A Lexicon Driven Method for Unconstrained Bangla Handwritten Word Recognition

U. Pal, K. Roy, F. Kimura

A Lexicon Driven Method for Unconstrained Bangla Handwritten Word Recognition

U. Pal
CVPR Unit
Indian Statistical Institute
Kolkata-108, India
Email: umapada@isical.ac.in

K. Roy
Dept. of CSE.
West Bengal University of Technology, Saltlake City,
Kolkata-64, India

F. Kimura
Faculty of Engg.
Mie University, 1577
Kurimamachiya-cho, TSU,
Mie 514-8507, Japan

Abstract

In this paper a lexicon driven segmentation-recognition scheme for unconstrained Bangla handwritten word recognition is proposed for Indian postal automation. In the proposed method, at first, binarization of the input document is done and slant correction of the individual words is performed. Next, using water reservoir concept words are pre-segmented into possible primitive components (characters or its parts). In order to merge these primitive components into characters and to find optimum character segmentation, dynamic programming (DP) is applied using total likelihood of characters as the objective function. To compute the likelihood of a character, modified quadratic discriminant function (MQDF) is used for the purpose. The features used in the MQDF are mainly based on the directional features of the contour points of the components. We tested our system on Bangla city-name images and at present an overall accuracy of 87.21% is obtained from the proposed system.

Keywords: *Handwriting recognition, Cursive script recognition, Bangla word recognition, Indian script*.

1. Introduction

Recognition of handwritten words has been a popular research area for many years because of its various application potentials. Some of its potential application areas are postal automation, bank cheque processing, automatic data entry, etc. There are many pieces of work towards handwritten recognition of Roman [4], Japanese/Chinese [7,9] and Arabic scripts [1]. Mainly two approaches (segmentation approach [13] and holistic approach [10]) are used for the purpose. Combination of these two approaches has also been used for word recognition [5]. Mostly neural networks [4], hidden markov model [2], modified quadratic discriminant functions [6-8], graph-based method [3] etc. are used for handwritten word recognition [13].

Although a few pieces of work have been done towards the recognition of isolated handwritten characters of Indian scripts but to the best of our knowledge, there exists only one work towards handwritten word recognition of Indian scripts. This work is done for Bangla script and NSHP-HMM approach was applied for recognition [14]. In this present paper we propose a lexicon driven method for unconstrained Bangla handwritten word recognition. This is the second work on unconstrained Bangla handwritten word recognition.

The word recognition approach proposed in this paper is as follows. At first, slant correction of the individual words is performed on the binarized images. In Bangla most of the characters have a horizontal line at the upper part. When two or more characters sit side by side to form a word in Bangla, these horizontal lines generally touch and generate big regions (spaces). For example see Figure 1. Because of this touching behavior, we use water reservoir based concept [12] for the segmentation of Bangla text words into primitives. Each primitive ideally consists of a single character or a sub-image of a single character. A lexicon driven segmentation-recognition scheme is applied here for our recognition purpose. In order to merge these primitive components into characters and to find optimum character segmentation of a word, dynamic programming (DP) is applied using the total likelihood of characters as the objective function. To compute the likelihood of a character, a modified quadratic discriminant function (MQDF) based on the directional features of the contour points of the components is used here. Block diagram of the proposed system is shown in Figure 2.

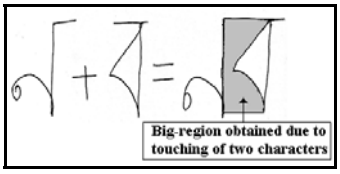


Figure 1. Example of touching Bangla characters

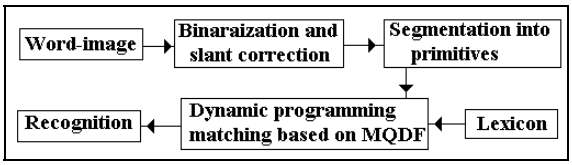


Figure 2. Block diagram of the proposed system.

The organization of the rest of the paper is as follows. In Section 2, properties of Bangla characters are described. Principle of water reservoir concept is discussed in Section 3. Section 4 deals with slant

correction and character pre-segmentation from Bangla words. Feature extraction for the use of recognition is discussed in Section 5. Section 6 deals with segmentation-recognition using dynamic programming technique. Finally, results and discussion are given in Section 7.

