2011 International Conference on Document Analysis and Recognition

Handwritten Street Name Recognition for Indian Postal Automation

Umapada Pal
Computer Vision and Pattern Recognition Unit; Indian Statistical Institute, Kolkata-108, India
Email: umapada@isical.ac.in

Ramit Kumar Roy
St. Xavier's College
30 Park Street
Kolkata-16, India
Email: ramitkumar.roy@gmail.com

Fumitaka Kimura
Graduate School of Engg., Mie University; 1577 Kurimamachiya-cho, TSU, 514-8507, Japan
Email: kimura@hi.info.mie-u.ac.jp

Abstract- **Although for postal automation there are many pieces of work towards street name recognition on non-Indian languages, to the best of our knowledge there is no work on street name recognition on Indian languages. In this paper we proposed a scheme for recognition of Indian street name written in Bangla script. Because of the writing style of different individuals some of the characters in a street name may touch with its neighboring characters. Accurate segmentation of such touching into individual characters is a difficult task. To avoid such segmentation, here we consider a street name string as word and the street name recognition problem is treated as lexicon driven word recognition. Some of the street names may contain two or more words and we have concatenated these words to have a single word. In the proposed method, at first, street names are binarized and pre-segmented into possible primitive components (individual characters or its parts) analyzing their cavity portions. Pre-segmented components of a street name are then merged into possible characters to get the best street name. Dynamic programming (DP) is applied for the merging using total likelihood of characters as the objective function. To compute the likelihood of a character, modified quadratic discriminant function (MQDF) is used. Our proposed system shows 99.03% reliability with 18.80% rejection, and 0.79% error rates when tested on 4450 handwritten Bangla street name samples.**

Keywords- Handwritten character recognition; Handwritten word recognition; Street name recognition; Bangla script; Indian postal automation.

I. INTRODUCTION

Postal automation is a topic of research interest for last two decades and many pieces of published article are available towards postal automation of non-Indian language documents [1,5]. At present postal sorting machines are available in several countries like USA, UK, Canada, Japan, France, Germany etc., but no postal automation machine is available for India. System development towards postal automation for a country like India is more difficult than other countries because of its multi-lingual and multi-script behavior. In India there are more than 21 official languages and 11 different scripts are used to write these languages. An Indian postal document may be written in any of these official languages. Although there are many languages and scripts in India, only a few pieces of work have been done towards postal automation in India [2,3].

For postal automation there are many pieces of work towards street name recognition on non-Indian languages [7-9]. To the best of our knowledge there is no work on street name recognition on Indian languages and in this paper we proposed a scheme for recognition of Indian street name written in Bangla script. Indian street name is a character string and we can locate a post office with this street name. Street name is generally written above the city-name of a postal document in India. Street name of a postal document is marked by a rectangular box in Fig.1. Because of the writing style of different individuals some of the characters in a street name may touch with its neighboring characters. The segmentation of touching characters is the main bottleneck in the handwritten recognition system and many algorithms on the segmentation of touching strings have been proposed in the past years [10]. To avoid such segmentation, here we consider a street name string as word and the street name recognition problem is treated as lexicon driven word recognition. Some of the street names may contain two or more words. Proper segmentation of the words from the street names is a difficult task and to get better accuracy we have concatenated these words to have a single string (word).

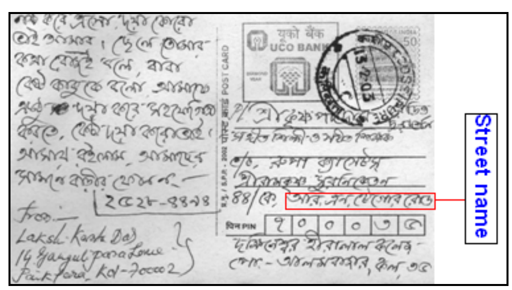

Figure1. Sample of an Indian postal document.

