An Analytic Scheme for Online Handwritten Bangla Cursive Word Recognition

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

A. Nigam  
Institute of Technology  
Banaras Hindu University  
Varanasi, India  
aditya.itbhu@ieee.org

Y. S. Rawat  
Institute of Technology  
Banaras Hindu University  
Varanasi, India  
yogesh.rawat.itbhu@gmail.com

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

Abstract

In this article, we describe a prototype system for recognition of online handwritten cursive words of Bangla, a script used by more than 200 million people of India and Bangladesh, two neighboring countries of Asia. To the best of our knowledge, in the literature, there does not exist any work on recognition of such online Bangla cursive words. Here, we propose an analytic recognition approach, which involves segmentation of the input Bangla word. Modified quadratic discriminant function classifier is used for recognition of segmented strokes based on a chain code histogram based feature vector. Finally, an input word is recognized by a verification module, which uses a set of rules for construction of characters from strokes. We carefully selected a set of 100 Bangla words such that each basic character and vowel modifier of this script occurs at least in two words. A total of 10000 handwritten online word samples provided by 50 native Bengali writers of different groups with respect to age, education, sex and income have been used in the present study.

Keywords: Online handwriting recognition, segmentation of online cursive handwriting, Bangla word recognition, MQDF classifier, chain code histogram feature.

1. Introduction

With the availability of electronic tablets at a cost affordable by common Indians, online handwriting recognition for Indian scripts has gained enough significance. On the other hand, both for standard and portable miniature computing devices, non-keyboard based methods for data entry have additional importance in the Indian context since its scripts have large symbol sets.

Several computing devices such as Tablet PC, PDA etc. are available in the market and often these are equipped with pen-based input technology for scripts of a few developed nations. Significant progress has already been achieved towards recognition of online handwriting in those non-Indian scripts. The present article presents a prototype system for recognition of online handwritten Bangla cursive words. To the best of our knowledge, this is a pioneering attempt and no existing work on online cursive handwriting recognition had dealt with an Indian script before. Input data to such a online handwriting recognition engine consist of (x, y) coordinates along the trajectory of the pen together with a few other possible information such as positions of pen-up, pen-down etc., India is a multilingual / multiscrypt country with Bangla its second most popular language and script. It is also the official language / script of its neighbouring country Bangladesh and is used by more than 200 million people across the two countries. Bangla, like other major Indian scripts, is a mixture of syllabic and alphabetic scripts, it came from the ancient Indian script, Brahmi, it run from left to right and it has no equivalent to capital letters of Latin scripts. The set of basic characters of Bangla consist of 11 vowels and 40 consonants. However, since the shapes of two consonant characters are the same, there are 50 different shapes in the Bangla basic character set. Ideal (printed) forms of these 50 different shapes of Bangla basic characters are shown in Fig. 1. In Bangla, a vowel other than उ following a consonant often take a modified shape called a vowel modifier (VM). Ideal (printed) shapes of these vowel modifiers corresponding to 10 vowels (other than उ ) are shown in Fig. 2. On the other hand, in an Indian script including Bangla, often a cluster of several consonants or a vowel in conjunction with a consonant form a large number of possible different shapes, called compound characters, making the total size of the alphabet as large as 300. However, in the present day Bangla text, the occurrence of compound characters is less than 5% and the rest is only basic characters and vowel modifiers.

![Figure 1. Set of Bangla basic characters.](image-url)
2. Brief survey

Extensive research on cursive handwriting recognition has been done during the last few decades. However, there has not been much work on handwriting recognition of Indian scripts. Particularly, recognition of online Bangla cursive handwriting has not been attempted before.

Recent survey works of handwriting recognition include [1], [2], [3] and [4]. Two main approaches to handwriting recognition are analytic approach and holistic approach ([1], [4], [5]). In an analytic approach, input word samples are segmented into smaller components (characters, graphemes, allographs, etc.), and these components are identified during the recognition stage. Finally, a word sample is recognized based on global combination of the resulting character scores. On the other hand, in holistic approaches no segmentation phase is involved. Strategies combining holistic and analytic approaches can improve the recognition rate over that of a single classifier ([5], [6]).

