

An Autonomous Learning System of Bengali Characters Using Web-Based Intelligent Handwriting Recognition

Nazma Khatun¹ & Jouji Miwa¹

¹ Department of Electrical Engineering and Computer Science, Graduate School of Engineering, Iwate University, Japan

Correspondence: Nazma Khatun, Department of Electrical Engineering and Computer Science, Graduate School of Engineering, Iwate University, Ueda 4-3-5, Morioka City, Iwate Prefecture, Japan. Tel: 81-19-621-6974. E-mail: nazmababu@gmail.com

Received: March 13, 2016

Accepted: April 24, 2016

Online Published: May 10, 2016

doi:10.5539/jel.v5n3p122

URL: <http://dx.doi.org/10.5539/jel.v5n3p122>

Abstract

This research project was aimed to develop an intelligent Bengali handwriting education system to improve the literacy level in Bangladesh. Due to the socio-economical limitation, all of the population does not have the chance to go to school. Here, we developed a prototype of web-based (iPhone/smartphone or computer browser) intelligent handwriting education system for autonomous learning of Bengali characters that allows students to do practice their handwriting at anywhere at any time. As an intelligent tutor, the system can automatically check the handwriting errors, such as stroke production errors, stroke sequence errors, stroke relationship errors and immediately provide colourful error feedback to the students to correct themselves. Bengali is a multi-stroke input characters with extremely long cursive shape where it has stroke order variability and stroke direction variability. Due to this structural limitation, recognition speed is a crucial issue to apply traditional online handwriting recognition algorithm. In this work, we have adopted hierarchical recognition approach to improve the recognition speed that makes our system adaptable for web-based language learning. We applied writing speed free recognition methodology together with hierarchical recognition algorithm. It ensured the learning of all aged population, especially for children and older people. Finally, we conducted a survey in Bangladesh for the performance analysis of our proposed education system. The experimental results showed that our autonomous learning methodology helped to improve the average recognition accuracy by 4.1% (from 87.2% to 91.4%) with average Mean-Opinion-Score 4.1. It confirmed that the successful use of web-based Bengali handwriting education system can be very helpful to improve the literacy level in Bangladesh within a very short period.

Keywords: autonomous learning, web-based intelligent handwriting education, hierarchical recognition algorithm, automatic stroke error detection, web-based client-server interface

1. Introduction

1.1 Motivation and Our Contribution

Literacy in Bangladesh is a key for socio-economic progress, and the Bengali literacy rate grew to 61.5% in 2015 from 5.6% at the end of British rule in 1947. Despite government programs, the literacy rate was improved very sluggishly, only about 10 times within 60 years (Bangladesh Literacy Survey Report by UNESCO, 2015). Because of socio-economical limitation, all of the population, especially children and older people, do not have the chance to go to school. Although government has various educational activities, the number of school was not adequate yet. Considering this educational background, traditional handwriting teaching system is not enough to improve the Bengali literacy rate at 100%. Because, in the traditional handwriting teaching system, the teacher must write a Bengali character on the blackboard and the students should rewrite the handwritten character on their copy notebooks. After that, the teacher tries to check the handwriting errors in the student's notebooks and provides a feedback in the next time, because it's impossible for a teacher to verify and check every student's handwriting in the limited time of the lesson. This system can be successfully acquired only though practice regularly and for long periods. In this context, Hu et al. (2009) defined three drawbacks of the traditional education, such as time-consumption, faultiness, teacher-oriented. In addition, these techniques have many more drawbacks in aspects of socio-economical view point. It motivated us to develop web-based

intelligent handwriting education system for autonomous learning of Bengali characters. The learning process becomes much more effective, if the handwritten character is checked just after the students have finished their handwriting. On the other hand, the students can learn without the teacher supervision and they can correct the committed errors. Also, the students can repeat the same exercise several times to speed up the learning process. In this research project, we are aiming to develop a web-based (iPhone/smartphone or computer browser) intelligent handwriting education system that can ensure the learning of Bengali characters at anywhere at any time for those population, especially children or older people, who do not have chance to go to school. Thus, 100% literacy improvement can be established within a very short period. To the best of our knowledge, this is a pioneering attempt for the development of web-based intelligent handwriting education system to improve Bengali literacy rate.

Bengali is a multi-stroke input characters with extremely long cursive shaped where it has stroke order variability and stroke direction variability. The difficulty in online recognition of handwritten Bengali characters arises from the facts that this is a moderately large symbol set, shapes are extremely cursive even when written separately. In addition, there exist quite a few groups of almost similar shape characters in their handwritten format. Fundamentally, multi-stroke recognition algorithm results very slow recognition speed in case of long cursive characters. For the structural limitation of Bengali characters, existing multi-stroke recognition algorithm is not applicable for the development of Bengali handwriting education system, because it needs to provide the real-time students feedback. To address this problem, we have developed hierarchical online recognition algorithm to improve the recognition speed with considerably higher accuracy. It makes our system adaptable for web-based language learning and ensured immediate feedback about student's handwriting errors. In this hierarchical online recognition algorithm, we applied a series of matching filters to reduce a small number of candidates characters for final Dynamic Programming Matching (DPM) where local features (angular feature) are used to guide DPM. Then, the character with lowest matching cost is selected as recognition results. Finally, it returns the recognition results. Using the structural information stored in a predefined structural dictionary, our algorithm can identify the handwriting errors automatically and feedback to students together with recognition results. Here, we have modified the traditional DPM algorithm that allows writing speed free variability and improve the recognition accuracy.

1.2 Related Background

In recent years, several research efforts have been done on e-learning system (Hiekata et al., 2007; Tang et al., 2006; Zein et al., 2007; Abdou et al., 2010; Ahmad et al., 2010) which aims to guide students to get more useful advice in their autonomous learning. They had developed an intelligence tutoring learning method to provide autonomous learning environment to the students. With the development of pen-based devices, it is now possible to apply e-learning techniques to handwriting education. Several handwriting education systems have been provided for different languages such as: Chinese, Latin and Arabic. It can be organized on three categories: read only systems, guided ones and systems with automatic errors detection. In case of Chinese handwriting education systems, the work proposed in (Tang et al., 2006) can find both the stroke production error and stroke sequence error but they did not consider the spatial relationship errors. In recent years, some researches on intelligent robot tutoring system has also been done where a robot teacher was used for autonomous learning (Deanna et al., 2015).

To develop a web-based handwriting education system for learning of handwritten Bengali characters, we need to develop an online recognition algorithm for cursive Bengali characters. Extensive research on cursive handwriting recognition has been done during the last few decades for different languages. However, there has not been much work on handwriting recognition of Indian scripts. Particularly, there have very few attempts for the recognition of online Bengali handwritten characters (Bhattacharya et al., 2008; Parui et al., 2008). But both of these two approaches are not applicable for the development of web-based handwriting education system, because of slow recognition speed. In our proposed education system, we have developed efficient hierarchical online recognition algorithm to speed up our system. Here, the student can practice their writing on the digital tablet accessed from both of iPhone/smartphone or computer browser. Then, our recognition engine can analysis the student's handwriting input and checks the handwriting errors to provide useful feedbacks.

