ANKUR: BANGLA ONLINE CHARACTER RECOGNITION

P. Mahmud[1], M. R. Rahman[2], M. J. Islam[3], R. M. Rahman[4], M. A. Matin[5]

[1,2,3,4] Department of Electrical and Computer Engineering, North South University
Plot-15, Block-B, Bashundhara, Dhaka 1229, Bangladesh
[5]Department of Electrical and Electronic Engineering Institut Teknologi Brunei, Brunei Darussalam
[1]pallmahmud@gmail.com, [2]refayetbd@gmail.com, [3]jahid1081@gmail.com,
[4]rashedur@northsouth.edu, [5]matin.mnt@gmail.com

Keywords: Bangla OCR, fuzzy features, character recognition, learning and recognition, cursive characters.

Abstract

Bangla alphabet consists of 50 characters. Most of these characters have twists and turns that is very complicated to trace down in real time for an online OCR (Optical Character Recognizer). The main challenge of making a reliable Bangla OCR is to generate analysable features of these curvaceous characters in a real time environment. For an online Bangla OCR, fuzzy logic could be one the most efficient way as it has very low computational requirement. This ensures highest efficiency with economical use of the processing power. The most important part of creating an online OCR is establishing a reliable fuzzy rule-base that would describe the characters to be recognized. However, for Bangla this job is quite cumbersome due to the curvaceous nature of the characters and, not to mention, variety of handwritings of different people. This paper aims to describe a way that can be used to generate a fuzzy-feature database that describes Bangla characters written in different handwritings. It also suggests a system for recognizing any given character with respect to the linguistic variable extracted from the fuzzy-feature database.

1 Introduction

With the advancement in computer technologies, people are more interested in a man-machine interface compared to the conventional ones. An OCR facilitates such an interface. The sole purpose of a handwritten character recognition system is the recognition of data that describes a handwritten object while online recognition system processes time sequenced data. There are already quite a few existing methodologies for recognizing handwritten characters. Most recent works related to character recognition are by techniques, namely, Neural Network [1], Genetic Algorithm [2] , Bayesian Inference [3], Stochastic HMM (Hidden Markov Model) [4], Fuzzy logic [5], Handwritten Alphanumeric Character Recognition [6]. However, all of these have some weaknesses when it comes to recognize a curvaceous script like Bangla.

The accurate recognition of Latin-script (from which the English alphabet is originated), typewritten text is now considered largely a solved problem for offline OCR on applications where clear imaging is available such as scanning of printed documents. Typical accuracy rates on these exceed 99% [7]; total accuracy can only be achieved by human review. Other areas—including recognition of hand printing, cursive handwriting, and printed text in other scripts (especially the East Asian language characters which have many strokes for a single character)—are still the subject of active research. As Bangla is one of those languages, it is highly likely that a successful implementation of an online Bangla character recognition system will provide a way to solve some of the issues that are blocking the way for implementing an online OCR for the other East Asian languages. The objectives of this paper are to segment the Bangla characters by calculating angle difference and eventually establishing a fuzzy rule-base for finding the fuzzy features of each segment for recognition.

The organization of the paper is as follows. Section 2 includes the related work in the area of character recognition. Section 3 describes the overall system by discussing the algorithms and methodologies conducted in this research. Section 4 shows the experimented results of the research. Finally, section 5 concludes by giving future directions of research.

2 Related Works

Even though it is now considered to be a solved problem, it was not always the same for English character recognition. Here are few of the previous works done for English character recognition. The techniques that are used in the research works are given below:

- Counter-propagation neural net (CPN) [8],
- Combination of directional and positional features [9],
- Representation and recognition of handwritten digits using deformable templates [10],
- Fuzzy method for handwritten character recognition [11],
- Use of fuzzy feature descriptions to recognize handwritten alphanumeric characters [6],

There are some works reported in literature that are related to Bangla character recognition as well. They are listed below.

