ISSN 2347-4289

Arab Kids Tutor (AKT) System For Handwriting Stroke Errors Detection

Ahmed El-Sawy, Mohamed Loey, Hazem EL-Bakry

Computer Science Department, Faculty of Computer & Informatics, Benha University, Benha, Egypt;
Information System Department, Faculty of Computers & Information, Mansoura University, Mansoura, Egypt.
Email: Ahmed.el_sawy@fci.bu.edu.eg, mohamed.loey@fci.bu.edu.eg, helbakry5@yahoo.com

ABSTRACT: This paper presents the architecture, components and evaluation of Arab kids tutor (AKT), an intelligent tutor system for learning handwriting Arabic alphabets. Today, children suffer from handwriting difficulty, so tutors hope to get rid of negative impact of traditional learning system. Our system contains an immediate feedback with error detection that can check multiple kinds of handwriting errors and provide an intelligent feedback to our children. Moreover, AKT use freeman chain code and mathematical algorithms to detect order and direction errors. Through the work, we indicate the children level of understanding of learning handwriting character using fuzzy sets. Experimental results indicate that the AKT successfully detect handwriting strokes errors with automatic feedback.

Keywords: Interactive Learning Environments, Human-Computer Interface, Intelligent Tutoring Systems, Improving Classroom Teaching

1 INTRODUCTION

Intelligent tutoring system (ITS) [1-3] is smart and intelligent system that imitates an educator teaching process based on a computer system that uses several techniques and artificial techniques. Moreover, the ITS help and assist a paedology through a software interface to interact between an artificial system, student, and pedagogue. 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 machine learning, fuzzy [4], unsupervised neural network [5], multi-agent systems [6-9], natural language processing [10, 11], ontology [12], semantic Web [13], social and emotional computing, multimedia, modelling, gaming and simulation [14], and statistics to provide an excellent platform for learning. In a classical 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 [15]. Therefore, the problem of this classical handwriting teaching system is impossible for the pedagogue to verify and check every children handwriting in the limited time of the lesson [16]. Moreover, children often practice incorrect of writing for extended periods of time before they are detected and corrected.

1.1 Handwriting-based Intelligent Tutor Systems

The rapid growth of the internet and mobile technology offers a new opportunities and challenges for many areas, one of them is teaching Handwriting-based learning. Many researchers had exerted much effort to design and improve handwriting learning systems for children. For example a web-based Chinese handwriting system with Automatic Feedback and Analysis proposed by Tang et al. [17] 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. Guey-Shya et al. [18] proposed a computer-based Chinese handwriting system that assist student to detect stroke order problems and the total number of strokes. The proposed system use methods to extract seven feature from Chinese characters to detect Chinese handwriting errors. They also proposed a computer-based assessment tool [16] that measure a stroke order of the Chinese Character handwriting based on gray measure scoring. Hu et al. [19, 20] 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 to detect. Neo Chin et al. [21, 22] 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. Priyankara et al. [23] proposed A Latin and Sinhala e-learning tool 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. Tang et al. [24] 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. In this study, we have developed an android system called as "AKT" 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 AKT: MOBILE HANDWRITING LEARNING SYSTEM

Our proposal, AKT, is an intelligent mobile platform assist Arabic children to learn the character handwriting. The AKT system was developed to detect handwriting alphabet stroke errors. In the next section we will describe the Arabic language to determine the characteristics of the language. Arabic is a kind of Semitic language used widely as a mother language of millions people [25]. Countries throughout the Middle East and northern and north-eastern Africa, and Classical Arabic use Arabic as the official or national language [26]. 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. we analysis each Arabic character to produce a major five key parameters (1) stroke number, (2) stroke order, (3) stroke likeness, (4) stroke direction (5) Dots stroke number. Our analysis helped to build a strong system to detect handwriting errors.

ISSN 2347-4289

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

Fig. 1. *Arabic alphabets*

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. AKT system accepts points of the written Arabic character stroke in x-y coordinate from a digital touch device. In next part we will explain two main parts of the AKT system (1) AKT architecture (2) AKT components.