2. Bengali Handwriting Education System

2.1 Handwritten Bengali Character Set

Bengali is official language/script of Bangladesh and used by 211 million people of India and Bangladesh. It is also second most popular language/script in India and 5th most popular language in the world. Bengali, like other major Indian characters, is a mixture of syllabic and alphabetic scripts. It came from the ancient Indian script,

Brahmi. The concept of upper/lower case is absent here and the direction of writing policy is left to right. Examples of Bengali characters are shown in Table 1. Bengali language consists of 50 basic characters including 11 vowels and 39 consonants. In Bengali basic characters (chad) is a nasalization marker that appears over the top of an independent vowel or consonant. In our experiment, we considered 49 basic Bengali characters except (chad). Most of the characters in Bengali language have a horizontal line at the upper part. We call this line as head-line or matra. Vowels have their modified shapes called Vowel Modifiers (VM). In Bengali script a vowel following a consonant takes a modified shape. Depending on the vowel, its modified shape is placed at the left, right (or both) or bottom of the consonant. These modified shapes are called modified or syllabic characters. In Bengali, there have 10 vowel modifiers which are joined with 35 of consonants and make 350 modified syllabic Bengali characters. On the other hand, several consonants or a vowel in conjunction with a consonant form a large number of possible different shapes, called compound characters. However, in the present day Bengali text, the occurrence of compound characters is less than 5% and the rest is only basic characters and vowel modifiers. So, our proposed autonomous learning system is focused on the learning and recognition of Bengali basic characters.

Table 1. Different shape of 50 Bengali basic characters where 49 of them are used in our recognition experiment

| Type | Characters | Number of Characters |
|------------|---|----------------------|
| Vowels | অ(a) আ(aa) ই(i) ঐ(ii) উ(u) ঊ(uu) ঋ(ri) এ(e) ঐ(ai) ও(o) ঔ(au) | 11 |
| Consonants | ক(ka) খ(kha) গ(ga) ঘ(gha) ঙ(nga) চ(ca) ছ(cha) জ(ja) ঝ(jha) ঞ(nya) | 50* |
| | ট(tta) ঠ(ttha) ড(da) ঢ(dha) ণ(na) ত(ta) থ(tha) দ(da) ধ(dha) ন(na) প(pa) | 39* |
| | ফ(pha) ব(ba) ভ(bha) ম(ma) য(ya) র(ra) ল(la) শ sha) ষ(ssa) স(sa) হ(ha) | 38 |
| | ড়(rra) ঳(rha) ঙ্গ(yya) (khandata) ঙ্গ(visarga) ঙ্গ(anus -vara) (chad)* | |

2.2 Learning System Architecture

Figure 1 represents the operational flow of our proposed Bengali handwriting education system for autonomous learning. The proposed system is composed of two modules: the guided writing mode and the free writing one. The architecture of the system is detailed in Figure 1. As shown in Figure 1, students have the choice to practice the guided handwriting mode or the free one. In case of free handwriting mode, writing will be done on a blank area, Figure 2(b). The guided writing mode is one of the beginning level of education designed for the children's who are in the early stages of learning. This tool displays a transparent image onto the digital web interface comprising this handwriting template, Figure 2(a). The student is then invited to follow this image to replicate the pattern of Bengali character.

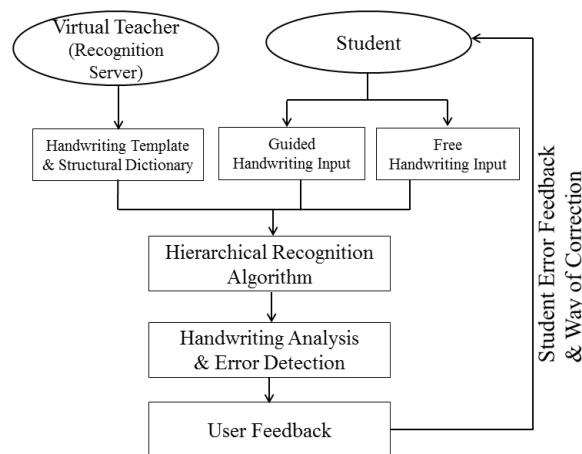


Figure 1. The architecture of the proposed Bengali handwriting education system

After student submits their sample character, handwriting input was received in our recognition server (saying as virtual teacher) through WWW client-server interface. Then, by matching the handwriting template and the handwriting input, the recognition of the inputted character will be carried out. Finally, the automatic stroke error detection engine can immediately locate the student's handwriting errors and provide an immediate feedback to the student about the location of the error; their type and how to correct them (Table 5 and Table 6). The details of the automatic stroke error detection and hierarchical recognition methodology will be described in the following sections.

We have developed a digital web interface that can access from both of iPhone/smartphone or computer browser. Figure 2 shows the snapshot of our web-based digital interface. It has three fields: (1) Handwriting character input field, (2) Recognized character output field, and (3) Options buttons field. While the users write Bengali characters with an input device (e.g., pen, mouse, finger, etc.) on the character input field. Then, our digital web interface gets the corresponding stroke data (sequence of points) and sent to recognition server. Those data stored at the database, later we used it to evaluate our proposed education system. Once the interface gets the recognition result from the server, result is displayed at the character output field with system font including student's error feedback.

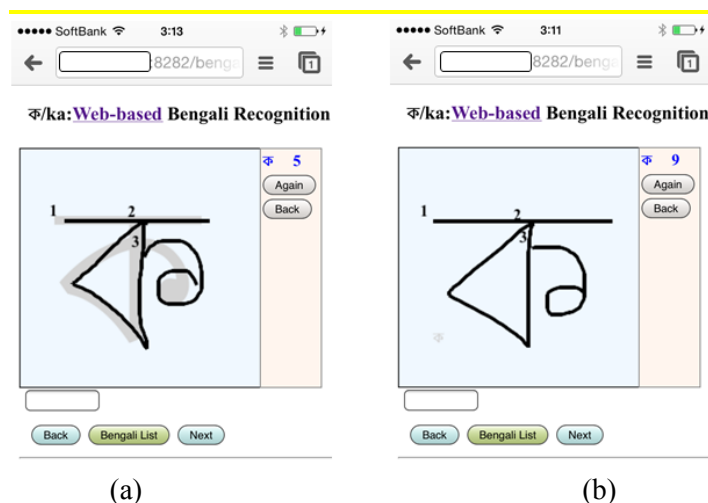


Figure 2. Digital web interface for Bengali handwriting education system that can be accessed from both of iPhone/smartphone or computer browser: (a) Guided handwriting mode (b) Free handwriting mode

2.3 Character Recognition Methodology

2.3.1 Web-Based (Client-Server) Recognition Architecture

In our bengali handwriting education system, a web-based handwriting client-server interface technique has been used for character recognition and student's feedback. We have designed the proposed system with the following distinctive features: (1) it is a web-based system developed by Java web application technology and works on WWW client browser (iPhone/smartphone or computer browser), (2) easy character input environment is provided according to use of rich editing functions and the input device (e.g., pen, mouse, finger etc.), (3) HTML5 canvas technology was used to detect and draw student input, (4) Apache Tomcat web server and PostgreSQL database were used for system implementation, (5) consecutive handwriting and recognition is possible, and (6) immediate student feedback.

