A Tool for Arabic Handwriting Training

Sherif Abdou , Aly Fahmy

Faculty of Computers and Information, Cairo University
sh.ma.abdou@gmail.com, aly.fahmy@gmail.com

Ahmed Elgammal

Computer Sciece Department, Rutgers University
elgammal@cs.rutgers.edu

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.

1. 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. It is generally recognized that correct stroke making techniques are essential to good hand writing skills [9]. 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 that consists of blackboard based demonstrations by teacher followed up by paper based examples and exercises for students. The forming of each character is a dynamic process and students need to become proficient in both the process and the final result. The process is only visible each time the teacher is actually writing on the blackboard. This process is not present whilst students are concentrating on their own work. Therefore the students are practicing the sequence, order and direction of the strokes of the character (s) based on what they can remember from watching the teacher's action. This is not reliable as the students may not remember all the process steps for the strokes. Moreover, in a typical class room when writing or drawing the teacher's body will be between the students and the blackboard, and therefore vision of some students will be obscured at certain stages. When teachers come to assess handwritten work, they only see the final result, not the process that was used. It is impossible to tell whether or not the correct stroke making techniques have been applied. For example, the teacher will not be able to tell whether a pencil was lifted too many times when forming the character, or whether the character was stroked in an incorrect direction, or whether the writing was fluent or jerky. Another important issue is that 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 some educational software tools for teaching handwriting have been developed [7]. 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 on the computer screen to find his errors and try to improve them using the try and error approach.

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 [4]. Most of the currently available tools for handwriting training only give a very rough estimate of the overall quality of the student writing [6]. 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.

In this paper we introduce a tool that provides solutions for the problems outlined above. The tool was developed for teaching Arabic handwriting for children using tablet PCs and on-line handwriting recognition technology [2] [3] [5] [8]. 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.

In the following sections, section 2 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 3 describes the guided writing module and the free writing module is described in section 4. Section 5 describes the handwriting data corpus that we used for training the tool models. Section 6 includes the results of several evaluations that we made for the first prototype version of our tool. Section 7 includes the final conclusions and our prospected future work and enhancements for our handwriting training tool.

2. 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. . Figure (1) includes a flow chart of the main modules of the handwriting tool.

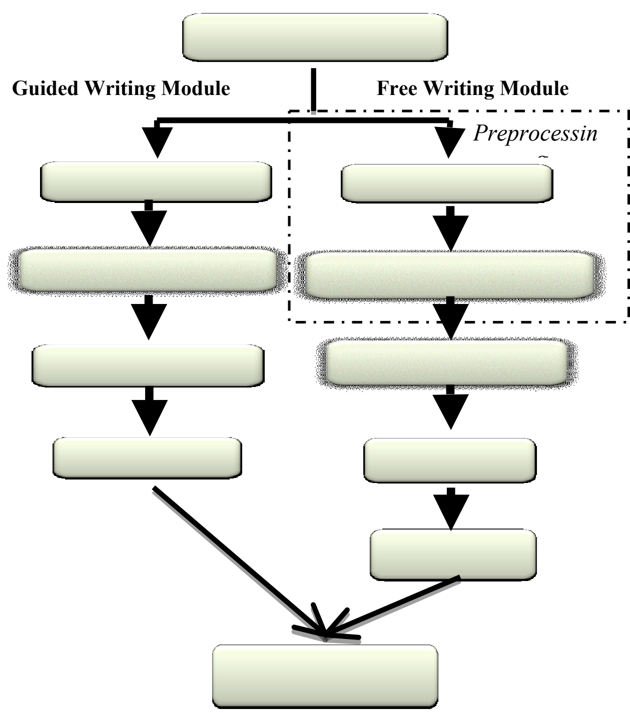

Figure (1): Handwriting Teaching Tool Flow chart

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

3. Guided writing

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. 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. Figure (2) shows an example for the guided writing training. The following sections include detailed description for those measures with illustrative examples.

