Artificial Tutor For Arabic Handwriting Training

Sherif Abdou , Aly Fahmy , Ibrahim Hosney , Iman Mostafa

Faculty of Computers and Information, Cairo University
sabdou@eng.miami.edu, a.fahmy@fci-cu.edu.eg

1. Abstract

The percentage of people who produce a neat and clear handwriting is declining sharply. The traditional approach for handwriting teaching is to have a dedicated teacher for long hours of handwriting practice. Unfortunately, this is not feasible in many cases. In this paper we introduce an automated tool for teaching Arabic handwriting using tablet PCs and on-line handwriting recognition techniques. This tool can simulate the tasks performed by a human handwriting teacher of detecting the segments of hypothesized writing errors and producing instructive real time feedback to help the student to improve his handwriting quality. The tool consists of two main components, the *guided writing* component and the *free writing* component. In the *guided writing* mode the student is required to write over transparent images for the training examples to limit his hand movements. After the student acquires the basic skills of handwriting he can practice the *free writing* mode where he writes with his own style, as he usually does in his daily handwritings. The first version of the tool was tested in several schools for children with edge ranging 7-12. The results are promising and show that this tool can help students to analyze their own writing and understand how they can improve it.

Introduction

The ratio of persons who produce a neat and clear handwriting is declining sharply. The problem can be traced to the early stages of handwriting learning. Many students struggle to produce neat, expressive written work. Young children want to write well, but are often frustrated by their own lack of coordination and discouraged because it requires so much more effort to please either the teacher or themselves than they thought it would. It is generally recognized that correct stroke making techniques are essential to good hand writing skills (Walter, 1994). These techniques can be successfully acquired only by practicing regularly and for long time periods. To date, methods of training handwriting in school mainly utilize the "blackboard and paper" approach. This consists of blackboard based demonstrations by teacher followed up by paper based examples and exercises for students. Having a dedicated teacher for long hours of practice is not usually available. For example in Egypt the number of the admitted students in the elementary schools stage can reach one million students per year. Schools simply do not have sufficient resources to teach all children the handwriting skills with the required interaction and attention.

To provide students with extra self practice besides the class teachers some educational software tools for teaching handwriting have been developed (Muroya , 2001). The handwriting lessons in these tools usually display some animations for the writing models on the computer screen associated with instructions to help the student to imitate the displayed model. These tools are not interactive and the educational load is on the student to compare his handwriting on paper with the ideal one the computer screen to find his own errors and try to improve them using the try and error approach. Some conventional mouse based personal computers have been used in some instances for teaching handwriting . These systems do not solve all of the problems outlined above. The mouse driven PC programs that teach writing do not reflect the way most students will write such characters in real life. Writing strokes with a mouse on a horizontal surface whilst watching them appearing on a vertical screen does not accurately mirror the process of writing with a pen, pencil or brush onto paper. Further these programs do not indicate the sequence and directions of students 'strokes.

Recently, systems with combined LCD display and digitizers have been available. With these systems children can write with a pen directly on-screen without having to lift up their heads to look at what has been written. With these new hardware tools, we have reached the technological capability needed to build interactive systems to assist in teaching handwriting to children. Although these new systems provide a learning environment very close to the real one for handwriting teaching they still have limited capabilities when compared with human teachers. Over the last two decades many studies using a digitizing tablet have emerged to improve the psychomotor behavior of children (Simner, 1995). These studies focused on several features of the human behavior that relates to the handwriting process such as understanding how a human makes a representation of a form, what strategies are used to coordinate sequences of movements to draw a form, how representation and fine motor system coordination capabilities evolve with age from youth to adulthood, and what kind of training exercises can improve these capabilities. The majority of these studies report results of experiments that highlight the complexity of the human process involved in handwriting. These experiments attempted to highlight some underlying mechanisms between the internal representation of a letter and the neuromotor system involved in the generation of that letter. Some theories formalize the motor control system involved in handwriting, e.g., (Sanguineti, 1998). There are also studies dealing with the efficiency of a training program in learning handwriting where a commonly used exercise is the copy exercise (Marse, 1991)

Most of the currently available tools for handwriting training only give a very rough estimate of the overall quality of the student writing. They measure how close the student writing to some ideal handwriting samples. Though this approach can judge the student handwriting quality and can evaluate his progress after some amount of practice, it can not provide any feedback about the regions of handwriting errors in the student writing. Also it does not provide any information on the types of errors the student have done and how he can avoid them in his next trails. This type of information is very crucial for any useful handwriting training tool. The tool should provide the student the capabilities to analyze his handwriting samples and detect the segments of hypothesized writing problems and produce instructive feedback to help the student to improve his handwriting. The recent techniques of on-line handwriting recognition can provide such type of detailed information (Bahlmann, 2004; Biem, 2006; Makhoul, 1994; Plamondon, 1999).