2. Properties of Bangla script

Bangla, the secondmost popular language in India and the fifthmost popular language in the world, originated from Brahmi script. Except Bangla, other languages like Assamese, Manipuri etc. are also written in Bangla script. The alphabet of the modern Bangla script consists of 11 vowels and 39 consonants. These characters are called *basic characters*. The basic characters of Bangla script are shown in Figure 3. The concept of upper/lower case is absent in this script. Writing style in Bangla script is from left to right. From the Figure 3 it can be seen that most of the characters in Bangla script have a horizontal line at the upper part. We call this line as *head-line* or *matra*. From a statistical analysis we notice that the probability that a Bangla word will have horizontal line is 0.994 [12].

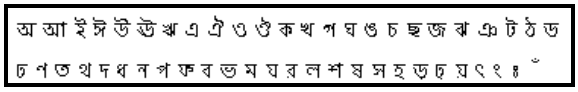


Figure 3. Basic characters of Bangla alphabet are shown (first eleven are vowels and rest are consonants).

Vowel	আ	ই	ঈ	উ	ঊ	ঋ	এ	ঐ	ও	ঔ
Modified shape	া	ি	ী	ু	ূ	ৃ	ে	ৈ	ো	ৌ
When attached to consonant ক	কা	কি	কী	কু	কূ	কৃ	কে	কৈ	কো	কৌ

Figure 4. Bangla vowels and their corresponding modified characters.

In Bangla script a vowel following a consonant takes a modified shape. Depending on the vowel, its modified shape is placed at the left, right (or both) or bottom of the consonant. These modified shapes are called *modified characters*. Examples of Bangla modified character are shown in Figure 4. A consonant or vowel following a consonant sometimes forms a different shape character and we call it as *compound character*. Compound characters can be combinations of two consonants as well as a consonant and a vowel. Compounding of three or four characters also exists in the Bangla script. To get an idea about Bangla compound characters some examples of compound characters formed by two and three characters are shown in Figure 5. Because of modified and compound characters in total there are about 280 characters in Bangla [12]. Because of the larger character set as well as complex-shaped structure of Bangla characters, development of Bangla handwritten word recognition system is more difficult than that of Roman script.

A Bangla text line can be partitioned into three zones. The *upper-zone* denotes the portion above the head-line, the *middle zone* covers the portion between head-line and base-line, the *lower-zone* is the portion below base-line. Different zones in a Bangla text line are shown in Figure 6. Mostly a modified or a part of a modified character sits in the upper zone and lower zone of a Bangla line.

Consonant characters		Compound characters
ক + র	=	ক্র
ক + ষ	=	ক্ষ
স + ক	=	স্ক
ক + ষ + ণ	=	ক্ষ্ণ
ন + ত + র	=	ন্ত্র

Figure 5. Examples of some compound characters.

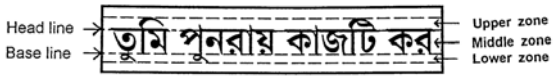


Figure 6. Different zones of a Bangla text line.

3. Water Reservoir principle

Water reservoir is a metaphor to illustrate the cavity region of a component. The water reservoir principle [12] is as follows. If water is poured from a side of a component, the cavity regions of the background portion of the component where water will be stored are considered as reservoirs of the component. These reservoirs are used for the segmentation of the words into primitives.

Some of the water reservoir principle based features used here are as follows.

Top (Bottom) reservoir: By top (bottom) reservoirs of a component we mean the reservoirs obtained when water is poured from top (bottom) of the component. A bottom reservoir of a component is visualized as a top reservoir when water will be poured from top after rotating the component by 180°.

Water reservoir area: By area of a reservoir we mean the area of the cavity region where water will be stored. The number of points (pixels) inside a reservoir is computed and this number is considered as the area of the reservoir.

Water flow level: The level from which water overflows from a reservoir is called as water flow level of the reservoir (see Figure 7).

