A Fuzzy Features Based Online Handwritten Bangla Word Recognition Framework

Kanchan Chowdhury, Lamia Alam, Shyla Sarmin, Safayet Arefin, and Mohammed Moshiul Hoque
Dept. of Computer Science & Engineering, Chittagong University of Engineering & Technology,
Chittagong-4349, Bangladesh
E-mail: moshiulh@yahoo.com

Abstract—**Handwriting recognition is one of the most important ways to ease the handling of information between man and machine. Online handwriting recognition can be a very attractive method when people feel inconvenient using keyboards to handle information with computing devices. The most complicated task associated with online Bangla handwritten recognition is to separate the adjacent characters and vowel signs from one another within a Bangla word. This problem becomes more complicated due to the variations of writing style of individuals. In this paper, we propose a framework to recognize handwritten Bangla words in real time considering different writing styles. We used fuzzy linguistic rules in order to recognize Bangla handwritten words. Evaluation result for various writing styles reveals that the propose framework can recognize Bangla handwritten words with 77% accuracy.**

Keywords—Handwritten recognition; online recognition; fuzzy features; segmentation; aggregation.

I. INTRODUCTION

The ability of a computing system to receive and interpret handwritten input from several sources such as paper documents, photographs, touch screens and other devices is known as handwritten recognition. Handwriting recognition plays an important role in the storage and retrieval of crucial handwritten information that also reduces the storage costs. Significant progress has already been achieved towards recognition of online handwriting in other scripts. However, recognition of Bangla online handwritten is in rudimentary stage. Handwritten recognition process can be classified into two ways: offline and online [1]. In offline process, the text is not recognized at the same time as it is produced. Instead it is recognized after the user has finished writing. In this case, the text is originally written on a surface such as paper and from there on it is recognized by the computer by scanning the surface. On the other hand, online handwriting recognition is done in real time. The surface used for handwriting is usually a digitized tablet and it is used along with a stylus in order to write on the surface. As the pen moves across the surface, the two-dimensional co-ordinates of successive points are collected and stored as a function of time.

There are a number of methodologies for online handwritten recognition. Some of these include the use of neural networks [2, 3], fuzzy logic [4] and genetic algorithms [5]. In terms of computational usage, fuzzy logic is most efficient among these methods [6]. A number of approaches are available that can recognize online handwritten Bangla characters, numerals and vowel signs using fuzzy logic. But the recognition of online Bangla handwritten words is still remained unexplored. It is a quite challenging task to isolate adjacent characters and vowel signs in a Bangla word as they remain very close to each other. Moreover, there are a plenty variations in writing styles of individuals. In this paper, we propose a framework to recognize handwritten Bangla words in real time using a set of fuzzy linguistic rules.

II. RELATED WORK

A significant amount of research activities have been carried out to recognize Bangla handwritten character. Some methods are proposed to recognize the printed Indian scripts such as, Bangla, Tamil, Telugu, Oriya, and so on [7, 8]. Another approach which uses Markov models for recognizing handwriting [9]. A rule based approach for segmentation of lower modifiers in complex Bangla scripts is available in [10], which solves the problem of segmenting lower modifiers under a wide range of document images by the use of dissection based lower modifier segmentation method. An approach to character recognition using applications of neural network is described in [11]. Most of the previous systems are based on offline reorganization approach and is not suitable for real time applications.

A very few research activities have been conducted on online Bangla handwritten recognition. Bhattacharya et al. [12] propose an analytic recognition method with some preprocessing operations such as smoothing and re-sampling of points. Some works are available on online isolated Bangla character or numeral recognition in [13]. A technique for segmentation of Bangla word images is available in [14]. Another approach for recognition of online handwritten Bangla characters is reported in [15], which uses direction code based features for the accomplishment of the target. Some works on online handwriting recognition are reported in [4, 16]. They use fuzzy rule base, fuzzy features of handwritten characters and fuzzy aggregation method to accomplish their task.