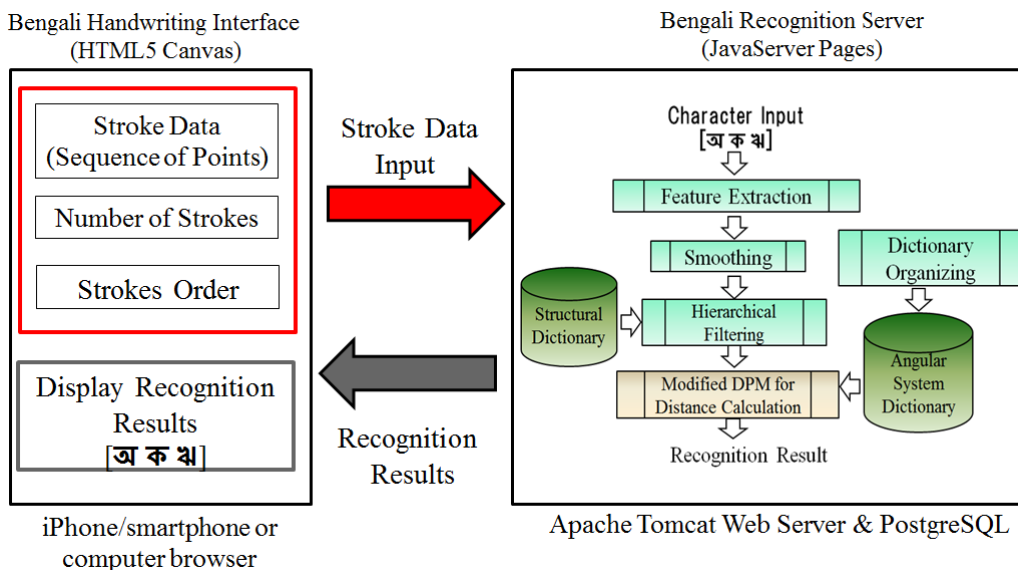


Figure 3. Handwriting recognition architecture for web-based Bengali handwritten education system

Figure 3 shows the handwriting recognition architecture that contains both of web-based handwriting interface and character recognition servers. Handwriting interface program is generated by Java Server Pages (JSP) and runs on WWW client, such as computer browser or iPhone/smartphone browser. On the other hand, JSP based character recognition engine works on Apache Tomcat web server. While the students write Bengali characters with an input device (e.g., pen, mouse, finger, etc.) on the character input field. Then, our digital web interface gets the corresponding stroke data as (x, y) coordinates and sent to the character recognition server. After that, the recognition engine converts the student’s stroke data into angular feature in feature extraction stage, see in right side of Figure 3. After applying smoothing to those extracted angular feature, our algorithm entered into hierarchical filtering stage. In this stage, we have applied a series of filters in a hierarchical manner to reduce the search space of final DPM. The first filter performs coarse classification on a large number of candidates based on the high level features of stroke patterns, such as stroke number. It reduces the candidate character models. Then the second filter performs structural preselection among the resulted samples of Filter 1, based on the structural information of Bengali characters stored in our predefined structural dictionary (Table 4). Again, it reduces to a small number of candidates for final DPM. In the final matching stage, low-level features (angular feature) are used to guide a DPM algorithm. In this stage, it calculates distance between input strokes and template strokes of each preselected characters by using our modified DPM. The character with optimal distance is selected as recognition result. After that, our recognition engine returns k top ranked characters as recognition results to client side browser. Then, it displayed the results into recognized character output field of our digital web interface.

2.3.2 The Flow of Hierarchical Recognition System

This research project was aiming to develop a web-based (iPhone/smartphone or computer browser) Bengali education system. We need an efficient recognition algorithm that gives higher accuracy with improved recognition speed. To speed up our recognition system, a series of filters have been applied in a hierarchical manner before applying into the final matching algorithm. It reduces the recognition search space and speed up our Bengali education system. Figure 4 shows the hierarchical recognition flow of our proposed system. The first filter performs coarse classification on a large number of candidates based on the high level features of stroke patterns, e.g., number of strokes, to reduce candidate character models. Then the second filter performs structural preselection based on the structural information of Bengali characters stored in our predefined structural dictionary, we named it as Bengali Structural Dictionary. It reduced to a small number of candidates for final Dynamic Programming Matching (DPM).

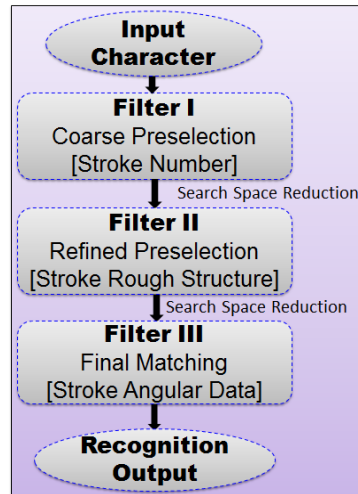


Figure 4. Operational flow of hierarchical recognition architecture for web-based Bengali handwriting education system

In the final matching, low-level features (angular feature) are used to guide DPM classification. The resulting k top ranked candidates are sent as recognition results. In this way, the proposed hierarchy can largely save the DPM calculation time, thus recognition speed and accuracy has improved that makes our system adaptable for real-time web-based Bengali education system.

Filter I: Coarse Preselection Based on Stroke Numbers

Bengali is a multi-stroke input characters with extremely long cursive shaped where it has stroke order variability and stroke direction variability. Bengali character can be categorized based on its stroke numbers (high-level feature) as shown in Table 2. Coarse preselection using this high-level feature (stroke numbers) among the large number of candidate characters, can reduce the recognition search space into five different levels, from 50 to 7, 20, 19, 3 and 1. In our hierarchical recognition architecture, Filter I performed coarse classification on a large number of candidates based on the stroke numbers and thus improved the recognition speed.

Table 2. Different categories of Bengali characters based on its stroke numbers

| Level | Number of Strokes | Bengali Characters | Number of Characters |
|-------|-------------------|--|----------------------|
| 1 | 1 | এ ও খ থ গ ঙ | 7 |
| 2 | 2 | খা ঙ ষ ম ঝ ঠ চ ত দ ফ ব ড ভ ল হ ঙ ঙ | 20 |
| 3 | 3 | অ ই ঙ উ ঐ ঐ ক ছ জ ঞ ট ত র শ ণ ন ষ য় স ড | 19 |
| 4 | 4 | উ ধ প | 3 |
| 5 | 5 | আ | 1 |

Filter II: Refined Preselection Based on Bengali Character’s Structural Patterns

Filter II is applied on the resulted candidates of Filter I for the further reduction of recognition search space that helps to speed up our web-based education system. In this stage, a hierarchical structured dictionary was used to perform structural preselection. It contained structural information of Bengali characters, e.g., stroke position, stroke combination, stroke crossing or not crossing etc. Figure 5 shows an example of structural difference between the Bengali characters “ন (na)” and “ম (ma)”. Both of these two characters have two numbers of strokes and belong to the same level 2 in Filter I preselection (Table 2). In our hierarchical recognition engine, Filter II is applied to classify between these two characters based on their structural patterns as shown in Figure 5, the second stroke’s starting and ending points of “ন (na)” and “ম (ma)” are opposite from each other. We used this structural information to classify between these two characters using our predefined structural dictionary (Table 4).

Table 3 represents the symbolic notation of Bengali character stroke’s position. We have used this notation to edit our predefined structural dictionary. Here, the co-ordinate for starting and ending point of 1st stroke represented as (x1, y1) and (i1, j1) respectively. Center point of 1st stroke was represented as (a1, b1) and calculated from (x1, y1) and (i1, j1). Here, the numbers of co-ordinate data are followed by the stroke number of any Bengali characters.