3 AKT Architecture

The main architecture of AKT consists of three interface components: children interface, tutor interface, learning interface. Moreover, AKT consist of three agent components: learning agent, feedback agent, statistical agent. Finally, AKT has one database contain four tables: children table, tutor table, learning table, feedback table. Conceptually, the relationship among the components: AKT interface, AKT agents and AKT 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. Children log to AKT system through the children 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 learning interface. Now, children choose the learning character from the interface.
- Step 4. The system loads the learning agent. The learning agent loads some tools for writing, deleting and creating new tracer.
- Step 5. The learning agent gets the contents of learning Arabic alphabet character from the learning table and exhibits them for the children.
- Step 6. The children 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 learning 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 teacher, who teaches these children.
- Step 12. Tutor log to AKT system through tutor 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 result agent obtains all data from feedback table.
- Step 16. The result agent show a statistical result about each child and all children related to tutor.

4 System Components

The Model–view–controller (MVC) [27] is a software architectural pattern used to design a software system. The AKT 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 AKT Interfaces

4.1.1 Children Interface

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

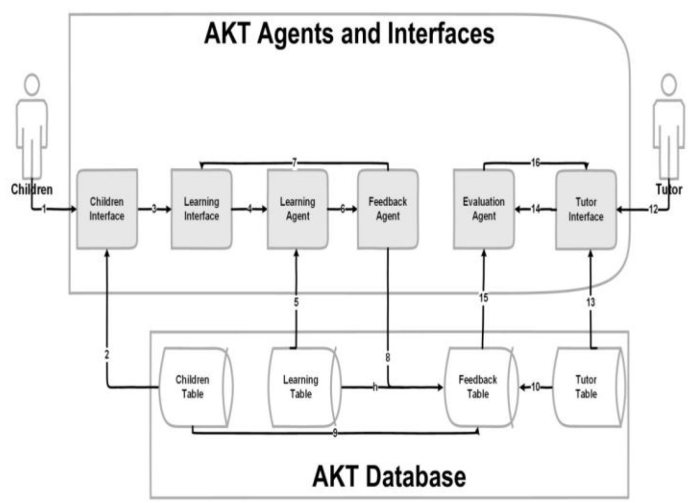
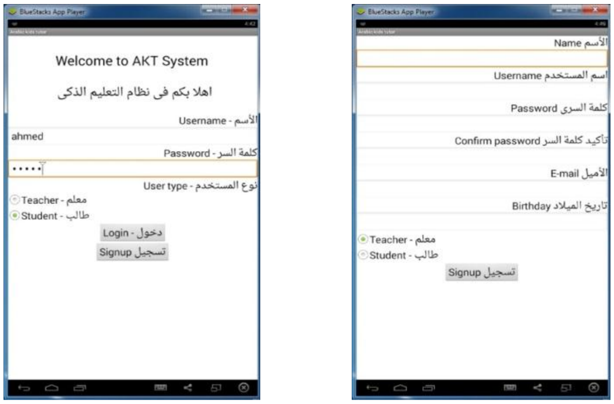


Fig. 2. *AKT system architecture*

4.1.2 Learning Interface

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

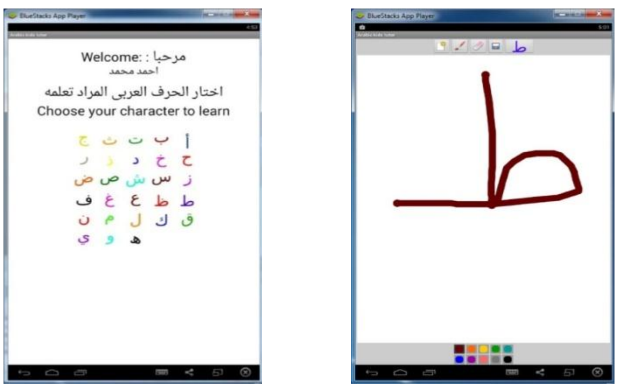


(a) Sign in screen *(b) sign up screen*

Figure 3. *Children and Tutor interface*

4.1.3 Tutor Interface

The Tutor interface provides a login and register interface to interact with tutor, the interface was illustrated in Fig. 3(a,b). Through the tutor interface agent, tutor can get a statistical result about children performance.



