Using Machine Learning and Image Processing for Character Recognition: An Application for Teaching Handwriting

Steven Sybenga[1] and Yves Rybarczyk[2,3]
[1]University of Twente, Enschede, The Netherlands
[2]MIST / Universidad Tecnológica Indoamérica, Quito, Ecuador
[3]CTS-UNINOVA / Universidade Nova de Lisboa, Lisboa, Portugal
y.rybarczyk@fct.unl.pt

Abstract

The ultimate advances in information technologies lead to new possibilities in terms of educational approaches. One of them, which are the focus of this paper, is to improve the handwriting skills of children by use of a tablet. The Android based application presented here promotes an autonomous learning without losing the input, expertise and preferences of the teacher(s). The app is based on the use of machine learning classifiers and image processing techniques to compare the written character with the character created or uploaded by the teacher and to grade the performance accordingly.

keywords: Apps, character recognition, handwriting, machine learning, image processing, self-learning

1 Introduction

The development of the information technologies (IT) has introduced new possibilities in teaching and learning, varying from e-learning and virtual classrooms to applications replacing (work) books in classrooms [1]. Many of these applications are in connection with the growth of serious games [2]. Basic math and reading exercises were quickly and in fast amounts available since the introduction of the first consumer tablets. Quality writing exercises are (due to the complexity) harder to find, especially those where the teachers' expertise is not lost.

A key element in learning how to write a character is repetition. Only this way a child can automate the correct muscle moments [3]–[6]. To learn such a new sensorimotor skill a teacher can hold the students hand but this teaching method is undesirable in full classrooms since this is usually time-consuming.

Using a tablet will change the focus of the writing exercise as we know it. From the age of 4, children will have to learn multiple skills at the same time. For them to be able to write the handwritten characters, they will have to learn how to hold a pen [6], [7]. Delays in mastering a correct grip or too much focus on it can harm the learning curve on typography and understanding of writing characters correctly [8].

The application reported in this paper is based on these assumptions, giving the student a way to learn to write correctly but keeping the teacher in control of how the characters should be written and evaluating the result of the student but without the necessity to be present. This way the teacher is able to focus on other aspects of teaching during the exercise or the student can do it as extra assignment at home.

The following sections will describe related work, followed by sections explaining the three faces of designing the app. Finally, preliminary tests on children and feedback from teachers are presented. The results are discussed in terms of the characteristics an app should have to be a useful tool for learning how to write.

2 Related Work

IWA (Improving Writing Ability) is an example of an application trying to keep some autonomy for both teacher and students. However, this application is built for the Tablet PCs (e.g. for the LG P100 Tablet PC) and not the current generation of tablets (e.g. Android and IOS based tablets) [4].

Research on tablet-based handwriting for Arabic words has been done by Aizan et al. [9]**.** They used an Android based application on a Samsung® note to test the potential of tablet-based handwriting evaluation and compared it with paper-based learning. Also, a comparison between using a finger and a stylus to write the words was analyzed. Unfortunately, their evaluation on how correct the children wrote the words was not automated yet but done afterwards using two digital cameras.

Some commercial applications using the current generation of tablets are for example "LetterSchool"[1] and "Digitaal Schrijfschrift" (Dutch)[2]. Although no data is available on how

[1] http://www.letterschool.com/
[2] https://itunes.apple.com/en/app/digitaal-schrijfschrift-ik/id495152971

well those apps improve the writing skills of the student, they seem to be engaging and fun. The downside of these apps is the fact that the designers preprogrammed the characters and teachers lack the ability to evaluate the exercise and automatic scoring.

In our approach, in order to eliminate the designers' preprogrammed characters and give this freedom to the teacher, the app will have the option to create or import character models. The app is built for Android tablets and will have the ability to score the character automatically.

3 Method

In the initial phase of the development, machine learning was used to research what the best approach in comparing the characters would be. Also, the optimal parameters for the image processing were found in this way, assuming the better the characters are classified, the better used approach and parameters would be to score the characters later.