Table 3. The symbolic notation of Bengali character stroke’s position using (x, y) co-ordinates

| # | Stroke position | Stroke’s X-Axis Notation for 1 st stroke | Stroke’s Y-Axis Notation for 1 st stroke |
|---|-----------------|---|---|
| 1 | Starting Point | x1 | y1 |
| 2 | Center Point | $a1=(x1+i1)/2$ | $b1=(y1+j1)/2$ |
| 3 | Ending Point | i1 | j1 |

In Figure 5, the x-axis data of 2nd stroke’s starting and ending points for “ন (na)” and “ম (ma)” can be represented as x2 and i2 respectively. As shown in Figure 5, the second stroke’s starting and ending points for Bengali characters “ন (na)” and “ম (ma)” are opposite from each other. In case of “ন (na)”, x2 is greater than i2 and it can be represented as $x2-i2!$ ($x2-i2 \geq 0$) in our predefined structural dictionary (Table 4). Oppositely, i2 is greater than x2 for Bengali character “ম (ma)” and it can be represented as $i2-x2!$ ($i2-x2 \geq 0$) in our structural dictionary (Table 4).

Table 4 represents the examples of our structural dictionary for Bengali character “ন (na)” and “ম (ma)”. Here, (x, y) and (i, j) with corresponding stroke number represents the starting and ending point of input stroke respectively. (a, b) is the central point and calculated as $a=(x+i)/2$; $b=(y+j)/2$. For example, (x1, y1), (i1, j1) and (a1, b1) represents the starting, ending and central point of 1st stroke. Similarly, the (x2, y2), (i2, j2), (a2, b2) represent the starting, ending and central point of 2nd stroke.

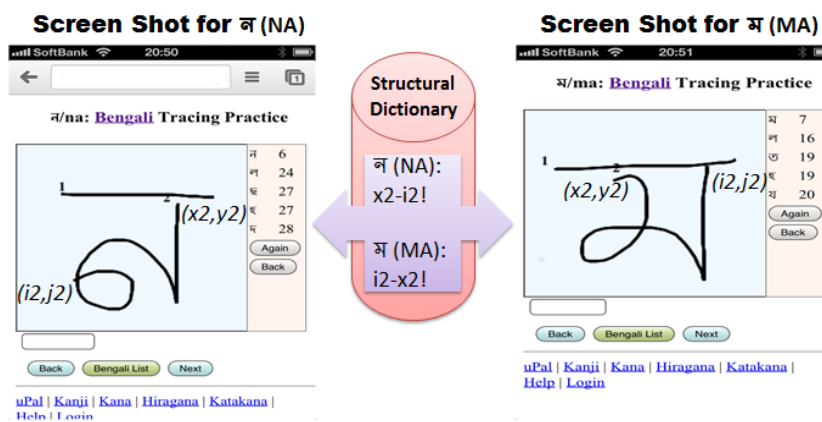


Figure 5. An example of structural difference between Bengali characters “ন (na)” and “ম (ma)”

In this structural dictionary, single template character was presented with multiple structural patterns depend on its probable handwriting errors (David et al., 1997). For example, the Bengali character “ন (na)” and “ম (ma)” has 6 different structural patterns considering its probable error case. Here, the 3rd column, index is used to identify the handwriting error pattern of corresponding student’s input. The 4th column, positional condition is used to locate stroke relationship error and provide correct recognition output together with necessary error feedback. The 5th column, stroke pattern is used to identify which stroke sequence was inputted and then provides colorful feedback to students about their handwriting stroke sequence. Also, it identifies the reverse stroke direction errors by checking the negative value in stroke pattern column (5th column in Table 4). The all of this error detection mechanism will be discussed detail in section “2.4 Automatic stroke error detection and stroke feedback”.

In our Bengali education system, recognition engine read the classification rule from predefined structural dictionary and applied those rules to classify between resulted candidates of Filter I. By using this classification rule, Filter II can successfully preselect the desired candidate characters for the final DPM matching. Thus, it further reduced the recognition search space and speed up our recognition engine to adapt with web-based Bengali education system.

Table 4. Examples of predefined structural dictionary for the Bengali character “ন (na)” and “ম (ma)”

| #Unicode | Char. | Index | Positional Condition | Stroke Pattern | Comments |
|-----------|-------|-------|----------------------|----------------|---|
| 09A8 0000 | — | 0 | b2-b1! x2-i2! | 1 2 | Black for all: Correct |
| 09A8 0000 | → | 1 | b2-b1! i2-x2! | 1 -2 | Red for 2 nd : Direction |
| 09A8 0000 | → | 2 | b2-b1! x2-i2! | -1 -2 | Red for all: Direction error |
| 09A8 0000 | → | 3 | b1-b2! x1-i1! | 2 1 | Brown for all: Order error |
| 09A8 0000 | → | 4 | b2-b1! x2-i2! | -2 1 | Red for 1 st :Direction error |
| 09A8 0000 | → | 5 | b1-b2! i1-x1! | -2 -1 | Red for all: Direction error |
| : | : | : | : | : | : |
| 09AE 0000 | ↖ | 0 | b2-b1! i2-x2! | 1 2 | Black for all: Correct |
| 09AE 0000 | ↖ | 1 | b2-b1! x2-i2! | 1 -2 | Red for 2 nd : Direction |
| 09AE 0000 | ↖ | 2 | b2-b1! i2-x2! | -1 -2 | Red for all: Direction error |
| 09AE 0000 | ↖ | 3 | b1-b2! i1-x1! | 2 1 | Brown for all: Order error |
| 09AE 0000 | ↖ | 4 | b2-b1! i2-x2! | -2 1 | Red for 1 st : Direction error |
| 09AE 0000 | ↖ | 5 | b2-b1! x2-i2! | -2 -1 | Red for all: Direction error |

Filter III: Final Matching by DPM with Writing Speed-Free Recognition Technique

In this section we explained about our proposed modified dynamic programming matching algorithm that support writing speed free recognition. In our proposed system, the recognition scheme is carried out using dynamic programming concept which is modified by accepting different length of input feature points to support writing speed free recognition. According to DPM algorithm, handwritten input pattern is matched with template patterns by calculating optimal matching cost, also known as character distance (Hu et al., 2007; Joshi et al., 2006; Prasanth et al., 2007; Tan et al., 2002; Shin et al., 2004; Tonouchi et al., 1997). In our recognition scheme, the term character distance stands for the angular difference between input stroke’s angles and corresponding template stroke’s angles. Then the character with optimal distance is selected as our recognition output and return back to students with necessary feedback. The mathematical notation for DPM is explained as follows.

To match handwritten input character with the template characters, we calculate character distance, D_k for corresponding template pattern k . A distance D_k for the candidate character k can be calculated as follows,

$$D_k = \frac{1}{L} \sum_{l=1}^L d_{kl} \quad (1)$$

Where, L is the number of total input strokes, k is the number of candidate template characters, and l is the number of handwritten strokes. The candidate character with smallest D_k is selected as the recognition result for current handwritten input character. The stroke distance for each template character can be calculated using dynamic programming matching technique as follows,

$$d_{kl} = \frac{g(I_l, J_{kl})}{I_l + J_{kl}} \quad (2)$$

Where, $g(I_l, J_{kl})$ represents the modified DPM distance between input feature vector I_l and k_{th} template feature vector J_{kl} for corresponding stroke l . We have modified the recurrence relation of DPM algorithm as below to find the character distance between two stroke sequences,

Initially,

$$g(i_l, j_{kl}) = \begin{cases} 0 & (i = 0, j_{kl} = 0) \\ \infty & (\text{other}) \end{cases}$$

Recursively,

$$g(i_l, j_{kl}) = \min \begin{cases} g(i_l - 2, j_{kl} - 1) + 2d(i_l - 1, j_{kl}) + d(i_l, j_{kl}) \\ g(i_l - 1, j_{kl} - 1) + 2d(i_l, j_{kl}) \\ g(i_l - 1, j_{kl} - 2) + 2d(i_l, j_{kl} - 1) + d(i_l, j_{kl}) \end{cases} \quad (3)$$

