multilayer Classification

Khondker Nayef Reza
Center for Research on Bangla Language Processing
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
e-mail: nayef.reza@gmail.com

Mumit Khan
Center for Research on Bangla Language Processing
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
e-mail: mumit@bracu.ac.bd

***Abstract*—For better performance in multilayer or hierarchical classification of handwritten text, appropriate grouping of similar symbols is very important. Here we aim to develop a reliable grouping schema for the similar looking basic characters, numerals and vowel modifiers of Bangla language. We experimented with thickened and thinned segmented handwritten text to compare which type of image is better for which group. For classification we chose Support Vector Machine (SVM) as it outperforms other classifiers in this field. We used both "one against one" and "one against all" strategies for multiclass SVM and compared their performance.**

Keywords-bangla basic characters numerals and vowel modifiers; handwritten character recognition; grouping of classes; support vector machine

I. INTRODUCTION

Bangla is the national language of Bangladesh. It is one of the most popular languages in the world and is used by more than 200 million people worldwide. Due to the recent initiatives of digitization of data included in hand filled voter list form, national ID card form, census survey form etc in Bangladesh, automated handwritten text recognition system has become even more important. But the field of machine recognition of handwritten Bangla text hasn't been developed to a satisfactory level because of its complex character pattern, huge symbol set, ambiguity and diverse writing styles. Bangla has 50 basic characters, 10 vowel modifiers, 7 consonant modifiers, 10 numerals and over 200 compound characters [1].

Most literature found in this field, study basic characters, numerals and modifiers separately whereas in the real world scenarios basic characters, numerals, vowel and consonant modifiers and compound characters are all present in documents. So we aim to incorporate basic characters, numerals and vowel modifiers in the same system of recognition although it creates a large symbol set to classify into. In case of large set of different symbols it has been found that sub grouping of similar looking symbols and recognizing group and inside group symbols in different layers improve the recognition rate [2, 3, 4]. In this paper we intend to derive a grouping scheme for the combination of basic characters, vowel modifiers and numerals which produces least amount of conflict among them. This scheme will be useful for building multi layer or hierarchical recognition of handwritten Bangla text and will be helpful to extend the grouping scheme to include consonant modifiers and compound characters as well. For the sake of simplicity we assume that the characters are already segmented.

For classification we used Support Vector Machine (SVM). As the use of SVMs is comparatively new to Bangla text recognition, we experimented with both 'one against one' and 'one against all' strategies of multi-class SVM classification and compared their performance in this context.

Fig 1. Samples of handwritten Bangla (a), (b) basic characters, (c) numerals and (d) segmented vowel modifiers.

II. RELATED WORKS

No paper could be found specifically on grouping of handwritten basic characters along with other symbols but some researches include grouping schema for basic characters as a part of the hierarchical classification.

Most previous work on handwritten Bangla text has been done on recognizing of handwritten Bangla numerals. In [5] two types of features are proposed for numeral recognition and achieved high level of accuracy. Reference [6] discusses a system for recognizing unconstrained Bangla numerals which shows accuracy rate of over 90%. In [7] a multi resolution wavelet analysis and majority voting scheme has

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.206

been used for recognizing Bangla numerals. Reference [8] experiments with different recognition methods on recognizing handwritten Bangla numerals and compares the results.

Works found on handwritten Bangla basic character recognition includes [2]. It proposes a multi stage approach for recognition of handwritten Bangla characters. It also proposes a sub grouping scheme for basic characters considering Matra, upper part of the character, disjoint section of the character, vertical line and double vertical line. Reference [3] implements an MLP based two stage classification for basic Bangla characters. The first stage determines groups of similar looking symbols and the second stage determines the actual symbol. The grouping proposed here is determined by the misclassification rate of one character with another. [4] uses a two stage classification for 45 Bangla characters and experiments with both disjoint and overlapping grouping schemas and compares the performance of different classifiers. Reference [1] proposes character recognition using superimposed matrices.