An app to record characters was developed and used to collect characters written by teachers. These characters are tested the same way and used as model for the children how to write the characters. Based on the conclusion of the two phases an app is designed, developed and finally tested in the last phase. A visual representation of these phases is visible in Figure 1.

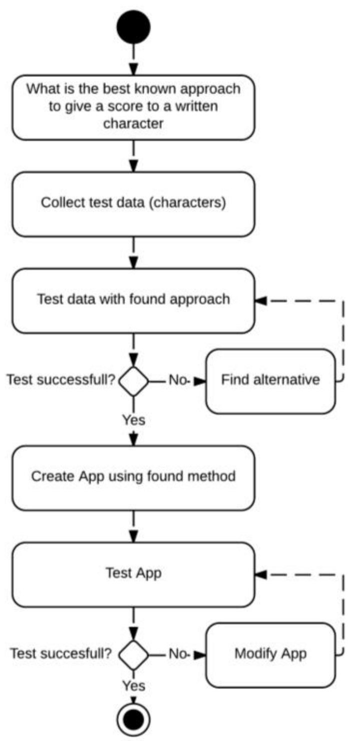

Figure 1: Method

4 Orientation

First, the MNIST (Mixed National Institute of Standards and Technology) handwritten number dataset[3] was tested. Apart from doing this 'manually' with Matlab, a tool called Weka[4] is used to test different classifiers in a quick and efficient way. The dataset is already preprocessed and transformed into a weka (.arff) file [10].

From all the classifiers tested using 10-fold cross-validation with Weka, a Decision Tree, a Nearest Neighbor and a Bayes classifier were selected for their relatively high success rates. The J48 Decision Tree classifier [11] classifies 82.5% of the numbers correctly. The Naïve Bayes classifier [12] was 84.3% of time correct and a K-Nearest Neighbor classifier (Lazy IBK) [13] manages to classify 96.9% of the characters correctly.

5 Recording Characters

A correctly detected character does not imply the character is in fact written correctly and in a way children learn to write. In fact, the characters (10 numbers) in the MNIST database are not even comparable with the (26 alphabetic) characters in the learning books of a child. Therefore, the conclusions of the previous analysis give a direction but do not tell what classifier or algorithm to use for the teaching application.

5.1 The Recorder App

To be able to have characters more comparable to the characters in the final application, a 'recorder app' has been developed. This app will have the same methods to record a written character as the final app will have. Teachers are asked to use this app to generate a set of handwritten characters and to give an idea of how writing on a tablet is perceived.

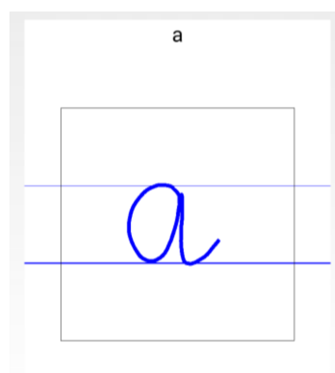

Figure 2: Drawing area

[3] http://yann.lecun.com/exdb/mnist/
[4] http://www.cs.waikato.ac.nz/ml/weka/index.html

On the drawing screen of the app are two visible lines representing the baseline and the small line above it, like in children's exercise book (Figure 2). Above this drawing area, a character is visible. This randomly generated character has to be written down on the drawing canvas as good as possible, since these characters will be used as reference to score the characters later and are assumed to be the 'perfect' way to write the character. The teacher can choose to redraw a character if he or she is not satisfied or press next. When all characters of the alphabet have passed, a message will inform the user a new series has started. Afterwards, a Weka (.arff) file can be created with the app and the raw images of the characters can be exported to a SD card.

5.2 Results

Unfortunately, these results were not as good when the same classifiers were used for a subset of the dataset with characters recorded by teachers (Table 1). With the highest success rate of almost 75% on characters that theoretically look alike, since this is how the teachers want the children to learn them, this is not a promising approach and an alternative method has to be found.