Where, $g(i_l, j_{kl})$ is the cumulative distance up to the current template character, $d(i_l, j_{kl})$ is local cost for measuring the dissimilarity between i_l^{th} and j_{kl}^{th} point of two sequences. We assumed that x_{il} represents the angular sequence of l^{th} input stroke and r_{klj} represents angular sequence of l^{th} template stroke of k^{th} character. The local distance as well as dissimilarity between the two stroke sequences can be measured as below:

$$d(i_l, j_{kl}) = \begin{cases} |x_{il} - r_{klj}| & \{-180^\circ \leq |x_{il} - r_{klj}| \leq 180^\circ\} \\ 180^\circ - |x_{il} - r_{klj}| & \{|x_{il} - r_{klj}| < -180^\circ\} \text{ or } \{180^\circ < |x_{il} - r_{klj}|\} \end{cases} \quad (4)$$

To establish writing speed free recognition, we modified the traditional dynamic programming matching algorithm. As we explained in previous section, our handwriting digital web interface extracts the user stroke data (feature points) and sent to our recognition server. Fundamentally, the number of extracted feature points is inversely proportional to student's handwriting speed. Slow writing speed provides large number of feature points, and oppositely fast writing has small number of feature points. Local cost as well as dissimilarity measurement between i_l^{th} and j_{kl}^{th} point of two sequences can be calculated by $d(i_l, j_{kl})$ where i_l is the number of angular feature points of l^{th} input stroke and j_{kl} is the number of angular feature points of l^{th} stroke of k^{th} template character. For slow handwriting case, i_l may greater than or equal to $2*j_{kl}$. Oppositely in case of fast handwriting, j_{kl} may greater than or equal to $2*i_l$. In this condition, the calculation of local stroke distance, dissimilarity measurement $d(i_l, j_{kl})$ (in equation 4) may fail due to the adaptability problem of adjustment window size in DPM algorithm. To avoid this problem, we modified the existing DPM to accept the input strokes data of any length wherever it greater or smaller than two times of corresponding template stroke's length.

For instance, we considered a slow handwriting case of children or older people where the number of angular data of input stroke i_l may greater than or equal to $2*j_{kl}$, the number of angular data for l^{th} stroke of k^{th} template character. We assume i_l is 24 and j_{kl} is 9. In this case, our modified DPM only consider the even sequences of angular data for input stroke i_l . In this way, modified DPM can successfully solved the adaptability problem of dissimilarity measurement $d(i_l, j_{kl})$ (in equation 4). Thus, our proposed Bengali education system can accept both of fast and slow handwriting of children and older people.

In addition, we modified the following settings of adaptive adjustment window size as equation 5 to accept any pattern of handwriting input. Here, W represents the adjustment window size. From the experimental analysis, we found the optimal value of $W=18$, that makes our system highly adaptable to recognize rough handwriting

characters. Here I_l represents the total number of input stroke sequence and J_{kl} represents the total number of template stroke sequences of corresponding stroke l .

$$1 \leq i \leq I_l, \max \{1, i \times J_{kl} / I_l - W\} \leq j_{kl} \leq \min \{J_{kl}, i \times J_{kl} / I_l - W\} \tag{5}$$

In practical, our intelligent Bengali handwriting education system was developed to improve the Bengali literacy rate by considering both of children and older students. Basically, children have slow handwriting speed and aged people have fast handwriting speed. By the above modification, our recognition engine can accept both of input patterns from children and older people. In this way, we can successfully implement the writers' independent recognition algorithm for our web-based Bengali handwriting education system. By using this writing speed free recognition technique, the accuracy was improved considerably. In a later section, we evaluate our proposed system using a rich Bengali handwritten character database.

2.4 Automatic Stroke Error Detection & Stroke Feedback

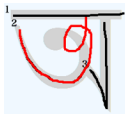
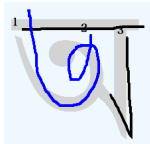
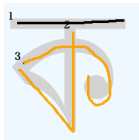
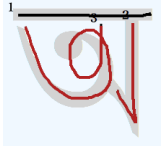
2.4.1 Student Feedback for Autonomous Learning

In our Bengali handwriting education system, we have developed an automatic stroke error detection methodology. It aims to identify the handwriting errors in student's handwriting and provide immediate feedback. We classified the handwriting errors as stroke production error and stroke relationship error and stroke order error. Stroke production error consists of reverse stroke direction, split stroke and merge stroke errors etc. On the other hand, stroke relationship error is the error where students write the stroke with extra length and the stroke order error is the error of wrong stroke sequence. Our automatic stroke error detection engine identifies the student's handwriting error and provides feedback to correct them. This error detection methodology was implemented using JSON: JavaScript Object Notation technology, see in Figure 6.

Table 5. Bengali handwriting error and idea of student's error feedback

| # | Error Category | Error Details | Color Marked Feedback |
|---|---------------------------|--------------------------|-----------------------|
| a | Correct handwriting | Correct Stroke | Black |
| b | Stroke production errors | Reverse direction | Red |
| | | Split/broken stroke | Purple |
| c | Stroke relationship error | Stroke with extra length | Blue |
| d | Stroke order error | Wrong Stroke Sequence | Brown |

Table 6. Examples of handwriting error patterns of Bengali characters: (1, 2) Stroke production error. (3) Stroke relationship error. (4) Stroke order error. (Numeric symbols means stroke no. and its start point)

| # | Student's Error Feedback | Error Type | # | Student's Error Feedback | Error Type |
|---|---|--|---|--|---|
| 1 |  | Reverse stroke direction (Red) | 3 |  | Stroke with extra length (Blue) |
| 2 |  | Split stroke (Purple) | 4 |  | Wrong stroke order (Brown) |

In this methodology, we marked erroneous strokes using different color models depend on the student's handwriting errors as shown in Table 5 and Table 6. The notation of “{“r”:[“অ”, “score”], “d”:[1,-2,3], “c”:[“black”,“red”,“black”]}” is an example to clarify our proposed JSON technique. Here, we marked the

reverse direction input stroke for character আ[a] by the red color. In this notation, “r” stands for the recognized character and its recognition score, “d” represents the stroke orders, and “c” is the color combination to mark the reversely inputted stroke. While the students write any stroke with reverse direction then our system detect it and feedback with appropriate color marking. In this case, the second stroke was marked as red color. Thus, our proposed education system feedback student’s error using different color models depend on their handwriting errors as shown in Table 5 and 6.

Figure 6 represents the snapshot of student’s feedback result to correct their error handwriting. Our proposed education systems can successfully feedback to the students about their error handwriting using colorful marking technology. We implemented this method to our system by JSON technology.

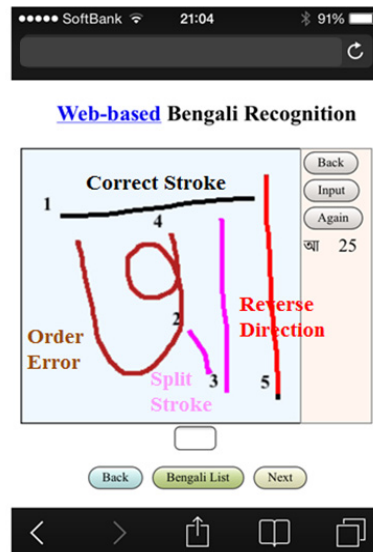


Figure 6. Automatic error detection and colorful feedback to correct student’s erroneous handwriting (Numeric symbols means stroke no. and its starting point)

2.4.2 Automatic Stroke Error Detection: How Does It Work

In our web-based intelligent handwriting education system, we developed predefined structural dictionary based on the structural information of Bengali characters. Table 4 is the examples of structural dictionary for a single Bengali characters “ন (na)” and “ম (ma)”. Based on this information, our recognition engine can successfully recognize the handwriting errors of Bengali characters and then feedback to students with necessary color marking.