Reference [9] discusses recognition of compound characters using gradient features. In [10] SVM is used for recognizing basic characters along with compound symbols and sub grouping has been done to perform a hierarchical recognition. Other than these, [11] implements a system which recognizes Bangla characters for postal automation and [12] suggests a lexicon driven method to recognize Bangla words.

III. METHODOLOGY

A. Dataset

For our experiment we couldn't find any standard dataset of handwritten Bangla texts. Though we found a standard numeral database for Bangla, we had to collect handwritten text and build our own dataset for the rest of the characters. These texts were collected mostly from students and office holders. For this purpose every individual was asked to fill up a form where for each symbol there were five rectangular boxes.

The forms were scanned at 300 d.p.i. as suggested in [13], using a flatbed Cannon CanoScan LiDE 110 scanner as grayscale images. Symbol images were then manually extracted from the scanned images. Removal of extra long headline for better recognition proposed in [3] was also performed manually during this stage. Then we incorporated the numerals from the numeral character dataset provided by Indian Statistical Institute, by manually selecting images with less background noise. For each of the 69 symbols we selected 210 samples. In total we had 14490 samples in the whole dataset for training and testing. We evenly divided the dataset in half for training and testing purposes.

B. Preprocessing

As mentioned before we considered already segmented character images and didn't perform any type of segmentation. As source images can be of any size, we resize them to 28 pixels X 28 pixels keeping in mind that the contours don't get broken during resizing process and to make sure of that we slightly blur the image before resizing. We binarize the gray-scale images using a global threshold value for the entire dataset. This threshold value was determined through a thorough manual examination of the dataset images. As most of our data had very low background noise, we skipped noise elimination for the images.

As we used both thinned and thickened images for performance analysis, we separately thinned and thickened binarized image in corresponding situations. Thinned images were thinned to 1 pixel width and thickened images were thickened to at least three pixel width.

C. Feature extraction

In this stage each character is represented as a feature vector, which becomes its identity and is used for training and testing with the classifiers mentioned in the next section. We studied for suitable feature for handwritten text and found that chain code feature universally performs well [8]. Also zonal directional chain code is widely used as feature extraction method [3, 14, 15, 5, 12] in contexts similar to ours. So we used this method for feature extraction. The character image is divided into 7x7 zones. From each zone directional features are extracted to form the feature vector. The goal of zoning is to obtain the local characteristics instead of global characteristics. For each zone the contour is followed and a directional histogram is obtained by analyzing the adjacent pixels in a 3x3 neighborhood. Previous experiments in [3] show that down sampling the feature vector results in better performance. So we down sampled the 7X7 feature vector to a 4X4 feature vector using Gaussian pyramid as during the pyramid generation process, for each dimension the density of nodes is reduced by half from one level to the next [16].

The pyramid generation equation can be represented as:

$$\mathrm{G_l}(\mathrm{i},\mathrm{j}) = \sum_{n=-2}^{n=+2}\sum_{m=-2}^{m=+2} W(m,n) * G_{l-1}(2\mathrm{i}+\mathrm{m})(2\mathrm{j}+\mathrm{n}) \quad (1)$$

Where W(m,n) is the weighting function. The weighting function is chosen subject to certain constraints [16]. We initially start with the 7X7 zonal feature as the base level G_0 of the pyramid, each point in the next level G_1 is computed as a weighted average of values in level within a 5-by-5 window, termed as the weighting function W(m,n) as mentioned in [16]. In the end the features were normalized to the limit [0, 1].

D. Classifier

For classification we have used support vector machines mainly because SVMs are known to generalize well even with small training samples and SVM based application largely outperforms other learning algorithms [17]. SVM is a relatively new classifier and very few have used it in Bangla handwritten text recognition.

Normally SVMs are for two class problems. The basic idea of SVM is to construct a hyper plane which separates the positive and negative classes. But it can be extended to multi class by combining binary classifiers. Two most common ways of using is 'one against one' and 'one against all'. In our experiment we have used both types and compared their performance.

One against one strategy consists in construction of one SVM for each pair of classes. For n classes it needs n*(n-1)/2 SVMs. Classification of a feature is done by maximum voting where each SVM votes for one. In cases where two or more classes get same number of vote we choose the class that has smaller class index just to break the tie.