- Online Bengali handwritten recognition with generated fuzzy linguistics rules [12],
- Bangla basic character recognition using digital curvelet transform [13],

- Optical character recognition for Bangla documents using HMM [14],
- Bangla vowel sign recognition by extracting the fuzzy features [15],
- Bangla numeral recognition engine (BNRE) [16].

3 System Overview

This section describes the techniques involved in our proposed online handwriting recognition system. This is a writer-independent system based on the automatically generated fuzzy rules of each character. For an online handwriting recognition system implemented with neural network or pattern recognition, the data obtained needs a good amount of pre-processing including filtering, smoothing, slant removing and size normalization before recognition process. Instead of doing such lengthy pre-processing, we present a simple approach to extract the useful character information. The whole process requires no pre-processing and size normalization whatsoever. A complete flowchart of the proposed system is shown in Figure 1.

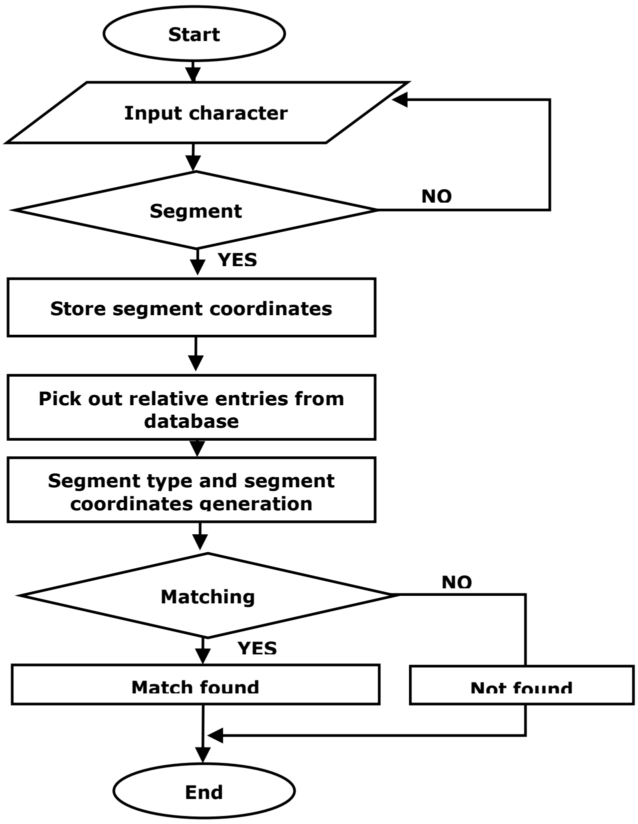

Figure 1. Flow chart of the recognition system

The input to the system is a sequence of handwritten character patterns. After receiving input from mouse or touch pad or light pen the input is broken down into some segments. Then different types of membership for each segment are calculated. After that, fuzzy rules are applied on the memberships and the results of these operations are compared to a pre-defined set of results. Then we choose the result with the highest degree of resemblance to the pre-defined result.

3.1 Data acquisition

An HP Mini 110 has been used to take the samples from different subjects. Each subject was given the choice to write on the touch pad or to use a mouse. No restriction was imposed on the content or style of writing. The simulation of each written character could be seen on computer screen as black digital ink on white background.

3.2 Segmentation

Segmentation is the most important part of our system as this is the mechanism that enables us to extract different fuzzy features from a given input character despite varieties of writing styles.

First of all, the given input is broken down into some segments according to a legitimate algorithm. In short, we check a set of pixels in real time which is done by different threads of the program in order to reduce the processing time. Then we check the change of direction from the set of pixels we found and calculate the angle difference between two sets. If the angle is greater than 90 degree then we consider it to be a new segment. Another point worth mentioning is that if the pen is down, we initialize a new segment and when the pen is up, we also consider the drawn part to be a segment. The algorithm for segmentation is given below.