In this paper we introduce an automated tool that was developed for teaching Arabic handwriting for children using tablet PCs and on-line handwriting recognition technology. The aim of this tool is to help young children to become good writers with fluent movements and a good quality of writing in shorter time frame. This handwriting teaching tool recognizes the student handwriting, detect the segments of hypothesized writing problems and produce instructive feedback to help the student to improve his handwriting. This handwriting teaching tool can increase the effectiveness of classroom teachers in several ways. It can provide positive, independent, individualized, and effective practice for students, and it can give the teacher detailed feedback on each student's progress. It can help free teachers' time by enabling students who need more individualized instruction to work independently with effective learning tools on a computer, while other students in the classroom receive more interaction and attention from the teacher.

In the following sections, section 3 includes the description of our "Handwriting Teaching Tool" and its overall architecture. The tool is composed of two main modules, guided writing module and free writing module. Section 4 describes the guided writing module and the free writing module is described in section 5. Section 6 describes the handwriting data corpus that we used for training the tool models. Section 7 includes the results of several evaluations that we made for the first prototype version of our tool. Section 8 includes the final conclusions and our prospected future work and enhancements for our handwriting training tool.

The Handwriting Teaching Tool

Following the methods used in schools for teaching handwriting, we designed our tool to consist of two main components, the *guided writing* component and the *free writing* component. The *guided writing* component is a preliminary level of education where students write characters or words on a transparent image for the training examples. This approach is equivalent to the method of writing over doted images, which is frequently used in the initial lessons of handwriting teaching. After the student acquires the basic skills of handwriting he can move on to the second level of practice using the *free writing* mode. In this mode students are shown an image or a video animation of a handwritten example, then they are asked to write that example on an empty panel that contains a single line. That panel is similar to writing books used at schools. In the free writing mode the student has more freedom to write with his own style, as he usually does in his daily handwritings, then the tool evaluate his handwriting and give him feedback messages about his errors. Figure (1) includes a flow chart of the main modules of the developed handwriting tool.

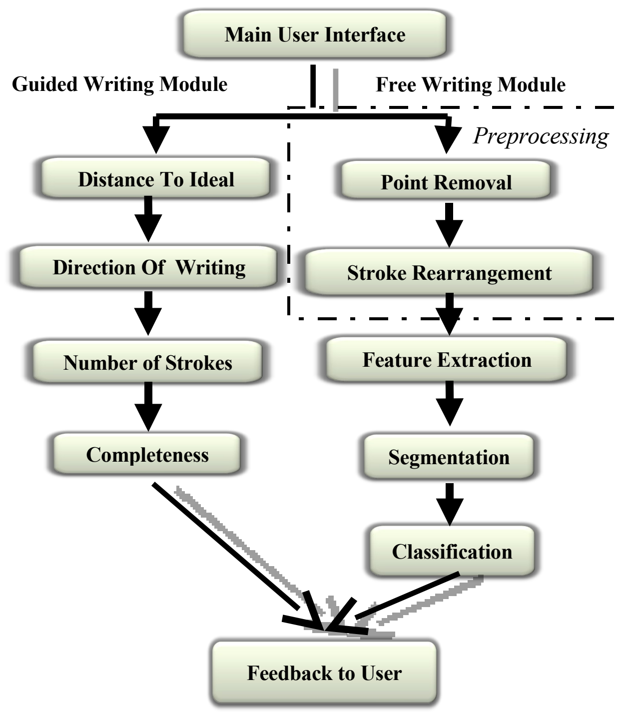

Figure (1): Handwriting Teaching Tool Flow chart

The following sections describe the detailed implementation of each one of these modules.