One against all strategy needs one SVM per class which is trained to distinguish one class from the samples of all remaining classes. For our experiment we have used probability estimation as a measure of confidence and in the cases where more than one class is identified for same features, the class with the highest probability wins. The probability measure is done using the built in probability measuring library of LIBSVM [18]. Here we only introduce some basic formulas for SVMs for details see [19].

Given the training samples $\{(X_i, y_i)\}$, i = 1, ... , n, $y_i \in \{-1,+1\}$, $X_i \in R^d$ where X_i is a d dimensional training sample, y_i is the class label for each X_i, and n is the number of training sample. As most problems are not linearly separable, to solve linearly unsolvable problems, the input space is mapped to a higher dimension using kernel function K. the decision function D for SVM can be denoted as

$$D(x) = f(\sum_{i=1}^{i=n} y_i\, a_i K(x, x_i) - b) \qquad (2)$$

$$\text{Where } f(u) = \begin{cases} 1, & if\ u > 0 \\ 0, & otherwise \end{cases} \qquad (3)$$

Training of SVM is to find the a_i values which can be achieved by minimizing the following quadratic function.

$$L(a) = \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i=1}^{i=n}\sum_{j=1}^{j=n} a_i\, a_j y_i y_j K(x_i, x_j) \qquad (4)$$

Which is subject to,

$$0 \le a_i \le C \qquad (5)$$

And

$$\sum_{i=1}^{n} a_i\, y_i = 0 \qquad (6)$$

.

Here C is a user chosen parameter.

Performance of SVM largely depends on kernel selection [17]. Although there are no theories on how to choose good kernel function in data dependent way, RBF kernel performs better than linear and polynomial due to better boundary response [17]. So for our current experiment we have used RBF kernel. For this kernel the equation for K is

$$K(x_i\,, x_j) = e^{-\Upsilon(|x_i - x_j|^2)} \qquad (7)$$

So for training we needed to find appropriate C and γ. Here we combined multiple binary classifiers for each multi class verification. We chose the C and γ of each binary classifier to minimize the generalization error of that particular binary classification problem, an alternative to choosing same C and γ for all classifier mentioned in [20].

To determine the optimal C and gamma for each binary classifier we first did a coarse grid search and then a fine grid search as mentioned in [20]. For coarse grid search we considered the values where C ∈ {1.0e-3, 1.0e-2,…, 1.0e+3} and γ ∈ {1.0e-3, 1.0e-2,…, 1.0e+3}. An optimal pair (C_0, γ_0) is selected from this coarse grid search. Then a fine grid search is conducted around (C_0, γ_0), with C ∈ $\{0.2C_0, 0.4C_0, 0.6C_0, 0.8C_0, C_0, 2C_0, 4C_0, 6C_0, 8C_0\}$ and γ ∈ $\{0.2\gamma_0, 0.4\gamma_0, 0.6\gamma_0, 0.8\gamma_0, \gamma_0, 2\gamma_0, 4\gamma_0, 6\gamma_0, 8\gamma_0\}$. The final optimal pair is selected from this fine search. In cases where there are several pairs that give the same cross validation accuracy, we prioritized pairs with smaller γ and smaller C. a fivefold cross validation was used to determine training accuracy.

The whole training and testing was done using LIBSVM [18].

E. *Initial grouping*

For initial grouping we considered some sub grouping methods suggested in [2] and also followed some parts of the grouping of basic characters mentioned in [3] and [4]. This schema was modified to include vowel modifiers and numerals along with basic characters. Also we considered the conflicting pairs found in other researches [11] and put them in the same group. For grouping in some cases we included some symbols which have common conflicting symbols but don't have much conflict among them, to keep the number of groups to a minimum while keeping each group mutually exclusive. In some cases we made some assumptions about the conflict of some symbols and tested for further modification as mentioned in the next section. For vowel modifiers which have two parts we considered each part for separate recognition. For example for ো we recognize ে and া separately. Similarly for ৌ we separately recognize ে and ৗ. In total we have a symbol set of 69 elements shown in Fig 1. The initial grouping is shown in Table 1.