Table 1: Successful classification of recorded characters for different classifiers using a 10-fold cross-validation

Classifier	Subset of characters
J48	38.6%
Naïve Bayes	73.9%
Lazy IBK	61.8%

5.3 Alternative Method

For every test round, one image is selected as test image and all others are used to create 26 different images called 'baselines'. This baseline images are combinations of images of the same character. After processing the pictures based on the method developed in [10], the baseline images are blurred and multiplied to create a set of images where the ideal line of the character has a fully black color and the further away from this line the lighter the color will be. All baseline images will be inverted resulting in an image where the ideal line is fully white and the bigger the distance of this line, the darker the pixels. A dot multiplication between these negative baseline images and a negative of the trial will make all the pixels in the trial white if they are on this ideal line or darker if they are further away (see Figure 3 and Figure 6). The trial (test image) is classified as the character of the baseline which generated a picture with the whitest pixels left.

The parameters for resizing (to reduce the time it takes), blurring (to make the ideal line wider) and multiplication (to make it thicker) are automatically adjusted to find the ideal combination resulting in the highest classification rate. By performing this test for all images, a 90% success rate was accomplished.

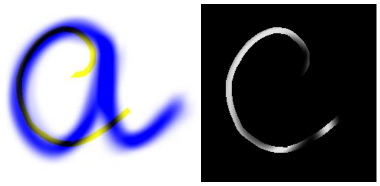

Figure 3: A trial character 'c' multiplied with baseline of 'a'

6 App Development

The main screen of the app displays two lists, one with the classes and one with names of the students in the selected class. After selecting the class and student, the user selects the type of exercise, the characters to practice and the amount of repetitions. Until all characters have been drawn for this number of times, the app will randomly present one of the selected characters to the student who has to draw it in the appropriate area. After each character an animation is displayed showing the progress of the student. After finishing the complete exercise the children see their score based on the average of all drawn characters and their score per character. This whole process is visible in the right side of Figure 4.

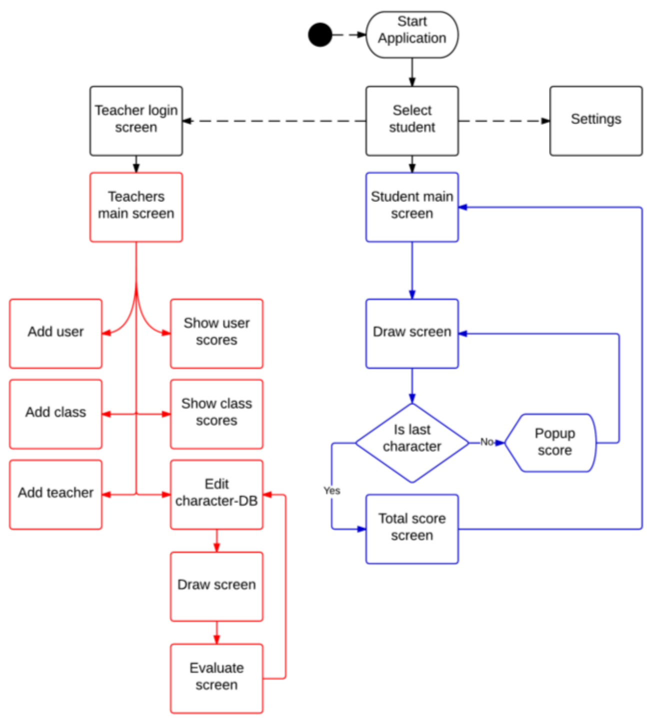

Figure 4: Relations between the different screens ("Activities") in the app

In addition, the editing ‘area’ (left side of Figure 4) enables the teacher to create new groups and new students and to consult their respective scores to the different exercises. Apart from that, teachers are able to change the character database. This can be done by uploading characters drawn by others or by writing the characters by themselves (Figure 5).