Guided writing

The traditional handwriting training at schools is usually a gradual process. It starts with simple exercises that ask the student to connect the dots of templates then proceed to more advanced exercises that allow more freedom in writing. In our tool we adopted the same gradual handwriting teaching approach. Beginner users start with a guided writing mode. In this mode the tool displays a transparent image for an ideally handwritten training example. The user is required to write over this transparent image. We decided to use transparent images instead of doted images for two reasons. First they are more convenient for a tablet PC based handwriting training than the doted images which are more suitable for paper and pencil handwriting practice. Second the student will be trained to write in a natural continuous movements

rather than just trying to connect the points. Figure (2) shows an example for the guided writing training.
On the transparent image the tool sets specific control points. These points aren't visible to the user but they are used for tracking the user handwriting. The tool evaluates the user performance using several measures. Each one of these measures uses a specific criterion to evaluate one of the properties that affect the quality of the user handwriting. The following sections include detailed description for those measures with illustrative examples.

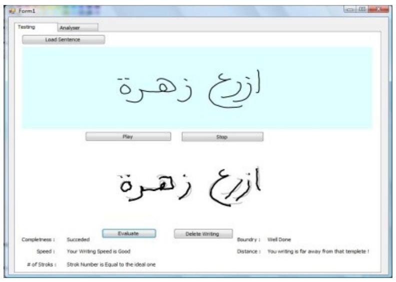

Figure (2): Example for the guided writing training

Distance

Some students write not exactly over the ideal sample. That means they can't follow an ideal sample and that they don't know the correct shape of characters. The distance measure mainly calculates how much the student writing is close or far from the ideal sample. This measure is calculated by measuring distance between the written text and the control points that lay on the ideal sample. If this distance is greater than a predefined threshold for a segment we consider that the student didn't manage to follow the template for that segment and we display that segment in a different color. Figure (3) shows a sample for using that measure.

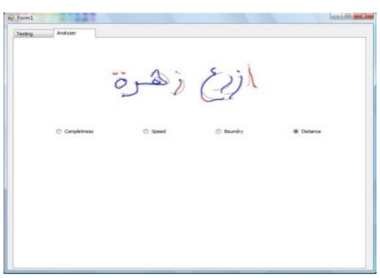

Figure 3: sample for using the distance measure

Number Of Stroke

Children tend to write in segmented style with large number of strokes. Figure (4) shows a sample for that segmented writing where the word "مدير" was written in 6 strokes instead of 4 as it should be. The reason for that phenomenon is that children tend to think while they write which results in interruptions their handwriting process. The increased number of strokes raises the possibility of making errors. Usually handwriting teachers encourage students to write words in paws, the ideal word parts, with each paw written in a single stroke if possible. Some exceptions are permitted for complex paws. In our tool we use the *Number Of Strokes* measure to detect segmented writings. This measure is calculated by counting the number of strokes in each paw. If it exceeds the expected number the user gets negative feedback.

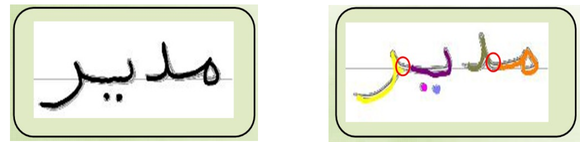

Figure 4: Sample of segmented writing

Direction

When students start to learn handwriting of complete words, if they have no guidance, they will develop their own way for the directions they follow. Sometimes these directions are odd and can complicate the handwriting process. Handwriting teachers usually advice their students to follow some ideal directions that will help them to do smooth and easy handwriting. In our tool we simulate that guidance by displaying an animation for the ideal handwriting directions for every training example. The student can play these animations whenever he wants. The *Directions* measure is used to check if the student followed the ideal writing direction or not. This measure is implemented in our tool by setting an order for the control points of the word. The student should pass over those points with the predefined order. If the student makes unexpected jumps he will receive low score with a feedback message that instruct him to follow the ideal directions. The segment where the student violated the ideal directions in his handwriting will be highlighted with different color as shown in figure (5).

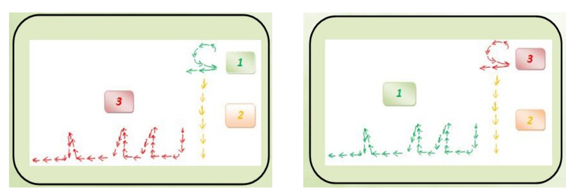

Figure 5: Sample of direction errors

Completeness

This measure is used to check whether the user has wrote the complete example or not, by checking which control points the user have visited and which ones he didn't. If the ratio of visited points to the total number of control points is over a specific threshold then the user writing is considered complete. Figure (6) shows an example for incomplete word.