As we described in Table 4, the single character pattern was presented with multiple structural patterns depend on its probable handwriting errors in our designed structural dictionary (David et al., 1997). For example, the Bengali character “ন (na)” and “ম (ma)” has 12 different structural patterns considering its probable error case. Here, the third column, index is used to identify the handwriting error pattern of corresponding student’s input. The fourth column, positional condition is used to locate stroke relationship error and provide correct recognition output together with necessary error feedback. The 5th column, stroke pattern is used to identify which stroke sequence was inputted and if they made any mistake then it provides colorful feedback to students about their wrong writing. Also, it identifies the reverse stroke direction errors by checking the negative value in stroke pattern column (5th column in Table 4). All of this error detection mechanism will be discussed as below.

In our online handwriting recognition engine, we used client-server interface to extract the feature points and relevant structural information as (x, y) coordinates along the trajectory of the input device (e.g., pen, mouse, finger etc.) onto the digital web interface. Then, we convert it to angular feature and match those angles with the angular features of preselected template characters and obtain a matching distance between them. In our hierarchical recognition algorithm, we applied multiple filters to reduce the recognition search space.

In Filter II, all of the handwriting input patterns were matched with the multiple patterns of predefined structural dictionary using their positional condition (Table 4, 4th column). First our recognition engine obtain a matching distance between input character and structural patterns, and then selects the candidate character for final matching which have the minimum matching distance. If the inputted character was matched with erroneous structural patterns, then our algorithm can detect the committed error by using the pattern index of structural dictionary (index 0~5, as shown in Table 4), and immediately send the colorful error feedback about their mistakes. Below is the mathematical notation of above algorithm:

$$k = \underset{k \in K}{\operatorname{argmin}} \{d_k\} \quad (6)$$

K =Total number of template characters

k =Student's k^{th} sample character

d_k =Matching distance between inputted character and structural patterns of predefined structural dictionary

As we discussed above, the character with optimal d_k is selected for final matching and thus the corresponding index number can easily be identified. After the identification of index number, the relevant stroke pattern can also be located from the 5th column of Table 4. Then we return the feedback to students about their writing mistake by marking the wrong strokes with different colors (6th column in Table 4). If the identified stroke pattern has any negative value then our recognition engine can detect that the student has inputted a stroke with reverse direction. Then it returns feedback to students by marking the reverse strokes with red color. After the identification of reverse stroke direction, our recognition algorithm reverses the angular feature of corresponding template characters and matches with the reversely inputted stroke's data of sample characters. In our system, angular feature of original template characters is stored into an array t . After the detection of reverse stroke direction, our algorithm automatically converted angular feature of original template characters using a common angular conversion rule (180°-angle) and stored into an array r . Then we match the student's input stroke with corresponding reversed stroke of template characters. Thus, our recognition engine can successfully accept the reverse stroke input and provide students the correct recognition result. In a similarly way, our algorithm can detect and send feedback for the stroke relationship errors, such as stroke with extra length as shown in Table 6. For example, in case of Bengali characters ঞ[a], the second stroke position is bottom of the first stroke. In correct recognition case, it satisfies the condition of $y_2 > b_1$ ($y_2 - b_1!$ in 4th column) where y_2 is the start point of second stroke and b_1 is the central point of first stroke. From the student input stroke data, our automatic error detection engine can judge that whether $y_2 > b_1$ or not. If $y_2 > b_1$ then handwriting was correct otherwise there have a stroke relationship error and then feedback to the students by marking the inputted stroke with blue color. In this way, our proposed intelligent handwriting education system can successfully provide the error feedback together with recognition output.

3. Learning System Evaluation

For the system evaluation, we have conducted the experimental analysis using the design data set and test data set. We have collected handwritten patterns of Bengali characters from different students where the students of design data set and test data set were different. Handwritten character samples of 21 students were used as design data set and 24 student's data were used as test data set.

Table 7. The experiment results of our web-based recognition engine (SCHEME1: DPM ALGORITHM WITH HIERARCHICAL PRESELECTION)

| Recognition Scheme | No. of Database | Recognition Accuracy (%) | | | Speed (ms/character) |
|--------------------|-----------------|--------------------------|-------|-------|----------------------|
| | | Top 1 | Top 2 | Top 3 | |
| Scheme1 | 10290 | 87% | 92% | 95% | 40 ms |

Table 7 gives the experimental results of our proposed system using the design data set. In our recognition engine, we applied writing speed free DPM algorithm together with hierarchical preselection. We noticed that our proposed recognition methodology achieved the highest recognition accuracy for every top choice; particularly it achieved 95% accuracy considering Top3 choice. Moreover, the recognition time is significantly reduced to 40 ms/character. These facts ensured that proposed hierarchical recognition scheme with DPM

reduced the inherent computational complexity and speed up our recognition engine to adapt with web-based Bengali education system.

Finally, we have conducted a survey for the acceptance of our web-based Bengali handwriting education system in Bangladesh. For test data set, we have collected handwritten character patterns from 24 Bengali native students of 4 different age groups with respect to age, education and gender together with their questionnaires. During this survey, each student has written almost two times of every Bengali character sample, 12 of them has written two or more times. In this survey, we have collected the student's individual scores in terms of "Simplicity", "Recognition Speed", "Colorful Error Feedback", and "Effectiveness" of our Bengali handwriting education system. Here, the student's evaluation score is ranged from 1 (Bad) to 5 (Best), and separated as "Bad" (Score 1), "Poor" (Score 2), "Fair" (Score 3), "Good" (Score 4) and "Best" (Score 5). Table 8 represents the student's MOS matrix based on 4 different evaluation terms. Here, the MOS (Mean-Opinion-Score) data was calculated by the arithmetic mean of all the individual scores for 4 different evaluation terms. As shown in Table 8, the average MOS value in each term is above 4.0. It confirmed that our proposed Bengali handwriting education system achieved "Good" in all evaluation categories. In addition, the students of age 5~15 years old and age 60~70 years old have the higher MOS of 4.6 and 4.9 respectively in terms "Effectiveness" whereas the students of age 15~30 years old and age 30~50 years old have the lower MOS of 3.0 and 3.8 respectively. It ensured that the children and older people who do not have the chance to go to school liked our education system more than the middle aged students. We further analyze the questionnaire's MOS data together with handwritten character recognition accuracy of 24 people of different age in next part.

