C.-C Liu et al. (Eds.) (2014). Proceedings of the 22nd International Conference on Computers in Education. Japan: Asia-Pacific Society for Computers in Education

A Web-based Intelligent Handwriting Education System for Autonomous Learning of Bengali Characters

NAZMA Khatun[a*], Jouji MIWA[a]
[a]*Department of Electrical Engineering and Computer Science, Graduate School of Engineering, Iwate University, Japan*
*nazmababu@gmail.com

Abstract: In this paper, we describe a prototype of web-based intelligent handwriting education system for autonomous learning of Bengali characters. Bengali language is used by more than 211 million people of India and Bangladesh. Due to the socio-economical limitation, all of the population does not have the chance to go to school. This research project was aimed to develop an intelligent Bengali handwriting education system. 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 a feedback to the students to correct themselves. Our proposed system can be accessed from smartphone or iPhone that allows students to do practice their Bengali handwriting at anytime and anywhere. Bengali is a multi-stroke input characters with extremely long cursive shaped 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 for Bengali language learning. 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 national. The experimental results showed that our proposed hierarchical recognition algorithm can provide higher accuracy than traditional multi-stroke recognition algorithm with more writing variability.

Keywords: Intelligent handwriting education system, Hierarchical recognition, Handwriting errors, Automatic stroke error detection, Modified Dynamic Programing Matching (DPM), JSP (JavaServer Page), WWW 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 55% in 2010 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) [1]. Because of socio-economical limitation, all of the population, especially children and older national, 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 books. 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, Z. Hu et al. [2] define 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 for those population, especially children or older national, 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.

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 group 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 character with low 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 [3, 4] 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 [5] can find both the stroke production error and stroke sequence error but they did not consider the spatial relationship errors.

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 [6, 7]. 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 fifth 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. 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 Bengali Characters

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

2.2 System Architecture

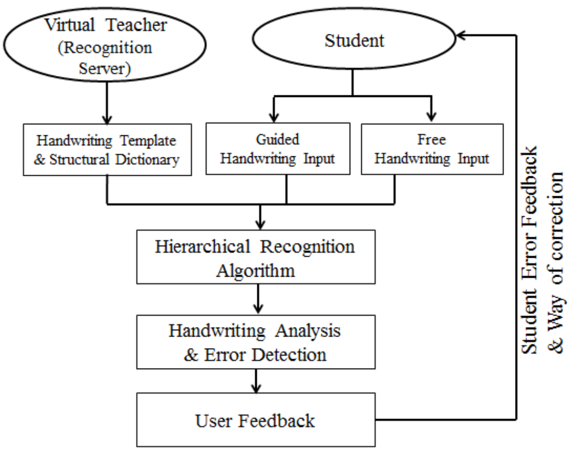
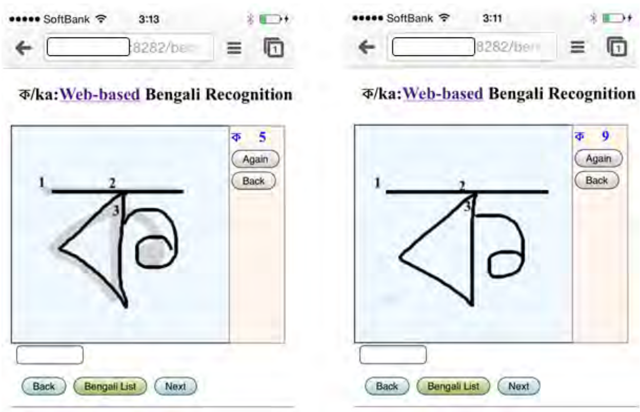


Figure 1: The architecture of the proposed Bengali Handwriting Education System

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 first levels 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). Then, the user is invited to follow this image to replicate the pattern of script. 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 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 2 and table 3). The details of the automatic stroke error detection will be described in the following section.

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.



(a) (b)

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 Automatic Stroke error detection & Stroke Feedback

2.3.1 Student Feedback for Autonomous Learning

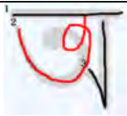
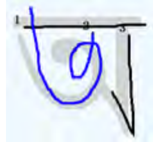
In our Bengali handwriting education system, we have developed an automatic stroke error detection methodology. It aims to identify the handwriting errors in student's handwriting and provide immediate feedback. We classified the handwriting errors as stroke production error and stroke relationship error 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 5. In this methodology, we marked erroneous strokes using different color models depend on the student's handwriting errors as shown in table 2 and 3. 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 অ[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

Table 2: Bengali handwriting error and idea of student's 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 3: Examples of handwriting error patterns of Bengali characters: (a) Stroke production error. (b) Stroke relationship error. (c) 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)**

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 2 and 3.