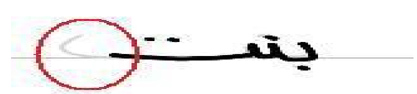

Figure 6: the character "taa' " isn't complete

Free Writing

After the student acquires the basic skills of handwriting using the guided writing mode he should move on to the second level of practice using the free writing mode. In this mode students can display an animation for the ideal handwriting of training examples. Then they can practice handwriting on an empty panel that contains a single line similar to the writing handbooks used at school. Before analyzing the user input for checking handwriting errors it is preprocessed. In this preprocessing step the points are removed to reduce the number of classes and the strokes are reordered to eliminate the delayed strokes effect as will be explained later. In the error analysis phase the user handwriting is segmented to the characters level, then these segmented characters passes through group of classifiers. Each one of these classifiers checks for the existence of a specific type of handwriting errors in the user handwriting. Figure (7) shows the processing steps for a free handwriting example.

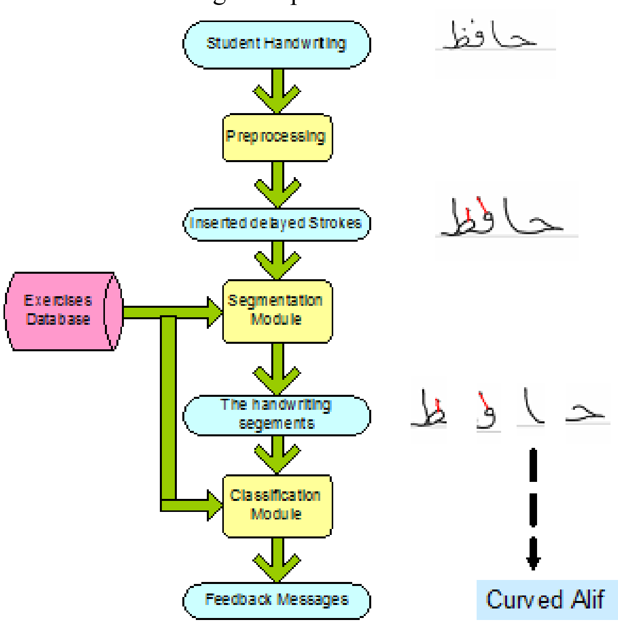

Figure 7: The processing of a free writing example

Preprocessing phase

In this phase characters are modified before they are delivered to the segmentation phase

a- Point removal

The Arabic language has groups of characters that only differentiate by the number and position of dots. For example in figure (8), we can easily notice that the three characters “ب” ‘baa’, ‘taa’ “ت”, and ‘thaa’ “ث” have the same body but different points so eliminating these points leads to the same character. In the preprocessing phase they have their points removed, and they are all assigned to the same class. Such process has significantly decreased number of classes to be recognized, leading to faster computation and better performance. Such process is developed using a specific recognizer to identify the points places. Such recognizer is easily trained with different shapes of dots as they are limited in Arabic script ranging between one and three dots. , and the three dots can be connected or separated as shown in figure (8). The detected points are stored for later processing to determine the handwriting errors related to the points.

Figure 8: Different shapes of dots

b - Rearrangement of strokes to solve the delayed stroke problem

Some characters in the Arabic language, and other languages, are written using delayed strokes. These cases happens when the writer moves back to complete some missing parts of a previously written character in a word. In the Arabic alphabet 20 out of 33 characters has delayed strokes. In some cases the delayed strokes is the only clue to differentiate between several characters. When we analyzed the children writing we found that they tend to use much more delayed strokes than the standard ones, in many cases they return to complete parts of the characters that they previously wrote or even rewrite several copies over the previously wrote characters. These features of children writing complicates the segmentation problem as the component strokes of a character will come scattered and interleaved with strokes from other characters. For adults we can enforce sort of handwriting restrictions, such as writing word parts in single strokes and forbidding back movements, to reduce the cases of delayed strokes. For children such kind of restrictions would be very hard and for sure they will not be able to follow them. We investigated some techniques proposed in literature for handling the delayed strokes but they didn’t provide an effective solution with accepted accuracy. We developed a new algorithm for handling the delayed stokes, more details can be found at (Abdou, 2009).

