An Intelligent Agent Tutor System for Detecting Arabic Children Handwriting Difficulty Based on Immediate Feedback

Ahmed El-Sawy
Computer Science Department
Faculty of Computer & Informatics,
Benha University
Benha, Egypt

Hazem El-Bakry
Information System Department
Faculty of Computer & Information Sciences,
Mansoura University
Mansoura, Egypt

Mohamed Loey
Computer Science Department
Faculty of Computer & Informatics,
Benha University
Benha, Egypt

Nikos Mastorakis
Technical University of Sofia,
BULGARIA

Abstract: - In this paper, an intelligent tutor application is built for Arabic preschool children called Arab Handwritten Children Educator (AHCE). AHCE allows Arab children to do practice at anytime and anywhere. As an intelligent tutor, the AHCE can automatically check the handwriting errors, such as stroke sequence errors, stroke direction error, stroke position error, and extra stroke errors. The AHCE provide a useful feedback to Arab children to correct their mistakes. In this paper, attributed mathematics and agents are used to locate the handwritten errors. The system applies a fuzzy approach to evaluate Arabic children handwriting. Experimental results indicate that the proposed intelligent system successfully detect handwriting strokes errors with immediate feedback.

Key-Words: - Human-Computer Interaction; Intelligent Mobile Tutoring Systems; Handwriting Difficulty; Improving Classroom Teaching; Interactive Learning Environments; Artificial Mobile System

1 Introduction

In a traditional approach of the handwriting teaching system, first, the pedagogue should write the Arabic character on the whiteboard, and then the children must rewrite the handwritten character in their copying notes. Finally, the pedagogue try to check and verify the handwriting errors in the children notebooks and provide a feedback [1]. Therefore, the problem of this traditional handwriting teaching system is impossible for the pedagogue to verify and check every children handwriting in the limited time of the lesson [2]. Moreover, children often practice incorrect of writing for extended periods of time before they are detected and corrected.

Intelligent tutoring system (ITS) [3-5] is smart and an intelligent system that helps an educator teaching process based on a computer and mobile system that uses artificial techniques. The ITS also help and assist a tutor through a software interface to interact between an artificial system, student, and tutor. However, the ITS provide students with a suitable way to study at their own way, the ITS improve student efficiency and ability to correct problems based on their study. The ITS use natural language processing [6, 7], unsupervised neural network [8], machine learning, fuzzy [9], multi-agent systems [10-13], semantic Web [14], ontology [15], social and emotional computing, multimedia, modeling, gaming and simulation [16], and statistics to provide an excellent platform for learning. The rapid growth of the internet and mobile technology offers a new opportunities and challenges for many areas, one of them is teaching handwriting. Many researchers had exerted much effort to design and improve handwriting learning systems for children.

Arabic is a kind of Semitic language used widely as a mother language of hundred millions people [17]. Countries throughout the Middle East and northern and north-eastern Africa, and Traditional Arabic use Arabic as the national language [18]. The Arabic alphabet consists of twenty eight characters shown in Fig. 1. Arabic language has many features to assist children to distinguish between alphabet characters. The Arabic character can be written with one or more strokes, which are necessarily sequential. Strokes are sequences of points trace between a finger-down and a finger-up of a hand on a touch screen. Moreover, the stroke is our basic unit and we assume that a stroke belongs to one or more characters. Most on-line characters consist of a sequence of strokes which can be written in a different order and direction and will still represent the same character.

خ	ح	ج	ث	ت	ب	أ
khah	hah	jeem	theh	teh	beh	alef
ص	ش	س	ز	ر	ذ	د
sad	sheen	seen	zain	reh	thal	dal
ق	ف	غ	ع	ظ	ط	ض
qaf	feh	ghain	ain	zah	tah	dad
ي	و	ه	ن	م	ل	ك
yeh	waw	heh	noon	meem	lam	kaf

Figure 1.Arabic characters

A computer-based assessment tool proposed by [2] that measure a stroke order of the Chinese Character handwriting based on gray measure scoring help Chinese children when learn how to write their characters. Another tool [19, 20] proposed an assessment tool to detect types and direction when forming Latin alphabets. The study used eight directional codes to represent stroke types that identify and classify simple and complex straight-line stroke, while more features are needed to further distinguish curve lines. A Latin and Sinhala e-learning tool proposed by Priyankara et al. [21] based on Android that facilitated self-learning of the preschool children. The proposed system based on a strong theoretical foundation to develop cognitive and psychomotor skills such as drawing, writing, and learning. The study use algorithms that identified the need to teach the child the correct way of writing a particular letter without guidance of parents. [22] proposed an interactive application on a smart device that designed to facilitate and encourage preschool children to learn and practice Chinese characters. The study built up a special relation matrix for each character to detect strokes errors and provide a visual feedback. A web-based Chinese handwriting system with Automatic Feedback and Analysis proposed by Tang et al. [23] that allows students to learn Chinese handwriting at anytime and anywhere. The proposed system can detect and check multiple stroke production errors at each time. A computer-based Chinese handwriting system proposed by [24] that help Chinese student to detect stroke order difficulty and the total number of strokes. The proposed system use methods to extract seven feature from Chinese characters to detect Chinese handwriting errors. [25, 26] proposed a graph matching technique to help the Chinese student in handwriting. The proposed system provides a student a feedback with error correction that detects all types of errors and even more errors that are complex.