2.3.2 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 snapshot of structural dictionary for a single Bengali characters অ[a]. 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.

Table 4: Snapshot of our predefined structural dictionary

#Unicode	Char.	Index	Positional Condition	Stroke order	Comments
--------------	------	--------	----------------------------	------------	-------------
0985 0000	অ	0	| y2-b1! y2-j1! a3-x2!	| 1 2 3	|#Correct
0985 0000	অ	1	| y1-b3! y1-j3! a2-x1!	| **2 3 1**	|#Order errors
0985 0000	অ	2	| y1-b2! y1-j2! a3-x1!	| **2 1 3**	|#Order errors
0985 0000	অ	3	| y1-b2! y1-j2! a3-x1!	| 1 **3 2**	|#Order errors
0985 0000	অ	4	| y2-b1! y2-j1! a3-x2!	| 1 **-2** 3	|#Reverse error

In table 4, (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 first stroke. Similarly, the (x2, y2), (i2, j2), (a1, b2) and (x3, y3), (i3, j3), (a3, b3) represent the starting, ending and central point of second and third stroke. In this structural dictionary, single template character was presented with multiple structural patterns depend on its probable handwriting errors [8]. For example, the Bengali character অ[a] has 5 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. Finally, the fifth column, stroke order is used to identify which stroke sequence was inputted and then provides colorful feedback to students about their wrong stroke sequence. Also, it identifies the reverse stroke direction errors by checking the negative value in stroke order column (5^{th} column in Table 4). The 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 system, each template character has its different structural patterns (index 0~4, as shown in table 4). Finally, it selects the optimal distance as the recognition results. Below is the mathematical notation:

$$k = \underset{k \in K}{\operatorname{argmin}} \{d_k\} \quad \text{(1)}$$

K = Total number of template characters
k = Student's sample character
d_k= Angular distance between sample characters and template characters considering multiple structural patterns of each template.
O_k = Stroke order patterns

As discussed above, the character with optimal d_k is selected as recognition result and thus the corresponding index number can easily be identified. After the identification of index number, the relevant stroke order, O_k can also be located from the column 5 in table 4. Then we return the feedback to students about their writing mistake by marking the wrong strokes with brown color.

If the identified stroke order O_k 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 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 table 4, the fourth column is used to identify the stroke relationship errors, such as Stroke with extra length. In case of Bengali characters অ[a], the second stroke position is bottom of the first stroke. In correct recognition case, it satisfies the condition of *y2>b1* (*y2-b1!* in 4th column) where *y2* is the start point of second stroke and *b1* is the central point of first stroke. From the student input stroke data, our automatic error detection engine can judge that whether y2>b1 or not. If y2>b1 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.

2.4 Recognition Methodology

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 (PC or iPhone/Smart phone 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 user input (4) Apache tomcat web server and PostgreSQL database was used for system implementation (5) Consecutive handwriting and recognition is possible (6) immediate student feedback.

Figure 3 shows the handwriting recognition architecture that contains both of web-based handwriting interface and character recognition servers. Handwriting interface was developed by JSP and runs on WWW client, such as PC browser or iPhone/Smartphone browser. On the other hand, JSP based character recognition engine works on Apache tomcat web server (Linux 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 DP matching. 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 DP matching. In the final matching stage, low-level features (angular feature) are used to guide a dynamic programming matching 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.