Feature Extraction

In our tool we used the chain code features to represent the online handwriting, Figure (9) shows an sixteen directions chain code. To consider longer directional segments we added two more features which are the difference between two successive chain codes which is named the “Delta” feature. The other one is the difference between two successive Deltas which is named “Double Delta|. This means we are modeling the directions of the previous 4 points in the feature vector for each online point.

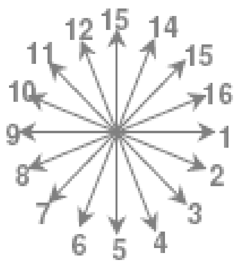

Figure 9: 16-directions chain code

Segmentation phase

Online handwriting recognition of Arabic script is a difficult problem since it is naturally both cursive and unconstrained. Arabic is written connected from right to left. Most letters are written in four different letter shapes depending on their position in a word. The analysis of Arabic script is further complicated in comparison to Latin script due to obligatory dots/stokes that are placed above or below most letters. The Hidden Markov Model (HMM) technique provide solutions for most of the difficulties inherent in recognizing Arabic script including letter connectivity, position-dependent letter shaping, and delayed strokes. The Hidden Markov Model is a finite set of states, each of which is associated with a (generally multidimensional) probability distribution. Transitions among the states are governed by a set of probabilities called transition probabilities. Figure (10) shows a sample HMM model. In a particular state an outcome or observation can be generated, according to the associated probability distribution. It is only the outcome, not the state visible to an external observer and therefore states are ``hidden'' to the outside; hence the name is “Hidden” Markov Model.

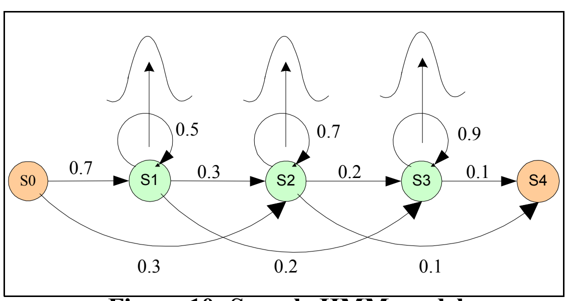

Figure 10: Sample HMM model

We used the HMM in our tool in the alignment mode to find the optimum segmentation points of a word to its composing characters. For t example the word “ولد” “boy” is composed of three characters “د ل و”. In our tool we report the handwriting errors for each one of these characters separately, so we need to locate the segment of each character before running the error analysis on it. The HMM is a flexible tool that can search all the possible segmentation hypotheses for a word to find the optimum one, the one with highest match with the training data that the model has seen before.

Classifiers

This is the main component in our handwriting teaching tool. This component is responsible for analyzing the student handwriting and giving him feedback on his performance and detailed messages on the types of errors that he has done and some guidance instructions to help him avoid these errors in his future trials.

We should keep in mind that at this stage:

- Classification is performed on single characters received from HMM.
- There are No dots
- Strokes are rearranged to avoid delayed strokes.

To prepare for building our classifiers we have done some analysis on the collected data. This included:

- *Manual Segmentation* **:** To analyze single characters.
- *Sample analysis***:** To identify classes of errors.
- *Assigning Errors***:** Manually assign errors for training purposes, with consultation from handwriting experts and personal knowledge.

After we got enough knowledge about the type and rate of handwriting errors in the children handwriting, we found that some errors can be detected using simple geometric rules. Some other errors required the design of more intelligent classifiers. We run some initial experiments using neural networks but we realized that our data is not enough to build robust classifiers. So we decided to use “Template Based” classifiers. This classifier does not require training data and can be tuned to be robust in specific areas of the space, where the writing errors are located. The following two sections describe the classifiers currently integrated in our tool

I. Geometric rules

Examples of the rules that we used to detect writing errors:

- Slope for characters which must be vertical or semi-vertical Such as: “alif’ ”
- Height of some characters to see if it is suitable or not proportional with the word containing that character such as: “alif’ ” in the middle position
- Intersection for character that may have a circular shape to determine whether the shape is closed or not such as: “waaw”.
- Determine equality of 2 parts in same stroke in same character such as the two sides of “daal”