A detailed approach based on fuzzy method to recognize alpha-numeric character has been described by Ranawana et. al. [6, 17]. These works mainly focus on recognizing alphanumeric characters of English language rather than words. In this paper, we propose a framework that is able to

978-1-4673-9930-2/15/$31.00 ©2015 IEEE

recognize online handwritten Bangla words considering various writing style.

III. Proposed Framework

The schematic representation of our proposed fuzzy feature based online handwritten Bangle word recognition system is illustrated in Fig. 1. The system consists of four main modules: segmentation, fuzzy feature extraction, learning phase and recognition phase.

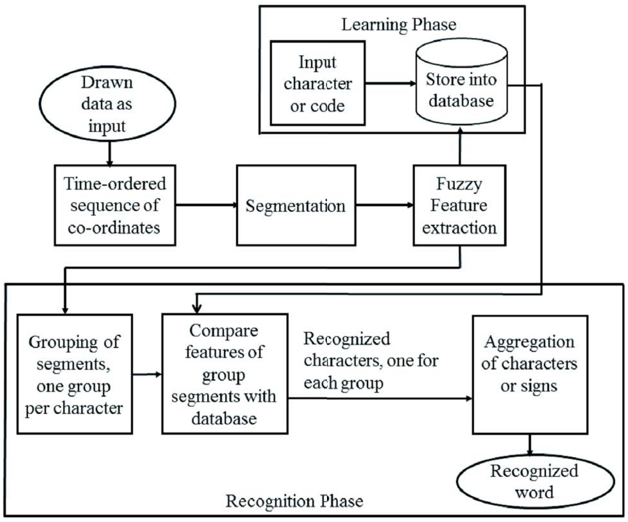


Fig. 1. Proposed framework for handwritten word recognition

The input word is drawn using a mouse on the screen or on a drawing pad. This data is collected as time ordered sequence of coordinates. The mouse-downs and mouse-ups at various points are also collected sequentially.

A. Segmentation

The segmentation process is based on following phase 1:

- Each mouse-up creates a new segment.
- If the change of direction of mouse movement is between 35 degrees and 145 degrees then a new segment is considered.

In the 2nd phase, we find the angle between first four points connected line and next four points connected line. The pseudo-code for the segmentation process is given below:

Step 1: Initialize variable no_segment=0.
Step 2: Initialize variable no_point=0.
Step 3: If mouse-down then
Step 3.1: Get the coordinate, store the coordinate and increment no_point by 1.
Step 3.2: If mouse-up then increment no_segment by 1 and go to step-2.

Step 3.3: Else if abrupt change in direction and no_point> 4 then increment no_segment by 1 and set no_point=0 and go to step-3.1.
Step 3.4: Else go to step-3.1.
Step 4: Else exit.

Fig. 2 shows segmented Bangla word "আমি"(I) as an example. It is seen that there is a mouse-up at the end of segment-1 and segment-2. So segment-2 and segment-3 starts according to step-3.2. There is an abrupt change in direction at the end of segment-3. So segment-4 starts according to step-3.3. For the other segments, there is mouse-up at the end of segment-4, segment-6, segment-8, segment-9, and segment-10 and there is abrupt change in direction at the end of segment-5, segment-7, segment-11, segment-12, and segment-13.

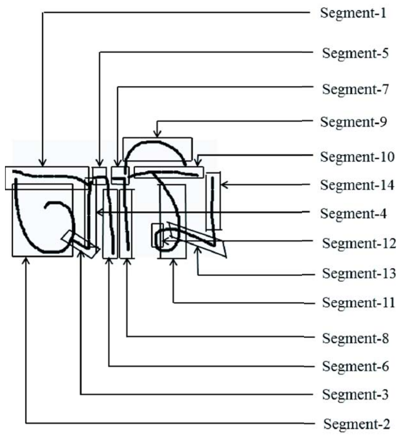


Fig. 2. Segmentation of Bangla word, "আমি" (I).

B. Fuzzy Feature Extraction

After dividing the input data into segments the next step is to determine the fuzzy features of each segment. This process consists of determination of universe of discourse, relative position and geometric feature of a segment.