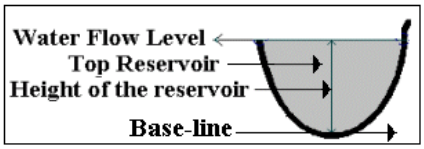


Figure 7. A top water reservoir and its different features are shown. Water reservoir is marked by gray shade.

Height of a reservoir: By height of a reservoir we mean the depth of water in the reservoir.

Base-line: A line, passing through the deepest point of a reservoir and parallel to its water flow level is the base line.

Base-area points: By *base-area* points of a reservoir we mean the border points of the reservoir having height less than R_L from the reservoir base-line. Here, R_L is the stroke width of the component. For a component, R_L is calculated as follows. A component is scanned both row-wise and column-wise. Suppose the component has n different runs of lengths $r_1, r_2, \ldots r_n$ with frequencies $f_1, f_2 \ldots f_n$, respectively. Then value of R_L will be r_i if $f_i = \max(f_j)$, $j = 1, 2, \ldots n$.

4. Slant correction and character pre-segmentation

Slant correction: A gray scale image of an input word is binarized by Otsu's algorithm [11]. The binary image is then slant estimated and corrected. The slant estimation is done based on the chain code information of the contour of the word image (chain codes are shown in Figure 8(a)). Since horizontal chain element n_0 does not interfere in slant hence we ignored horizontal chain elements in slant estimation. Slant angle θ is estimated based on the equation 1.

$$\theta = \tan^{-1}\left(\frac{n_1 + n_2 + n_3}{n_1 - n_3}\right) \cdots\cdots\cdots (1)$$

where n_i is the number of chain elements at an angle of i x 45^o (/ or | or \) (n_i are shown in Figure 8(a)). Here $n_1+n_2+n_3$ is the total value in vertical/slanted direction and n_1-n_3 is the change in horizontal direction as shown in Figure 8(b). It is easy to show that the average orientation over all chain segments is also given by (1). An example of a Bangla word and its slant corrected version is shown in Figure 9.

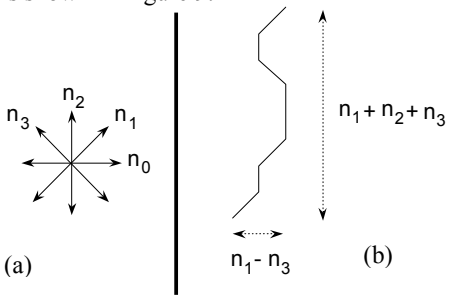


Figure 8. Orientation of a chain segment.

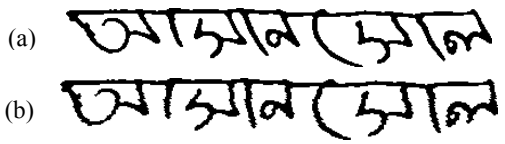


Figure 9. (a) Input word (b) its slant corrected version.

Character pre-segmentation: To find optimal character segmentation by the segmentation-recognition scheme using dynamic programming, we pre-segmented the words into primitives. As mentioned earlier, when two or more characters sit side by side to form a word in Bangla text, these horizontal lines touch and generate big regions (spaces). Because of the touching nature of Bangla characters in a word we use water reservoir concept for character pre-segmentation. For segmentation we find top and bottom water reservoirs from an input word (see Figure 10(b)). Based on the base-area points of a reservoir the pre-segmentation points are decided (the pre-segmentation points are marked by small vertical lines in Figure 10(b)). Please note that all the reservoirs are not considered for pre-segmentation. Only those reservoirs whose height is greater than 2* R_L are considered for segmentation (R_L is stroke width and discussed earlier). Because of the structural shape of some characters, very small reservoirs may be obtained where segmentation should not be done. We use this threshold to ignore such segmentations. In character pre-segmentation our aim was to segment a word into individual characters as much as possible.

After detection of pre-segmentation points from an input word image, the image is split vertically at each pre-segmentation point and is separated into horizontally non-overlapping zones. A connected component analysis is applied to the split image to detect the boxes enclosing each connected component. These boxes are usually disjoint and do not include parts of other connected components. Connected components in the split word image and their enclosing boxes are shown in Figure 10(c). These boxes are numbered (from left to right) and these numbers are shown in Figure 10(c). These connected components are regarded as primitive segments and each of which corresponds to a full character or a part of a character.