(a) welcome screen *(b) learning screen*

Figure 4. *AKT wellcome and learning screen*

4.2 AKT Database

The AKT database contain four main tables: (1) children table (2) tutor table (3) learning table (4) feedback table , our database design was illustrated in Fig. 5.

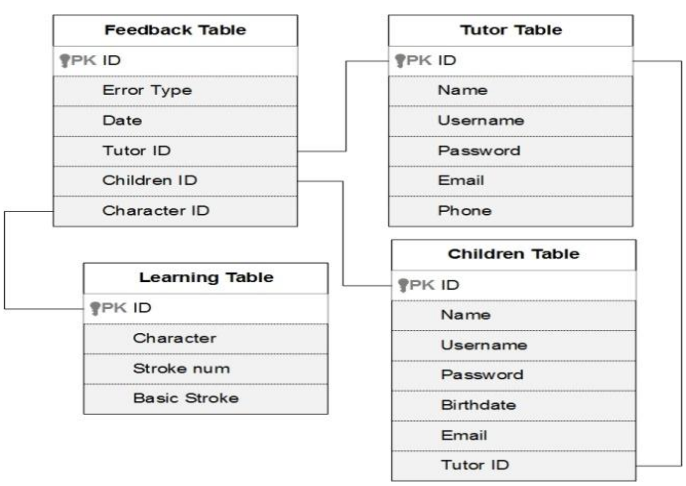


Fig. 5. *Arabic alphabets stroke order*

4.3AKT Agents

AKT 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 Learning Agent

In this part, learning agent load knowledge data from learning table that contained the characteristic of the chosen Arabic character. Moreover, learning agent load tools and the interface that interact with children. Learning agent based on four components (1) stroke number , (2) stroke likeness, (3) stroke order, and (4) stroke direction.

4.3.1.1 Stroke Number

In our method, we classify Arabic alphabet characters to four categories based on stroke number. Therefore, Our classification of Arabic characters 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 teh (ت) has three strokes (ب) and two dots (.), so any stroke more this three strokes is an extra stroke. 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.

4.3.1.2 Stroke Order

In this research, AKT used a stroke order feature to write any Arabic characters correctly. Our AKT system has a stroke counter to detect stroke order and extra stroke. Therefore, Arabic alphabet character thal (ذ) has two strokes (د) and (.). Stroke dal (د) wrote before stroke dot (.). When preschool children wrote stoke dot (.) before stroke dal (د), the system will give quick feedback with stroke order errors detection. Order of stroke in Arabic characters illustrated in Fig. 6.

4.3.1.3 Stroke Likeness

In this part, we classify Arabic character based on similarity of major stroke, stroke likeness 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).

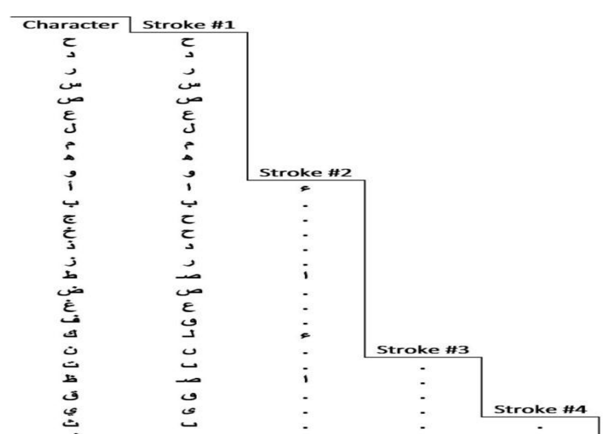


Fig. 6. *Arabic alphabets stroke order*

ISSN 2347-4289

4.3.1.4 Stroke Direction

Any character has a special direction to represent each stroke, the direction of Arabic alphabet characters shown in Fig. 7. Arabic alphabet character thal (ذ) has two strokes (د) and (.). Preschool children must write stroke (ذ) in a sequence down-left. Stroke (.) should write after stroke (د) and above stroke (د). When preschool children wrote dal (د) in an inverse sequence right-up. Our AKT system pop up error message that says wrong direction, then remove wrong stroke.