In this study, we have developed an android system called as Arab Handwritten Children Educator ‘‘AHCE” for tutoring of handwriting. The present paper is structured as follows: in Section 2 the proposed system. Section 3 describes the architecture of the system. Section 4 describes the components of the system. Section 5 show results of a research study. Finally, conclusions and future work are presented in Section 6.

2 AHCE Architecture

Our proposal, AHCE, is an intelligent mobile platform assist Arab children to learn handwriting their character. The AHCE system was developed to detect handwriting alphabet stroke errors. AHCE system accepts points of the written Arabic character stroke in x-y coordinate from a digital touch device. In next part we will explain AHCE architecture.

The main architecture of AHCE consists of three interface components: kid interface, teacher interface, teaching interface. Moreover, AHCE consist of three agent components: teaching agent, feedback agent, evaluation agent. Finally, AHCE has one database contain four tables: kid table, teacher table, teaching table, feedback table. Conceptually, the relationship among the components: AHCE interface, AHCE agents and AHCE database can be viewed as in Fig. 2. The following section details the system architecture operating procedure. Based on the system architecture, the details of system operation procedure are described as follows:

- Step 1. Kids log to AHCE system through the kid interface.
- Step 2. If the children have owned a legal account in our system, the interface will load his/her information; otherwise, the system will add a new record into children account table for this learner.
- Step 3. After successful log in, the system automatic go to teaching interface. Now, children choose the learning character from the interface.
- Step 4. The system loads the teaching agent. The teaching agent loads some tools for writing, deleting and creating new tracer.
- Step 5. The teaching agent gets the contents of learning Arabic alphabet character from the teaching table and exhibits them for the children.
- Step 6. The kids his/her finger on touch screen, the system detected the sequence of x-y point

coordinate. The Feedback agent has intelligent error detection with automatic feedback.

- Step 7. The automatic feedback return back to teaching interface, the feedback show the detecting stroke error.

- Step 8/9/10/11. The feedback agent automatic record the type of error in a feedback table. The feedback table also gets the record of information about children and his/her age, also the educated character. Moreover, the feedback children obtain information record about a tutor, who teaches these children.

- Step 12. Tutor log to AHCE system through teacher interface.

- Step 13. If the tutor has owned a legal account in our system, the interface will load his/her information; otherwise, the system will add a new record into tutor account table for this teacher.

- Step 14. The teacher enters the result to show all his/her children performance.

- Step 15. The evaluation agent obtains all data from feedback table.

- Step 16. The evaluation agent show a statistical result about each child and all children related to teacher.

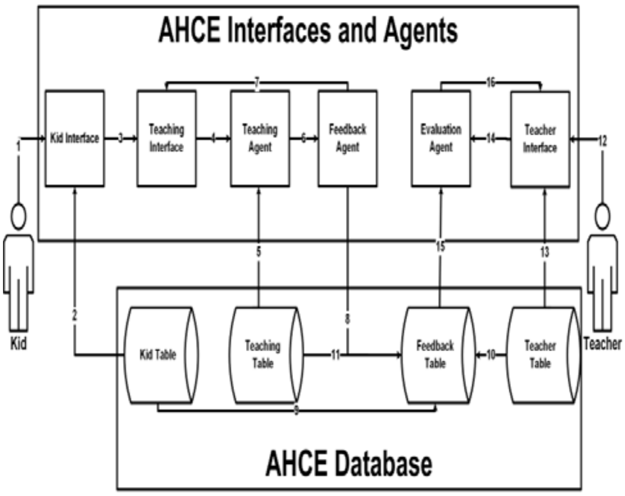


Figure 2. AHCE system architecture

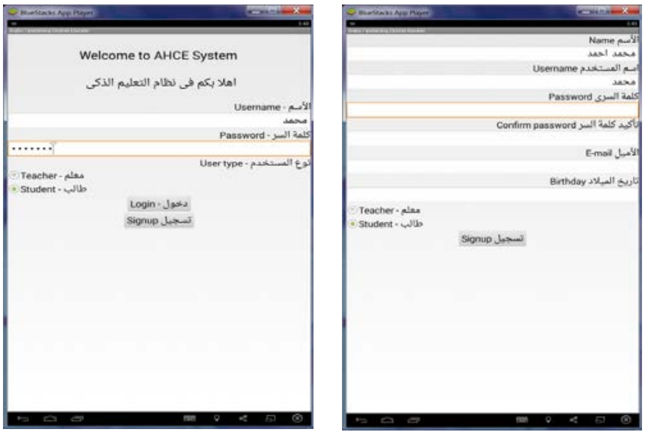
3 AHCE Components

The Model–view–controller (MVC) [27] is a software architectural pattern used to design a software system. The AHCE system implementation follows the MVC design pattern in which the system is divided into the data component and business logic using this data (Model), the user interface (View).

4.1 AHCE Interfaces

4.1.1 Kids interface

In our work, the main screen of AHCE which appears when the system works. Children enter user name and password in this screen. The kid interface provides a login form Fig. 3 (a) and register screen Fig. 3 (b) to interact with children.