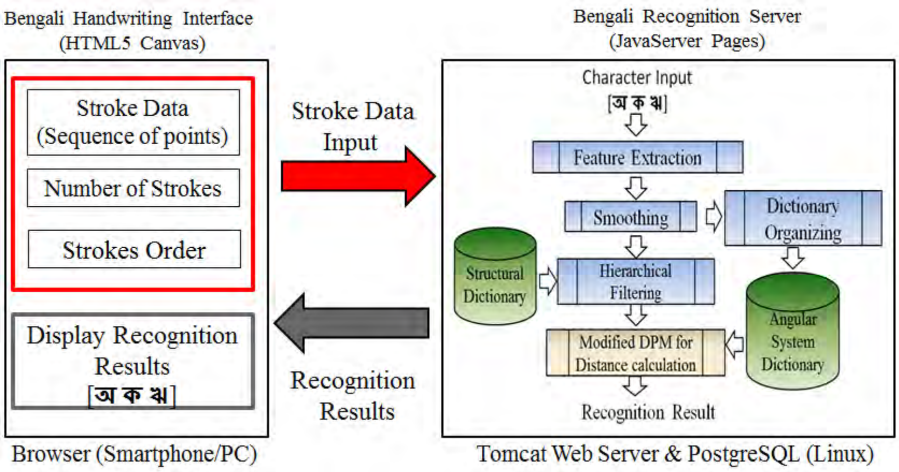


Figure 3: Handwriting recognition architecture for Bengali handwritten education system

2.4.1 *Modified Dynamic Programming for writing speed-free recognition*

In this section we explained about our proposed modified dynamic programming 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 dynamic programming matching algorithm, handwritten input pattern is matched with template patterns by calculating optimal matching cost, also known as character distance [9, 10, and 11]. 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 DP matching 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 normalize distance D_k for the candidate character k can be calculated as follows,

$$D_k = \frac{1}{L}\sum_{l=1}^{L} d_{kl} \quad \text{.......................................} (2)$$

Where, L is the number of total input strokes, k is the 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,

$$d_{kl} = \frac{g(I_l, J_{kl})}{I_l + J_{kl}} \quad \text{.......................................} (3)$$

Where, $g(I_l , J_{kl})$ represents the DPM distance between input feature vector I_l and k_{th} template feature vector J_{kl} for corresponding stroke l. The following recurrence relation is used to find the DPM distance between two sequences,

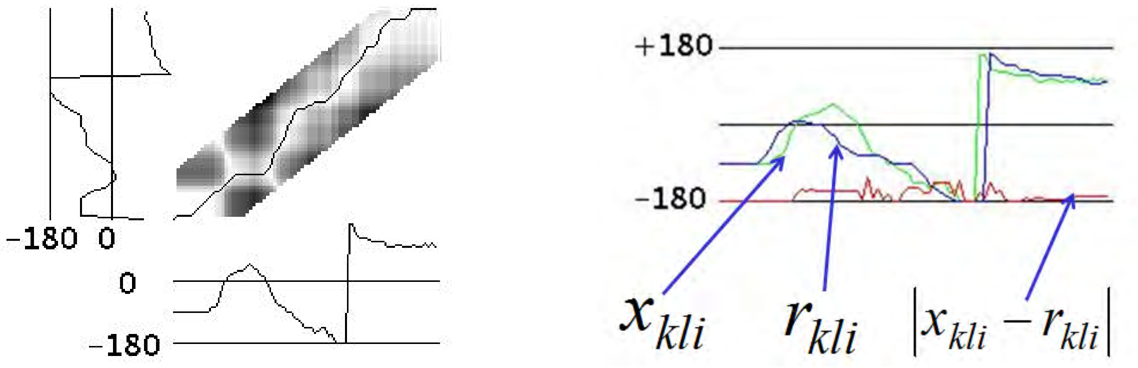
initially,

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

recursivelly,

$$g(i, j_{kl}) = \min \begin{Bmatrix} g(i, j_{kl} - 1) + d(i, j_{kl}) \\ g(i - 1, j_{kl}) + d(i, j_{kl}) \\ g(i - 1, j_{kl} - 1) + 2d(i, j_{kl}) \end{Bmatrix} \quad \ldots\ldots(4)$$

Where, $g(I, J_{kl})$ is the cumulative distance up to the current template character, $d(i, j_{kl})$ is local cost for measuring the dissimilarity between i^{th} and j_{kl}^{th} point of two sequences.



(a) (b)

Figure 4: Dynamic programming matching: (a) illustration of wrapping path for the 2nd stroke of Bengali character ড[da] (b) Calculation of local distance also known as dissimilarity between two strokes sequences of input and template.