In the proposed method, at first, street names are binarized and pre-segmented into possible primitive components (individual characters or its parts). Each primitive ideally consists of a single character or a sub-image of a single character. In order to merge these primitive components into characters and to find optimum character segmentation of a street name, dynamic programming (DP) is applied using the total likelihood of characters as the objective function. To compute the likelihood of a character, an MQDF based

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.103

on the directional features of the contour points of the components is used. Block diagram of the proposed system is shown in Fig.2

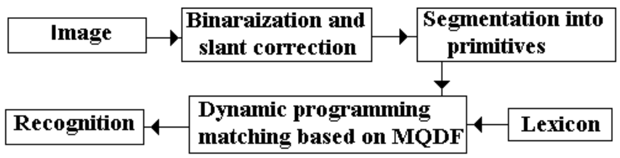

Figure 2. Block diagram of the proposed system.

II. PROPERTIES OF BANGLA SCRIPT

Bangla, the secondmost popular language in India and the fifthmost popular language in the world, originated from Brahmi script. About 200 million people in the eastern part of Indian subcontinent speak in this language. Except Bangla, other languages like Assamese, Manipuri etc. are also written in Bangla script. Bangla is also the national language of Bangladesh. The alphabet of the modern Bangla script consists of 11 vowels and 39 consonants. These characters are called *basic characters*. The concept of upper/lower case is absent in this script. Writing style in Bangla script is from left to right. Most of the characters in Bangla script have a horizontal line at the upper part. We call this line as *head-line* (*shirorekha*). In Bangla script a vowel following a consonant takes a modified shape, called *modified characters*. Depending on the vowel, its modified shape is placed at the left, right (or both) or bottom of the consonant. A consonant or vowel following a consonant sometimes takes a new orthographic shape, which we call as *compound character*. There are about 300 compound characters.

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.

Main difficulty of any recognition system is shape similarity. In Bangla some characters have similar shapes. Examples of some similar shaped characters are group-wise shown in Fig.3. From the figure it can be seen that shapes of two characters of a group is very similar and such shape similarity reduces the accuracy of a recognition system. Moreover, because of the larger character set as well as complex-shaped structure of the compound characters of Bangla, development of Bangla handwritten street-name recognition system is more difficult than Roman script.

Figure 3. Examples of some similar shaped characters in Bangla.

III. SLANT CORRECTION AND CHARACTER PRE-SEGMENTATION

A. Slant correction

Slant correction is an important preprocessing step for handwriting recognition. In this paper the gray scale image of an input street name is binarized by Otsu algorithm [4] and the binary image is then slant estimated and corrected using the method proposed by Kimura et al. [6]. The slant estimation is done based on the chain code information of the character contours of an input street name image.

B. Character pre-segmentation

To find optimal character segmentation by the segmentation-recognition scheme using dynamic programming, we pre-segmented the street names into primitives. When two or more characters sit side by side in a street name the neighboring characters may touch and generate big cavity region. We find this cavity region based on the profile information and the deepest point of each cavity region is considered for pre-segmentation. Please note that all the cavities are not considered for pre-segmentation. We compute the heights of different cavities obtained in a street name and those cavities having height greater than $4*R_L$ (Here R_L is stoke width and its computation is discussed later) are considered for segmentation. Because of the structural shape of some characters, some small cavities may be obtained where segmentation should not be done and we use this threshold to ignore the segmentations of such small cavities. The value of this threshold is decided from experiment and it depends on the stroke width of an input street name. In character pre-segmentation our aim was to segment a street name into individual characters as much as possible avoiding much over-segmentation. The upper (lower) cavity portions of a Bangla street name shown in Fig. 4(a) are marked by light (deep) grey in Fig.4(b). After detection of pre-segmentation columns from the cavities of an input street name image, the image is split vertically at each pre-segmentation column and is separated into 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 street name image and their enclosing boxes are shown in Fig.4(c). These boxes are numbered (from left to right) and these numbers are shown at the upper/lower side of each box (see Fig.4c). These connected components are regarded as primitive segments and each of which corresponds to a full character or a part of a character.

The stroke width (R_L) is calculated as follows. The image is scanned in row and column-wise and different run-lengths (of foreground pixels) with their frequencies are computed. If a component has n different run-lengths r_1, r_2,..r_n with frequencies f_1, f_2 ...f_n, respectively, then R_L=r_i where f_i = max(f_j), j = 1...n.