(a) login screen (b) register screen

Figure 3. AHCE kid interface

4.1.2 Teaching interface

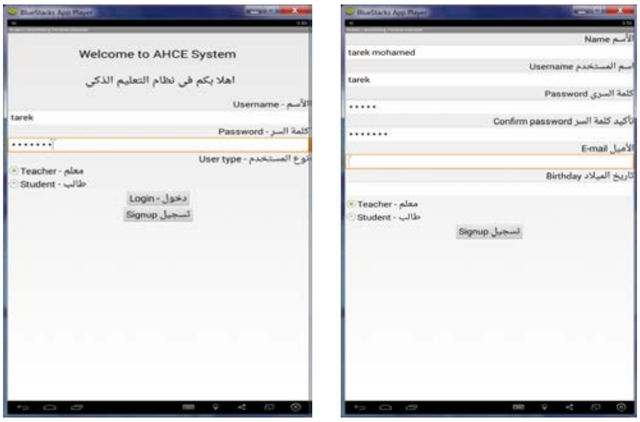
The teaching interface provides a friendly learner interface to interact with a kid, the interface illustrated in Fig. 4(a). Through the teaching interface, children can choose between Arabic alphabets. Moreover, children put his/her finger on screen, and then draw the teaching character shown in Fig. 4(b).

(a) (b)

Figure 4. AHCE teaching interface

4.1.3 Teacher Interface

The teacher interface provides a login and register interface to interact with tutor, the interface was illustrated in Fig. 5(a,b). Through the teacher interface agent, tutor can get a statistical result about children performance.



(a) login screen (b) register screen

Figure 5. AHCE teacher interface

4.2 AHCE Database

The AHCE database contain four main tables: (1) teacher table (2) kid table (3) teaching table (4) feedback table , our database design was illustrated in Fig. 6.

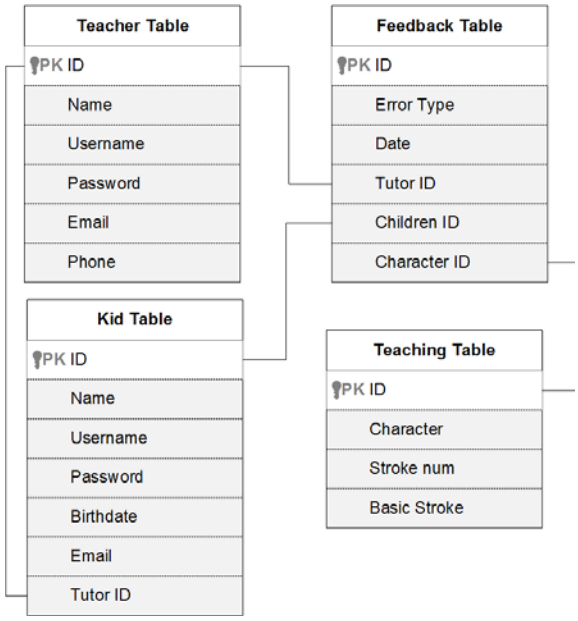


Figure 6. AHCE database

4.3AHCE Agents

AHCE is a multi-agent system based on three components: learning, feedback, evaluation. Here, agents use knowledge and methods to achieve our goals. Agents adapted online, learn and improve through interaction with the environment. In the next part, we will illustrate our intelligent agents.

4.3.1 Teaching agent

In this part, teaching agent load knowledge data from learning table that contained the characteristic of the chosen Arabic character. Moreover, teaching agent load tools and the interface that interact with children. Teaching agent based on five categories stroke number , stroke similarity, stroke order, stroke direction, and number of dots.

The first category is stroke number, Arabic alphabet characters were classified to four classifications based on stroke number. Therefore, the classification of Arabic characters were based on stroke number to (1) one stroke character such as (ح،د،ر،س،ص،ع،ل،م،ه،و), (2) two strokes character such as (أ،ب،ج،خ،ذ،ز،ض،ط،غ،ف،ك،ن), (3) three strokes character such as (ت،ق،ظ،ي), (4) four strokes character such as (ث،ش). Alphabet character qaf (ق) has three strokes (ى) and two dots (.), so any stroke more this three strokes is an extra stroke.

The second category is stroke number, The system stored stroke number for each Arabic alphabet character. The main purpose of this stroke number was to assist our educational system to detect stroke order error and extra stroke error. AHCE used a stroke order feature to write any Arabic characters correctly. Our AHCE system has a stroke counter to detect stroke order and extra stroke. Therefore, Arabic alphabet character zah (ظ) has two strokes (ط) and (.). Stroke (ط) wrote before stroke dot (.). When preschool children wrote stoke dot (.) before stroke (ط), the system will give quick feedback with stroke order errors detection. Order of stroke in Arabic characters illustrated in Fig. 7.

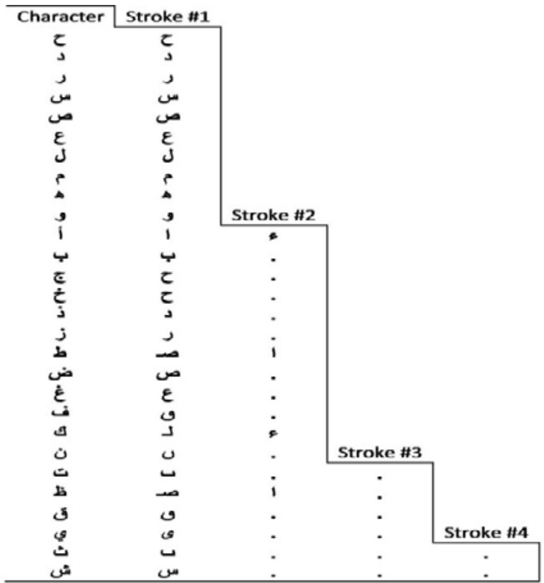


Figure 7. Stroke order

The third category is stroke similarity, Arabic characters are classified based on similarity of the master stroke, stroke similarity shown in Table 1. Alphabet characters jeem (ج), hah (ح) and khah (خ)

have the same major structure stroke (ح). The difference between them is the dot (.) is inside letter jeem (ج), the dot (.) above letter khah (خ).