Figure 4 represents illustration of wrapping path for DP matching and dissimilarity measurement between two strokes sequence. Here, x_{kli} represents the student's input stroke angles and r_{kli} represents the template stroke angles. The local distance as well as dissimilarity between the two stroke sequences can be measured as below:

$$d(I_l, J_{kl}) = \begin{cases} |x_{kli} - r_{kli}| & \{-180^\circ \le |x_{kli} - r_{kli}| \le 180^\circ\} \\ 180^\circ - |x_{kli} - r_{kli}| & \{|x_{kli} - r_{kli}| < -180^\circ\} \text{ or } \{180^\circ < |x_{kli} - r_{kli}|\} \end{cases} \quad \ldots\ldots (5)$$

To establish writing speed free recognition, we modified the traditional dynamic programming 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.

$$\textit{Number of input stroke data} \propto \frac{1}{\textit{Students handwriting speed}} \quad \ldots\ldots (6)$$

As in equation 5, local cost as well as dissimilarity measurement between i^{th} and j_{kl}^{th} point of two sequences can be calculated by $d(I, J_{kl})$ where I is the number of input stroke and J_{kl} is the number of k^{th} template stroke. For slow handwriting case, I may greater than or equal to $2*J_{kl}$. Oppositely in case of fast handwriting, J_{kl} may greater than or equal to $2*I$. In this condition, the calculation of local stroke distance, dissimilarity measurement $d(I, J_{kl})$ (in equation 5) 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. By using the following settings of adaptive adjustment window size as equation 6, our modified DPM can accept any length 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.

$$1 \le i \le I_l, \max\{1, i \times {}^{J_{kl}}\!/_{I_l} - W\} \le j_{kl} \le \min\{J_{kl}, i \times {}^{J_{kl}}\!/_{I_l} - W\} \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{(7)}$$

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 next section, we evaluate our proposed system using a rich Bengali handwritten character database.

3 Experimental Results

For our experiment, handwritten patterns were collected from 24 Bengali native writers of different groups with respect to age, education and gender, where each has written almost 10 times of every character sample. There are 50 basic characters in Bengali and therefore the proposed system has been evaluated by using 12,500 handwritten online character samples. We study the performance of proposed web-based intelligent handwriting education system by the three criteria: average recognition accuracy, average recognition speed, and automatic error detection capability.

Table 5: The Experiment Results

Scheme1: Modified Dynamic Programing Algorithm
Scheme2: Traditional Dynamic Programing Algorithm

Recognition Scheme	No. of Database	Top1 (%)	Top2 (%)	Top3 (%)	Speed (ms/char)
Scheme1	12500	87%	92%	95%	40ms
Scheme2	12500	78%	80%	83%	880ms

Average recognition accuracy can be calculated by dividing the number of correctly recognized test patterns with the total number of test patterns. Average recognition speed is calculated by dividing the total recognition time with number of recognized test patterns and its unit is characters per millisecond (ms). Average recognition accuracy is given up to top 3 choices. In this experiment, we used two different recognition schemes, Scheme1 and Scheme2, writing speed free modified DPM was used for final matching stage in Scheme1 whereas traditional DPM was applied in Scheme2. Table 5 gives the experimental results of our proposed system. We noticed that Scheme1 achieved the highest recognition accuracy for every top choice; particularly it achieved 95% accuracy considering Top3 choice. Moreover, the recognition time for Scheme1 is about 22 times lower than that of Scheme2. These facts ensured that proposed hierarchical recognition with modified DPM reduced the inherent computational complexity and speed up the recognition.

Figure 5 represents the snapshot of student's feedback result to correct their error handwriting. As we described in section 2.3, our proposed education system can successfully feedback to the students about their error handwriting using colorful marking technology. We implemented this method to our system by JSON: JavaScript Object Notation technology. In our preliminary experiment, we gathered stroke data from high level native Bengali speakers. In future we will do the experiment for various levels of Bengali learners with different age. Finally, we made a survey among 100 (almost) of native Bengali speakers for the acceptance of our education system. We considered the user's viewpoints about the system effectiveness. The user's comments confirmed that the proposed Bengali handwriting education system can be very helpful to improve the Bengali literacy rate in near future.

