ISSN No. 0976-5697

Volume 2, No. 6, Nov-Dec 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A Novel Scheme for Online Bangla Handwriting Recognition

K. Roy
Dept. of Computer Science
West Bengal State University, Barasat
Kolkata, India
Kaushik.mrg@gmail.com

Abstract – Online handwriting recognition is very challenging for Indic scripts because of its huge no of shapes in total character set (isolated, modified and compound). To handle such situation we plan to use strokes for recognition rather than character. This paper describes a procedure to recognize online Bangla handwriting recognition in unconstrained domain based on strokes. For simplicity of recognition he strokes divided into five different categories in the first phase and in second phase detailed classification is done. Next based on the recognized stroke we plan to recognize the characters/words. We have obtained encouraging results in the experiment.

Keywords: stroke, special character, Valid character set, May be valid character set, modifier character set, not character set, Neural Network, classifier.

I. INTRODUCTION

The goal of handwriting recognition [1-3] is to identify an input character's image correctly and it is an application area of pattern recognition, in addition the conversion of handwritten text image to editable form is important to many automated systems. Two main techniques are used for character recognition depending on the intended application. The first is off-line optical recognition (OCR), which accepts it's input from a digital scanner or from a picture using some image processing algorithm needed before classification step [2-3]. The alternative approach is online character recognition, which accepts input data in real time and then computes the relationship between points to extract the features in real time [1]. The work proposed in this paper is based on online approach.

Data entry using pen-based devices is gaining popularity in recent times. This is so because machines are getting smaller in size and keyboards are becoming more difficult to use. Also, data entry for Indian scripts having large alphabet size is difficult using standard keyboard. Moreover, there is an attempt to mimic the pen and paper metaphor by automatic processing of online handwriting. Work on online character recognition started gaining momentum about forty years ago.

Many techniques are available for on-line recognition of English, Arabic, Japanese and Chinese [1-10] characters but there are only a few pieces of work [11-14] available towards Indian characters although India is a multi-lingual and multi-script country. Connell et al. [11] also proposed a work on Devnagari on-line character recognition. Joshi et al. [12] proposed an elastic matching based scheme for on-line recognition of Tamil character recognition. Garian et al. [13] presented a preliminary study on online character recognition. Roy et al. [14] presented a preliminary study on online Bangla character recognition. They considered only the main characters that occur in the core-strip neglecting the ascending and the descending parts of the characters. In this paper we propose a bi-stage system for unconstrained on-line Bangla handwritten recognition based on strokes.

It is yet to get full attention from researchers for online Bangla handwriting recognition. There are some works are available on online isolated Bangla character/numeral recognition in the literature include Garain et al. [13], Roy et al. [14], Parui et al. [15]. Garian et al. [13] presented a preliminary study on online character recognition. Roy et al. [14] presented a preliminary study on online Bangla character recognition.

There are twelve scripts in India and in most of these scripts the number of alphabets (basic and compound characters) is more than 250 [16], which makes keyboard design and subsequent data entry a difficult job. Hence, online recognition of such scripts has a commercial demand. Although, a number of studies [17-18] have been done for offline recognition of a few printed Indian scripts like Devnagari, Bangla, Gurumukhi, Oriya, etc. with commercial level accuracy, but to the best of knowledge no system is commercially available for online recognition of any Indian script.

In this work only a total 59 basic strokes are considered to recognize purpose. By stroke we mean collection of pen points between a pen down and pen up. In other words the number of sample points collected by a continuous writing of pen without lifting it. Recognition is based on neural network. After segmentation of Bangla word and feature extraction when a segmented part goes to neural network as input for recognition, it falls under any one of possible 59 classes. In that case success rate is not too good. So, in this work it has been tried to recognize Bangla word using two-phase neural network. Main target is to improve success rate by reducing the number of output classes.

The organization of the paper is as follows. In Sec. 2 Data collection and segmentation described. Sec. 3 deals with feature extraction and the classifier is in Sec. 4. Finally, in Sec. 5 the result and their analysis is given.