The fourth category is stroke direction, Any character has a special direction to represent each stroke, the direction of Arabic alphabet characters. Arabic alphabet character zain (ز) has two strokes (ر) and (.). Preschool children must write stroke (ز) in a sequence down-left. Stroke (.) should write after stroke (ر) and above stroke (ر). When preschool children wrote (ر) in an inverse sequence right-up. Our AHCE system pop up error message that says wrong direction, then remove wrong stroke.

The Final category is number of dots on characters, in Arabic language dots are very important to classify alphabet characters. Therefore, 15 of 28 Arabic letters have one or more dots. The Arabic characters (ب،ج،خ،ذ،ز،ض،ظ،غ،ف،ن) have one dot. Moreover, the Arabic characters (ت،ق،ي) have two dots. Finally, the Arabic characters (ث،ش) have three dots.

Table 1. Stroke similarity

Similar characters	Master stroke
ب،ت،ث	ب
ج،ح،خ	ح
د،ذ	د
ر،ز	ر
س،ش	س
ط،ظ	ط
ص،ض	ص
ع،غ	ع
ف،ق	ف
ك،ل	ل

4.3.2 Feedback agent

The main part of AHCE system is feedback agent, feedback agent is the brain of AHCE. In this part, feedback agent diagnoses and fixes many type of Arabic character stroke error. Moreover, AHCE successfully send immediate feedback with automatic error detection to children. Feedback agent based on stroke direction detector and stroke order detector.

4.3.2.1 Stroke direction detector

our AHCE system used chain code [28] to encode a movement. Chain code is a widely used technique that provides an algorithm to represent a direction. The chain code encodes a stroke character as a sequence of movements through the neighboring boundary pixels based on 8-connectivity. The main purpose of chain code was to determine the direction of stroke in Arabic alphabet character. The freeman chain code algorithm steps defined as below:

Step 1: The absolute difference between y-axis

$$\Delta y = |Y_i - Y_{i-1}|$$

Step 2: The absolute difference between x-axis

$$\Delta x = |X_i - X_{i-1}|$$

Step 3: Calculate the angle

$$\theta = \tan^{-1}\frac{\Delta y}{\Delta x}$$

Moreover, the direction of next coordinate from current coordinate is determined according to chain code. Based on the chain code, the written stroke direction determined according to the analysis of angle θ. The angle θ produced eight quadrants of angle represented to determine the direction of points in stroke as shown below in table 2.

Table 2. Direction code angle

Code	Angle (θ)	Direction
C0	355°< θ <5°	Right
C1	5°< θ <85°	Up Right
C2	85° < θ < 95°	Up
C3	95°< θ < 175°	Up Left
C4	175° < θ < 185°	Left
C5	185° < θ < 265°	Down Left
C6	265° < θ < 275°	Down
C7	275° < θ < 355°	Down Right

4.3.2.2 Stroke order detector

When children put his/her finger on touch screen, the system detected the sequence of x-y point coordinate. Until the children move his/her finger up, the system store those sequence of point as Pi=(xi,yi) such that i=[0,1,...,m-1,m] where m number of x-y coordinate. Each stroke had number of points as Sn= [p0,p1,...,pm-1,pm], where S is stroke, P is point in x-y coordinate, n is number of stroke.

4.3.3 Evaluation agent

In this part, AHCE indicates the Arab children level of understanding of learning handwriting character concepts. AHCE use fuzzy logic [29] to evaluate Arabic children. Fuzzy models successfully handle information, and enable representation of children handwriting in the same way human teachers do. The rules used in AHCE system have a form similar to the one in Fig. 8.

IF
(Character-Diffucullity is Very-Difficult)
and
(Character-Error is Very-Small)
and
(Time-Consumed is Very-Short)
and
(Children-Age is Kindergarten)
Then
(Children-Evaluation is Expert)

Figure 8. Example of a rule used

The generalized bell-shaped membership function is describing the linguistic variables of input and output for the fuzzy inference of AHCE system. The bell-shaped membership function can be defined as follows:

$$f(x;a,b,c) = \frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}}$$

Where $f(x;a,b,$ denotes the fuzzy degree of input x under the linguistic variable xi, a is the half width, b controls the slopes at the crossover points, c determines the centre of the corresponding membership function.

In Fig. 9 (a), 'very-large' is one of the possible values of the linguistic input variable 'character error' and it reflects the number of error when children write the character. It takes values in the range from 0 to 10, and is defined by four fuzzy sets. 'Character error' has the range from zero ('very small) to 10 ('very large)

(Fig. 9 (b)) 'very-easy' is one of the possible values of the linguistic input variable 'character difficulty' described by the membership function of the corresponding fuzzy set. The 'character difficulty' variable depends on the difficulty of each particular character handwriting. It is calculated as the value of character handwriting difficulty that the children have to write character. 'Character difficulty' has the range from zero ('very easy') to 100 ('very difficult'), and is described by five fuzzy sets (Fig. 9 (b)).