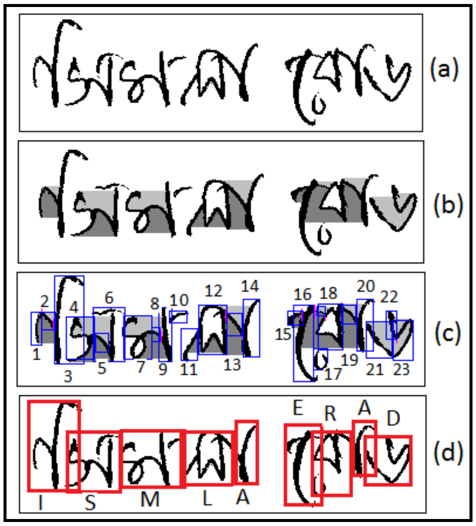

Figure 4. Examples of street name primitive segmentation. (a) An input street name (b) upper (lower) cavity portions of the street name are marked by light (deep) grey. (c). Bounding boxes of pre-segmented individual components are numbered from left to right (d) Optimum character segmentation. Segmented characters and their code are shown.

IV. FEATURE EXTRACTION

Histograms of direction chain code of the contour points of the components are used as features for recognition [5]. Extraction procedure of the 64 dimensional features is described below.

At first the bounding box is divided into 7 × 7 blocks (as shown in Fig.5c). 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 Fig.6. [direction n_0 (horizontal), n_1 (45 degree slanted), n_2 (vertical) and n_3 (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 Fig.5(d). Thus, for 7 × 7 blocks we get 7 × 7 × 4 =196 features. To reduce the feature dimension, after the histogram calculation in 7 × 7 blocks, the blocks are down sampled into 4 × 4 blocks using a Gaussian filter. As a result we have 64 (4 × 4 × 4) dimensional features for recognition. Histogram of these values of all the four directions obtained after down sampling is shown in Fig.5(e). The feature vector is divided by the height of the bounding box to make it size independent.

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 for feature extraction and orientation labeling are performed only once to an input street name 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 [6].

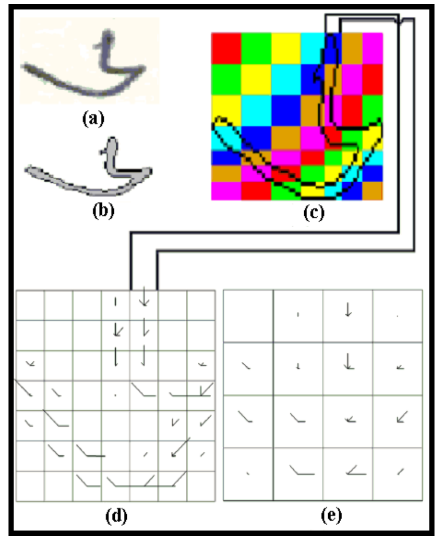

Figure 5. Example of feature extraction (a) A Bangla character. (b) its contour (c) 7 x 7 segmented blocks shown in the zoomed version of 5(b). (d) Block-wise chain code histogram of contour points. (e) Chain code histogram after down sampling into 4 x 4 blocks from 7 x 7 blocks.

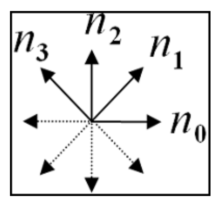

Figure 6. Different chain codes used are shown.

V. SEGMENTATION RECOGNITION USING DYNAMIC PROGRAMMING

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

A. *Markov chain representation*

The number of the primitives of an input street name image is usually 1.2 to 2.5 times the number of characters in the street name image. 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 [6]. The ISCII (or Bangla code) lexicon of the considered street names of Bangla 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 Fig.4(c) show the order of the sorted boxes. It is worth observing that the disjoint box segmentation and the box sorting process reduce the segmentation problem to a simple Markov process, in most cases. For example, the boxes 1, 2 and 3 correspond to character "I" of the Bangla street name 'ISMLAERAD' (shown in Fig.4a), boxes 4, 5 and 6 correspond to character "S", boxes 7 to 10 correspond to character "M", boxes 11 and 13 correspond to character "L", box 14 corresponds to character "A" etc. See Fig.4(d) where characters' codes of this street name image are given. These assignments of boxes to character are represented by:

	I	S	M	L	A	E	R	A	D
i -->	1	2	3	4	5	6	7	8	9
j(*i*) -->	3	6	10	12	13	14	16	18	21

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

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

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:

$$L^* = L(n,j(n)^*) = \text{Max } L(n,j(n)) \quad \text{------(2)}$$

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

$$L(k,j(k)) = \underset{j(1),j(2)......j(k-1)}{Max} \{\sum_{i=1}^{k} l(i, j(i-1)+1, j(i))\}$$

$$= \underset{j(k-1)}{Max}[l(k, j(k-1)+1, j(k)) + L(k-1, j(k-1))] \cdots\cdots\cdots(3)$$

$$and\, L(0, j(0)) = 0 \quad for = j(0) = 1,2,\ldots m \cdots\cdots\cdots(4)$$

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

B. MQDF for Character likelihood

Character likelihood is calculated by the following modified quadratic discriminant function [11].

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

$$\ln[h^{2(n-k)}\prod_{i=1}^{k}(\lambda_i+h^2)] \quad \text{............(5)}$$

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 20 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 street name lexicon.

VI. RESULT AND DISCUSSIONS

Data details: For the experiment of the street name recognition scheme proposed in this paper we collected 4450 handwritten street name samples in Bangla script. Number of total street name class was 89. Number of samples in a class was 50. These data are collected from individuals and digitized at 300 DPI. The dataset will be available freely to the researchers on request to the authors. From the dataset we noted that 52 street names contain two words, 31 street names contain three words, 5 street names contain four words, and one street contains five words. The learning samples of Bangla characters for the MQDF classifier were extracted from city name samples collected independently [2].

Measures of result computation: For recognition result computation we used different measures and they are defined as follows: Recognition rate = $(N_C*100) / N_T$, Error rate = $(N_E*100) / N_T$, Rejection rate = $(N_R*100) / N_T$, Reliability = $(N_C*100) / (N_E+N_C)$, Where N_C is the number of correctly classified street names, N_E is the number of misclassified street names, N_R is the number of rejected street names and N_T is the total number of street names tested by the classifier. Here $N_T = (N_C + N_E + N_R)$.

Global recognition results: From the experiment we noted that the street name recognition accuracy of the proposed scheme was 91.13%, when no rejection was considered. Also, from the experiment we noted that 93.92% (94.49%) accuracy was obtained when first two (three) top choices of the recognition results were considered without any rejection. Detailed results with different choices are shown in Table I. From the table it can be noted that recognition accuracy increases 2.79% when we consider two top choices instead of one top choice.

TABLE I. STREET NAME RECOGNITION RESULTS BASED ON DIFFERENT TOP CHOICES (WITHOUT ANY REJECTION)

Number of top choices	1	2	3	4	5	6
Recognition rate (%)	91.13	93.92	94.49	94.89	95.28	95.55

From the experiment we also noted the street name accuracy in terms of number of words present in a street name. Details results are shown in Table II. From the table it can be noted that accuracy of street name recognition decreases when number of words in a street name increases.

TABLE II. STREET NAME ACCURACY IN TERM OF NUMBER OF WORDS PRESENT IN STREET

Number of words in street name	Number of such street names in our dataset	Recognition rate
2	52	91.37%
3	31	91.17%
>=4	6	88.88%

To give the idea of accuracy based on different number of street name classes, we also computed such results. We obtained 94.37% (without any rejection) accuracy when only 33 street names with their 1650 (33x50) samples are used. Accuracy of 93.65% obtained when 54 street names with their 2700 (54x50) samples are used. We noted that accuracy reduces only 0.72% (94.37-93.65) when number of street name classes increases from 33 to 54.

Rejection versus reliability results: From the system we also computed rejection vs. reliability of our system. We noted that system provides 99.03% (99.52%) reliability when error and rejection rates are 0.79% (0.35%) and 18.80% (28.29%), respectively. Street name recognition reliability with different rejection rates is given in Table III. Rejection is done based on: (i) optimal likelihood value of the best recognized street name, and (ii) difference of the optimal likelihood values of the best and the second-best recognized street names.