II. DATA COLLECTION & SEGMENTATION

A. *Data Collection:*

On-line handwriting recognition involves the automatic conversion of text as it is written on a special digitizer or A4 take note where a sensor picks up the pen-tip movements X (t), Y (t) as well as pen-up/pen-down switching. That kind of data is known as digital ink and can be regarded as a dynamic representation of handwriting. The ink signal is captured by either:

A paper based capture device
A digital pen on patterned paper
A pen-sensitive surface such as a touch screen

In this work we use pen positions (x, y) and pen pressure (z) sampled at a certain interval from the pen tablet. If the stroke is continuing i.e. pen pressure is ‘on’ i.e. ‘pen-down’, the value of z at a particular point or pixel will be 1 and x, y denotes the pixel’s x coordinate values and y coordinate values when using pen we write any Bangla word or character.

The information on strokes and trajectories is mathematically represented in an ink signal composed of a sequence of 2D points ordered by time. No matter what the handwriting surface may be, the digital ink is always plotted according to a matrix with x and y-axes and a point of origin.

Online data acquisition captures just the information needed, which is trajectory and strokes, to obtain a clear signal [19]. This effective information makes the data easier to process. For our proposed stroke based bi-stage recognition system we require isolated stokes for training but as we naturally used to write character/modifiers in together not as stroke, so we plan to collect the isolated character and extract isolated strokes from them for the training of our proposed module. For testing purpose we plan to collect unconstrained handwritten words. To collect the data we use Wacom tablet and A4 take note with pre-designed the datasheets. The pen pressure represents pen ups and downs in a continuous manner. In our conventional research, we did not use pen-up-down information whether the pen leaves (pen-up) or touches (pen-down) the tablet surface or A4 take note surface. For online data collection, the sampling rate of the signal is considered fixed for all the samples of all the classes of character. Thus the number of points M in the series of co-ordinates samples of all the classes of character.

Thus the number of points M in the series of co-ordinates for a particular sample is not fixed and depends on the time taken to write the sample on the pad. As the number of points in actual trace of the characters are generally large and varies greatly due to high variation in writing speed, a fixed lesser number of points, regularly spaced in time are selected for further processing. The digitizer output is represented in the format of $B\ P_{i=1}^{M} \in R^{2} x\{0,1\}_{B}$, where pB_{iB} is the pen position having x-coordinate (xi) and y-coordinate (yi) and M is the total number of sample points. Let (pi) and (pj) be two consecutive pen points. We retain both of these two consecutive pen points (pi) and (pj) if the following condition is satisfied:

$xP^{2P} + yP^{2P} > mP^{2P}$ .... (i)

Where $x = xB_{iB} - xB_{jB}$ and $y = yB_{iB} - yB_{jB}$. The parameter m is empirically chosen. We have set m equal to zero in Equation (i) to removes all consecutive repeated points. Analyzing a total of 22,372 Bangla character it was found that, for writing Bangla characters, the number of points varies from 14 (৩) to 189 (Rঈ) points. The average number of points in a Bangla character is 72. It was also noted that the character (Rঈ) uses the maximum number of points in average and its value is 115. It is closely followed by ‘Rঊ’ (108), ‘ক্ষ’ (105), & ‘Rআ’ (104). The minimum number of points in an average is used by the character ‘Rচ’ (47) and is closely followed by ‘Rব’ (49) ‘Rদ’ (51).

B. *Preprocessing:*

Analyzing above mentioned 22,372 isolated characters of Bangla it was found that Bangla characters are formed by combination of one or more basic strokes. The recognition of Bangla script is more difficult compare to roman script due to its large size of character set and compound characters. It gets tougher due to presence of multiple strokes while writing Bangla character. By stroke we mean a collection of pen points that are collected between pen down and pen up (without lifting it in between).

The problem of online Bangla handwriting recognition gets more complex due to stroke order variation and variation in number of strokes used to write a character. . For example, let us consider the character ‘আ’. Again it may be seen that 2-6 of the 7 strokes are used for writing the same (as found from statistical analysis of the database). For example see figure 1. So with possible 6 strokes (out of 7) and their order variation makes the recognition process more complicated. Some ways of writing the character ‘আ’ is shown in figure 2.