1. **If pen down start segment and save coordinates and set num_of_pixels = 0**
2. **If being dragged get the coordinates**
3. **If change in direction and num_of_pixels>=8 add coordinate index to angle_check queue**
4. **Save the coordinate and Add 1 to num_of_pixels**
5. **If 8 more pixels registered after an angle_check is triggered go to step 6, else go to step 2**
6. **If the angle is smaller than 90 degree then initialize a new segment and move last 8 coordinates from previous segment and set num_of_pixels = 8 Else Go to step 2**
7. **If pen up Save segment and go to step 1**

To follow our algorithm, we need every coordinates of mouse dragging event. However, a mouse handler never stores all the co-ordinate values, creating a gap between two points. It takes the values up to a certain threshold and truncates the rest. To overcome this complication, we have used Bresenham's Line Algorithm [17] to fill the gap found in a segment. An example of the segmentation of Bangla letter 'ka' is shown in Figure 2.

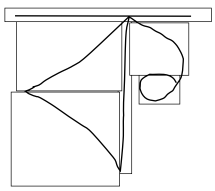

Figure 2. Segmentation of Character 'Ka'

3.3 Fuzzy Features

Ranawana et al [6, 11] derived various fuzzy features for English character recognition. In this research we also use the relevant fuzzy features from [6, 11] that are useful for Bangla character recognition and divide the features of the characters: universal features, regional features and contour features. Universal features are the features that are found in every character in the alphabet. These features are very important as we will need them for calculating both regional and contour features.

3.3.1 Universal Features

The features common to every character in the alphabet are the universal features. The universal features that we have used are stated in equations (1) to (4).

$$x_{\min} = \bigwedge_{i=1}^{N} x_i \bigcup_{j=1}^{P} seg(j) \quad (1) \qquad y_{\min} = \bigwedge_{i=1}^{N} y_i \bigcup_{j=1}^{P} seg(j) \quad (2)$$

$$x_{\max} = \bigvee_{i=1}^{N} x_i \bigcup_{j=1}^{P} seg(j) \quad (3) \qquad y_{\max} = \bigvee_{i=1}^{N} y_i \bigcup_{j=1}^{P} seg(j) \quad (4)$$

The Universal features of first segment of 'গ' are described in Figure 3.

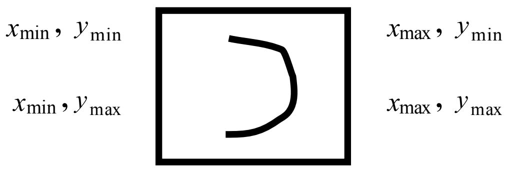

Figure 3. Global features of first segment of 'ga'

3.3.2 Regional Features

Regional feature determines the relative position of an identified segment with respect to the universe of discourse. When a segment has been identified, the next step is to determine the center of that segment. The co-ordinates of the center of a segment are calculated by the equations (5) and (6).

$$x_{center}^{seg(n)} = \frac{x_{\max}^{seg(n)} + x_{\min}^{seg(n)}}{2} \quad (5)$$

Universe of discourse is then divided into two linguistic variables Vertical Position (VP) and Horizontal Position (HP). The relative position of the identified segment is expressed by equations (7) and (8).

$$\mu_{HP} = \frac{x_{center}^{seg(n)} - x_{\min}^{seg(n)}}{x_{\max}^{seg(n)} - x_{\min}^{seg(n)}} \quad (7)$$

$$\mu_{VP} = \frac{y_{center}^{seg(n)} - y_{\min}^{seg(n)}}{y_{\max}^{seg(n)} - y_{\min}^{seg(n)}} \quad (8)$$

Then these membership values are compared with pre-defined linguistic terms for horizontal and vertical positions.

3.3.3 Contour Features

Contour features are extracted for each segment and divided into two main categories- Straight line and Arc. The fuzzy values involved with these feature are arcness and straightness [12]. Straightness ($\mu_{straightness}$) of a given segment is calculated by fitting a straight line with minimum 'Least Squares' error. The membership function for straightness of a segment is calculated by equation (9).

$$\mu_{straightness} = \frac{D_{P(O)P(N)}}{\sum_{K=1}^{N} D_{P(K)P(K+1)}} \quad (9)$$

Here, $D_{P(K)P(K+1)}$ is the straight line distance between point K and point $(K+1)$ on the n^{th} segment. The number of elements in the segment is depicted by N. If $\mu_{straightness}$ is greater than or equal to 0.7, then the segment is a straight line. Otherwise, it is an arc.