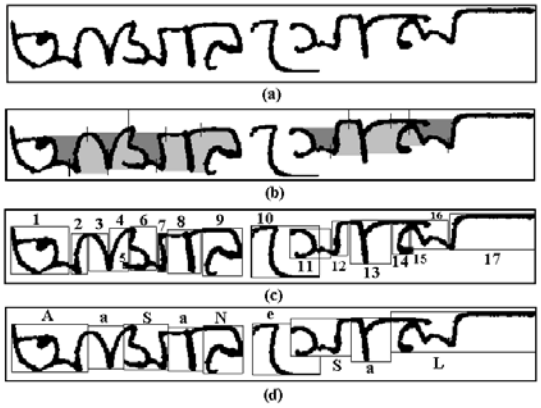


Figure 10. Example of character segmentation. (a) An input word (b) top and bottom reservoirs obtained from the word. Top (bottom) reservoirs are shown by deep (light) gray shade (c) Individual component are marked by disjoint boxes (d) optimum character segmentation. Here segmented individual characters and their respective codes are shown.

5. Feature extraction

Histograms of direction chain code of the contour points of the components are used as features for recognition [7]. For recognition we use 64 dimensional features and the features extraction procedure is described below.

At first the bounding box is divided into 7 x 7 blocks (as shown in Figure 11c). In each of these blocks the direction chain code for each contour point is noted and frequency of direction codes is computed. Here we use chain code of four directions only as shown in Figure 8(a). [directions 1 (horizontal), 2 (45 degree slanted), 3 (vertical) and 4 (135 degree slanted)]. Thus, in each block, we get an array of four integer values representing the frequencies of chain code in these four directions. These frequencies are used as feature. Histogram of the values of these four direction codes in each block of a Bangla character is shown in Figure 11(d). Thus, for 7 x 7 blocks we get 7 x 7 x 4 =196 features. To reduce the feature dimension, after the histogram calculation in 7 x 7 blocks, the blocks are down sampled with Gaussian filter into 4 x 4 blocks. As a result we have 64 (4 x 4 x 4) features for recognition. Histogram of these values of all the four directions obtained after down sampling is shown in Figure 11(e). The feature vector is height normalized and the height normalization is simply performed by multiplying each vector component by the ratio of standard height to the actual height of the letter. The standard height is considered as 76 in our experiments.

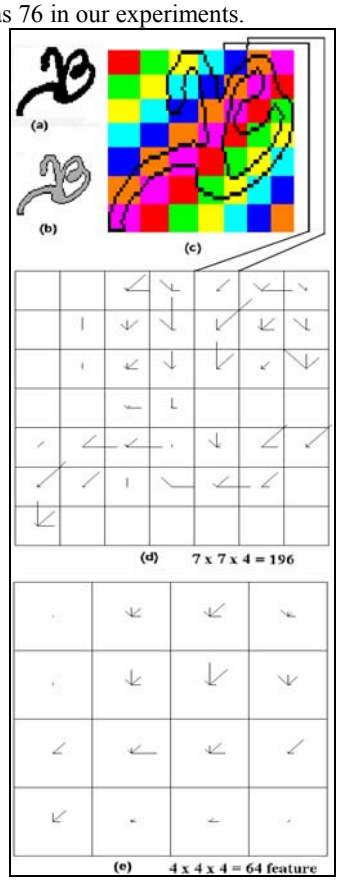


Figure 11. Example of feature extraction (a) A Bangla character. (b) Contour of the character. (c) 7 x 7 segmented blocks shown in the zoomed version of the character of 11(b). (d) Block-wise chain code histogram of contour points. (e) Histogram of chain code direction of the contour points after down sampling into 4 x 4 blocks from 7 x 7 blocks.

One critical point in segmentation-recognition techniques using dynamic programming is the speed of feature extraction, because the correct segmentation points have to be determined in optimization process with respect to the total likelihood of the resultant characters. The use of the cumulative orientation histogram enables one to realize high-speed feature extraction. Border following and orientation labeling are performed only once to an input word image, and the orientation feature vector of a rectangular region including one or more boxes is extracted by a small number of arithmetic operations for high-speed feature extraction [7].