Table 1. Initial Grouping

Group No.	Symbols
0	অ আ
1	২ ছ হ
2	ই ঈ
3	উ ঊ এ ঐ ও ঔ
4	ক ফ ব র
5	খ ঘ থ য ষ য়
6	০ ৩ ৫ ৬ ঙ ড ত ভ ড়
7	গ ণ প শ া
8	চ ঢ ঢ় ে
9	ট ঠ ৈ
10	ন ল
11	স ম

Group No.	Symbols
12	ঋ ঝ ধ
13	১ ৯
14	৪ ঃ
15	ৎ ং ু ূ
16	৭ ৌ ী
17 (others)	৮ জ ঞ দ ঁ ি ৃ

F. *Final grouping*

With this initial grouping we trained and tested the classifiers to find out which groups have high conflicts and merged or re organized groups to solve that. Also we determined inside group recognition error rate for each symbol to determine if the symbols included in the same group are actually error prone to one another and are rightfully in the same group. For wrongly grouped symbols we extensively searched for conflicting symbols and if none was found then we grouped it in the 'others' group.

In the initial grouping scheme we found high conflict among group 2, 3 and 9 due to their similarity in the upper part over Matra. 75% of the group detection error of test data of alphabet ই (group 2) is misclassified either as group 3 or 9. The same is seen in the case of alphabet ঠ (group 9), 69% of the group detection error of test data of the character either misclassified it as group 2 or 3. Also we found the grouping error in group 0 mostly results in group 6. For group 11, 69% of the misclassified test data are grouped as group 5. In group 12 ধ didn't have much conflict with other members of the group and we couldn't find any symbol which has exceptional conflict with this character. In group 15, ং and ৎ don't show conflict with other members of the group and are not error prone to any other group as well. In group 17 we found that ৃ is mostly misclassified as group 4 and ঁ mostly misclassified as group 6. From these findings we rearranged the initial grouping and formed the final grouping. The final grouping is shown in Table 2.

Table 2. Final Grouping

Group No.	Symbols
0	০ ৩ ৫ ৬ অ আ ঙ ড ত ভ ড় ঁ
1	২ ছ হ
2	ই ঈ উ ঊ এ ঐ ও ঔ ট ঠ ৈ
3	ক ফ ব র ৃ
5	গ ণ প শ া
6	চ ঢ ঢ় ে
7	ন ল
8	১ ৯
9	৪ ঃ
10	ু ূ
11	৭ ৌ ী
12	ঋ ঝ
13 (others)	৮ জ ঞ দ ি ধ ৎ ং

With the final grouping we train and test the classifiers and determine the grouping performance.

IV. RESULTS

From Table 3 we see that the recognition error rate is at a tolerable limit. In overall accuracy using thickened image do slightly better than thinned image and one against all strategy performed a little better than one against one strategy though not decisively. For some of the groups thinned images work better than thickened image. Table 4 shows the comparison.

Table 3. Summary of group recognition accuracy rate on test data

SVM Method	Feature type	Accuracy Rate %
One against one	Thinned	88.61
One against one	Thickened	89.59
One against all	Thinned	89.60
One against all	Thickened	89.99

Table 4. Group detection error rate (%) for test data

Grp.	Thinned image / one against one SVM	Thinned image / one against all SVM	Thickened image / one against one SVM	Thickened image / one against all SVM
0	7.42	7.42	8	7.67
1	15	12.33	18.33	19
2	8.72	10.18	6.72	7.27
3	7.4	4.6	9.6	7.2
4	11.5	11.38	8.63	8
5	8.2	8	12.2	12.2
6	10.25	10	8.75	7.5
7	12	9.5	12.5	10.5
8	20	16	16	10.5

Grp.	Thinned image / one against one SVM	Thinned image / one against all SVM	Thickened image / one against one SVM	Thickened image / one against all SVM
9	38.5	28.5	13.5	14
10	15	15	14.5	17
11	14.67	13	14.33	10
12	19	15.5	18	18

Also from the confusion matrix in fig 2 we don't see any major conflict between any two groups. This indicates that the grouping is considerably efficient.