The ratio of the distance between two end points and total arc length determines arcness ($\mu_{arcness}$) of a particular segment. The membership function for arcness is expressed by equation (10).

$$\mu_{arcness} = 1 - \mu_{straightness} \quad (10)$$

If the given segment is determined as an arc, then we categorize it into one of the different types; horizontal line, Vertical line, Positive slant, Negative slant, A-like curve, U-like curve, C-like curve, D-like curve and O-like curve. Their membership functions are given through equations (11) to (19).

For vertical line:

$$\mu_{VL} = \vee(\Delta(\theta,90,90), \Delta(\theta,90,270)) \quad (11)$$

For horizontal line:

$$\mu_{HL} = \vee(\Delta(\theta,90,0), \Delta(\theta,90,180), \Delta(\theta,90,360)) \quad (12)$$

For positive slant:

$$\mu_{PS} = \vee(\Delta(\theta,90,45), \Delta(\theta,90,225)) \quad (13)$$

For negative slant:

$$\mu_{NS} = \vee(\Delta(\theta,90,135), \Delta(\theta,90,315)) \quad (14)$$

For A-like curve:

$$\mu_{AL} = \wedge(1, \sum_{i=0}^{n} \frac{ay_i}{n}) \quad (15)$$

Where $ay_i = 1$ if $y_i > (y_s + y_e)/2$; otherwise 0

For U-like curve:

$$\mu_{UL} = \wedge(1, \sum_{i=0}^{n} \frac{by_i}{n}) \quad (16)$$

Where $by_i = 1$ if $y_i < (y_s + y_e)/2$; otherwise 0

For C-like curve:

$$\mu_{CL} = \wedge(1, \sum_{i=0}^{n} \frac{lx_i}{n}) \quad (17)$$

Where $lx_i = 1$ if $x_i < (x_s + x_e)/2$; otherwise 0

For D-like curve:

$$\mu_{DL} = \wedge(1, \sum_{i=0}^{n} \frac{r_{x_i}}{n}) \quad (18)$$

Where $rx_i = 1$ if $x_i > (x_s + x_e)/2$; otherwise 0

For O-like curve:

$$\mu_{OL} = \frac{\sum_{K=1}^{N} D_{P(K)P(K+1)}}{2 \times 3.1416 \times r} \quad (19)$$

Here r is the radius of the curve, x_s, x_e are the start point and end point of a segment on the X-axis, y_s, y_e are the start point and end point of a segment on the Y-axis.

Table 1. Some of the geometric fuzzy feature [6, 11]

SHAPE	NAME	FUNCTION
	Vertical line	$\mu_{VL} = \vee(\Delta(\theta,90,90), \Delta(\theta,90,270))$
	Horizontal line	$\mu_{HL} = \vee(\Delta(\theta,90,0), \Delta(\theta,90,180), \Delta(\theta,90,360))$
	Positive slant	$\mu_{PS} = \vee(\Delta(\theta,90,45), \Delta(\theta,90,225))$
	Negative slant	$\mu_{NS} = \vee(\Delta(\theta,90,135), \Delta(\theta,90,315))$
	A-like curve	$\mu_{AL} = \wedge(1, \sum_{i=0}^{n} \frac{a_{y_i}}{n})$
	C-like curve	$\mu_{CL} = \wedge(1, \sum_{i=0}^{n} \frac{l_{x_i}}{n})$
	D-like curve	$\mu_{DL} = \wedge(1, \sum_{i=0}^{n} \frac{r_{x_i}}{n})$

After we are done calculating all the fuzzy features, we will map these values into linguistic terms according to pre-defined range. The range of values of membership function for evaluating angle difference and different linguistic terms are shown in Figure 4.