Table 8. Student's MOS matrix based on 4 different evaluation terms

| Student's Age (Years) | Average MOS for Simplicity | Average MOS for Speed | Average MOS for Feedback | Average MOS for Effectiveness | Average MOS of Each Age Group |
|--------------------------|----------------------------------|-----------------------------|--------------------------------|-------------------------------------|-------------------------------------|
| 5-15 | 4.0 | 4.3 | 4.8 | 4.6 | 4.4 |
| 15-30 | 4.3 | 3.7 | 3.7 | 3.0 | 3.7 |
| 30-50 | 4.4 | 3.6 | 4.2 | 3.8 | 4.0 |
| 50-70 | 4.3 | 4.3 | 4.4 | 4.9 | 4.5 |
| Average MOS | 4.3 | 4.0 | 4.3 | 4.1 | 4.1 |

As we described above, all of the handwritten character samples and questionnaire's MOS data were stored in our database table. We have executed an experimental analysis of that test data set (2,352 handwritten characters samples) by using our developed recognition batch program and stored the recognition results of each student in another database table. Then, we calculated the recognition accuracy (%) of every writer together with their provided MOS value from the database table where we have stored all the recognition results during the execution of recognition batch program. Table 9 shows the surveyed results of student's evaluation score in terms of average MOS together with the average handwriting recognition accuracy (%) of 4 different age groups. For the performance analysis, we have calculated the student's recognition accuracy into two separate parts, 1st trial handwriting recognition accuracy (%) and 2nd trial handwriting recognition accuracy (%). It was calculated by using the 1st time and 2nd time handwriting test data set of each writer respectively. We considered the student's MOS value and their 1st trial and 2nd trial handwriting recognition accuracy as system evaluation parameters.

As shown in Table 9, the students of age 5~15 years old and age 60~70 years old have the MOS value of 4.4 and 4.5 with 2nd trial handwriting recognition accuracy of 91.1% and 95.3% respectively. Oppositely, the middle age students (15~30 and 30~50 years old) have the low recognition accuracy with low MOS value. It ensured that the students with higher recognition accuracy have the higher MOS values and the students with low recognition accuracy have the lower MOS values. This result also confirmed that the average of 2nd trial handwriting recognition accuracy (average 91.4%) was improved compare to 1st trial handwriting recognition accuracy (average 87.2%) for each age group, and the average improvement was 4.1% by using our autonomous learning methodology.

Table 9. Student's survey results for the acceptance of web-based Bengali handwriting education system together with the average recognition accuracy of each age group

| Student's Age (Years) | No. of Students | 1 st Trial Handwriting Recognition Accuracy (%) | 2 nd Trial Handwriting Recognition Accuracy (%) | Improved Recognition Accuracy (%) | Average MOS of Each Age Group | Standard Deviation(S.D.) |
|-----------------------|-----------------|--|--|-----------------------------------|-------------------------------|--------------------------|
| 5~15 | 9 | 89.1 | 91.1 | 2.0 | 4.4 | 0.5 |
| 15~30 | 3 | 81.1 | 89.2 | 8.1 | 3.7 | 0.6 |
| 30~50 | 5 | 82.5 | 85.2 | 2.7 | 4.0 | 1.0 |
| 50~70 | 7 | 88.9 | 95.3 | 6.4 | 4.5 | 0.5 |
| Average | 6.0 | 87.2 | 91.4 | 4.1 | 4.1 | 0.7 |

This improvement was achieved due to the colorful error feedback about student's handwriting errors for their 1st time handwriting. The students can successfully notice their handwriting mistakes of stroke order or stroke direction. Then, they can correct their handwriting in 2nd trial, and thus the 2nd trial handwriting recognition accuracy was improved. Figure 7 shows the scatter plot of 1st trial and 2nd trial handwriting recognition accuracy (%) of every student to analysis the performance improvement of our proposed Bengali education system. As shown in Figure 7, the trend line is moving in the positive direction and it achieved the positive gradient with "strong" positive correlation where the larger values of 1st trial handwriting recognition accuracy data (x-axis) are associated with the larger values of the 2nd trial handwriting recognition accuracy data (y-axis). It depicts that 2nd trial handwriting recognition accuracy (%) was improved compare to 1st trial handwriting recognition accuracy (%) for almost every students, and the average accuracy was improved by 4.1% (Table 9).

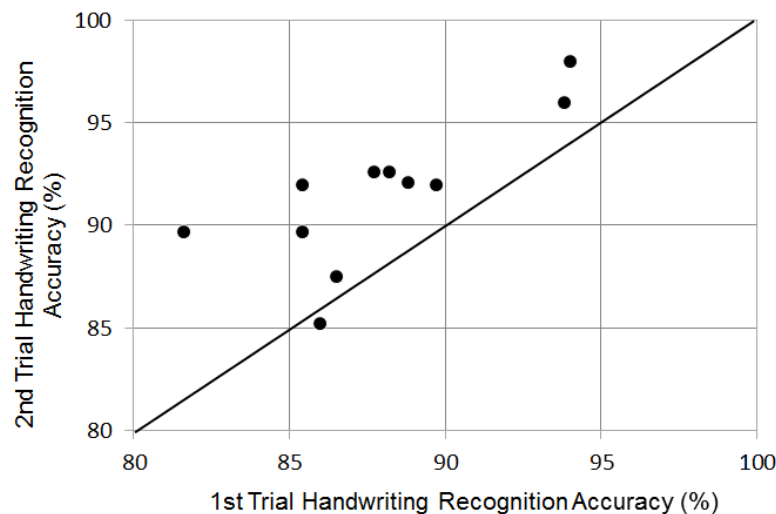


Figure 7. The scatter plot of 1st trial and 2nd trial handwriting recognition accuracy (%) of 12 students to analysis the performance improvement for our proposed Bengali education system

Moreover, the students of age 5~15 years old and age 60~70 years old have the MOS value of 4.4 and 4.5 with standard deviation 0.5, and the students of age 15~30 years old have the MOS value of 3.7 with standard deviation 0.6 (Table 8, 9). It depicted that most of the students in Bangladesh, especially children or older people, who do not have chance to go to school, liked our Bengali education system to practice Bengali handwriting characters. On the other hand, the middle age students (15~30 years old) have the low MOS value with higher standard deviation, it illustrates that most of the people aged 15~30 years old are literate and their opinion is fluctuating. But, the literacy rate of children or older people are very lower compare to middle age people, and

they can learn without teacher supervision at anywhere at any time and they can correct their committed error using real time colorful error feedbacks. In addition, they can repeat the same exercise several times to speed up their learning process. The above analytic results confirmed that our proposed web-based Bengali education system is highly appreciated by the illiterate people in Bangladesh. Moreover, the total average MOS for all aged people has the value of 4.1 with the standard deviation of 0.7. This value ensured that almost every user evaluated our system as “Good” (Score 4).

The following Figure 8 shows an example of autonomous learning of Bengali characters অ (a) which has the three individual strokes. Since the second stroke of this character has a similar shape with other Bengali characters, handwritten mistake is very easy to occur. In our survey, almost 8 people made the stroke order mistakes between 1st and 2nd strokes of Bengali characters অ (a) for almost two or three times. Also, 4 of them have made the stroke direction mistake for the 2nd stroke of অ (a) for three times. Our autonomous learning tool can successfully send the colorful feedback about the student’s handwriting errors Figure 9 (left side), and the students can correct their own mistake by themselves as shown in Figure 9 (right side). In our proposed Bengali education system, we are aimed to teach our students correct and attractive handwriting style automatically that makes the students be able to write good balanced Bengali handwriting characters. For instance, the second stroke of Bengali character ভ (bha) and ষ (cha) has similar handwriting shape. It’s starting and ending points are opposite from each other. Reverse handwriting input of second stroke of “ভ(bha)”, turns it into the Bengali characters “ষ(cha)” and vice versa. This kind of miss handwriting input creates lots of miss understanding between Bengali characters. So, the stroke’s error (e.g., stroke direction, order, split or merge errors) should be detected to teach our students correct and attractive handwriting style.