1) Determination of Universe of Discourse

The universe of discourse of a character is the smallest rectangular area where the character fits fully. To calculate these features, we use the following equations:

$$minX = {}_{\mathrm{i}=1}^{\mathrm{N}}min(\, x_i)$$

$$minY = {}_{i=1}^{N}min(y_i)$$

$$maxX = {}_{i=1}^{N}\max(x_i)$$

$$maxY = {}_{i=1}^{N}\max\,(y_i)$$

Here *i* stands for the serial number of points starting from 1 to N where N denotes total number of points and x_i and y_i are x-coordinates and y-coordinates respectively of point *i*. *minX*, *minY, maxX, maxY* stands for minimum x-coordinate, minimum y-coordinate, maximum x-coordinate and maximum y-coordinate respectively [6, 17].The universe of discourse of Bangla character ”ক“ is shown in Fig. 3.

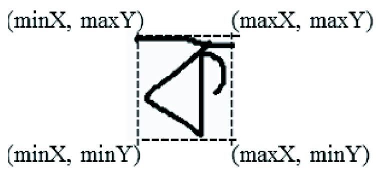


Fig. 3. Universe of discourse of Bangla character "ক".

2) Determination of Relative Position of a Segment

To determine the relative position of a given segment with respect to the universe determined earlier, the first step is to determine the maximum and minimum values of all coordinates of that segment i.e., $minX^{seg\ (n)}$, $minY^{seg(n)}$, $maxX^{seg(n)}$, $maxY^{seg(n)}$. After the center point of each segment are determined as follow:

$$centerX^{seg(n)} = \frac{(minX^{seg(n)} + maxX^{seg(n)})}{2}$$

$$centerY^{seg(n)} = \frac{(minY^{seg(n)} + maxY^{seg(n)})}{2}$$

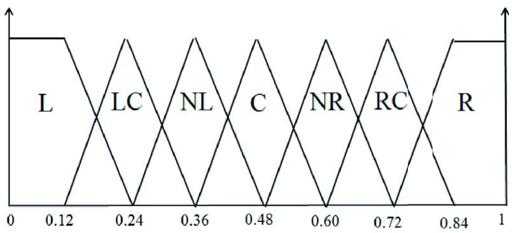
Later, the relative position of each segment is calculated as follows:

$$\mu_{HP}^{seg(n)} = \frac{centerX^{seg(n)} - minX^{seg(n)}}{maxX^{seg(n)} - minX^{seg(n)}}$$

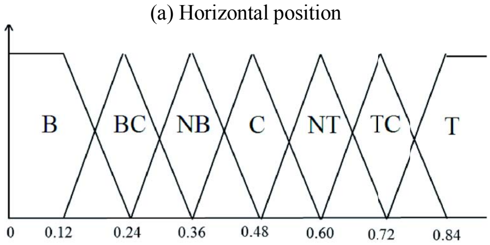
$$\mu_{VP}^{seg(n)} = \frac{centerY^{seg(n)} - minY^{seg(n)}}{maxY^{seg(n)} - minY^{seg(n)}}$$

Here the terms HP and VP stand for 'Horizontal Position' and 'Vertical Position' respectively.

Fig. 4 expresses the membership function for the relative position of a segment.



(a) Horizontal position



(b) Vertical position.

Fig. 4. Membership functions of a segment.

In Fig. 4(a), the linguistic variables L, LC, NL, C, NR, RC and R stand for 'left', 'left center', 'nearly left', 'center', 'nearly right', 'right center' and 'right' respectively. In Fig. 4(b) the variables B, BC, NB, C, NT, TC and T stand for 'bottom', 'bottom center', 'nearly bottom', 'center', 'nearly top', 'top center' and 'top' respectively.