TABLE 1. *ARABIC ALPHABET CHARACTERS SIMILARITY*

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

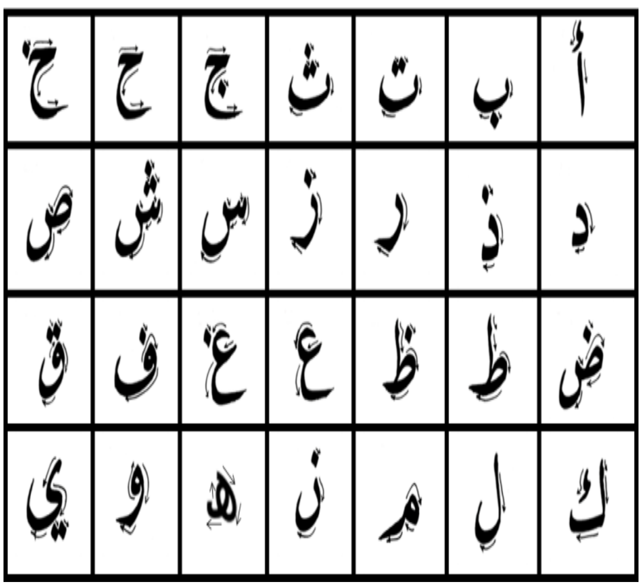


Fig. 7. *Arabic alphabets stroke direction*

4.3.1.5 Dot Strokes Number

In Arabic language, dot is 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.

4.3.2 Feedback Agent

The main part of AKT system is feedback agent, feedback agent is the brain of AKT. In this part, feedback agent diagnoses and fixes many type of Arabic character stroke error. Moreover, AKT 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 AKT 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 neighbouring 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. *ANGLE THAT REPRESENTS THE DIRECTION CODE*

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

4.3.2.1 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 $P_i=(x_i,y_i)$ such that i=[0,1,...,m-1,m] where m number of x-y coordinate. Each stroke had number of points as $S_n= [p_0,p_1,...,p_{m-1},p_m]$, where S is stroke, P is point in x-y coordinate, n is number of stroke.

4.3.3 Evaluation agent

In this part, AKT indicates the children level of understanding of learning handwriting character concepts. In AKT, we 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 AKT system have a form similar to the one in Fig. 8. Moreover, the generalized bell-shaped membership function is describing the linguistic variables of input and output for the fuzzy inference of AKT system. The bell-shaped membership function can be defined as follows:

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

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.

IF

(Character-Diffucullity is Very-Difficult)

and

(Character-Error is Very-Small)

and

(Time-Consumed is Short)

and

(Children-Age is Young)

Then

(Children-Evaluation is Excellent)

Fig. 8. *Example of a rule used*

In Fig. 9a, 'very-difficult' 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. 9a). 'Very small' 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 (Fig. 9b). The 'short' 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. 8c) 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. '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. 8d) 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. 9d).