Though we couldn't find any other similar study to compare the results of grouping, we can relate our results to the grouping results of [3] which only considers basic characters, with accuracy rate almost similar to ours. We can also compare our results with the Inter-Group classification accuracy for disjoint grouping scheme shown in Table 6 of [4] which has higher accuracy (94.07%) than ours. As we had a bigger symbol set we may conclude that the accuracy rate achieved in our grouping scheme is satisfying.

Target Groups	Output Groups													
G	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	92	0	3.1	0.5	1.5	0.3	0.8	0.1	0	0.2	0	0.3	0.1	1.2
1	3.7	81.7	5.3	1	1	0	4.7	0	0	0.3	0	0.7	0	1.7
2	2.8	0.4	93.3	0	0.5	0.3	0.3	0.1	0	0.1	0.1	0.1	0	2.1
3	1.4	0.8	2.6	90.4	2.4	0.4	0	0	0	0	0.8	0	0	1.2
4	1.9	0.1	1.9	1.1	91.4	2	0	0.3	0	0	0	0.1	0.5	0.8
5	1	0	1.8	0	3.8	87.8	0	1.6	0	0	0	2.6	0.8	0.6
6	4.3	0.8	2	1.3	0.3	0	91.3	0	0	0	0	0	0	0.3
7	1	0	1	0.5	4	4.5	0	87.5	1	0	0	0.5	0	0
8	0	0	6	0.5	2	2.5	0.5	0.5	84	1	0	3	0	0
9	4	1	0.5	0.5	2	0	0	0	1	86.5	0	0	0	4.5
10	1.5	0.5	1	0.5	0	3.5	1	1	0	0	85.5	2	0	3.5
11	1.7	0	2	0	1.4	5	0	0	0.3	0.3	1.3	85.7	0	2.3
12	2	0	0.5	6	7	0	0	0	0	0	0	0	82	2.5
13	4.3	1.4	5.4	0.5	1.4	0.3	1	0	0.1	0.6	0	0.1	0.3	84.8

Fig. 2. Confusion Matrix for group detection (in percentage) of final grouping (using thickened image and 'one against one' SVM method).

V. CONCLUSION

This paper has come up with a satisfying grouping scheme for the combination of handwritten Bangla basic characters, numerals and vowel modifiers. We started by deriving an initial grouping scheme combining the previously found conflicts of similar looking symbols and grouping schemes proposed for only basic characters. The initial grouping showed high conflict rate so we tested the scheme and made thorough analysis to identify the changes needed to improve the performance. The modified grouping was tested and showed high accuracy. We also found that both "one against one" and "one against all" schemes for multiclass SVMs performed with almost similar accuracy rate in our experiments.

ACKNOWLEDGMENT

We acknowledge the help of Indian Statistical Institute for providing us with the database for handwritten Bangla numerals [21]. We would also like to thank all the volunteers who contributed their handwritten text for our experiment.

REFERENCES

[1] Ahmed Shah Mashiyat, Ahmed Shah Mehadi, Kamrul Hasan Talukder, "Bangla off-line Handwritten Character Recognition Using Superimposed Matrices", http://research.banglacomputing.net/iccit/ICCIT_pdf/7th%20ICCIT-2004_112.pdf

[2] A.F.R. Rahman, R. Rahman, M.C. Fairhurst, "Recognition of Handwritten Bengali Characters: a Novel Multistage Approach," Pattern Recognition, vol. 35, p.p. 997-1006, 2002

[3] U. Bhattacharya, M. Shridhar and S. K. Parui, "On Recognition of Handwritten Bangla Characters", Computer Vision, Graphics and Image Processing, p.p. 817-828, 2006

[4] Tapan Kumar Bhowmik, Pradip Ghanty, Anandarup Roy, and Swapan Kumar Parui, " SVM-based hierarchical architectures for handwritten Bangla character recognition". International Journal on Document Analysis and Recognition archive Volume 12 Issue 2, June 2009