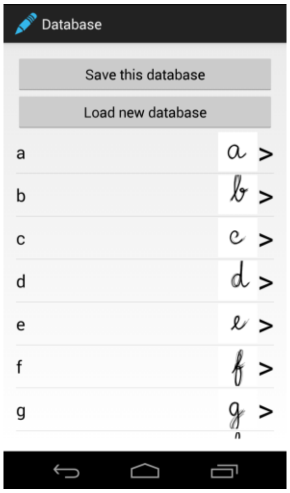

Figure 5: Character database with a list of all baseline characters (part of the teachers’ editing area). The row on the left shows the characters and the row on the right displays the corresponding baseline images.

6.1 Image Processing

The same image processing steps used to classify the characters are also used to resize, combine and compare the characters to score written characters in the app. Optimizing this process is important to give appropriate feedback to the user. Matlab® is first used to design and optimize an algorithm for both creating the ‘baseline character’ as well as scoring a character (Figure 6).

In the first step of creating a baseline the (resized) pictures used to make up this baseline are imported. One by one the characters will be shifted horizontally to find the position in which they have the best overlap with the previous pictures. A blur will be added in the next step to get rid of the lines that are not exactly overlapping and, in combination with the multiplication in the next step, this blurring will also determine the precision as definition of correct and the according score. The bigger the blur and multiplication, the thicker and blacker the baseline is.

When a test image of a character handwritten by a child is presented, the first step will be to shift it again to have the best overlap. This also means the horizontal position of the character inside the drawing area is not important but size and vertical position are. Both images will be inverted followed by a dot multiplication, which results in an image with a black background and with a white character. The parts where the test character overlaps the dark (most ideal) parts of the original baseline will be fully white in the result. At the parts of the character where the test image diverts from the ideal path the resulting image will be darker or black. The final score is based on the difference between the negative of the original test image (where all pixels drawn are fully white) and this result image (where pixels will be darker if they were not on the ideal line).

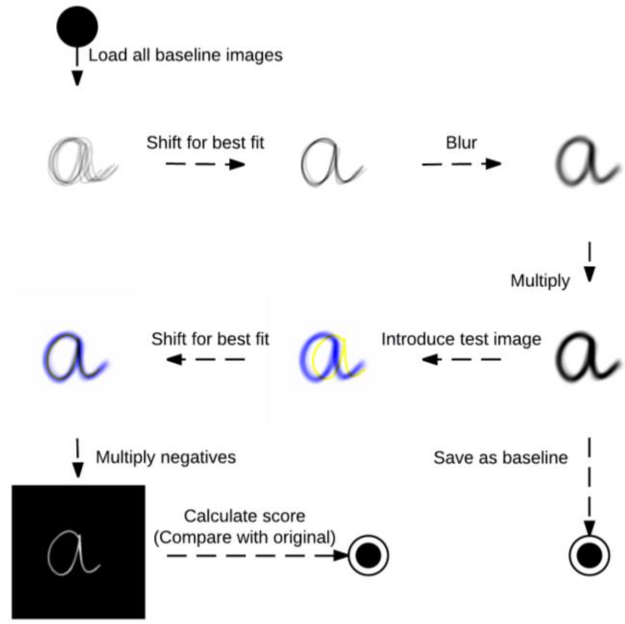

Figure 6: Baseline creation and scoring algorithm

Java (Android) libraries such as, but not limited to, JMagick and OpenCV were compared to see which of these libraries is able to do this and in the least amount of time. Eventually, the last mentioned was implemented in the app to handle the image processing. The downside of using OpenCV is the extra app it needs to have on the device to work but the reason for this is so it can provide the best and fastest processing because it is specifically compiled for the devices’ core.

6.2 Testing

A small group of 13 students ranging from 6 till 11 years old was asked to perform two exercises. One of the two exercises was to draw (write) ten given characters each two times as they learned it in school. The other exercise was to copy the same ten characters by drawing over the character displayed in the background of the drawing area. Half of the students were asked to do the copying first, followed by the drawing and the others did the drawing first and afterwards the copying exercise.

A rocket launch was used to give a funny feedback to the children regarding their performance. The rocket would go up from the baseline to the top of the screen and would reach the top if all characters were given the maximum score. Otherwise it would eventually reach the height based on the total average score. The rocket is only visible and rises a bit while the score of last written character is shown, then disappears until the next character has been written.