Figure (2): Example for the guided writing training

3.1 Distance

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.

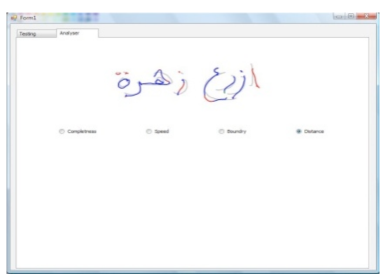

Figure 3: sample for using the distance measure

1.2 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 "____" 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 interrupt 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.

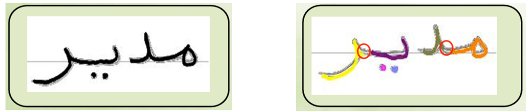

Figure 4: Sample of segmented writing

1.3 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).

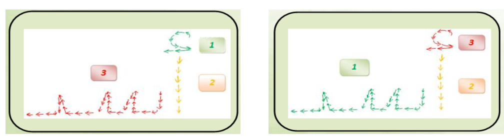

Figure 5: Sample of direction errors

1.4 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

4. 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 illustrated on a handwriting example.

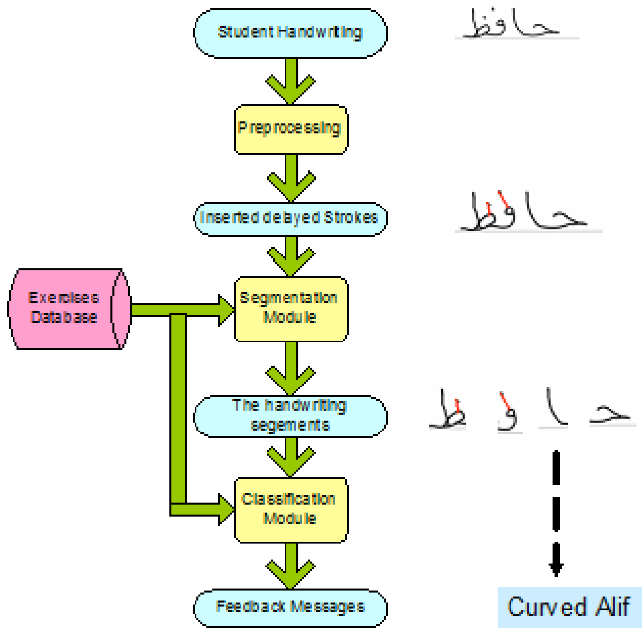

Figure 7: The processing of a free writing example

4.1 Pre-processing 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 “_” ‘baa’, ‘taa’ “_”, and ‘thaa’ “_” 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 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. 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 are scattered and interleaved with strokes from other characters. For adults we can enforce some 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 [1].

1.2 Feature Extraction

In our tool we used the chain code features to represent the online handwriting. 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.

1.3 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 (9) 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.

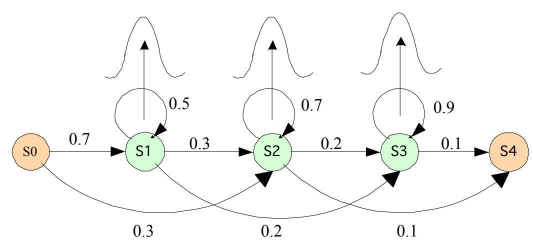

Figure 9: Sample HMM model

We use the HMM in our tool in the alignment mode to find the optimum segmentation points of a word to its composing characters. For the example in figure 10 the word "___" "boy" is composed of three characters "_ _ _". 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, with highest match with the training data that the model has seen before.