In the earliest available work on segmentation of handwritten cursive Bangla words [8], offline word images were considered. In this work, a recursive contour following approach was proposed. In [9], a certain water reservoir technique was used for segmentation of handwritten Bangla word images. In another work [10], a candidate path (representing the matra or headline of the word sample) was computed from a skew corrected word image obtaining the feature points for segmentation purpose. A fuzzy feature based segmentation technique for Bangla word images was proposed in [11]. In another related work [12], segmentation of touching characters of printed Bangla and Devanagari scripts were considered.

Reports on online isolated Bangla character/numeral recognition are found in [13-15]. In the earliest work on online handwritten Bangla script recognition [13], isolated basic characters of Bangla were considered. In [14], a Hidden Markov Model was used for recognition of online handwritten isolated Bangla numerals. A direction code based feature vector was used in [15] for recognition of online handwritten Bangla basic characters.

3. Present work

3.1. Bangla handwriting

Unlike other Indian scripts, unconstrained handwriting in Bangla is usually very cursive. Also, an important feature of several Indian scripts such as Devanagari and Bangla is the presence of a headline (Siorekha or Matra). This often plays an important role in printed OCR of these scripts. Although this headline is usually present in the handwritten form of Devanagari script, the same is not generally true for Bangla handwriting. This is illustrated in the samples of a few handwritten Bangla words shown in Fig.3.

![Figure 3. Bangla handwritten word samples. Samples in the leftmost column have no headline; headline is present partially in samples in the middle column; headline is present significantly in samples in the right column.](image-url)

Also, it may be observed from the above samples that the cursiveness in Bangla handwriting occur in the region of its headline (siorekha) in contrast to English handwriting where the cursiveness occurs in the lower part of the word shape.

3.2. Online handwritten Bangla word database

Since there is no available database of online handwritten Bangla word samples, we collected 10000 such samples written by 50 native writers. These writers belong to different groups with respect to age, education, sex and income. We carefully selected a lexicon of 100 Bangla words of various lengths consisting of 1 to 7 basic characters and 0 to 5 vowel modifiers. Also, each basic character and vowel modifier occurs in at least two words of the above lexicon set. In the present work, we do not consider any word involving a compound character.

Samples of the present database have been collected using WACOM Intuous 2 tablet. The maximum sampling rate is 200 points per second. No restriction was imposed on the writers. Each person was asked to write each word at two different points of time. Each sample is stored as a text file consisting of the (x, y) coordinates and pressure among a few other information of the pen tip along the trajectory. In the preset study, we consider only the coordinates of the positions of pen tip having non-zero pressure.

We randomly selected 15 writers and samples provided by them are used to form the test set. Handwritten word samples obtained from the remaining 35 writers form the training set. Thus, our training and test sets consist of 7000 and 3000 word samples respectively. These samples include completely or partially cursive words as well as handprinted words.
3.3. Line and word segmentation

Since the present work is the first ever attempt for recognition of handwritten online Bangla cursive words, we used simple methods providing acceptable results on the handwritten data collected by us. Further studies of sophisticated methods are necessary before its use into real-life applications.

Devices used for collecting samples of handwriting stores data in a page-wise format. For extraction of individual lines from de-skewed pages of online handwritten data, we assume that each new line starts near the left margin. In fact, this is generally true for all document pages collected by us. But, in more realistic situations, such an assumption is not valid. However, we just locate valleys in the histogram of $x$-coordinates of successive points captured by the device as shown in Fig. 4. Separate lines are obtained by segmenting the document at these valleys. This approach does not get affected either by spatial overlapping of consecutive lines or presence of out-of-order diacriticals and/or parts of modifiers (two such possible situations shown in Fig. 5) creating only smaller peaks and/or closer valleys in the above histogram.