6. Segmentation-recognition using dynamic programming

The core of a dynamic programming algorithm is the module that takes a word image, a string to incorporate contextual information, and a list of the primitives from the word image and returns a value that indicates the confidence that the word image represents the string. Given a lexicon word, the primitive segments are merged and matched against the characters in the lexicon word so that the average character likelihood is maximized using dynamic programming.

6.1. Markov chain representation

The number of the boxes (or the primitives) is usually 1.2 to 2 times as many as the number of actual characters in the word. In order to merge these primitive components into characters and find the optimum character segmentation, dynamic programming (DP) is applied using the total likelihood of characters as the objective function [7]. The ASCII lexicon of possible words is utilized in the process of dynamic programming to incorporate contextual information. The likelihood of each character is calculated using the modified quadratic discriminant function. To apply the DP, the boxes are sorted from left to right according to the location of their centroids. If two or more boxes have the same x coordinates of their centroids, they are sorted from top to bottom. Numbers at the top/bottom of the boxes in Figure 10(c) show the order of the sorted boxes. It is worth observing that the disjoint box segmentation and the box sorting process reduces the segmentation problem to a simple Markov process, in most cases. For example, the boxes 1 and 2 correspond to letter "A" of the Bengali word *AaSaNeSaL,* box 3 and 4 correspond to "a", box 5 to 7 correspond to "S" ….. and so on. See Figure 10(d) where characters' codes of this word image are given. These assignments of boxes to letters are represented, for example, by

A-a –S--aNe-S-aL--

Where "-" is used to denote the continuation. The assignment is also represented, for example, by

	A	a	S	a	N	e	S	a	L
i -->	1	2	3	4	5	6	7	8	9
j(i) -->	2	4	7	8	9	10	12	13	17

where i denotes the letter number, $j(i)$ denotes the number of the last box corresponding to the i-th letter. Note that the number of the first box corresponding to the i-th letter is $j(i-1)+1$. Given $[j(i),i=1,2,..,n]$ the total likelihood of characters is represented by

$$L = \sum_{i=1}^{n} l(i,j(i-1)+1,j(i)) \quad \ldots\ldots(2)$$

where $l(i,j(i-1)+1,j(i))$ is the likelihood for i-th letter.

The optimal assignment (the optimal segmentation) that maximizes the total likelihood is found in terms of the dynamic programming as follows:

The optimal assignment $j(n)^*$ for n-th letter is the one such that

$$L^* = L(n,j(n)^*) = \max_{j(n)} L(n,j(n)) \quad \ldots\ldots(3)$$

where $L(k,j(k))$ is the maximum likelihood of partial solutions given $j(k)$ for the k-th letter, which is defined and calculated recursively by

$$L(k,j(k)) = \max_{j(1),j(2)\ldots\ldots j(k-1)} \{ \sum_{i=1}^{k} l(i,j(i-1)+1,j(i)) \}$$

$$= \max_{j(1),j(2)\ldots j(k-1)} [l(k,j(k-1)+1,j(k)) + \sum_{i=1}^{k-1} l(i,j(i-1)+1,j(i)) \}]$$

$$= \max_{j(k-1)} [l(k,j(k-1)+1,j(k)) + \max_{j(1),j(2)\ldots\ldots j(k-2)} \{ \sum_{i=1}^{k-1} l(i,j(i-1)+1,j(i)) \}]$$

$$= \max_{j(k-1)} [l(k,j(k-1)+1,j(k)) + L(k-1,j(k-1))] \quad \ldots..(4)$$

and $L(0,j(0)) = 0$ for $j(0)=1,2,...,m$ $\ldots\ldots(5)$

Starting from (5), all $L(k,j(k))$'s are calculated for $k=1,2,...,n$ using (4) to find $j(n)^*$ using (3). The rest of $j(k)^*$'s ($k=n-1,n-2,...,1$) are found by back tracking a pointer array representing the optimal $j(k-1)^*$'s which maximizes $L(k,j(k))$ in (4).