3) Geometric Features of a segment

In order to determine the geometric shape of a segment, we have to determine its straightness and arc-ness. To find the straightness and arcness, we use the following equations [6].

$$\mu_{Straightness} = \frac{d_{P(0)P(N)}}{\sum_{k=0}^{N-1} d_{P(k)P(k+1)}}$$

$$\mu_{Arc-ness} = 1 - \mu_{Straightness}$$

Here, N stands for total number of points of a given segment and $d_{P(k)P(k+1)}$ stands for distance between point k and point *k+1* on that segment. If the straightness is greater than a threshold value (e.g. 0.6) the segment is considered as a straight line otherwise it is considered as an arc. If a segment is a straight line it may one of 'Horizontal line', 'Vertical line', 'Negative slanted line' or 'Positive slanted line'. To determine the category of the straight line we use the following formula described in [15]:

$$\Delta(\varphi; b, c) \begin{cases} 1 - 2 \cdot \left|\frac{\varphi - c}{2}\right|; (c - \frac{b}{2}) \leq \varphi \leq (c + \frac{b}{2}) \\ 0; otherwise \end{cases}$$

Here, φ is the angle that a segment forms with positive x-axis of O-x-y plane, b stands for bandwidth and is always 90º and c at the maximum membership value, which is unity. Table I describes types of straight lines according to the formula described above.

TABLEI TYPES OF STRAIGHT LINES

Name	Shape	Function
Horizontal line		μ_{HL}=max(Δ(φ;90°,0°), Δ(φ,90°,180°),Δ(φ;90°,360°))
Vertical line		μ_{VL}=max(Δ(φ;90°,90°), Δ(φ;90°,270°))
Positive slanted line		μ_{PS}=max(Δ(φ;90°,45°), Δ(φ;90°,225°))
Negative slanted line		μ_{NS}=max(Δ(φ;90°,135°), Δ(φ;90°,315))

The relative length of a line with respect to universe of discourse is another important feature. This is determined using three variables: horizontal length, vertical length and slant length. The following formulas can be used for calculating these values:

$$\mu_{HLEN} = \frac{d_{P(0)P(N)}}{WIDTH}$$

$$\mu_{VLEN} = \frac{d_{P(0)P(N)}}{HEIGHT}$$

$$\mu_{SLLEN} = \frac{d_{P(0)P(N)}}{SLANT\_LENGTH}$$

If a segment is an arc it can be one of the curve shown in Fig. 5.

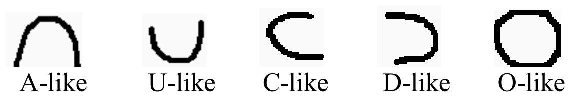


Fig. 5. Various types of curves.

These curve-nesses are calculated in the following way:

$$\mu_{AL} = min(1, \frac{\sum a_y}{n}); \quad \text{where,} a_y = \begin{cases} 1; y > \frac{(y_s + y_e)}{2} \\ 0; Otherwise \end{cases}$$

$$\mu_{UL} = min(1, \frac{\sum a_y}{n}); \quad \text{where,} a_y = \begin{cases} 1; y > \frac{(y_s+y_e)}{2} \\ 0; Otherwise \end{cases}$$

$$\mu_{CL} = min(1, \frac{\sum l_x}{n}); \quad \text{where,} l_x = \begin{cases} 1; x > \frac{(x_s+x_e)}{2} \\ 0; Otherwise \end{cases}$$

$$\mu_{DL} = min(1, \frac{\sum l_x}{n}); \quad \text{where,} l_x = \begin{cases} 1; x > \frac{(x_s+x_e)}{2} \\ 0; Otherwise \end{cases}$$

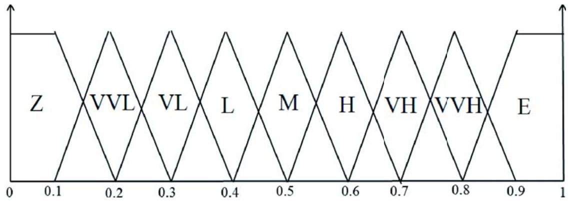
$$\mu_{OL} = \sum_{k=0}^{N-1} \frac{d_{P(k)P(k+1)}}{(2 \times 3.1416 \times r)}$$

Here, r is the radius of the curve and x_s, x_e, y_s, y_e are starting point and ending point of x-axis and y-axis respectively. Table II shows the four variations of O-like curves.