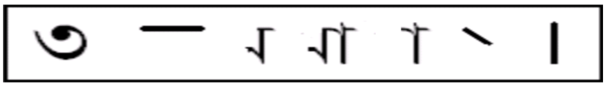


Figure 1. strokes use to write আ

The problem is more complex due to stroke order variation. If only 4 strokes are considered to write the character ‘আ’, they may be again in different order. For example see figure 3.

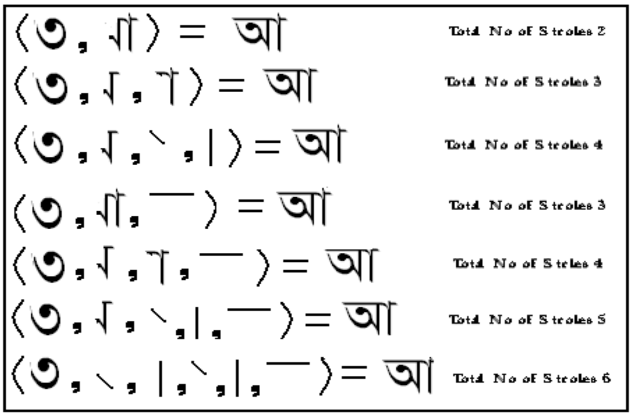


Figure 2. Some of the ways of writing ‘আ’ using some of the possible strokes in some possible combinations.

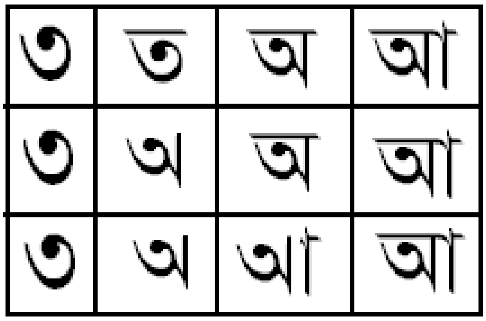


Figure 3. Example of part of the different stroke-order for a character having four strokes

Most of the classification techniques assume that the data is given in a predetermined form, which satisfy certain requirements as to quality, size, invariance, etc. However, these characteristics are commonly not satisfied by on-line handwritten data. The low quality of the data is due basically to the combination if three facts. One is the addition of noise during digitalization, which is generally generated by a badly configured digital tablet. The other is the irregularity generated by inexperienced users having an erratic handwriting. The last are variations in handwriting styles.

To overcome these problems, we use preprocessing, which involves the substitution, removal, reordering, and/or extraction of the data. Preprocessing eliminates noise, normalizes handwriting, and reduces the amount of redundant information, in order to fix the variations of handwriting and facilitate encoding of raw data into feature vectors. (I) Noise and Data reduction, (II) Smoothing, (III) Dehooking, (IV) Feature Extraction.

C. Segmentation:

In case of online character segmentation we have designed an algorithm and the procedure to design the algorithm is as follows:

Each and every online word document file contains x and y coordinates and third value (as pen-up or pen-down) as 1 or 0. If the third value is 0 then a new stroke begins, if it is 1 the stroke continues. In case of online data segmentation the data can be segmented by changing the value of pen stroke feature from 1 to 0. But the major challenges here to identify the coordinate where to segment the word and find out the complete character according to bangle script. In the process of character segmentation there is a possibility of improper segmentation, which is called either over segmentation or under segmentation. In case of under segmentation the coordinates where the segmentation is required could not be chosen and in case of over segmentation the coordinates where it is not required but has been segmented.

In Bangla handwriting the movement of each stroke is generally downside. By keeping this concept in mind it has been seen that in a downside movement stroke the point from where that downside movement starts at that point we have to split that stroke. This should be done only in the upper zone i.e. first 33% portion of the total height of the image. In the remaining 67% of the image segmentation is not needed. Generally people write any word in a manner where more than one alphabet is joined with one another. This joining is generally found in the upper 1/3rd. portion of the image (exception in few cases).