6.2. MQDF for Character likelihood

Character likelihood is calculated by the following modified quadratic discriminant function (MQDF) [6].

$$g(X) = \{|X-\hat{M}|^2 - \sum_{i=1}^{k} \frac{\lambda_i}{\lambda_i + h^2} [\phi_i^T (X-\hat{M})]^2\}/h^2 + \ln[h^{2(n-k)} \prod_{i=1}^{k} (\lambda_i + h^2)] \quad \text{---(6)}$$

where X denotes the input feature vector, $\hat{M}$ denotes the sample mean vector for each character class, and λ_i and ϕ_i denote the eigenvalues and eigenvectors of the sample covariance matrix. Values of constants h^2 and k are selected experimentally to achieve the best performance. In the following experiments, k is set to 10 and h^2 to $3/8*\sigma^2$, where σ^2 is the mean of eigenvalues λ_i's over i and character classes.

Given a feature vector, $g(X)$ is calculated for a character class specified by a word lexicon. Based on the character likelihood, total likelihood of a word is found in terms of the dynamic programming technique as discussed above using the lexicon words. Please note that segmentation-recognition is repeated for all lexicon words. For an input word, its likelihood corresponding to each dictionary word is computed and the dictionary word for which the input word gets optimal likelihood is the recognition result of that input word.

7. Results and discussion

For the experiment of the word recognition scheme proposed in this paper we collected 8625 handwritten data of 84 Indian city-names written in Bangla script. Number of samples of each city-name was at least 100. These data are collected at 300 DPI from handwritten address block of Indian postal documents as well as from some individuals of different professionals like students, researchers, businessmen etc.

For the computation of word recognition results, we have used 5-fold cross validation technique dividing the above data into 4:1 ratio for training and testing. From the above experiment we noted that the word recognition accuracy of the proposed scheme was 87.21%, when 0.07% rejection is considered. Also, from the experiment we noted that 90.56% (92.14%) accuracy was obtained when first two (three) top choices of the recognition results were considered. Details results with different choices are shown in Table 1.

Also, from the experiment we noted that 91.21% accuracy is obtained when 3.73% rejection is considered in the proposed system. Word recognition results with different rejection rates are given in Table 2. Rejection is done based on the following criteria (i) optimal likelihood value of the best recognized city-name is low, and (ii) difference of the optimal likelihood values of the best and the second-best recognized city-names is very small.

Table 1: Word recognition results based on different choices from top (with 0.07% rejection rate).

Number of choices from top	Accuracy
Only 1 choice	87.21%
Only first 2 choices	90.56%
Only first 3 choices	92.14%
Only first 4 choices	93.00%

Table 2: Word recognition results with different rejection rates.

Rejection rate	Accuracy	Rejection rate	Accuracy
3.73%	91.21%	18.76%	98.02%
9.94%	94.06%	24.32%	99.05%

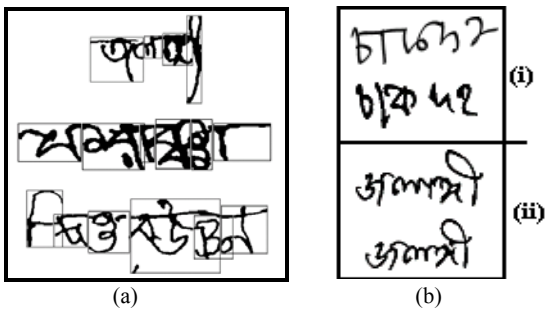
From the experiment we noticed that one of the main reasons for not getting higher recognition rates was the complex structure of some of the city-names due to presence of compound characters. Recognition of

Bangla handwritten compound characters is a complex problem because of the handwritten variability in the compound characters of different individuals. In English there is no compound character and number of characters is only 52 (26 upper-case and 26 lower-case). In total there are more than 280 character shapes in Bangla. Thus, recognition task of Bangla handwritten word is more difficult than that of English.

From the experiment we also noticed that much higher result was obtained when city-names written only by basic characters were considered. We obtained 95.38% accuracy (with 0.12% rejection) by our proposed scheme when it was tested on the city-names written in basic characters only (note that we considered 84 city-names and out of them 39 city-names were written only by basic characters). Also we noticed that 98.22% word recognition accuracy was obtained from the city-names written in basic characters when 5.10% rejection rate was considered. We obtained 100% accuracy when rejection rate is 15.83%. Details results are shown in Table 3.