II. Template matching

In this approach recognition of errors is performed according to a stored prototypical version of each letter (called a template) and compare input letter with each template, taking the closest match using Dynamic Time Wrapping (DTW). Figure (11) shows an example of the template based classifier. We provide several templates for each error type so the tool can recognize the several ways for committing that type of error. The more the templates we provide the more robust the tool will be, but this will increase the processing time. Also it is better to select the templates from different users. Also the ideal templates should be provided from several persons to accommodate the natural differences in handwriting. The template that gets the minimum DTW score make the tool decision and feedback message. We have done some modifications to the standard DTW distance to match our application as explained in the following section.

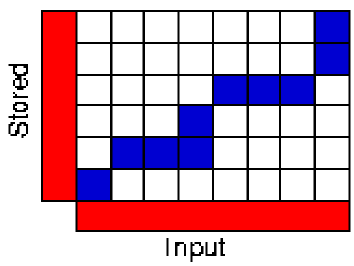

Figure 11 :Dynamic Time Warping process

DTW "Dynamic Time Warping"

Dynamic time warping (DTW) is a technique that calculates the optimal alignment between two time series if one time series may be "warped" non-linearly by stretching or shrinking it along its time axis. This warping between two time series can then be used to find i regions between the two time series or to determine the similarity between the two time series. Dynamic time warping is often used in speech recognition to determine if two waveforms represent the same spoken phrase. From our data analysis we found that the handwriting error is localized in small parts of the template. So this part should receive the highest attention while calculating the warping score. To add this effect in our tool we added two markers, the *Error_Segement_Start* and the *Error_Segement_End* for each error template. These two markers are used to locate the segment of the template that will be included in the DTW score.

The Data Corpus

One of the valuable outputs of this project is the handwriting data corpus that serves the target of building educational tools for handwriting. This data corpus include three types of collected samples for

1- Separate letters.
2- Single words.
3- Sentences.

The list of words and sentences were selected to be simple enough for children. We wanted to make the child concentrate on the handwriting practice and not spend much effort in understanding the meaning of the training examples. The data include samples that represent the left and right handed subjects. Also the data include balanced numbers of male and female samples. The data was collected for the two styles of the tool exercises, the doted templates and free writing. The data was collected from 9 schools and from 340 student. It is known from classical studies of human behavior that the process of learning handwriting skills begins around age five and finishes approximately at age fifteen. In this project we collected data from students in the age range 7-10 as we expect this would be the optimum range for improving handwriting skills. The collected data size is around 20,000 samples that included 100,000 characters. Table (1) include the details of the collected data corpus.

	No. of children	Age Ranges	Samples	Right Handed students	Left Handed students
Male	197	7-10	70235	170	27
Female	144	7-9	32517	110	34
Total	341	7-10	102752	280	61

Table (1): Details of the collected handwriting corpus

A small portion of the corpus, around 10%, were selected to be manually segmented and annotated for the purpose of models initialization. In this process each word is divided to its composing characters. If the character was written using multiple strokes they are grouped together and attached with the character label. We developed a special tool for data segmentation and annotation. This tool allows the user to do the segmentation by hand using the touch screen pen which accelerates the segmentation process.

Results

We run several internal tests with ourselves to check the functionality of the tool components and to make sure they perform as excepted. For the formal test we selected five children from an elementary school, there edges were in the range 6-11 years old. We created a test form that included 7 columns which are the word under test, the remove points result, the HMM segmentation result and the classifier result. We tested the accuracy of the main three components of the free writing tool: Remove point, Segmentation and the Errors Classifiers. Table 2 includes this test results.

Remove points Accuracy	**74%**
Segmentation Accuracy	**56%**
Classifier Accuracy	
Correct Feedback Message	**79%**
False Feedback Message	**21%**

Table (2): The free writing tool test results

We noticed that the most significant improvement was for the 6-7 years old children. They managed to copy the ideal writings with perfect performance. They required around three hours of practice to reach that level with no teacher guidance. The older children had harder time trying to change there writing style and the final result didn't show much improvement compared to the younger ones. We plan to expand` this test on a larger pool of students to get more generalized results and receive more reach evaluation of the tool from its real users.