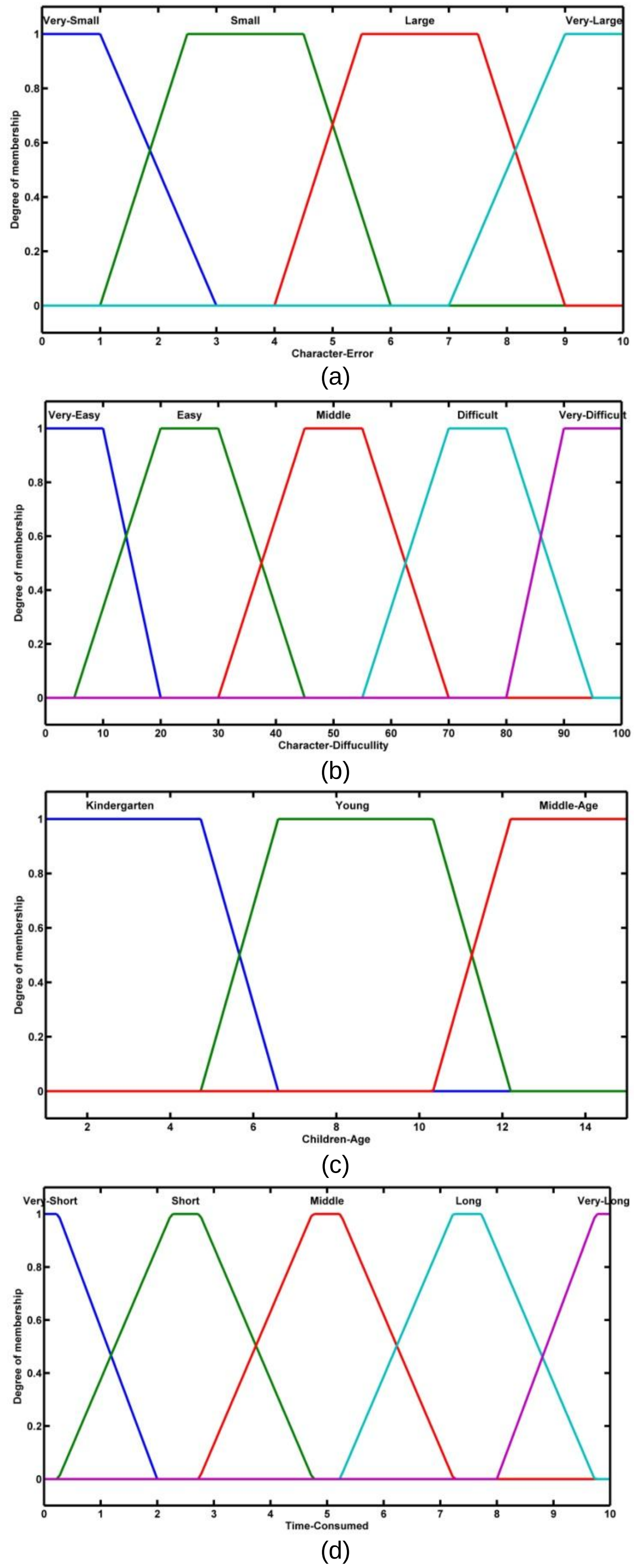


Fig. 9. *AKT input fuzzy set*

'Excellent' 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). By analysing the behaviour of the AKT 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,

ISSN 2347-4289

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 [30].

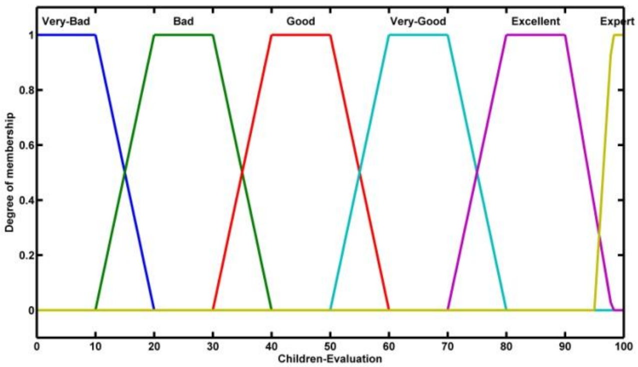


Fig. 10. *AKT fuzzy set output*

In our AKT, we have successfully evaluate children handwriting by encoded a set of fuzzy rules using the jfuzzylogic Toolkit [31, 32]. 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.

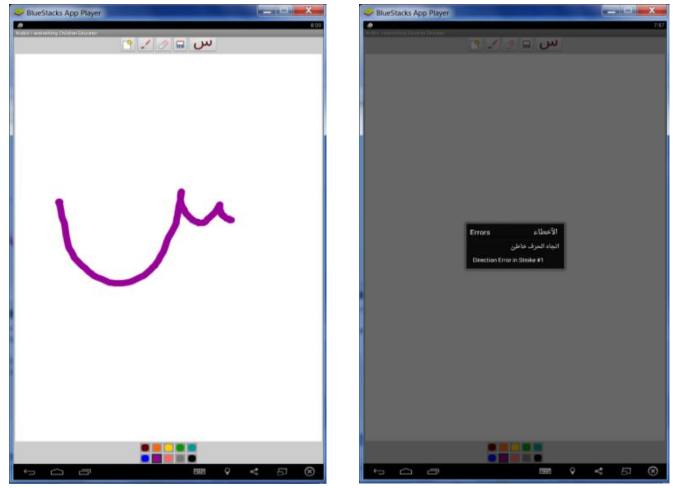
5 Result

5.1 Experimental Environment

Through the work, the proposed prototype of AKT is implemented on the platform of Android. Moreover, the front-end script language of java is used to implement this system. The AKT 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 10 64bit operating system with an Intel® core i7 CPU(2 GHz) and 8GB of RAM.