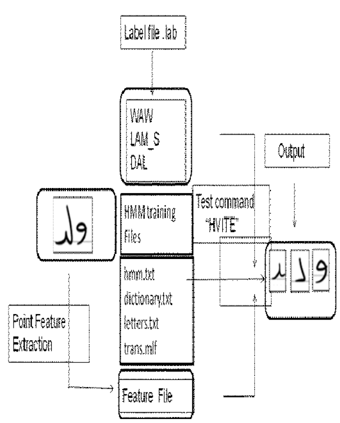

Figure 10: Segmentation process of the word "___" (Boy)

1.4 Classifiers

This is the main component in our handwriting teaching tool. It is responsible for analyzing the student handwriting and giving him feedback on his performance. We collected a large data set of children handwriting samples from all the target grades. With help of some handwriting experts we made an analysis for this dataset to get 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 (NN) and support vector machines (SVM) 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 in proportional with the word containing that character such as: "alif' " in the middle position.
- Closed loops intersections for character that may have a circular shape to determine whether the shape is closed or not such as: "waaw".
- Equality of two parts in some characters such as the two sides of " daal ".

II. **Template matching**

In this approach the student input handwriting is matched against sample ideal writing templates and also against other templates that are representatives for the possible writing errors of that character. The matching score is calculated using Dynamic Time Wrapping (DTW). Figure (11) shows an example of the template based classifier. We provide several templates for each handwriting 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 derives the tool decision and feedback message.

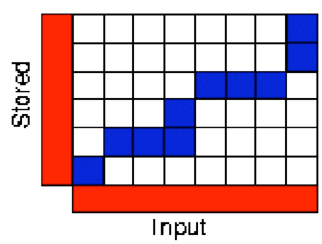

Figure 11 :Dynamic Time Warping process

We have done some modifications to the standard DTW distance to match our application. Dynamic time warping (DTW) is a technique that calculates the optimal alignment between two time series. 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.

5. The Training and Evaluation Data Corpus

This data corpus included three types of collected samples for Separate letters, Single words and 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 included samples that represent the left and right handed subjects. Also the data included 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.

Table (1): Details of the collected handwriting 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

A small portion of the corpus, around 10% , were selected to be manually segmented and annotated for the purpose of HMM 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 accelerated the segmentation process. Figure.12 shows a screen shot for the data annotation tool.

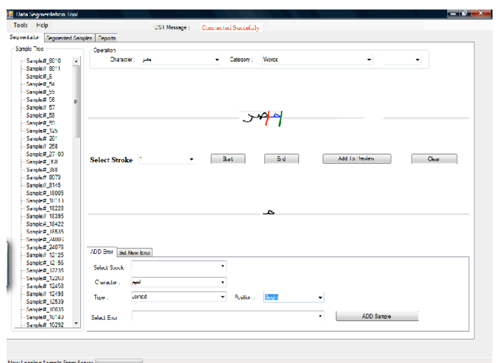

Figure 12: The data annotation tool

6. 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 fifty 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 pre-processing 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.

Table (2): The free writing tool test results

Preprocessing Accuracy	94%
Segmentation Accuracy	66%
Classifier Accuracy	
Correct Feedback Message	79%
False Feedback Message	21%

From the results in table 2 we see that the preprocessing module achieved good performance and managed to detect the points and reorder most of the strokes correctly. The segmentation module still need more enhancement. Its performance has large variance as some words were segmented with accuracy more than 90% while others are still poorly segmented with accuracy under 30%. Figure 13 displays the segmentation result of some words from our database. Our inspection for the segmentaion results has shown that most of the errors resulted from the chratcers that have a more complicate shape. We plan to increase the training data for those characters and try HMM models adaptaion techniques such as MLLR and MAP to capture the inter-person variability and boost the performance for those characters.