'Young' is one of the possible values of the linguistic input variable 'children age' and it reflects the age of children on test. Specifically, three fuzzy sets (Fig. 9(c)) are defined for the 'children age' variable: 'kindergarten' corresponds to the age of children between 1 to 6, 'young' for the age of children from 6 to 11, and 'middle age' for the age between 11 to 15. It takes values in the range from 1 to 15, and is defined by four fuzzy sets (Fig. 9(c)).

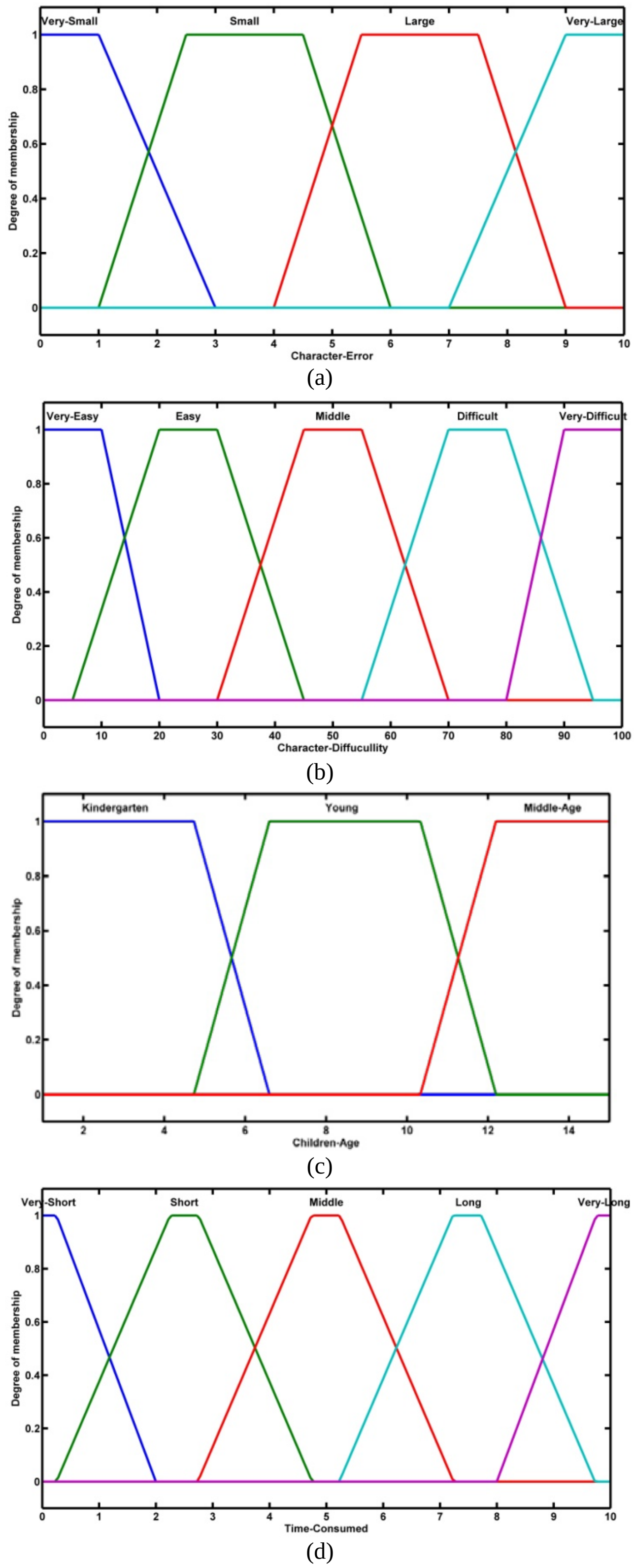


Figure 9 AHCE input fuzzy set

The 'long' value in the example means that the children has spent less time solving the test than he/she should have spent. It is one of the possible values of the 'time consumed' variable. Specifically, five fuzzy sets (Fig. 9(d)) are defined for the 'time consumed' variable: 'very short' corresponds to very fast resolution of a character handwriting, 'middle' for average time consumed of resolving a test, and 'very long' for very slow character handwriting resolving. The value of the variable depends on the average time that the tutor has defined for solving a particular set of questions in a test.

'Expert' is one of the values that can be used to represent children evaluation. This value belongs to the 'children evaluation' variable and takes values in the range from 0 to 100. It is defined by six fuzzy sets (Fig.10).

AHCE have successfully evaluate children handwriting by encoded a set of fuzzy rules using the jfuzzylogic Toolkit [30, 31]. The jfuzzylogic Toolkit is a Java API for representing and reasoning with fuzzy information. The toolkit consists of a set of classes that allow us to build logic fuzzy systems. By analysing the behavior of the AHCE feedback response, several basic fuzzy rules can be summarized to infer the children understanding degree for the learned character. To infer the children evaluation degree, the reasoning process of Mandani's minimum fuzzy implication. Moreover, the defuzzification method of center of gravity is employed to obtain the crisp value of children evaluation degree [32].