TABLE III. ERROR AND RELIABLITY RESULT OF THE PROPOSED SYSTEM WITH RESPECT TO DIFFERENT REJECTION RATES

Reliability (%)	Error rate (%)	Rejection rate (%)
96.53	3.14	9.63
97.05	2.62	11.26
97.52	2.17	12.45
98.05	1.68	13.83
98.56	1.21	15.81
99.03	0.79	18.80
99.52	0.35	28.29

Comparison of results: To the best of our knowledge there is no work on handwritten street name recognition on Indian scripts. Hence we cannot compare our results. However to get idea about the accuracy on the street names of the earlier reported work, result reported by Kim and Govindaraju [7] is presented here. Kim and Govindaraju [7] obtained 83% accuracy where 107 US street names are considered. We obtained 91.13% accuracy on 4450 samples of 89 Bangla street names.

Error analysis: From the experiment we noted that most of the errors occurred because of wrong segmentation i.e. when the set of character pre-segmentation points obtained from our pre-segmentation module did not contain actual character segmentation points. Another reason of misrecognition was the shape similarity of some of the street names. 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 street names due to the presence of compound characters in Bangla.

VII CONCLUSION

In this paper we proposed a system for Bangla handwritten street name string recognition and the street name string recognition problem is treated as lexicon driven word recognition. We obtained 99.52% reliability from our proposed system when error and rejection rates are 0.35% and 28.29%, respectively. This is the first work on street name recognition among Indian languages and the proposed method can be used for other Indian scripts. The dataset developed for this work is the first dataset on Indian street name and the dataset will be available freely to the researchers on email request to the authors.

REFERENCES

[1] S. N. Srihari and E. J. Keubert, “Integration of handwritten address interpretation technology into the United states Postal service Remote Computer Reader System”, In Proc. 4th Int. Conference on Document Analysis and recognition, pp. 892-896, 1997.
[2] U. Pal, K. Roy and F. Kimura, “A Lexicon Driven Handwritten City Name Recognition Scheme for Indian Postal Automation”, IEICI Trans. Inf. and Systems, Vol.E92-D, pp. 1146-1158, 2009.
[3] S. Vajda, K. Roy, U. Pal, B. B. Chaudhuri and A. Belaid, “Automation of Indian Postal Documents written in Bangla and English”, Int. Journal of PRAI, Vol. 23, pp. 1599-1632, 2009.
[4] N. Otsu, “A threshold selection method from gray-level histograms”, IEEE Trans. SMC, vol.9, pp.62-66, 1979.
[5] F. Kimura, Y. Miyake and M. Sridhar, “Handwritten Address Interpretation Using Extended Lexicon Word Matching”, In Proc. Int. Workshop on Frontiers in Handwriting Recognition, pp. 369-372, 1996.
[6] F. Kimura, S. Tsuruoka, Y. Miyke and M. Shridhar, “A Lexicon Directed Algorithm for Recognition of Unconstrained Handwritten Words”, IEICE Trans. Inf. and System, Vol.E7-D, pp.785-793, 1994.
[7] G. Kim and V.Govindaraju, “Handwritten phrase recognition as applied to street name images”, Pattern Recognition, vol. 31, pp. 41-51, 1998.
[8] S. Madhvanath, E. Kleinberg and V. Govindaraju, “Empirical Design of A Multi-Classifier Thresholding/Control Strategy for Recognition of Handwritten Street Names”, Int. Journal of PRAI, Vol. 11, pp. 933-946, 1997.
[9]A. El Yacoubi, J. M. Bertille and M. Gilloux, "Conjoined location and recognition of street names within a postal address delivery line", Proc. 3rd Int. Conference on Document Analysis and recognition, 1024-1027, 1995.
[10] R. Plamondon and S. N. Srihari, “On-Line and off-line handwritten recognition: A comprehensive survey”, IEEE Trans on Pattern Analysis and Machine Intelligence, Vol.22, pp.62-84, 2000.
[11] F. Kimura, K. Takashina, S. Tsusuoka and Y. Miyake, “Modified quadratic discriminant functions and the application to Chinese character recognition, IEEE Trans on Pattern Analysis and Machine Intelligence, Vol.9, pp.149-153, 1987.