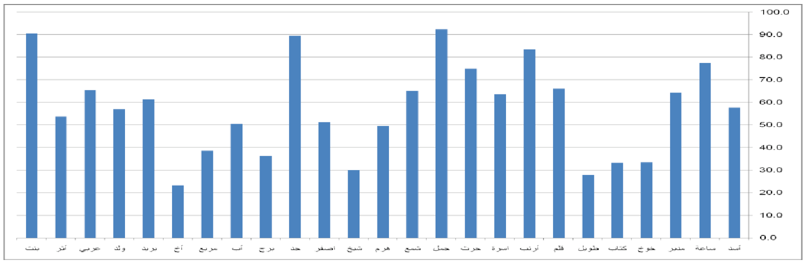

Figure 13: The segmentation result of some words

For the words that were correctly segmented we measured the accuracy of the tool feedback messages. The correct message means the acceptance of good written charaecters or the rejection of the badly writtin charaecters. The other cases, false alarms and false acceptance, are counted as incorrect feedback from the tool. The tool had variable performance for the different charaecters. As shown in Figure 14, the tool accuracy ranged from 62%-90% with average accuracy 79%. This means for each 4 out of 5 charecters the tool mangaed to provide correct feedback which can be considered a promissing performance for the first version of the tool. An inspection of the errors showed some new writing problems for some charecters that didn't exsist in the training data. Also we need to increase the number of representive templetes for some charecters to capture their different writing shapes.

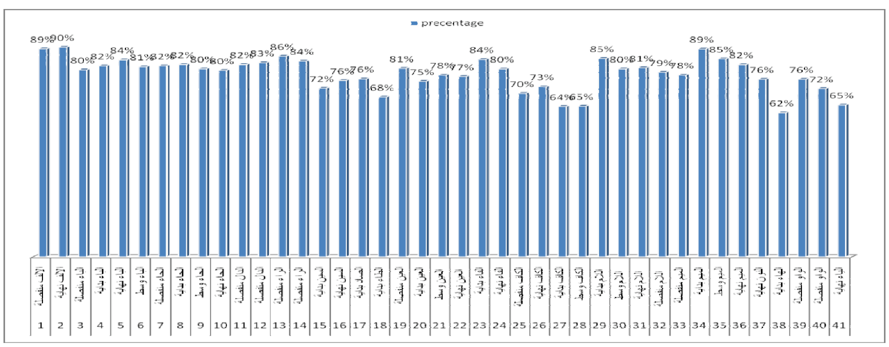

Figure 14: The classification accuracy for some characters

We also 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. Figure 15 shows the improvement rate for the test students.

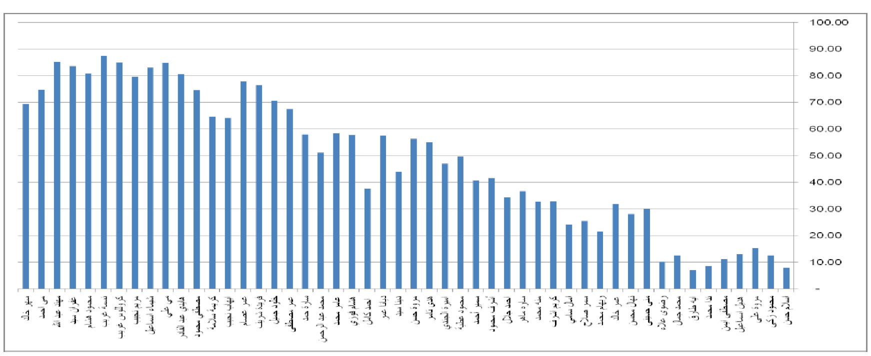

Figure 15: Students improvement rates

7. Conclusion

Handwriting does not have to be a battleground. 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. We tested the components of the application in real usage scenarios. The tool works with reasonable accuracy considering it is 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.

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. 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.

In our future work we plan to use other segmentation and classification algorithms that may enhance the accuracy. Extend the application to include non-native Arabic students. Increase the types of handled handwriting errors. Extend application's capability to be used on mobile phones enabled with handwriting input.

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 all project team 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

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

[2] 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

[3] 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

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] Walter B. Barbe, Virginia H. Lucas and Thomas M. Waslylyk , "Basic Skills for Effective Communication", copyright 1994