TABLEII VARIATIONS OF O-LIKE CURVE

Name	Shape	Function
O-like Top		μ_{OLT}=min(μ_{OL},μ_{UL})
O-like Bottom		μ_{OLB}=min(μ_{OL},μ_{AL})
O-like Left		μ_{OLL}=min(μ_{OL},μ_{DL})
O-like Right		μ_{OLR}=min(μ_{OL},μ_{CL})

The fuzzy values calculated for the geometric shapes are mapped to linguistic terms Z, VVL, VL, L, M, H, VH, VVH, E which stand for zero, very very low, very low, low,



medium, high, very high, very very high and excellent respectively. This mapping is shown in Fig. 6.

Fig. 6. Membership function for geometric features of a segment.

C. *Training Phase*

The entire system is managed using a database. The database consists of a table named 'character info'. This table has individual columns for character code, total number of segment, segment serial and each of the fuzzy features. In learning phase characters, numerals and vowel signs are learned. So user draws a character or a vowel sign in the drawing pad and also inputs the real character. After segmentation and feature extraction the linguistic terms of the fuzzy features are stored in the database. The number of rows is equal to total number of segments in the drawn data. The serial of a segment is stored in 'segment_serial' column and the fuzzy features of a segment are stored against the serial number of that segment. The character code and total number of segments are also stored.

D. *Recognition Phase*

In recognition phase, the user draws a word which is at first segmented and later fuzzy features are extracted. After that, for the recognition of the drawn word three more steps need to be accomplished which are described below:

1) *Grouping of Segments*

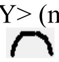
In this stage, all the segments of input data are categorized into a number of groups. All the segments of a single character belong to a single segment group and total number of segment group is equal to total number of characters in the input data. The pseudo-code for segment grouping process is given here:

Step 1: Initialize no_group=0, seg_no=0, startX and maxX with initial x-coordinate, startY and maxY with initial y-coordinate.

Step 2: If seg_no<= total number of segment then

Step 2.1: If mouse-down at beginning of this segment then

Step 2.1.1: Initialize holdX and holdY with x- and y-coordinate of starting point of this segment.

Step 2.1.2: If holdY> (maxY-3) &holdY> (startY+12) & previous character !=  and previous character != then

Step 2.1.2.1: If no_group> 0 then set end_seg =seg_no - 1.

*Step 2.1.2.2:*Incrementno_group by 1 and set start_seg = seg_no.

Step 2.1.2.3: Update maxX and maxY with maximumx- and y-coordinate of this segment respectively, increment no_seg by 1, go to step 2.

Step 2.1.3: Else if (holdX> (maxX-8) &holdY> = (startY – 10) &holdY<= (startY+10)) or holdX>= (maxX-3) then go to step 2.1.2.1.

Step 2.1.4: Else if previous character =  or  or  then go to step 2.1.2.1.

Step 2.1.5: Else go to step 2.1.2.3.

Step 2.2: Else go to step 2.1.2.3.

Step 3: Else set end_seg = seg_no – 1, exit.

Here *start_seg* is the starting segment and *end_seg* is the ending segment of a group and previous character indicates the character or part of character which is recognized using the previous segments of that group. In Fig. 2 there is a mouse-down at the beginning of segment-1 and holdX > (maxX-3). So segment-1 is the starting segment of group-1. Segment-5 and segment-7 and segment-9 also fall in step-2.1.3. So group-1 ends at segment-4, group-2 ends at segment-6 and group-3 ends at segment-8. When segment-10 is reached it is seen that the previous character is . So group-4 ends at segment-9 and group-5 starts at segment-10. Since segment-14 is the last segment according to step-2, group-5 ends at segment-14. So there is a total of 5 segment-group. Group-1consists of segments 1 to 4, group-2 consists of segments 5 to 6, group-3 consists of segments 7 to 8, group-4 consists of segment 9 and group-5 consists of segments 10 to 14.