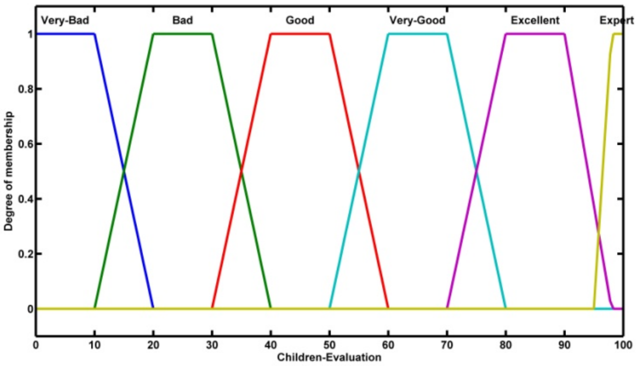


Figure 10 AHCE fuzzy set output

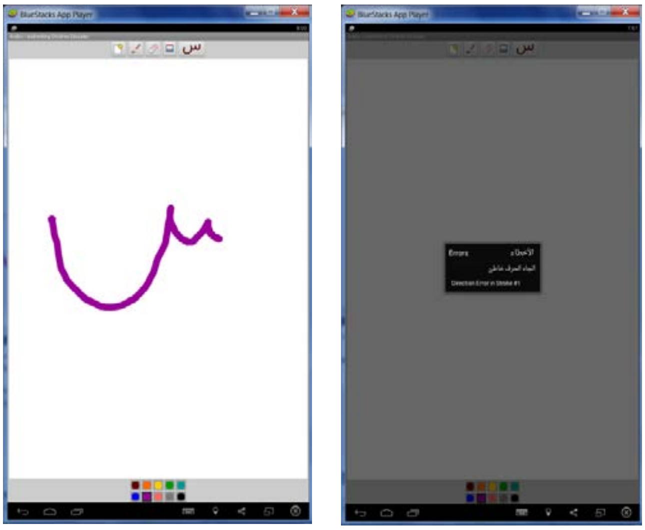
5 Result and Discussion

5.1 Experimental environment

Through the work, the proposed prototype of AHCE is implemented on the platform of Android. Moreover, the front-end script language java is used to implement this system. The AHCE has been tested on bluestacks simulator. Bluestacks is designed to enable Android applications to run on Windows personal computers, Macintosh computers. Therefore, this implementation is tested on Windows 7 64bit operating system with an Intel® core i7 CPU(2.5 GHz) and 8GB of RAM.

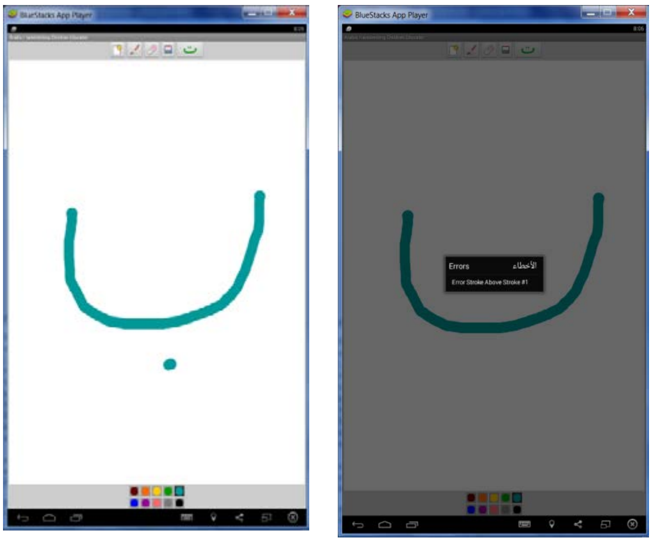
5.2 Experimental test

In this part, AHCE was tested with a set of inputs containing all Arabic alphabet letters using a tablet and Smartphone's. The evaluation of AHCE was performed in our faculty of Computer Science and Informatics. The AHCE system performs analysis to preschool children's handwriting with proposed methods and automatically AHCE returns immediate feedback result. The result for sample Arabic characters illustrated in Fig. 11, 12, 13.



(a) direction from left to right (b) AHCE system feedback

Figure 11. Directional stroke error of character Seen



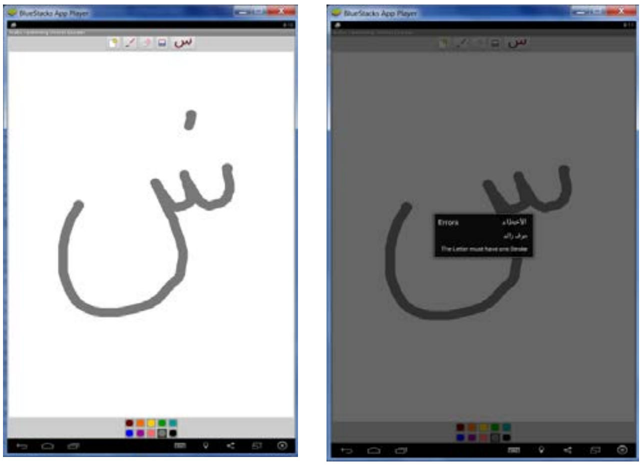
(a) stroke under (ب) (b) AHCE feedback

Figure 12. Stroke position of Arabic character Theh (ث)

Different kinds of feedback presented depending on the scenario. When the input character is correct, the screen will display no problem. Otherwise, the system will give immediate feedback with error detection. When children wrote a stroke in inverse direction, the children have got a direction error message, and then AHCE system removes a wrong stroke. The directional stroke error in letter seen (س) show in Fig. 11.