5.2 Experimental Test

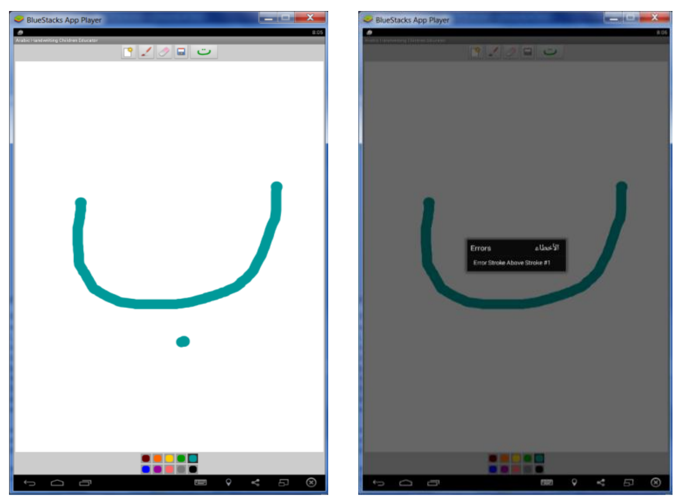
In this part, we tested AKT with a set of inputs containing all Arabic alphabet letters using a tablet and Smartphone's. The evaluation of AKT was performed in our faculty of Computer Science and Informatics, Benha University, Qalyubiyah, Egypt. The AKT system performs analysis to preschool children's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



(a) direction from left to right *(b) AKTsystem feedback*

Figure 11. *Directional stroke error of character Seen*

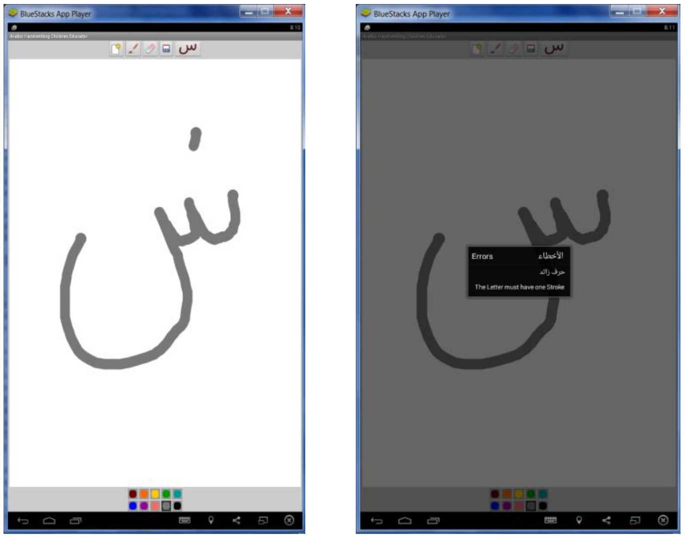
The Egyptian preschool children and their tutors from Benha city were asked to use the AKT as part of their learning of write Arabic alphabet. After the learning was completed, the children and tutors were interviewed. A part of the questions were asked to the educator illustrated in Table 3. 100 questionnaires were collected from the educators in the experimental groups after the computerized training program. The values of Table 4 are the result of analyses of the questionnaire. The educators and children found the whole handwriting learning software application satisfiable and useful. The results specify that educators rated the AKT very highly on acceptability for both likeability and ease of use.



(a) stroke under (ب) *(b) AKTfeedback*

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

ISSN 2347-4289



(a) extra stroke above seen (س) *(b) AKTfeedback*

Figure 13. *Extra stroke error of Arabic character Seen*

TABLE 3. *A PART OF THE QUESTIONNAIRE THAT WERE ASKED*

1	Improve children handwriting		
	(a) Disagree	(b) Neutral	(c) Agree
2	Easy to learn		
	(a) Disagree	(b) Neutral	(c) Agree
3	Easy to use		
	(a) Disagree	(b) Neutral	(c) Agree
3	Interactive		
	(a) Disagree	(b) Neutral	(c) Agree

TABLE 4. *THE RESULT OF QUESTIONNAIRE*

	Agree	Neutral	Disagree
1	85%	10%	5%
2	90%	5%	5%
3	90%	5%	5%
4	75%	10%	15%

6 CONCLUSION AND FUTURE DIRECTIONS

This study proposed an intelligent tutor system called Arab Kids Tutor (AKT) based on automatic error detection and correction. Through the work, we have discussed the system architecture and discuss the system components. The AKT is a multi-agent system designed by using Model–view–controller (MVC) model. In the work, we have built and developed AKT system that supported immediate feedback that detect and fix stroke order and direction errors. Moreover, the proposed system has successfully reduced time and effort for Arab tutors and children. For the future work, we will continue to improve our system to detect all types of Arabic handwriting learning errors.