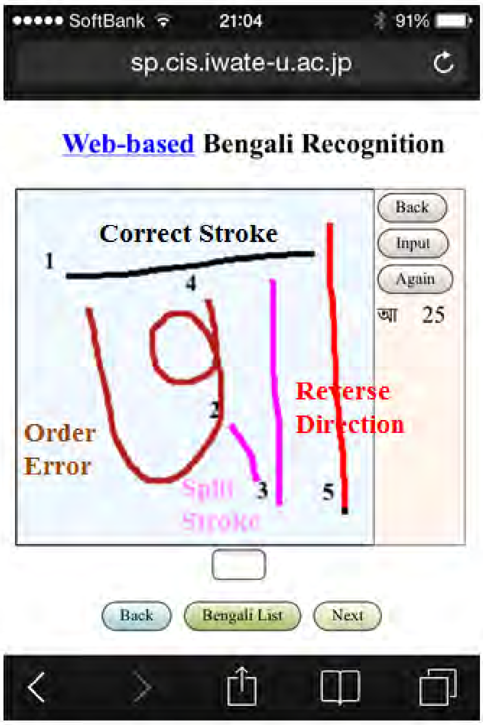


Figure 5: Automatic error detection and colorful feedback to correct student's error handwriting. (Numeric symbols means stroke no. and its starting point)

4 Conclusion and Future Work

In this paper, we have described about the implementation details of web-based intelligent handwriting education system for autonomous learning of Bengali handwritten characters. Here, we developed a web-based (iPhone/Smartphone or computer browser) handwriting client-server interface using JavaServer Page (JSP) technology to improve the Bengali literacy rate. We used 12500 of Bengali isolated character database for system evaluation. Our experimental results showed that the use of hierarchical recognition algorithm together with writing speed free modified DPM improved the recognition accuracy to 95% as well as recognition speed of 40ms. It makes our recognition algorithm adaptable for the application of web-based language learning application. Our automatic error detection methodology ensured the necessary feedback to the students to learn about their handwriting mistake autonomously. In future, we will focus our research on the development of Bengali handwriting education system considering Bengali word learning. Furthermore, partial function of our system can be accessed from the following URL: "http://www.sp.cis.iwate-u.ac.jp/icampus/b/"

Acknowledgements

We would like to thank all the people who supported to accomplish this work.

References

[1] Report on the Bangladesh Literacy Survey, 2010 "http://www.bbs.gov. bd/webtestapplication/userfiles/ image/Survey%20reports/Bangladesh%20Literacy%20Surver%202010f.pdf"

[2] Hu Z., Xu Y. , Huang L., Leung H., (2009) "A Chinese Handwriting System with Automatic Error Detection", Int. J. of Software, Special Issue of Advanced Distance Learning Technologies, vol. 4, Issue 2, pp 101-107.

[3] O. K. Zein and Y. Kermarrec, November 2007, "Description and composition of e-learning", Journal of software, vol.2. no.5, pp.74-83.

[4] K.Hiekata, October 2007 "A framework for design engineering education with workflow-based e-learning system", Journal of software, vol.2. no.4, pp.88-95.

[5] K. T. Tang, K. K. Li and H. Leung. , July 2006 " A web-based Chinese handwriting education system with automatic feedback and analysis" , Lecture Notes in Computer Science, vol. 4181, (Proceedings of the 5th International Conference on Web- based Learning (ICWL 2006)), pp.176-188.

[6] U. Bhattacharya, A. Nigam, Y. S. Rawat, S. K. Parui, 19-21 August, 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.

[7] S. K. Parui, K. Guin, U. Bhattacharya and B. B. Chaudhuri, 2008 "Online Handwritten Bangla Character Recognition Using HMM", Proceedings of 19th ICPR published by IEEE Computer Society Press.

[8] Todd David Rudick: Japanese/English dictionary and character dictionary browser featuring handwriting recognition-based character lookup, ACM "Quest for Java Contest", JavaDict (1997). "http://www.cs.arizona.edu/projects/japan/JavaDict/ "

[9] J. Shin, 2004 "On-line cursive hangul recognition that uses DP matching to detect key segmentation points" Elsevier Pattern Recognition 37, 2101–2112.

[10] L. Prasanth, V. Jagadeesh Babu, R. Raghunath Sharma and G.V. Pradhakara Rao, Sep. 2007 "Elastic Matching of Online Handwritten Tamil and Telugu Scripts Using Local Feature," *Proc. of ICDAR'07*, Vol. II, pp.1028-1032.

[11] N. Joshi, G. Sita, A. G. Ramakrishnan, 2004 "Comparison of Elastic Matching Algorithms for Online Tamil Handwritten Character Recognition" IEEE Computer Society, Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9).