Step 1: Store each pixel of the online data in three variables corresponds to X and Y coordinates and pen feature value of 0 or 1 in third variables for identifying strokes.

Step 2: For each third variable value 0 separates each strokes scanning pixels of the word. Calculate the 30% of the height of the entire word image.

Step 3: Select at which point of stroke segmentation is needed. We have to finally segment those points of same or different strokes, which required to be segmented. So, we use one function to check at which pixel it is feasible to segment a stroke. We have to check few features of Bangla characters for this process such as (i) each pixel's distance from the start and end of the stroke, (ii) the width of the stroke up to the pixel in question from the start and end of the stroke, (iii) the height of the stroke up to the pixel in question, (iv) Total stroke distance, (v) Total width of the word. After finding these features we have to take some ratio of (a) each pixel's distance & Total stroke distance, (b) the width of the stroke up to the pixel in question & Total width of the word and thus to decide at which pixel of a particular stroke segmentation is feasible.

Step 4: Now if at a particular pixel it is feasible to segment the stroke, then first we check whether that pixel's y co-ordinate value is 30% of the height or not. If it is not then there will be no segmentation. If it is, then we check whether at that pixel downside movement of the stroke starts or not. For this checking We are taking two points pB_{i-1B} and pB_{i-2B} before the point in question and similarly two points pB_{i+1B} and pB_{i+2B} after that point. Then we have to determine the angle between two line segments pB_{i-2B} to pi and pi to pB_{i+2B}. If this angle satisfies certain range then only at point pi stroke value will be made as 0, that is, in this point the stroke will be splitted. If the y-coordinate of pB_{i-1B} is <= pB_{i-2B} and pB_{iB} <= pi-1 and simultaneously if the y-coordinate of pB_{i+1B} >= pB_{iB} and pB_{i+2B} >= pB_{iB} (i.e. downside movement of stroke) then only angle is calculated. If at a particular point stroke is splitted then we skip next 9 or 10 pixels for checking of feasibility of segmentation.

Step 5: Repeat step 3 and 4 for each pixels and each strokes of the entire word.

Step 6: Now after the final segmentation find out the final strokes of the words and all the strokes need to be added and converted to offline file for compare with original word file. By this approach we have done segmentation on all the words covering all the vowel and consonant modifiers and also covering all the alphabets in Bangla language..

III. FEATURE GENERATION

The accuracy of the classifier is directly dependent on its feature set. For ease of computation we have used a smaller dataset to make them robust. A total feature set of of 105 features (15 + 90) are used for recognition.

The features used are
Structural features (15) [19]
Point based feature (90) [19]

The processed character is transformed into a sequence t = [tB_{1B}… tB_{NB} tB_{N+1B} …. tB_{N+15B} $tB_{N+15+1B}$…..$tB_{N+15+128B}$] of feature vectors tB_{iB} = (tB_{i1B}, tB_{i2B}, tB_{i3B})T (Where I <= N). We calculated the following features:

A. Structural Features:

Gradient (tBN+1B): tN_1=

$$n\sum_{i=0}^{n} x_i y_i - \sum_{i=0}^{n} x_{\,i} \sum_{i=0}^{n} y_i n \sum_{i=0}^{n} x^2 - \sum_{i=0}^{n} x^2$$

$$tN\_2= \sum_{i=0}^{n} y_i \sum_{i=0}^{n} x^2 - \sum_{i=0}^{n} x_i \sum_{i=0}^{n} x_i y_i n \sum_{i=0}^{n} x_{i^2} - \sum_{i=0}^{n} x^2{}_i$$

Here t_{N+1} and t_{N+2} are the gradient and the intercept in the y-axis of the straight line, constituting by the consecutive 3-points respectively.