From the experiment we noted that some errors occurred when the set of character pre-segmentation points obtained from our pre-segmentation module does not contain actual character segmentation points. Examples of some of such erroneous words with their segmentation are shown in Figure 12a. Another reason of mis-recognition was the shape similarity of some of the city-names. Examples of some similar-shaped city-names are shown in Figure 12b. In Figure 12b(i) there are two different city-names and these two city-names differ only by the last character of the city-name. Although the last characters are different but their shape is very similar because of the writing style of different individuals. Similarly, in Figure 12b(ii) there are two different city-names but they are very similar. Because of such shape similarity of two or more words some errors occur in our proposed scheme.

To the best of our knowledge there is no work on Bangla city-name recognition except the NSHP-HMM based method [14]. From the experiment we noticed that our current method gives about 2% better results than the NSHP-HMM method.



(a) (b)

Figure 12. Examples of some mis-recognition results. (a) because of improper segmentation. (b) because of similar shapes.

Table 3: Word recognition results of city-names written in basic characters only.

Rejection rate	Accuracy	Rejection rate	Accuracy
2.85%	97.10%	8.36%	99.33%
5.10%	98.22%	15.83%	100%

References

[1] A. Amin, "Off-line Arabic character recognition: The state of the art", *Pattern Recognition*, vol. 31, 1998, pp. 517-530.

[2] M. Y. Chen, A. Kundu and J. Zhou, "Off-Line Handwritten Word Recognition Using a Hidden Markov Model Type Stochastic Network", *IEEE Trans. on PAMI*, vol. 16, 1994, pp. 481-496.

[3] J. T. Favata, "Offline General Handwritten Word Recognition Using an Approximate BEAM Matching Algorithm", *IEEE Trans. on PAMI*, vol. 23, 2001, pp. 1009-1021.

[4] P. D. Gader, M. Mohamed and J. H. Chiang, "Handwritten Word Recognition with Character and Inter-character Neural Networks", *IEEE Trans. on SMC*, vol. 27, 1997, pp. 158-164.

[5] J. Hull, T. Ho, J. T. Favata, V. Govindaraju and S. Srihari, "Combination of segmentation-based and holiostic handwritten word recognition algorithms", *In Proc. 2nd IWFHR*, 1992, pp. 229-240.

[6] F. Kimura, K. Takashina, S. Tsuruoka and Y.Miyake, "Modified quadratic discriminant functions and the application to Chinese character recognition", *IEEE Trans. on PAMI*, vol. 9, 1987, pp. 149-153.

[7] F. Kimura, S. Tsuruoka, Y.Miyake and M. Shridhar, "A lexicon directed algorithm for Recognition of Unconstrained Handwritten words", *ICICE Trans. on Information and Systems,* vol. 77, 1994, pp. 785-793.

[8] H. Liu and X. Ding, "Handwritten character recognition using gradient features and quadratic classifier and multiple discrimination scheme", *In Proc. 8th ICDAR*, 2005, pp. 19-23.

[9] C. L. Liu and M. Koga and H, Fujisawa, "Lexicon Driven Segmentation and Recognition of Handwritten Character Strings for Japanese Address Reading", *IEEE Trans. on PAMI*, vol. 24, 2002, pp. 1425-1437.

[10] B. S. Madhvanath and V. Govindaraju, "The Role of Holistic Paradigms in Handwritten Word Recognition", *IEEE Trans. on PAMI*, vol. 23, 2001, pp.149-164.

[11] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms", *IEEE Trans. on SMC*, vol. 9, 1979, pp. 62-66.

[12] U. Pal and P. P. Roy, "Multi-oriented and curved text lines extraction from Indian documents", *IEEE Trans. on SMC-B*, vol. 34, 2004, pp. 1676-1684.

[13] R. Plamondon and S. N. Srihari, "On-line and off-line handwritten recognition: A comprehensive survey". *IEEE Trans. on PAMI*, vol. 22, 2000, pp. 62-84.

[14] K. Roy, S. Vajda, U. Pal, B. B. Chaudhuri, and A. Belaid, "A System for Indian Postal Automation", *In Proc. 8th ICDAR*, 2005, pp. 1060-1064.