7 Results & Discussion

In Table 2, the scores of all 13 participants are shown. In the second and third column, the average scores of both exercises (20 characters each) are given. This can be the 'Copy' or 'Draw' exercise depending on the participant. Odd participants carried out 'Copy' first whereas even participants did 'Draw' first. The next two columns show the average of both trials, if the student had the task to do the 'Copy' exercise first followed by the 'Draw' it is displayed in column 4, otherwise in column 5. The following two columns show the score of trial 1 but separated into two columns to be able to show the difference in scores for both types of exercises. Finally, the last column displays the score to 'Draw' in trial 2.

Table 2: Test results

Student	Trial 1	Trial 2	T1: Copy T2: Draw	T1: Draw T2: Copy	Trial 1: copy	Trial 1: draw	Trial 2: draw
1	0.97	0.64	0.81		0.97		0.64
2	0.66	0.93		0.80		0.66	
3	0.97	0.71	0.84		0.97		0.71
4	0.66	0.97		0.81		0.66	
5	0.91	0.70	0.81		0.91		0.70
6	0.64	0.95		0.80		0.64	
7	0.91	0.71	0.81		0.91		0.71
8	0.62	0.92		0.77		0.62	
9	0.94	0.58	0.76		0.94		0.58
10	0.68	0.91		0.79		0.68	
11	0.93	0.67	0.80		0.93		0.67
12	0.60	0.92		0.76		0.60	
13	0.88	0.65	0.76		0.88		0.65
Average	**0.797**	**0.790**	**0.798**	**0.788**	**0.930**	**0.643**	**0.667**

The difference in averages between sessions 1 and 2 suggests there is not a strong improvement when a student did the exercise before, as confirmed by a Nonparametric Wilcoxon test ($Z = 0.21$).

A slightly larger difference is visible between the averages of students who did the copy exercise first followed by the draw (no background) exercise, compared to the students who wrote the character by themselves first followed by copying the background (see Figure 7). However, both tests on having the background in the first session or the background in the second session show no significant improvement in performance (Mann-Whitney, $U = 14.5$ for overall performance and $U = 13.5$ for comparing only the two background sessions).

The only significant difference in performance is visible when we compare the scores with and without background (Wilcoxon: $Z = 3.18$; $p < 0.01$). As expected, the performance to draw the letter is better with than without a model of the letter in background.

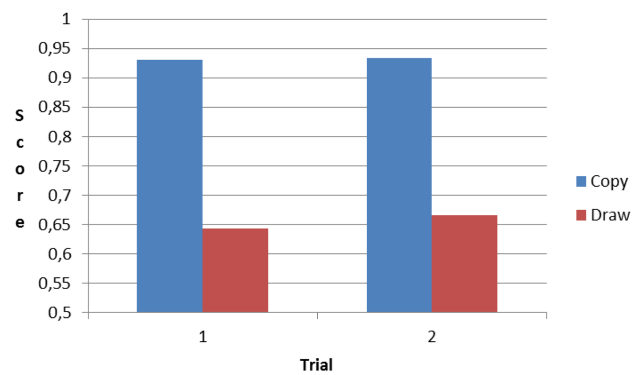

Figure 7: Averages of both sessions

As expected, the averages of the copying exercise itself are a lot better than the exercises where a student has to write the character without any example.

The difference between the averages of the last two columns does suggest an improvement when they practiced with the example first. The group doing the drawing in the second round, in which they have to do it without example, tend to be better than the group doing the drawing exercise in the first round. The experiment has to be carried out with more participants and a longer period of time to draw further conclusions.

Some interesting comments and observations were made by both teachers and researchers while performing the test. During the copy exercise some students started at the wrong position or drew part of the character in a way they were not taught to do so, even if they drew this character correctly during the draw exercise. It looks like these children are so focused on exactly copying the character they do not think about how to write it. The acquisition of the difference between writing and simply copying might take several years for the children [5].