Length by Width ratio (tB_{N+3B}):

tB_{N+3B} = (max (xB_{iB}) - min(xB_{iB})) / (max(yB_{iB}) – min(yB_{iB})) i = 0, 1,…,N

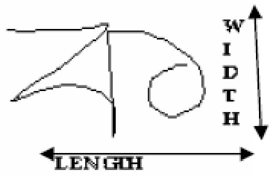


Figure 4. Length and Width of a character

By using this feature the ratio of the length and width of the corresponding stroke is calculated.

a. ***Standard Deviation* (tB_{N+4B}):** The standard deviation measures the spread of the data about the mean value. It is useful in comparing sets of data which may have the same mean but a different range. Here the deviation of each co-ordinate is calculated with respect to its mean value.
b. ***Normalized Start Co-ordinates and End Co-ordinates* (tB_{N+4B}):** In this feature only the first and last co-ordinates in the strokes of a character considered. Taking the first and last co-ordinates normalized them and stored them as feature.
c. ***Crossing of the lines:*** Here the co-ordinate position of the crossing of the stroke is stored with itself as shown in figure 5. In this system only first two crossing are considered.

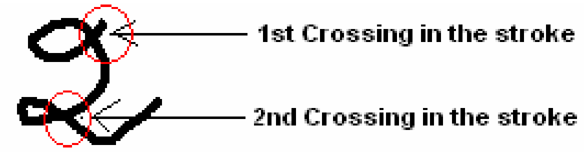
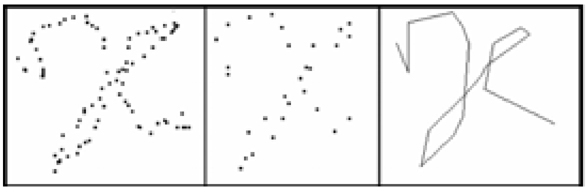


Figure 5. Crossing points of a stroke.

B. *Point Based Feature:*

The strokes are first normalized in to 30 points. The normalization is done in two stages. First the points are re-sampled to fixed number points and then they are converted from equal time sample to equal distant points. The processed character is transformed into a sequence t = [tB$_{1B}$,.., tB$_{NB}$] of feature vectors tB$_{iB}$ = (tB$_{i1B}$,tB$_{i2B}$,tB$_{i3B}$)T [19]. The following features were calculated:



(a) (b) (c)

Figure 6. Feature extraction from a sample stroke is shown. (a) Original stroke, (b) its normalized 30 points used as feature, (c) the normalized stroke.

a. Normalized horizontal (ti1) and vertical (ti2) co-ordinates:

$t_{i_1} = (x_i - \mu_x) / \sigma_y$ And $t_{i2} = (y_i - \mu_y) / \sigma_y$ are the pen co-ordinates normalized by the sample mean $\mu = \frac{1}{n}\sum_{i=1}^{n} p_i$ and standard deviation, $\sigma_y = \sqrt{\frac{\sum_{i=1}^{N}(\mu_y - y_i)^2}{n-1}}$ of the character's sample points.

Tangent slope angle (tB$_{i3B}$): tB$_{i3B}$ = arg ((xB$_{i+1B}$ – xB$_{i-1B}$) + j*(yB$_{i+1B}$ – yB$_{i-1B}$)), with jP^{2P} = -1 and "arg " the phase of the complex number above, is an approximation of the tangent slope angle at point i.

Thus finally, a feature vector sequence is defined as t = [tB$_{1B}$… tB$_{NB}$ tB$_{N+1B}$ …. tB$_{N+15B}$], each vector of it as tB$_{iB}$ = (tB$_{i1B}$,tB$_{i2B}$,tB$_{i3B}$)T is obtained. Here the number of points in which the character is normalized is (N) 30. So a total of 105 (30 X 3 [3 for each point] + 15 [15 local features based on Stroke]) features are used.

IV. RECOGNITION

As we are planning for an unconstrained recognition we have used a bi-stage Neural Network [20] based scheme for recognition. This is because based on property and occurrence we have divided the strokes in to following five categories:

- ***a. Special_character_class:*** only matra is within this class.
- ***b. Valid_character_class:*** strokes that can represent a valid Bangla character such as 'ক'.
- ***c. May_be_valid_character_class:*** strokes that can represent a valid character and also can be a part of any valid character such as 'ব', 'য'.
- ***d. Modifier_character_class:*** ' '.' ' etc are the examples of modifier character class.
- ***e. Not_character_class:*** strokes that are the part of any character such as ' '.