2) *Compare Features of Group Segments with Database*

In this step we search for a character for each group found in the previous step. Part of a character may also be found from a group. For a group, we search such characters in the database where total number of segments in the character is equal to total number of segments in the group. If multiple such characters exist we choose the one where segments match highest with the segments of that group. Fig. 7 shows the groups of word "আমি".

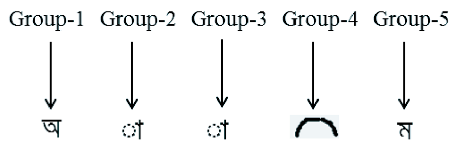


Fig. 7. Comparing result of handwritten Bangla word “আমি” (I).

3) *Aggregation of Recognized Characters*

In this step, we aggregate the characters one after another to form the desired word. After recognizing all characters they are aggregated sequentially and the desired word is formed. The rules of aggregation are:

- Rule 1: Characters are aggregated sequentially.
- Rule 2: Vowel signs ে, ৈ, িare aggregated after the next character of the sign and all other vowel signs are aggregated after the previous character of the sign.
- Rule 3: If েis found before a character and াor is found after the same character then both vowel signs are replaced with ো or ৌ respectively and the replaced sign is aggregated after that character.
- Rule 4: Some parts of characters are aggregated to form the whole character. Sequential াand forms ি, and া forms প, and া forms গ, and া forms ণ, and া forms শ, ব and forms ঋ, and forms ঋ, এ and forms ঞ, এand forms ঐ, and forms ং, two consecutive forms ঃ.
- Rule 5: If is found after ব, ড, ঢ or য then they are replaced with র, ড়, ঢ় or য় respectively.

In the example of Fig. 8, অ and া are aggregated to form আ. Then আ and া can’t be aggregated. So া and are aggregated and ি is formed. Since িis aggregated with its next character so ি will be aggregated after ম and মি will be formed. After aggregating আ and মি the desired word আমি is formed.

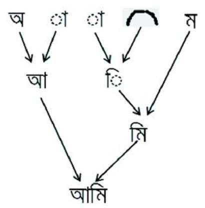


Fig. 8. Aggregation of Bangla word “আমি”(I).

IV. Experimental Results

The proposed system is built to detect handwritten Bangla words. We have tested it with 500 words of varying length and writing styles. Sample words were collected from a total of 10 participants. The average age of the participants were 25 years (SD=3.75). Fig. 9 shows some of the snapshot of our system while recognizing Bangla words such as “আমি” (I), “টাকা”(Money), and “দিন” (Day).

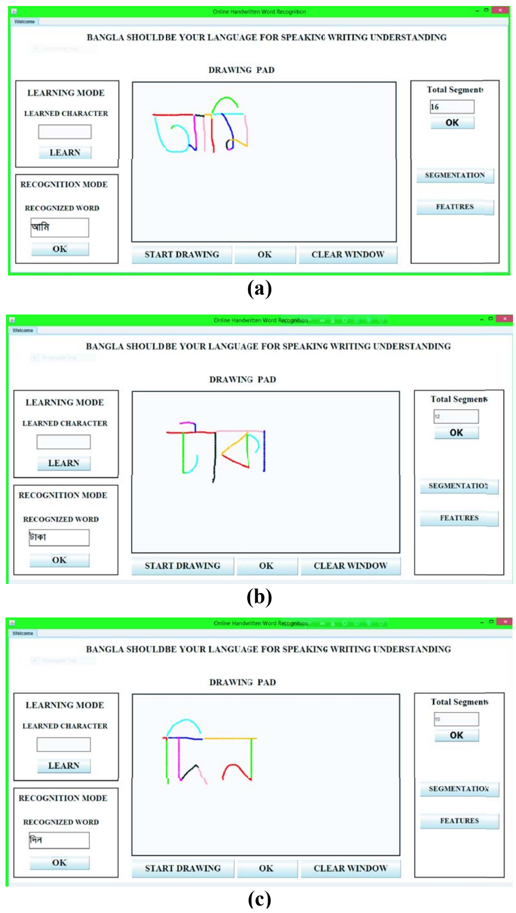


Fig. 9. Recognition snapshot of the proposed system for Bangla words (a) “আমি” (I) (b) “টাকা”(Money) and (c) “দিন”(Day) respectively.