![Figure 4. Segmentation of handwritten text into lines.](image)

![Figure 5. Example strokes that may appear out-of-order in the online data.](image)

For segmenting a text line into words, we compute horizontal distance between the right extreme point of a left stroke and left extreme point of the stroke (sharing no common $x$-value) to its immediate right. Words along a text line are isolated based on a heuristically selected threshold value of this horizontal distance. We observed that such a heuristic approach sometimes (as in Fig. 6(a)) fails resulting in under-segmentation caused by parts of the concerned strokes appearing either in the lower and/or upper part (see Fig. 6(a)) of the word shape. So, for each segmented word obtained as above, the horizontal distances are recomputed only for the part of the word appearing in the middle (called the busy zone as described in Section 3.4.1) and verified for possible further segmentation. For example, the under-segmentation shown in Fig. 6(a), is taken care of by this approach resulting in proper segmentation as shown in Fig. 6(b).

![Figure 6. (a) Horizontal gap between adjacent words causes under segmentation, (b) Horizontal gap between the middle parts of adjacent words is enough.](image)

3.4. Cursive stroke segmentation

Usually, two different approaches (external or internal) [16] are used for segmentation of cursive words. In the literature, several methods are available for external segmentation of online handwritten cursive words. In [17], possible segmentation points had been decided by obtaining intersections, cusps, points of inflection and endpoints. Points of minima in the tangential pen-tip velocity were used for word segmentation in [18] and [19]. In the present work, we considered an external approach in which an input online cursive Bangla word is segmented into characters or their parts before the recognition phase.

![Figure 7. Ideal (printed) shapes of Bangla words. (a) the shape has three zones, (b) the shape has no upper zone, (c) the shape has no lower zone, (d) the shape has only middle zone.](image)

Ideal (printed) shapes of Bangla words have generally three distinct zones. This is illustrated in Fig. 7. The middle zone is found in the shape of every Bangla word while the other two zones (upper and lower) may or may not be present. Also, in printed forms of Bangla words, a distinct headline (matra or sirorekha) separating the upper and middle zones is always present except in a few rare words. Consequently, segmentation of printed Bangla words is often based on detection of its headline (Matra) [20]. On the other hand, it is clear from handwritten samples shown in Fig. 3, that computation of the headline for handwritten Bangla words is a tricky job.

3.4.1. Estimation of headline in handwritten Bangla words

The present segmentation approach is based on estimation of the positions of headline and busy zone of the input word sample. The algorithm is described below.

Step 1: Compute height ($H = y_{\text{max}} - y_{\text{min}}$) of the word.
Step 2: Set $HT_{Lim} = [A \ast H]$, where $A$ (0 < $A$ < 1) is selected empirically.

Step 3: Compute frequency distribution of all those $y$-values for which $y < HT_{Lim}$ ($y_{min}$ corresponds to the top-most point(s) and $y$ increases downwards).

Step 4: Set $M =$ modal value of the above frequency distribution.

Step 5: Obtain $S = \{y \mid \text{freq}(y) > B \ast M\}$, where $B$ (0 < $B$ < 1) is selected empirically.

Step 6: $y_{Top} = \min(y \mid y \in S)$

Step 7: The busy zone is obtained as the horizontal strip bounded by $y = y_{Top}$ and $y = HT_{Lim}$.

We set $A = 0.75$ and $B = 0.5$ based on extensive simulation runs using training samples of the present database. The headline is indicated by the row $y_{Top}$. In Fig. 8, we show an example to describe how the above algorithm works. In this sample word, $H = 18$ and $HT_{Lim} = 14$. Here, the successive frequencies (arranged according to increasing $y$) of the said distribution are 9, 13, 14, 9, 10, 8, 6, 7, 8, 4, 6, 4, 7. Thus, $M = 14$ and $S = \{i \mid 0 \leq i \leq 13\}$. Here, $y_{Top} = 0$ and this is justified by the fact that this particular word does not have any part in the upper zone (see Fig. 4).

Figure 8. Estimated position of headline and busy zone of a Bangla word sample

Here, we like to mention that the above method for detection of headline may fail in several situations such as when different parts of the word has different amount of rotations.

3.4.2. Computation of segmentation points

Here we obtain the points (along the trajectory of the pen movement) where the pen-tip after traveling through the busy zone crosses / touches the headline (say, at point $S_1$) and after some more time it again enters the busy zone (say, at point $S_2$) without lifting the pen-tip from the writing surface. Segmentation points include (i) midpoints of the parts of trajectories between $S_1$ and $S_2$ and (ii) endpoints of each constituent strokes save for the last stroke.