[5] U. Pal, N. Sharma, T. Wakabayashi, F. Kimura, "Handwritten Numeral Recognition of Six Popular Indian Scripts," icdar, vol. 2, pp.749-753, Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2, 2007

[6] U. Pal, A. Belaid and B. B. Chudhuri, "A system for Bangla Handwritten Numeral Recognition", Available: hal.archives.ouvertes.fr/docs/00/11/09/73/PDF/kbcs-cam-final.pdf

[7] U. Bhattacharya, B. B. Chaudhuri, "A Majority Voting Scheme for Multiresolution Recognition of Handprinted Numerals," icdar, vol. 1, pp.16, Seventh International Conference on Document Analysis and Recognition (ICDAR'03) - Volume 1, 2003

[8] Liu, Cheng-Lin , Nakashima, Kazuki, Sako, Hiroshi, Fujisawa, Hiromichi, "Handwritten digit recognition: benchmarking of state-of-the-art techniques", Pattern Recognition 2003, Volume 36, pp. 2271-2285

[9] U. Pal, T. Wakabayashi, F. Kimura, "Handwritten Bangla Compound Character Recognition Using Gradient Feature," ICIT, pp.208-213, 10th International Conference on Information Technology (ICIT 2007), 2007

[10] Nibaran Das 1, Brindaban Das, Ram Sarkar, Subhadip Basu, Mahantapas Kundu, Mita Nasipuri. "Handwritten *Bangla* Basic and Compound character recognition using MLP and SVM classifier", Journal of Computing, Volume 2, Issue 2, February 2010, ISSN 2151-9617. https://sites.google.com/site/journalofcomputing/

[11] K. Roy, S. Vajda, U. Pal, B. B. Chaudhuri, "A System towards Indian Postal Automation," iwfhr, pp.580-585, Ninth International Workshop on Frontiers in Handwriting Recognition (IWFHR'04), 2004.

[12] U. Pal, K. Roy, F. Kimura, "A Lexicon Driven Method for Unconstrained Bangla Handwritten Word Recognition" , In Proc. 10th International Workshop on Frontiers in Handwriting Recognition, pp. 601-606, 2006

[13] Bhattacharya, U.; Chaudhuri, B.B.; , "Databases for research on recognition of handwritten characters of Indian scripts," *Document Analysis and Recognition,*

2005. Proceedings. Eighth International Conference on, vol., no., pp. 789- 793 Vol. 2, 29 Aug.-1 Sept. 2005 doi: 10.1109/ICDAR.2005.84
[14] N. Sharma, U. Pal, F. Kimura, "Recognition of Handwritten Kannada Numerals," icit, pp.133-136, 9th International Conference on Information Technology (ICIT'06), 2006-of-the-art techniques", Pattern Recognition 2003, Volume 36, pp. 2271-2285.
[15] Sharma, N. , Pal, U. ,Kimura, F., Pal, S., "Recognition of Off-Line Handwritten Devnagari Characters Using Quadratic Classifier", p.p.805-816, Computer Vision, Graphics and Image Processing, Lecture Notes in Computer Science , Vol 4338, 2006.
[16]http://sepwww.stanford.edu/data/media/public/docs/sep124/ssen1/paper_html/node3.html.
[17] Byun Hyeran , Lee Seong-Whan, "A survey on pattern recognition applications of support vector machines", International Journal of Pattern Recognition and Artificial Intelligence, 2003, volume 17, pp.459-486..
[18] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM : a library for support vector machines", ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
[19] Burges, Christopher J.C., "A Tutorial on Support Vector Machines for Pattern Recognition", Data Mining and Knowledge Discovery, 1998, pp. 121-167, DO - 10.1023/A:1009715923555.
[20] Kai-Bo Duan, S. Sathiya Keerthi. "Which Is the Best Multiclass SVM Method? An Empirical Study", Proceedings of the Sixth International Workshop on Multiple Classifier Systems, 2005, pp. 278--285.
[21] U. Bhattacharya and B. B. Chaudhuri, Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 31(3), 2009, 444-457