The names are self explanatory we have relied on the characteristics of the strokes for such grouping. For the recognition of Bangla handwritten words, firstly by closely looking the shapes and characteristics of basic Bangla strokes, that is if the stroke alone can describe any valid Bangla character, can describe valid character or also may be a part of any valid character, can be a modifier character, or can be part of a character or a special character, depending on that, basic strokes. After recognition on the first phase e.g. stroke category identification they are feed to appropriate Neural network in the second stage (expect for stroke of first category), for final recognition of the strokes. We have used Multi Layer perceptron Neural Network for the present work with one hidden layer. The number of neurons on the hidden layer are finalized based on some trial runs.

V. RESULT AND DISCUSSION

As mentioned earlier a total of 22,372 characters are collected for our present work. Along with that fro testing of the proposed system 200 words (50 each) were also collected. A total of 30000 manually extracted strokes form those characters were used for the current experiment.

A. *Recognition result on isolated Stroke:*

The recognition rate of the isolated strokes was found to be 88.10 on the test set. The detailed results are shown in Table 1.

B. *Recognition results on Bangla Word:*

Statistical analyses of the result of the second phase of neural Network where actual id of the strokes are determined from its corresponding class category are given in Table 2. The Statistical analysis of result on word recognition after testing 200 Bangla words is shown in Table 3.

After analyzing the errors we found that the main source of errors is due to poor quality of the data, huge stroke number and stroke order variation. We have not used any sort of preprocessing algorithm here, which we plan to add in future. We also plan to add some grammatical rules to validate the output gram using Bangla dictionary using 2nd and higher order of recognition results. We also plan to enhance the stroke segmentation procedure, use more appropriate Bangla specific features for recognition. We plan to incorporate all modifiers and compound characters to make it a full-fledged recognition system Statistical analysis of the result of first phase of neural network [20] where the constituent strokes of the given Bangla word are categorized in terms of their corresponding class is shown in as Table 1. Also the result analysis of the second phase of neural Network where actual id is determined from its corresponding class category. Result for each category is given in Table 2. The Statistical analysis of result after testing 200 Bangla words is shown in Table 3.

Some problems occur due to improper segmentation (Over or Under-Segmentation), in some cases segmentation is proper i.e. correct basic strokes are generated from segmentation but recognizer can't recognize properly because in the second phase it fails to recognize the correct character id from the correct class and it mainly happens in modifier and not character class category. Valid character, may be valid character and special class category strokes are recognized well. Few misclassifications between strokes are noticed between ‘ফ’ and ‘থ’ , ‘ন’ and ‘ল’ etc. The features for modifier and not character class category are also not satisfactory. So there is a scope for further study on modifier and not character class category type strokes and its feature generation to increase recognition rate.

Table I. Results of first phase of neural network

	Training set	**Test set**
Success rate	80.4	78.6
Confidence	84.2	82.7
Misclassification rate	15 .0	16.4
Rejection rate	4.5	4.9

Table II. Rresults of second phase of neural network for different class category

	Special char set		**Valid char set**		**May be valid char set**		**Modifier char set**		**Not char set**	
	training	***Test***	***training***	***Test***	***Training***	***test***	***training***	***test***	***Training***	***test***
Classification	3285	906	4044	1917	4876	2370	1900	895	1340	656
Success rate (%)	100	100	94.8	90	98.6	95.9	92.1	86.6	81.2	79.5
Confidence	100	100	100	96.9	99.9	98.1	97.8	93.6	98.6	96.5
Misclassification rate (%)	0	0	0	2.9	0.1	1.8	2.0	5.9	1.1	2.9
Rejection rate (%)	0	0	5.1	7.0	1.2	2.3	5.8	7.4	17.6	17.5

Table III. Results on a few Bangla Words (50 each)

Words (no. of instance)	**Correctly recognized**	**One stroke error**	**Two stroke error**
ওল (50)	34	12	4
ফল (50)	31	13	6
জল (50)	28	11	11
□□□ **(50)**	32	08	10

VI. REFERENCE

[1] C. C. Tappert, C. Y. Suen, T. Wakahara, “The state of the art in on-line handwriting recognition”, IEEE Trans. PAMI, vol. 12, no. 8, pp. 787-808, 1990.