Conclusion

Handwriting does not have to be a battleground. The handwriting teacher must be patient, choose reasonable objectives, and stand firm. Fine motor skills develop more slowly, then gross motor skills. By targeting specific and narrow objectives, praising efforts that are well-done as well as pointing out errors to be corrected, and scheduling regular, supervised practice, progress can be made much more rapidly than if students are left on their own to complete handwriting workbooks. Through this project we were able to explore, and also enjoy, an important problem which is teaching handwriting for kids. The Arabic language had its own challenges of cursive writing, the many dots and delayed strokes. Also the recognition of the handwriting of children is much more challenging than adult handwriting due to the increased irregularities, the lossy control of the pen movements and the fragmented writings. We implemented some standard techniques for Arabic handwriting recognition and also developed new techniques that can handle the challenging handwriting of children. The availability of the hardware, the tablets PC, allowed us to test the components of the application in real usage scenarios which gave us some confidence of our work. We almost have achieved the promised objectives of the project. We delivered a working version of the application that includes all the proposed functionalities. The tool works with reasonable accuracy considering it the first version. Also considering this application is a new one and there are no similar products in the market that we can compare with.

Our future plan is to finish the field tests and integrate the test results as enhancements in the application. We plan to use other segmentation and classification algorithms that may enhance the accuracy. Improve HMM accuracy by increasing its training data. Extend the application to include non-native Arabic students. Include sentences in the application. Increase the types of handled handwriting errors. Extend application's capability to be used on mobile phones to recognize writing. All the used approaches and implemented techniques in our tool are language independent and can easily be applied to any other language. The tool was developed for the Arabic language but can easily be ported to other languages since all the language related information are stored in external databases.

Acknowledgements

We express our gratitude's to Cairo Microsoft Innovation Center(CIMIC) team for the funding of this project and keen efforts to make this application a successes story. Also the students of the graduation project of year 2008 at the Faculty of Computers and Information at Cairo University (FCI) for sharing in the development of that application and in the data collection and annotation phase.

References

Abdou, S.M, Fahmy, A. A. (2009) "Unconstrained Arabic Online Handwriting Recognition", pending patent application.

Bahlmann, C. (2004). The Writer Independent Online Handwriting Recognition System frog on hand and Cluster Generative Statistical Dynamic Time Warping . IEEE Transactions on Pattern Analysis and Machine, (Vol. 26, No. 3) pp. 299-310

Barbe, W. B., Virginia H. Lucas and Thomas M. Waslylyk, (1994), "Basic Skills for Effective Communication", Morgan Kaufman Publishers.

Biem, A. , (2006) . Minimum classification error training for online handwriting recognition", IEEE Transactions on Pattern Analysis and Machine, (Vol. 28, No. 7) pp. 1041- 1051

Greer, T. , Lockman, J. J. (1998). Using Writing Instruments: in variances in Young Children and Adults, Child Devolopment, vol. 69, no. 4, pp. 88-902.

Makhoul, J., Starner, T., Shartz, R. and G.Lou, (1994). On-Line Cursive HandWriting Recognition Using Speech Recognition Models, Proc. IEEE Int'l Conf. Acoustics, Speech ,and Signal Proccesing, pp. v125-v128, Adelaide, Australia.

Marse, J. F. et al, (1991). Handwriting Training: Computer-Aided Tools for Premedial Teaching, Development of Graphic Skills. J. Wann, A.M. Wing, N. Suvik, eds., pp. 249-258, Academic Press.

Muroya et al, (2001) U. S. Patent No. 5,040,222

Plamondon, R., Lopresti, D., Schomaker, L.R.B. and R. Srihari(1999), On- Line Handwriting Recognition, Encyclopedin of Electrical and Electronics Eng., J.G. Webster, ed., vol. 15, pp 123-146, New York: Wily, 1999.

Simner, M.L. (1995). Basic and applied issues in handwriting and drawing research, Proceedings Seventh Biennial Conf. Int'l Graphonomics Soc., , ed., London, Ontario, Canada.

Sanguineti, V. , Frisone, F. , Bruni, S. ,and P. Morasso, (1998). Can Non-Linear Muscle Dynamics Explain the Smoothness of Handwriting Movemts?. Acta Psychological, vol. 100, no. 1-2, pp. 217-227.