In Figs. 9(a) and 9(d), two samples of cursive handwritten Bangla words are shown. Estimated headlines of both the words are shown in Figs. 9(b) and 9(e) respectively. Both type (i) and type (ii) segmentation points are shown in Figs. 9(c) and 9(d). Here, type (i) segmentation points are enclosed by circles while type (ii) segmentation points are enclosed by squares. In the first sample, there are 5 segmentation points ($S_1, S_2, S_3, S_4$ and $S_5$) of type (i) and 6 segmentation points ($E_1, E_2, E_3, E_4, E_5$ and $E_6$) of type (ii). In the second sample, these numbers are 4 and 4 respectively.

Figure 9. Results of segmentation. (a) and (d) Two cursive word samples are shown; (b) and (e) estimated headlines are shown; (c) and (f) both types of segmentation points are shown.

4. Recognition methodology

Two persons in consultation with each other manually partitioned the set of strokes obtained by segmenting 7000 training samples based on shape similarities. The number of resulting groups is only 73. As an example, the character of Fig. 10(a) appearing in various word samples (segmented) contributed 6 strokes (Fig. 10(b)) of different shapes giving rise to 6 groups in the above set. Samples of this character class (Fig. 10(a)) are formed by combining either the first three strokes or the 1st and 4th strokes or 5th, 6th and 3rd strokes in a left-to-right order. On the other hand, the same stroke may appear in images of different character classes such as the 3rd stroke in Fig. 10(b) may be obtained from different character classes as shown in Fig. 10(c). Finally, it should be noted that the present set of samples collected from 50 writers represent only 100 different words involving 50 basic characters and 10 vowel modifiers. We did not consider any compound character which are large in number but constitute less than 5% of any Bangla corpus.

Figure 10. (a) A basic character; (b) strokes obtained after segmentation of word samples containing the character in (a); (c) basic characters / vowel modifiers which often use the 3rd stroke in (b) for formation of respective shapes.
This module uses a set of rules and these are designed based on script knowledge and training samples of the present database. Implementation of these rules in the verification module is done in the form of two look-up tables. In one of them, there are 60 entries corresponding to 50 basic and 10 vowel modifier characters. This table, called character table, stores information about possible stroke classes corresponding to each character. It also provides information whether a stroke alone forms the character or contributes only a part of the character shape. In another table, called stroke table, there are 73 entries corresponding to the possible 73 stroke classes. It stores information of possible character classes in which a given stroke may appear.

### 4.1. Preprocessing

In the preprocessing stage, initially, we compute the length (cumulative sum of successive points) of the input stroke and if it is smaller than a small value set a priori, we ignore it during the next few phases but store the same for possible use in the verification module. This approach is considered for noise removal. On the other hand, if the stroke carries important information such as a diacritic mark, then it can be reused at a latter stage. Other preprocessing operations include shift of origin, smoothing and resampling of points. For each stroke, the origin is shifted to its top-left point. For smoothing we use a mean filter of length 5 successively for two times. Finally, a sequence of approximately equidistant points is generated during resampling.

### 4.2. Feature extraction

For extraction of features, we divide a pre-processed stroke into 7 sub-strokes of approximately equal length. Let the sequence of points in a sub-stroke be \(P_1, P_2, \ldots, P_N\). Now, let the angle made with the x-axis while moving from \(P_{r-1}\) to \(P_r\) (anticlockwise) be \(\alpha_r\), \(r = 2, \ldots, N\) (\(0 \leq \alpha_r < 360^\circ\)). Here, the change in direction while moving from one point to the next one is important and we quantize these directions into 9 different values, viz. 0, 1, 2, \ldots, 8. In particular, if \(337.5^\circ \leq \alpha_r < 360^\circ\) or \(0^\circ \leq \alpha_r < 22.5^\circ\), then the direction code corresponding to the point \(P_r\) is obtained as 1, or if \(22.5 + (k-1) \times 45^\circ \leq \alpha_r \leq 22.5 + k \times 45^\circ\), then the direction code corresponding to \(P_{r+1}\) is obtained as \(k+1\), for \(k = 1, \ldots, 7\), \(r = 2, \ldots, N\). The direction code corresponding to \(P_1\), the starting point of a stroke, is assumed to be 0.