Teachers also suggested to not only score the end result of writing the character correctly but also the process. This includes the correct starting position and direction of writing. Introducing this would not only give a better motivation and indication in how to write the character but also reduce the score of children who just copy the character instead of correctly writing them as mentioned before.

8 Conclusion & Future Work

In this paper was proposed an approach for teaching children handwriting through a character recognition based on image processing and machine learning. Different classifiers were analyzed and tested on handwritten characters provided by teachers. This way a classifier was found recognizing 90% of the characters correctly. Using this principal as foundation, an Android application has been developed giving children a way to learn how to write characters autonomously without losing input and expertise of a teacher.

The test performed on a small group of students shows promising preliminary results. Besides character recognition, other parameters, such as starting point and movement direction, have to be taken into account in the development of a system aiming to assist children in the learning process of handwriting skills. A further study regarding the effect on the learning curve, motivation and engagement of the students and teachers will be carried as future work.

Acknowledgment

We are thankful for the assistance provided by schools in the Lisbon area providing us with test data and advice. In particular, we are grateful for the ability to test the App at the Escola Branca of Caparica Monte and Colégio Campo de Flores.

References

[1] F. V. C. Ficarra, “Virtual Classroom and Communicability: Empathy and Interaction for All,” in *Human-Computer Interaction. Interacting in Various Application Domains*, Springer, 2009, pp. 58–67.

[2] G. Carrasco, Y. Rybarczyk, T. Cardoso, and I.P. Martins, “A serious game for multimodal training of physician novices,” in *Proceedings of the 6th International Conference of Education, Research and Innovation*, Seville, Spain, 2013.

[3] M. A. Eid, M. Mansour, A. H. El Saddik, and R. Iglesias, “A haptic multimedia handwriting learning system,” in *Proceedings of the International Workshop on Educational Multimedia and Multimedia Education*, 2007, pp. 103–108.

[4] J. Pereira, L. Carriço, and C. Duarte, “Improving children’s writing ability,” in *Human-Computer Interaction. Interacting in Various Application Domains*, Springer, 2009, pp. 186–195.

[5] R. Palluel-Germain, F. Bara, A. H. de Boisferon, B. Hennion, P. Gouagour, and E. Gentaz, “A Visuo-Haptic Device-Telemaque-Increases Kindergarten Children’s Handwriting Acquisition,” *Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2007, pp. 72–77.

[6] M. Mansour, M. Eid, and A. E. Saddik, “A multimedia handwriting learning and evaluation tool,” in *Proceedings of the 4th Conference on Intelligent Interactive Learning Object Repositories (I2LOR)*, Montreal, Canada, 2007.

[7] M. H. Tseng and S. A. Cermak, “The influence of ergonomic factors and perceptual–motor abilities on handwriting performance,” *American Journal of Occupational Therapy*, vol. 47, no. 10, pp. 919–926, 1993.

[8] S. Graham, K. R. Harris, and B. Fink, “Is handwriting causally related to learning to write? Treatment of handwriting problems in beginning writers,” *Journal of Educational Psychology*, vol. 92, no. 4, pp. 620–633, 2000.

[9] N. L. K. Aizan, E. I. Mansor, and R. Mahmod, “Preschool Children Handwriting Evaluation on Paper-Based and Tablet-Based Settings,” *International Journal of Computer and Information Technology*, vol. 3, no. 2, pp. 256–262, 2014.

[10] A. K. Seewald, “Digits-a dataset for handwritten digit recognition,” *Austrian Research Institut for Artificial Intelligence Technical Report, Vienna (Austria)*, 2005.

[11] D. W. Aha, D. Kibler, and M. K. Albert, “Instance-based learning algorithms,” *Machine learning*, vol. 6, no. 1, pp. 37–66, 1991.

[12] G. H. John, and P. Langley, “Estimating continuous distributions in Bayesian classifiers,” in *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, 1995, pp. 338–345.

[13] J. R. Quinlan, *C4. 5: Programs for Machine Learning*. Elsevier, 2014.