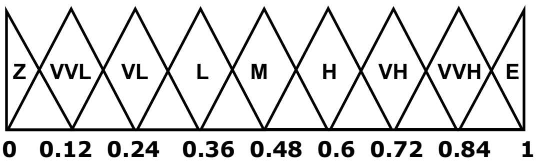

Figure 4. Linguistic variable mapping from fuzzy values

3.3.4 Creation of Fuzzy Rule Base

The rule base depends on the number of segments contained in a given character. Our work comprises of quite a few fuzzy rules, each one unique for a particular character. The general idea of our fuzzy rule base for each character is – the fuzzy features of the same segments of the input and stored data are combined with AND (MIN) operator and the fuzzy features of different segments of the input and stored data are combined with OR (MAX) operator.

3.3.5 Modes of Recognition

Our system operates in two modes- Learning mode and Recognition mode. These are described in the subsequent sections.In learning mode, user draws a character in the input palette and while drawing is being done, we segment the character and do calculations simultaneously with the help of multiple threads. We describe our approach step by step.

1. Segmentation is done by the algorithm explained in section 3.2.
2. Then fuzzy features are extracted by the equations described in section 3.3
3. We generate a table like Table 2 from the data found in step 2
4. In the end we save all the analyzed data in different database (.csv-comma separated value format) according to the segment numbers found from a sample.
5. We analyze the data to create fuzzy rule base and then save it in a database.

Table 2. Extracted fuzzy features

SAMPLE	μVL	μHL	μOL	μDL	μCL	μAL
1	*Z*	*VH*	*H*	*M*	*L*	*VL*
2	*M*	*H*	*L*	*M*	*M*	*VL*
3	*VVH*	*L*	*M*	*L*	*L*	*M*
4	*VVH*	*H*	*M*	*VVH*	*VH*	*L*
5	*L*	*H*	*VVH*	*VH*	*H*	*VH*
Total = 5	***VVH***	***H***	***M***	***M***	***L***	***VL***

In recognition mode, our system tries to recognise any given character. The character recognition approach of our system is given below in step by step.

1. Recognition mode segmentation is same as the learning mode segmentation.
2. Recognition mode feature extraction is same as the learning mode extraction.
3. Check the segmentation number and then open an existing database (comma separated value format) matching with the segmentation number found from the input.
4. In comparison stage, data extracted from the given input are compared with the data found from the database. Comparison is done in terms of linguistic variable. This mapping is depicted in Fig. 4.
5. From the comparison, we get the highest matching value and according to that we give conclusion character in Unicode format.

4 Experimental Results

In the experimentation, we have collected different samples for each character from different individuals. In Bangla, there are 50 characters. Therefore, the total numbers of samples are pretty big in number. We got very high recognition accuracy for the proposed system, which is more than 72%.

4.1 Segmentation Results

To find out attractive fuzzy features it is necessary to divide each character into a number of meaningful segments. In this paper, each character can be divided into maximum fifteen segments. But maximum accuracy is found in different specific number of segments for each different vowel sign. The segmentation result of Bangla character 'ba' is depicted in Figure 5.

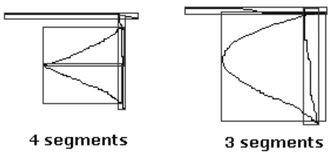

Figure 5. Different segments of 'ba'

4.2 Regional Features Extraction

We have extracted horizontal membership function (μHP) and vertical membership function (μVP) from the equations described in section 3.3. An example of the extracted regional feature tabular format is given in Table 3.

Table 3. Regional features

SAMPLE	SEGMENT NUMBER	μHL	μOL
	6	*VH*	*H*
	7	*H*	*L*
	3	*L*	*M*
	4	*H*	*M*

4.3 Contour Features

First we calculated Arcness ($\mu_{arcness}$) and Straightness($\mu_{straightness}$) from the equations described in section 3.3.3 and according to that, we extracted vertical line membership function(μVL), horizontal line membership function(μHL), positive slant membership function(μPS), negative slant membership function(μNS), A-like curve membership function(μAL), U-like curve membership function(μUL), C-like curve membership function(μCL), D-like curve membership function(μDL), O-like curve membership function(μOL).
An example of the extracted regional feature tabular format is given in Table 4.