A. *Measures*

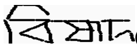
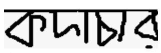
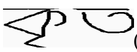
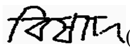
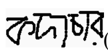
To evaluate the proposed framework, we consider the two aspects: (i) varying word length i.e. total number of characters or vowel signs in a word, and (ii) different writing styles of participant.

Table III shows few word samples that were collected from two different participants with corresponding recognition accuracy. Recognition rate refers to the ratio between the summation of recognition rates for all characters of a sample word and total number of characters in that sample word.

B. *Recognition rate*

We have calculated recognition rate for different word lengths such as word lengths 2, 3, 4, 5 and 6 respectively. Fig. 10 represents the recognition rate versus word length of the proposed framework. Fig. 10 also shows that recognition rate decreases with increasing word length. Fig. 11 shows the resultant graph of recognition rate for different writing style (i.e., participant). The result reveals that the recognition accuracy does not vary so much depending on the writing style of different participants. We have collected a total of 500 sample handwritten words of 10 participants. Among them, the proposed framework can recognize 386 words accurately. Therefore the overall accuracy is 77%.

TABLE III SOME SAMPLES COLLETED FROM PARTICIPANTS

User No.	Sample words	Word length	Recognition rate (%)	Average recognition rate (%)
1	 (মা)	2	88	77.8
	 (কৃত)	3	80	
	 (বিষাদ)	5	73	
	 (কদাচার)	6	70.5	
2	 (মা)	2	82	72.5
	 (কৃত)	3	76.5	
	 (বিষাদ)	5	68	
	 (কদাচার)	6	64.5	

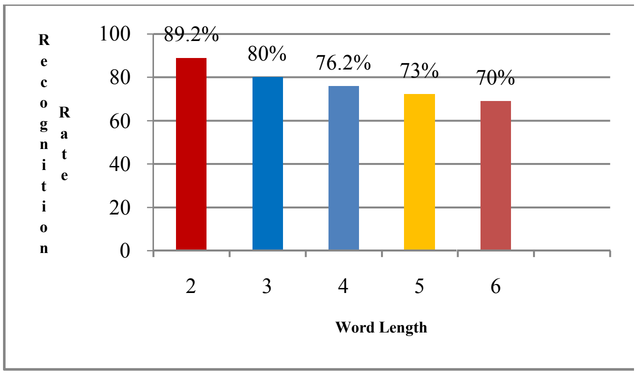


Fig. 10. Recognition rate vs. word length.

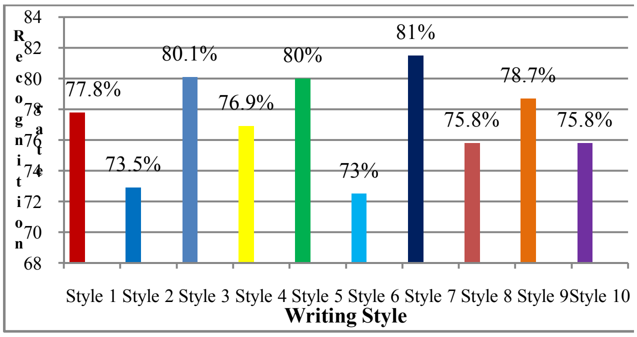


Fig. 11. Recognition rate vs. writing style.

V. CONCLUSION

The main focus of our work is to develop a framework that can recognize online handwritten Bangla words so that people can more easily interact with machines or digital devices. Although the handwriting varies from individuals our propose module can perform with fewer variations for various writing styles. Moreover, the recognition performance depends on the word length and writing styles and the evaluation result shows that the proposed framework is functioning quite satisfactory. Improving the segmentation process for detecting and eliminating unnecessary points and incorporating more fuzzy features to get more classified and more precise information about the segments that will be improves the overall accuracy of the framework. These are left as future issues.