Another kind of feedback is when children wrote the second stroke in wrong position. The AHCE will give immediate feedback with automatic error detection in Fig. 12. The last kind of feedback error detection shown in Fig.13. When children entered an extra stroke to our system, the system automatic detects and remove extra stroke.

The preschool children and their tutors were asked to use the AHCE as part of their learning of write Arabic alphabet. Arabic children were classified to four level based on their ages. After the learning was completed, the children and tutors were interviewed. A part of the questions was asked to the educator illustrated in Table 3. 200 questionnaires were collected from four levels of children groups. The values of Fig. 14 are the result of analyses of the questionnaire. The educators and children found the whole handwriting learning software application improve children handwriting and useful. The results specify that educators rated the AHCE very highly on acceptability for both interactive and ease of use.



(a) extra stroke above seen (س) (b) AHCE feedback

Figure 13. Extra stroke error of Arabic character Seen

Table 3. A part of the questionnaire that were asked

1)	It is improve handwriting		
	(a) Disagree	(b) Neutral	(c) Agree
2)	It is user friendly		
	(a) Disagree	(b) Neutral	(c) Agree
3)	It is useful		
	(a) Disagree	(b) Neutral	(c) Agree
4)	It is interactive		
	(a) Disagree	(b) Neutral	(c) Agree

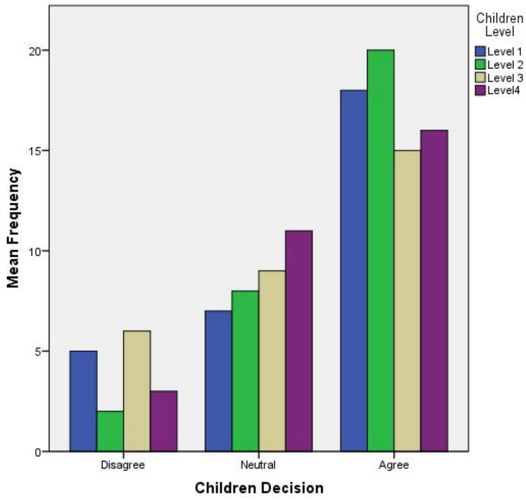


Figure 14. The result of questionnaire

6 CONCLUSION

In this study, An intelligent tutoring system for handwriting Education has been developed, called ‘‘AHCE”. Such system has attempted to imitate the behavior of an intelligent human tutor and a domain expert. In addition, AHCE can be used for the purpose of either individual or real classroom environment with the guidance of a tutor during a formal learning process. Furthermore, it helps Arab preschool children to diagnose their handwriting mistakes efficiently. Moreover, the proposed AHCE has successfully reduced time and effort for Arab preschool educators and their children. In the future work, the authors will try to develop the proposed AHCE system to detect all types of the Arabic handwriting learning mistakes.

7 REFRENCES

[1] Hu, Z.-H., Xu, Y., Huang, L.-S. and Leung, H. 2009. A Chinese Handwriting Education System with Automatic Error Detection. Journal of Software, 4, 2 (Apr 2009), 101-107.

[2] Guey-Shya, C., Yu-Du, J., Der-Bang, W. and Hei-Tsz, C. .2009. Stroke order computer-based assessment with grey measure scoring. Grey Systems and Intelligent Services City, (Nov. 2009), 763 - 768.

[3] Noh, N. M., Ahmad, A., Halim, S. A. and Ali, A. M. .2012. Intelligent Tutoring System using Rule-based And Case-based: A Comparison. Procedia - Social and Behavioral Sciences, 67 (Oct 2012), 454-463.

[4] Sanchez, R. P., Bartel, C. M., Brown, E. and DeRosier, M. 2014. The acceptability and efficacy of an intelligent social tutoring system. Computers & Education, 78 (Sep 2014), 321-332.