REFERENCES

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

[2] R. P. Sanchez, C. M. Bartel, E. Brown, and M. DeRosier, "The acceptability and efficacy of an intelligent social tutoring system," Computers & Education, vol. 78, pp. 321-332, 2014.

[3] Z. Jeremić, J. Jovanović, and D. Gašević, "Student modeling and assessment in intelligent tutoring of software patterns," Expert Systems with Applications, vol. 39, pp. 210-222, 2012.

[4] M. Badaracco and L. Martínez, "A fuzzy linguistic algorithm for adaptive test in Intelligent Tutoring System based on competences," Expert Systems with Applications, vol. 40, pp. 3073-3086, 2013.

[5] R. Z. Cabada, M. L. Barrón Estrada, and C. A. Reyes García, "EDUCA: A web 2.0 authoring tool for developing adaptive and intelligent tutoring systems using a Kohonen network," Expert Systems with Applications, vol. 38, pp. 9522-9529, 2011.

[6] J. Cardoso, G. Bittencourt, L. Frigo, E. Pozzebon, and A. Postal, "Mathtutor: A Multi-Agent Intelligent Tutoring System," in Artificial Intelligence Applications and Innovations. vol. 154, M. Bramer and V. Devedzic, Eds., ed: Springer US, pp. 231-242, 2004.

[7] Y. Sun and Z. Li, "A multi-agent intelligent tutoring system," in Computer Science & Education, 2009. ICCSE '09. 4th International Conference on, pp. 1724-1728, 2009.

[8] E. Sierra, A. Hossian, P. Britos, D. Rodriguez, and R. Garcia-Martinez, "A Multi-agent Intelligent Tutoring System for Learning Computer Programming," in Electronics, Robotics and Automotive Mechanics Conference, 2007. CERMA 2007, pp. 382-385, 2007.

[9] E. Lavendelis and J. Grundspenkis, "MASITS – A Tool for Multi-Agent Based Intelligent Tutoring System Development," in 7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009). vol. 55, Y. Demazeau, J. Pavón, J. Corchado, and J. Bajo, Eds., ed: Springer Berlin Heidelberg, pp. 490-500, 2009.

[10] K. Günel and R. Aşlıyan, "Extracting learning concepts from educational texts in intelligent tutoring systems automatically," Expert Systems with Applications, vol. 37, pp. 5017-5022, 2010.

[11] M. Arevalillo-Herráez, D. Arnau, and L. Marco-Giménez, "Domain-specific knowledge representation and inference engine for an intelligent tutoring system," Knowledge-Based Systems, vol. 49, pp. 97-105, 9// 2013.

[12] J. Marciniak, "Building Intelligent Tutoring Systems Immersed in Repositories of E-learning Content," Procedia Computer Science, vol. 35, pp. 541-550, 2014.

[13] M. A. Abbas, W. F. W. Ahmad, and K. S. Kalid, "OntoCog: A Knowledge based Approach for Preschool Cognitive Skills Learning Application,"

Procedia - Social and Behavioral Sciences, vol. 129, pp. 460-468, 2014.

[14] M. C. Angelides and R. J. Paul, "Developing an intelligent tutoring system for a business simulation game," Simulation Practice and Theory, vol. 1, pp. 109-135, 1993.

[15] Z.-H. Hu, Y. Xu, L.-S. Huang, and H. Leung, A Chinese Handwriting Education System with Automatic Error Detection vol. 4, 2009.

[16] C. Guey-Shya, J. Yu-Du, W. Der-Bang, and C. Hei-Tsz, "Stroke order computer-based assessment with grey measure scoring," in Grey Systems and Intelligent Services, 2009. GSIS 2009. IEEE International Conference on, pp. 763-768, 2009.