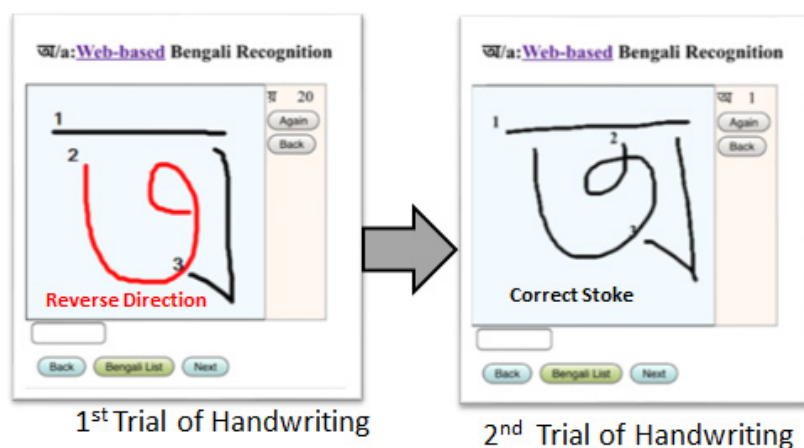


Figure 8. An example of autonomous learning of Bengali characters অ (a)

4. Conclusion and Future Work

In this paper, we have described the effectiveness of autonomous learning methodology for the literacy improvement by using our proposed web-based Bengali handwriting education system. It ensured the autonomous learning of Bengali handwriting characters at anywhere at any time for those population, who do not have chance to go to school, especially children or older people. Here, we developed a web-based (iPhone/smartphone or computer browser) intelligent handwriting client-server interface using JavaServer Pages technology for autonomous learning of Bengali handwriting characters. Our experimental analysis showed that the use of colorful error feedback methodology helped to improve the average recognition accuracy by 4.1% (improved from 87.2% to 91.4%) with average Mean-Opinion-Score of 4.1. Also, our proposed hierarchical recognition algorithm together with writing speed free DPM improved the average recognition accuracy up to 95% as well as recognition speed of 40ms/character for Bengali basic characters. It makes our recognition algorithm adaptable for the application of web-based language learning. Our automatic error detection methodology ensured the necessary feedback to the students to learn about their handwriting mistake autonomously. Since all of the population, especially children and older people do not have the chance to go to school in Bangladesh. So, schooling system is not enough to achieve the 100% literacy improvement. The successful use of web-based Bengali handwriting education system can help to achieve 100% literacy improvement in Bangladesh within a very short period.

In future, we will focus our research to improve the recognition accuracy. To implement student's feedback methodology regarding stroke order free and stroke direction free recognition, miss-recognition possibility was increased. In our future work, we will apply the fuzzy rule to build structural dictionary. In addition, we will focus our research on the development of Bengali handwriting education system considering Bengali word learning.

Acknowledgements

We would like to thank all the people in our laboratory for their important comments and the subjects for the learning system evaluation.

References

- Abdou, S., Fahmy, A., & Elgammal, A. (2010). A Tool for Arabic Handwriting Training. In *The Fifth Conference of Learning International Networks Consortium (LINC)* (pp. 23-26). MIT, Cambridge, Massachusetts, USA.
- Ahmad, T., Al-Taani, & Al-Haj, S. (2010). Recognition of On-line Arabic Handwritten Characters using Structural Features. *International Journal of Pattern Recognition Research, 1*, 23-37.
- Bhattacharya, U., Nigam, A., Rawat, Y. S., & Parui, S. K. (2008). An Analytic Scheme for Online Handwritten Bangla Cursive Word Recognition. In *the Proceedings of the International Conference on Frontiers in Handwriting Recognition (ICFHR-2008)* (pp. 320-325). Montreal, Canada. Retrieved from <http://www.iapr-tc11.org/archive/icfhr2008/Proceedings/papers/cr1063.pdf>
- Bangladesh Literacy Survey Report by UNESCO.* (2015). Retrieved from https://en.wikipedia.org/wiki/List_of_countries_by_literacy_rate
- Deanna, H., Séverin, L., & Pierre, D. (2015). When Children Teach a Robot to Write: An Autonomous Teachable Humanoid Which Uses Simulated Handwriting. In *Proceedings of the 10th Annual ACM/IEEE International Conference on Human-Robot Interaction* (pp. 83-90). ACM New York, NY, USA. <http://dx.doi.org/10.1145/2696454.2696479>
- David, T. R. (1997). *Japanese/English dictionary and character dictionary browser featuring hand writing recognition-based character lookup*. ACM Quest for Java Contest, JavaDict. Retrieved from <http://www.cs.arizona.edu/projects/japan/JavaDict/>
- Hiekata, K. (2007). A framework for design engineering education with work flow-based e-learning system. *Journal of software, 2*(4), 88-95. <http://dx.doi.org/10.4304/jsw.2.4.88-95>
- Hu, Z., Xu, Y., Huang, L., & Leung, H. (2009). A Chinese Handwriting System with Automatic Error Detection. *International Journal of Software, Special Issue of Advanced Distance Learning Technologies, 4*(2), 101-107. <http://dx.doi.org/10.4304/jsw.4.2.101-107>
- Hu, Z. H., Leung, H., & Xu, Y. (2007). Stroke Correspondence Based on Graph Matching for Detecting Stroke Production Errors in Chinese Character Handwriting. In *Lecture Notes on Computer Science, Proceedings of Pacific-Rim Conference on Multimedia* (pp. 734-743). http://dx.doi.org/10.1007/978-3-540-77255-2_89
- Joshi, N., Sita, G., & Ramakrishnan, A. G. (2004). Comparison of Elastic Matching Algorithms for Online Tamil Handwritten Character Recognition. In *IEEE Computer Society, Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR-9)* (pp. 1-6). <http://dx.doi.org/10.1109/iwfh.2004.30>
- Parui, S. K., Guin, K., Bhattacharya, U., & Chaudhuri, B. B. (2008). Online Handwritten Bangla Character Recognition Using HMM. In *19th International Conference on Pattern Recognition* (pp. 1-4). <http://dx.doi.org/10.1109/ICPR.2008.4761835>
- Prasanth, L., Jagadeesh, B. V., Raghunath, S. R., & Pradhakara, R. G. V. (2007). *Elastic Matching of Online Handwritten Tamil and Telugu Scripts Using Local Feature*.
- Shin, J. (2004). On-line cursive hangul recognition that uses DP matching to detect key segmentation points. *Elsevier Pattern Recognition, 37*, 2101-2112. <http://dx.doi.org/10.1016/j.patcog.2004.05.002>
- Tang, K. T., Li, K. K., & Leung, H. (2006). A web-based Chinese handwriting education system with automatic feedback and analysis. In *Lecture Notes in Computer Science, Proceedings of the 5th International Conference on Web-based Learning* (pp. 176-188). http://dx.doi.org/10.1007/11925293_17

- Tan, C. K. (2002). An algorithm for online strokes verification of Chinese characters using discrete features. In *8th International Workshop on Frontiers in Handwriting Recognition* (pp. 339-344).
- Tonouchi, Y., & Kawamura, A. (1997). An On-line Japanese character recognition method using length-based stroke correspondence algorithm. In *Proceedings of the Fourth Intl. Conf. on Document Analysis and Recognition (ICDAR)* (Vol. 2, pp. 633-636). <http://dx.doi.org/10.1109/icdar.1997.620582>
- Zein, O. K., & Kermarrec, Y. (2007). Description and composition of e-learning. *Journal of software*, 2(5), 74-83. <http://dx.doi.org/10.4304/jsw.2.5.74-83>

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).