Table 4. Contour features

Sample	no	μ VL	μ HL	μ PS	μ NS	μ AL	μ UL	μ CL	μDL	μOL
	7	*H*	*VVH*	*VH*	*H*	*M*	*L*	*L*	*Z*	*M*
	4	*M*	*L*	*H*	*VVH*	*M*	*Z*	*E*	*H*	*VVH*
	5	*VH*	*M*	*L*	*VL*	*H*	*M*	*L*	*Z*	*E*
	5	*L*	*M*	*H*	*M*	*VL*	*M*	*Z*	*E*	*L*

4.4 System Performance

The performance of the system is described by the curve of number of segments versus percentage of accuracy for each individual character, which are shown in Figure 6. The highest, lowest and average percentage value of accuracy for each character with respect to number of segments has been mentioned in the Figure 6.

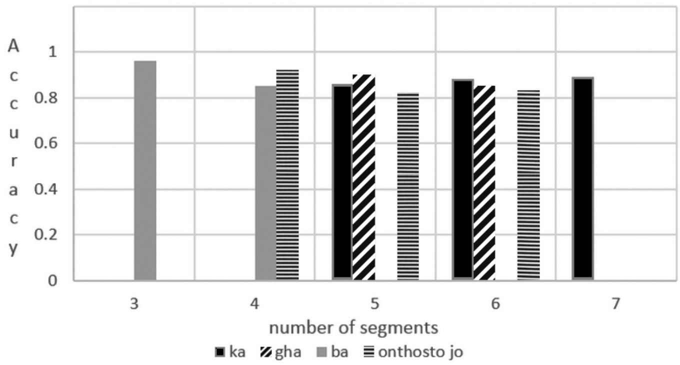

Figure 6. Performance analysis for different characters

Average system performance is shown in Figure 7. Here we are depicting overall system performance. It is a graph of accuracy over segment numbers.

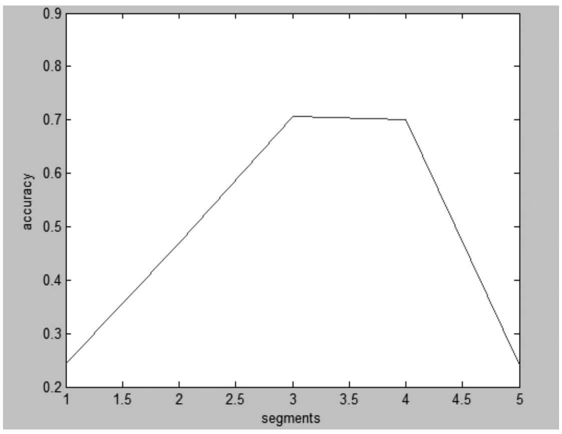

Figure 7. Average system performance

5 Conclusion

With time and tide, the word 'interactivity' is taking a whole new meaning. People are more interested to interact with machines in a more natural way. Online handwriting recognition is one of the most effective ways of natural communication. The main goal of our system is to develop a clear idea so that it can be possible to develop software capable of recognizing Bangla handwritten inputs regardless of the different writing styles of the user. In this work, we are interested to develop such system that will mainly focus on economical resource consumption which will enable us to implement it in handheld devices. This will facilitate easy communication for a very large population native to Bangla Language. The efficiency of this system can be further improved by implementing a dynamic learning algorithm that will integrate both the learning and recognizing modes.

References

[1] I. Guyon, "Application of Neural Networks to Character Recognition", *Int. J. Pattern Recognition. Artificial Intelligence,* **Vol.** 5*,* pp. 353-382, 1991

[2] G. Kim and S. Kim, "Feature Selection using Genetic Algorithm for Handwritten Character Recognition", *Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition*, September 11-13 2000, Amsterdam, Netherlands, pp. 103-112, 2000.