[5] Jeremić, Z., Jovanović, J. and Gašević, D. 2012. Student modeling and assessment in intelligent tutoring of software patterns. Expert Systems with Applications, 39, 1 (Jan. 2012), 210-222.
[6] Günel, K. and Aşlıyan, R. 2010. Extracting learning concepts from educational texts in intelligent tutoring systems automatically. Expert Systems with Applications, 37, 7 (Jul. 2010), 5017-5022.
[7] Arevalillo-Herráez, M., Arnau, D. and Marco-Giménez, L. 2013. Domain-specific knowledge representation and inference engine for an intelligent tutoring system. Knowledge-Based Systems, 49, 0 (Sep. 2013), 97-105.
[8] Cabada, R. Z., Barrón Estrada, M. L. and Reyes García, C. A. 2011. EDUCA: A web 2.0 authoring tool for developing adaptive and intelligent tutoring systems using a Kohonen network. Expert Systems with Applications, 38, 8 (Aug 2011), 9522-9529.
[9] Badaracco, M. and Martínez, L. 2013.A fuzzy linguistic algorithm for adaptive test in Intelligent Tutoring System based on competences. Expert Systems with Applications, 40, 8 (Jun 2013), 3073-3086.
[10] Cardoso, J., Bittencourt, G., Frigo, L., Pozzebon, E. and Postal, A. 2004. Mathtutor: A Multi-Agent Intelligent Tutoring System. Springer US, City, 154, 2004, 231-242.
[11] Sun, Y. and Li, Z. 2009. A multi-agent intelligent tutoring system. Computer Science & Education ICCSE '09. 4th International Conference on, (Jul 2009), 1724 - 1728.
[12] Sierra, E., Hossian, A., Britos, P., Rodriguez, D. and Garcia-Martinez, R. 2007. A Multi-agent Intelligent Tutoring System for Learning Computer Programming. City, (Sep. 2007), 382 - 385.
[13] Lavendelis, E. and Grundspenkis, J. 2009. MASITS – A Tool for Multi-Agent Based Intelligent Tutoring System Development. 7th International Conference on Practical Applications of Agents and Multi-Agent Systems. Springer Berlin Heidelberg, City, 55 (2009), 490-500.
[14] Abbas, M. A., Ahmad, W. F. W. and Kalid, K. S. 2014. OntoCog: A Knowledge based Approach for Preschool Cognitive Skills Learning Application. Procedia - Social and Behavioral Sciences, 129, 0 (May 2014), 460-468.
[15] Marciniak, J. 2014. Building Intelligent Tutoring Systems Immersed in Repositories of E-learning Content. Procedia Computer Science, 35, 0 (2014), 541-550.
[16] Angelides, M. C. and Paul, R. J. Developing an intelligent tutoring system for a business simulation game. Simulation Practice and Theory, 1, 3 (Dec. 1993), 109-135.
[17] Elarian, Y., Ahmad, I., Awaida, S., Al-Khatib, W. G. and Zidouri, A. 2015. An Arabic handwriting synthesis system. Pattern Recognition, 48, 3 (Mar 2015), 849-861.
[18] Hackett, J. 2006. Semitic Languages. Elsevier, City, (2006), 229-235.
[19] Neo Chin, C., Ming, E. S. L., Khalid, P. I. and Yeong Che, F. 2012. Algorithm for Identifying Writing Stroke and Direction. Computational Intelligence, Modelling and Simulation (CIMSiM), 2012 Fourth International Conference on City, (Sep 2012), 94 - 98.
[20] Neo, C. C., Su, E. L. M., Khalid, P. I. and Yeong, C. F. 2012. Method to Determine Handwriting Stroke Types and Directions for Early Detection of Handwriting Difficulty. Procedia Engineering, 41 (2012), 1824-1829.
[21] Priyankara, K. W. T. G. T., Mahawaththa, D. C., Nawinna, D. P., Jayasundara, J. M. A., Tharuka, K. D. N. and Rajapaksha, S. K. 2013. Android based e-Learning solution for early childhood education in Sri Lanka. Computer Science & Education (ICCSE), 2013 8th International Conference on City, (Apr 2013) 715 - 718.
[22] Tang, W. W. W., Leong, H. V., Ngai, G. and Chan, S. C. F. 2014. Detecting handwriting errors with visual feedback in early childhood for Chinese characters. In Proceedings of the Proceedings of the 2014 conference on Interaction design and children (Aarhus, Denmark, 2014). ACM, 273-276.
[23] Tang, K.-T., Li, K.-K. and Leung, H. A Web-Based Chinese Handwriting Education System with Automatic Feedback and Analysis. Springer Berlin Heidelberg, Springer Berlin Heidelberg, 4181 (2006), 176-188.
[24] Guey-Shya, C., Yu-Du, J. and Li-Fon, L. Computer-based Assessment for the Stroke Order of Chinese Characters Writing. City, 2007.
[25] Hu, Z., Leung, H. and Xu, Y. Stroke Correspondence Based on Graph Matching for Detecting Stroke Production Errors in Chinese Character Handwriting. Advances in Multimedia Information Processing – PCM 2007 .Springer Berlin Heidelberg, City, 4810 (2007), 734-743.
[26] Hu, Z. H., Xu, Y., Huang, L. S. and Leung, H. A Chinese handwriting education system with automatic error detection. J. Softw. Journal of Software, 4, 2 (2009), 101-107.
[27] Leff, A. and Rayfield, J. T. Web-application development using the Model/View/Controller design pattern. Enterprise Distributed Object Computing Conference, 2001. EDOC '01. Proceedings. Fifth IEEE International, 2001, 118 - 127.

[28] Kui Liu, Y. and Žalik, B. 2005. An efficient chain code with Huffman coding. Pattern Recognition, 38, 4 (Apr 2005), 553-557.
[29] Zadeh, L. 1965. A. Fuzzy sets. Information and Control, 8, 3 (Jun 1965), 338-353.
[30] Cingolani, P. and Alcalá-Fdez, J. 2013. jFuzzyLogic: a Java Library to Design Fuzzy Logic Controllers According to the Standard for Fuzzy Control Programming. International Journal of Computational Intelligence Systems, 6, sup1 (Jun 2013), 61-75.
[31] Cingolani, P. and Alcala-Fdez, J. 2012. jFuzzyLogic: a robust and flexible Fuzzy-Logic inference system language implementation, (Jun 2012), 1-8.
[32] Zimmermann, H.-J. 1996 Fuzzy set theory\—and its applications (3rd ed.). Kluwer Academic Publishers, 1996, 435.