[2] R. Plamondon and S.N. Srihari, “On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey,” IEEE Trans. PAMI, vol. 22, no. 1, pp. 68-89, Jan. 2000.

[3] I. Guyon, M. Schenkel and J. Denker, “Overview and synthesis of on-line cursive handwriting recognition techniques”, in Handbook of Character Recognition and Document Image Analysis, 1997.

[4] Z. L. Bai and Q. Huo, “A Study on the Use of 8-Directional Features for Online Handwritten Chinese Character Recognition”, In Proc. 8th ICDAR’2005.

[5] S. Madhvanath and V. Govindaraju, “The role of holistic paradigms in handwritten word recognition”, IEEE Trans. PAMI, vol. 23, no. 2, pp. 149–164, 2001.

[6] L. Koerich, R. Sabourin and C. Y. Suen, “Recognition and Verification of Unconstrained Handwritten Words”, IEEE PAMI, vol. 27, no. 10, pp.1509-1522, 2005.

[7] C. A. Higgins, D. M. Ford. “On-line recognition of connected handwriting by segmentation and template matching”, In Proc. 11th ICPR, vol. II, pp. 200-203, 1992.

[8] L. Schomaker, “Using stroke- or character-based self-organising maps in the recognition of on-line, connected cursive script”, PR, vol. 26, no. 3, pp. 443-450, 1993.

[9] J. Shin, “Online cursive hangul recognition that uses DP matching to detect key segmentation points”, Pattern Recognition, vol. 37, no. 11, pp. 2101-2112, 2004..

[10] R. I. Elanwar, M. A. Rashwan, S. A. Mashali, “Simultaneous segmentation and recognition of Arabic characters in an unconstrained online cursive handwritten document”, Proceedings of World Academy of Science, Engineering and Technology, vol. 23, pp. 288-291, 2007.

[11] S. D. Connell, R. M. K. Sinha, and A. K. Jain, “Recognition of Unconstrained Online Devnagari Characters”, In Proc. 15th ICPR, pp. 312-324, 2000.

[12] N. Joshi, G. Sita, A. G. Ramakrishnan, V. Deepu, Sriganesh Madhvanath: “Machine Recognition of Online Handwritten Devanagari Characters” In Proc. 8th ICDAR, pp. 1156-1160, 2005.

[13] U. Garain, B. B. Chaudhuri, T. Pal, “Online Handwritten Indian Script Recognition: A Human Motor Function Based Framework,” In Proc. 16th Int. Conf. on Pattern Recognition, pp. 164-167, 2002.

[14] K. Roy and N. Sharma, T. Pal and U. Pal, “Online Bangla Handwriting Recognition System” , In Proc. 6th International Conference on Advances in Pattern Recognition, pp. 121-126, 2007.

[15] S. K. Parui, U. Bhattacharya, B. Shaw, K. Guin, “A Hidden Markov Models for Recognition of Online Handwritten Bangla Numerals”, Proceedings of the 41st National Annual Convention of Computer Society of India, pp 27-31, 2006.

[16] http://www.ethnologue.com/ethno_docs/distribution.asp?by=country, visited on 11/11/2011.

[17] B. B. Chaudhuri and U. Pal, “A complete printed Bangla OCR system. Pattern Recognition” Vol. 31, No. 5, pp. 531-549 1998.

[18] U. Pal and B. B. Chaudhuri, “Indian Script Character Recognition A Survey.”, Pattern Recognition, 37, pp. 1887-1899, 2004.

[19] K. Roy, “Stroke-Database Design for Online Handwriting Recognition in Bangla”, In Proc. International Conference on C^{3}IT, pp. 190-198, 2009.

[20] K. Roy, C. Chaudhuri, M. Kundu, M. Nasipuri and D. K. Basu, "Comparison of the Multilayer perceptron and the nearest neighbor classifier for handwritten digit recognition", Journal of Information Science and Engineering, Vol. 21, No. 6, pp. 1245-1257, 2005.