[3] K. W. Cheung and D. Y. Yeung, "Bidirectional Deformable Matching with Application to Handwritten Character Extraction", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol.** 24, No. 8, pp.1133-1139, 2002 .

[4] C. Bahlmann and H. Burkhardt, "Measuring HMM Similarity with the Bayes Probability of error and its Application to Online Handwriting Character Recognition", *6th International Conference on Document Analysis Recognition*, pp. 406-411, 2001

[5] A. Malaviya and L. Peters, "Extracting Meaningful Handwriting Features with Fuzzy Aggregation Method", *3rd. International Conference on Document Analysis and Recognition*, Montreal, Canada, pp.841-844,1995.

[6] R. Ranawana, V. Palade and G. E. M. D. C. Bandara, "Automatic Fuzzy Rule Base Generation for On-line Handwritten Alphanumeric Character Recognition", *International Journal of Knowledge-based and Intelligent Engineering Systems*, **Vol.** 9, No. 4, pp.327-339, 2005.

[7] "Optical character recognition," available online from: http://en.wikipedia.org/wiki/Optical_character_recognition [20/8/2010].

[8] M.F. Zafar, D. Mohammad, and R.M. Othman, "On-line Handwritten Character Recognition: An Implementation of Counterpropagation Neural Net", *International Journal of Computer, Information, Systems and Control Engineering,* **Vol.** 1, No.10, pp. 232-236, 2007.

[9] T. Long, L. Jin, L. Zhen, and J. Huang, "One Stroke Cursive Character Recognition Using Combination of Directional and Positional Features", *Thirtieth IEEE International Conference on Acoustics, Speech, and Signal Processing(ICASSP),* pp. 449-452, 2005.

[10] A.K. Jain, and D. Zongker, "Representation and Recognition of Handwritten Digits Using Deformable Templates", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **Vol.** 19, No. 12, pp.1386-1390, 1997.

[11] R. Ranawana, V. Palade and G. E. M. D. C. Bandara, "An Efficient Fuzzy Method for Handwritten Character Recognition", *Proceedings of KES Conference*, Amsterdam, Netherlands, pp.698-707, 2004.

[12] M. M. Islam, S. M. F. Rahman, and M. M. Hoque, "Online Bengali Handwritten Recognition with Automatically Generated Fuzzy Linguistics Rules", *8th International Conference on Computer and Information Technology*, Dhaka, Bangladesh, pp.673-678, 2005.

[13] A. Majumder, "Bangla Basic Character Recognition Using Digital Curvelet Transform", *Journal of Pattern Recognition Research,* **Vol.** 1, pp. 17-26, 2007.

[14] M.S. Monjel, and M. Khan, "Optical Character Recognition For Bangla Documents Using HMM". available online from: http://www.panl10n.net/english/outputs/Working%20Papers/Bangladesh/Microsoft%20Word%20-%2028_N_170.pdf [20/08/2010].

[15] M.S. Kamal, M.M. Hoque, M.M.U. Hasan, and M.S. Arefin, "Bangla Vowel Sign Recognition by Extracting the Fuzzy Features", *Eleventh International Conference on Computer and Information Technology(ICCIT)*, Khulna, Bangladesh, pp. 306-311, 2008.

[16] M.M. Hoque, M.R. Karim, M.G. Hossain, M.M.U. Hasan, and M.S. Arefin, "Bangla Numeral Recognition Engine (BNRE)", *Proceedings of Fifth International Conference on Electrical and Computer Engineering(ICECE)*, pp. 644-647, 2008.

[17] "Bresenham's Line Algorithm", available online from: *http://en.wikipedia.org/wiki/Bresenham%27s_line_algorithm*[20/10/ 2010].

[18] L. H. Tsoukalas and R. E. Uhrig, *Fuzzy and Neural Approaches in Engineering*, John Wiley, New Jersey, USA, pp.13-185, 1997.