[17] K.-T. Tang, K.-K. Li, and H. Leung, "A Web-Based Chinese Handwriting Education System with Automatic Feedback and Analysis," in Advances in Web Based Learning – ICWL 2006. vol. 4181, W. Liu, Q. Li, and R. W.H. Lau, Eds., ed: Springer Berlin Heidelberg, pp. 176-188. , 2006

[18] C. Guey-Shya, J. Yu-Du, and L. Li-Fon, "Computer-based Assessment for the Stroke Order of Chinese Characters Writing," in Innovative Computing, Information and Control, 2007. ICICIC '07. Second International Conference on, pp. 160-160, 2007.

[19] Z. Hu, H. Leung, and Y. Xu, "Stroke Correspondence Based on Graph Matching for Detecting Stroke Production Errors in Chinese Character Handwriting," in Advances in Multimedia Information Processing – PCM 2007. vol. 4810, H. S. Ip, O. Au, H. Leung, M.-T. Sun, W.-Y. Ma, and S.-M. Hu, Eds., ed: Springer Berlin Heidelberg, pp. 734-743, 2007.

[20] Z. H. Hu, Y. Xu, L. S. Huang, and H. Leung, "A Chinese handwriting education system with automatic error detection," J. Softw. Journal of Software, vol. 4, pp. 101-107, 2009.

[21] C. Neo Chin, E. S. L. Ming, P. I. Khalid, and F. Yeong Che, "Algorithm for Identifying Writing Stroke and Direction," in Computational Intelligence, Modelling and Simulation (CIMSiM), 2012 Fourth International Conference on, pp. 94-98, 2012.

[22] C. C. Neo, E. L. M. Su, P. I. Khalid, and C. F. Yeong, "Method to Determine Handwriting Stroke Types and Directions for Early Detection of Handwriting Difficulty," Procedia Engineering, vol. 41, pp. 1824-1829, 2012.

[23] K. W. T. G. T. Priyankara, D. C. Mahawaththa, D. P. Nawinna, J. M. A. Jayasundara, K. D. N. Tharuka, and S. K. Rajapaksha, "Android based e-Learning solution for early childhood education in Sri Lanka," in Computer Science & Education (ICCSE), 2013 8th International Conference on, pp. 715-718, 2013.

[24] W. W. W. Tang, H. V. Leong, G. Ngai, and S. C. F. Chan, "Detecting handwriting errors with visual feedback in early childhood for Chinese characters," presented at the Proceedings of the 2014 conference on Interaction design and children, Aarhus, Denmark, 2014.

[25] Y. Elarian, I. Ahmad, S. Awaida, W. G. Al-Khatib, and A. Zidouri, "An Arabic handwriting synthesis system," Pattern Recognition, vol. 48, pp. 849-861, 2015.

[26] J. Hackett, "Semitic Languages," in Encyclopedia of Language & Linguistics (Second Edition), K. Brown, Ed., ed Oxford: Elsevier, pp. 229-235, 2006.

[27] A. Leff and J. T. Rayfield, "Web-application development using the Model/View/Controller design pattern," in Enterprise Distributed Object Computing Conference, 2001. EDOC '01. Proceedings. Fifth IEEE International, pp. 118-127, 2001.

[28] Y. Kui Liu and B. Žalik, "An efficient chain code with Huffman coding," Pattern Recognition, vol. 38, pp. 553-557, 2005.

[29] L. A. Zadeh, "Fuzzy sets," Information and Control, vol. 8, pp. 338-353, 1965.

[30] H.-J. Zimmermann, Fuzzy set theory\—and its applications (3rd ed.): Kluwer Academic Publishers, 1996.

[31] P. Cingolani and J. Alcalá-Fdez, "jFuzzyLogic: a Java Library to Design Fuzzy Logic Controllers According to the Standard for Fuzzy Control Programming," International Journal of Computational Intelligence Systems, vol. 6, pp. 61-75, 2013.

[32] P. Cingolani and J. Alcala-Fdez, "jFuzzyLogic: a robust and flexible Fuzzy-Logic inference system language implementation," in Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on, pp. 1-8, 2012.