A histogram of the direction codes is calculated for each sub-stroke and its values are divided by \(N\) for normalization purpose. Also, we obtain coordinates of the C.G. (center of gravity) of all points constituting a sub-stroke and we normalize these values by width and height of the stroke respectively. Thus, the number of feature components generated from a sub-stroke is 11 (9+2) and the feature vector for a stroke has 77 (11×7) components.

### 4.3. Classification

Quadratic discriminant function (QDF) classifier has been widely used for recognition of offline handwriting [21]. Under the assumption of equal prior probabilities, the QDF may be obtained as

\[
g(x, \omega_i) = \frac{1}{2} (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) + \log \mid \Sigma_i \mid ,
\]

where \(T\) denotes transpose, \(x\) is the feature vector, \(\mu_i\) and \(\Sigma_i\) denote respectively the mean vector and covariance matrix of class \(\omega_i\). The QDF in (1) may be rewritten as

\[
g(x, \omega_i) = \frac{1}{2} \sum_{j=1}^d \frac{1}{\lambda_{ij}} [(x - \mu_i)^T \phi_{ij}]^2 + \frac{1}{2} \sum_{j=1}^d \log \lambda_{ij} ,
\]

where \(\lambda_{ij}, j = 1, 2, \ldots, d\), denote the eigenvalues of class \(\omega_i\) stored in decreasing order, and \(\phi_{ij}, j = 1, 2, \ldots, d\) are the corresponding eigen vectors. Replacing the minor eigen values by a constant \(\delta_i\), the modified quadratic discriminant function (MQDF) [21] may be obtained as

\[
g(x, \omega_i) = \frac{1}{2} \sum_{j=1}^m \frac{1}{\lambda_{ij}} [(x - \mu_i)^T \phi_{ij}]^2 + \frac{1}{2} \delta_i \left(\|x - \mu_i\|^2 - \sum_{j=1}^m [(x - \mu_i)^T \phi_{ij}]^2\right)
\]

\[
+ \sum_{j=1}^m \log \lambda_{ij} + (d - m) \log \delta_i
\]

where \(m\) denotes the number of principal eigen vectors. Usually, \(\delta_i\) is set to a class-independent constant. MQDF improves the efficiency with respect to the computational and classification performance over QDF by considering only the principal eigen vectors and the corresponding eigen values of covariance matrices.

### 4.4. Verification module

In this module, we start with the top choice for each stroke provided by MQDF towards construction of characters. For this choice, we first consider the stroke table and identify the possible parent character class(es). Then, we use the character table for confirmation whether any other stroke is necessary to construct a character. If so, we continue considering latter strokes in the temporal order before we can construct a character. However, it may so happen that the expected stroke is missing from its temporal order. In such a situation, we search for it in any other temporal location and if found, we verify the candidature of the out-of-order stroke by comparing positional information. If the search for an out-of-order stroke also fails, we look for the possibility that it had been ignored by the preprocessing stage due to its small size. In
case of a further failure, we consider second top choice of the stroke provided by MQDF and repeat the above procedure. If it still does not form a character, we reject the input word. The actual implementation of this verification module considers several other situations but a complete account of all those are avoided due to space limitation.

5. Results and discussions

Evaluation of the proposed segmentation scheme is done manually on the basis of the test set. Segmentation of words from individual lines failed in 1.8% cases even after using the additional check for possible out-of-order strokes across the words. We exclude the above failure situations for evaluation of the stages following it. 3.1% of the strokes segmented by the proposed scheme have suffered from under segmentation. We have used only the properly segmented strokes for training and testing of the classifier. Recognition error is 1.22% at stroke level, and 1.96% at character level. Overall word level recognition accuracy on the test set is 82.34%. This recognition performance has been achieved without using any postprocessing scheme. Preliminary investigations show that segmentation performance may be improved by combining offline and online information while recognition accuracy could be improved by using a dictionary and/or n-gram. Finally, there are many other avenues, to be studied in future, towards improvement of performance during both segmentation and recognition stages.

6. References