REFERENCES

[1] A. Bhushan, P. Kalyani and J. Supriya, "Handwritten Script Recognition," *IOSR Journal of Computer Engineering (IOSR-JCE),* 2012, vol. 2, pp.30-33.

[2] A.L. Koerich and Y. Leydier, "A Hybrid Large Vocabulary Word Recognition System using Neural Networks with Hidden Markov Models," in *Proc. IWFHR*, 2002, pp. 99-104.

[3] S. W. Lee, Y. J. Kim, "A new type of recurrent neural network for handwritten character recognition," in *Proc. 3rd Int. Conference on Document Analysis and Recognition*, 1995, vol. 1, pp. 38-41.

[4] A. Malaviya and L. Peters, "Extracting meaningful handwriting features with fuzzy aggregation method", in *Proc. ICDAR*, 1995, pp. 841-844.

[5] G. Kim and S. Kim, "Feature Selection Using Genetic Algorithms for Handwritten Character Recognition, " in *Proc. 17th Int. Workshop on Frontiers in Handwriting Recognition*, 2000, pp. 103-112.

[6] R. Ranawana, V. Palade and G. Bandara, "An Efficient Fuzzy Method for Handwritten Character Recognition," *Knowledge-Based Intel. Info. and Eng. Syst.*, 2004, vol. 3214, pp. 698-707.

[7] R. Jayadevan, S. R. Kolhe, P. M. Patil, and U. Pal, "Offline Recognition of Devanagari Script: A Survey," *IEEE Trans. Systems, Man, and Cybernetics*, 2011, vol. 41, no. 6, pp. 782-796.

[8] A. Hasnat and M. Khan, "Elimination of splitting errors inprinted Bangla scripts," in *Proc.Conference on Language and Technology*, 2009, pp. 102-107.

[9] T. Plotz and G. A. Fink, "Markov Models for Offline Handwriting Recognition: A Survey," *International Journal on Document Analysis and Recognition*, 2009, vol. 12, no. 4, pp. 269–298.

[10] A. Hasnat and M. Khan, "Rule based segmentation of lower modifiers in complex Bangla scripts," *in Proc. Conference on Language and Technology,* 2009, pp. 94-101.

[11] G. Rasul and C. M. Rahman, "Genetics Algorithm Based Model for Bengali Character Recognition," in *Proc. ICCIT*, Dhaka, 1998.

[12] U. Bhattacharya, A. Nigam, Y. S. Rawat and S. K. Parui, "An analytic scheme for online handwritten Bangla cursive word recognition," in *Proc. International Conference on Frontiers in Handwriting Recognition*, 2008, pp. 320-325.

[13] K. Roy and N. Sharma, T. Pal, and U. Pal, "Online Bangla Handwriting Recognition System," in *Proc. International Conference on Advances in Pattern Recognition*, 2007, pp. 117-122.

[14] S. Basu, R. Sarkar, N. Das, M. Kundu, M. Nasipuri, and D. K. Basu, "A fuzzy technique for segmentation of handwritten Bangla word images," *in Proc. International Conference on Computing: Theory and Applications*, 2007, pp. 427–433.

[15] U. Bhattacharya, B. K. Gupta, and S. K. Parui, "Direction Code Based Features for Recognition of Online Handwritten Characters of Bangla," in *Proc. ICDAR*, 2007, vol. 1, pp. 58-62.

[16] A. Malaviya, H. Surmann and L. Peters, "Automatic Generation of a Fuzzy Rule Base for Online Handwriting Recognition," in *EUFIT*, 1994, pp. 1060-1065,.

R. M. Ranawana, V. Palade and G. Bandara, "Automatic fuzzy rule base generation for on-line handwritten alphanumeric character recognition," *Know. based Int. Eng. Sys.* vol. 9, pp. 327-339, 2005.