مجلة العلوم البحتة والتطبيقية
Journal of Pure & Applied Sciences
www.Suj.sebhau.edu.ly ISSN 2521-9200
Received 26/04/2019 Revised 17/08/2019 Published online 17/11/2019

Off-line Optical Character Recognition System for Arabic Handwritten text

Mohamed A. Fadeel
Department of Computer Science/Sabha University, Libya
Corresponding author: fadeel1@sebhau.edu.ly

Abstract This paper introduces an Optical Character Recognition (OCR) system, where different stages; pre-processing, thinning, segmentation, features extraction, classification and recognition were designed, tested and implemented in the overall system. Learning Vector Quantization algorithm was first used as classifier for Arabic handwritten characters recognition. Two classification strategies were performed; one classifier for all Arabic Alphabets and three classifier (one for ascenders, one for descenders and one for embedded). The later strategy was adopted in classification stage of the proposed system. Numbers of experiments were run on different samples of Arabic handwritten scripts for different writers, a very satisfactory recognition accuracy was obtained.

Keywords: OCR system, handwritten recognition, Neural Networks, Features extraction, classification.

نظام التعرف بالحاسوب على الحروف العربية المكتوبة بخط اليد

محمد عبدالسلام علي فضيل

قسم علم الحاسوب – كلية تقنية المعلومات – جامعة سبها، ليبيا

للمراسلة: fadeel1@sebhau.edu.ly

الخلاص تقدم هذه الورقة نظام التعرف الضوئي على الحروف (OCR) ، حيث توجد مراحل مختلفة ؛ تم تصميم واختبار وتنفيذ ما قبل المعالجة ، والترقيق، والتجزئة، واستخراج الميزات، والتصنيف و التعرف في النظام العام. تم استخدام خوارزمية Learning Vector Quantization لأول مرة كمصنف للتعرف على الأحرف العربية المكتوبة بخط اليد. تم تنفيذ استراتيجيات التصنيف في طريقتين؛ مصنف واحد لكل الحروف الهجائية العربية وثلاثة مصنفات (واحد للحروف الصاعدة، وواحد للحروف النازلة والآخر للحروف الواقعة على خط الكتابة). تم تبني الاستراتيجية اللاحقة في مرحلة تصنيف النظام المقترح. تم إجراء عدد من التجارب على عينات مختلفة من النصوص العربية المكتوبة بخط اليد لكُتاب مختلفين، وتم الحصول على دقة تمييز مرضية للغاية.

الكلمات مفتاحية: نظام التعرف الضوئي على الحروف، التعرف على خط اليد، الشبكات العصبية، استخراج الميزات، التصنيف.

Introduction

Arabic Off-line handwriting character recognition has been a difficult problem to machine learning. It is hard to mimic human classification where specific writing features are utilized [1]. Recent surveys have shown that present technology has still a long way to catch up in terms of robustness and accuracy [1,9]

Although it was briefly mentioned that recognition systems for reading handwriting have existed for many years, the enthusiasm for developing new and improved systems for recognizing Arabic handwritten characters and words has not been extinguished[1], [3]. Character recognition can be used to automate many tasks that previously require manual human interaction, such as mail sorting, reading of amount on bank checks, office automation, data entry and many more.

Handwritten Samples Database Sets

The developed Arabic text recognition system has been tested using Arabic handwritten samples collected from 63 writers, most of them are Arabic native, three of them are written by Arabic non-native writers. Database has two types of samples; 63 handwritten sample forms and 21 samples of each Arabic character basic-shapes CBS (i.e. 21x53=1113 characters). Arabic character basic-shape CBS is a skeleton of a character without dot/dots or madda, (e.g. س is CBS of س and ش, likewise, ف is CBS of ف and ق). Each handwritten sample form has a paragraph consists of 43 words; word in turn may consist of two or more sub-words. The total number of sub-words of that paragraph is 88 sub-words, which equivalent to 176 characters. Some constrains are imposed on all writers e.g. writing with a pen not a pencil, writing in regular font size, and to write in a proper way.

Pre-processing

The images to be tested are captured using a 1200 dpi scanner, they are stored and passed to the system as a bitmap files. The captured gray-scale image is changed to binary image. The use of a black-and-white representation found to be greatly simplifies processing in the rest of the system modules [5].

During the digitization process, some spurious pixels may result in the character image. These pixels are noise pixels that add irregularities to the outer boundary of the characters and may have undesired effects on the recognition system. For the purposes of OCR, Arabic text is extremely sensitive to “salt and pepper” noise and speckle noise [4], than are Latin-character-based languages because some Arabic characters share a common body (CBS) and are differentiated mainly by the location and the presence or absence of dot/dots. A smoothing algorithm [11] is

implemented to eliminate the noise of the binary image.

Segmentation

Three segmentation steps are performed on the database sets; text-lines segmentation, text-line to words/subwords segmentation and words/subwords to characters segmentation. The first and second segmentation steps are performed with high accuracies, however the third step is performed with lower accuracy. Nevertheless, the words/subwords-to-characters segmentation algorithm is considered very satisfactory compared with other algorithms found in the literature [5].

A modified version of base-line technique called Base-Area has been introduced to assist in word/subword-to-character segmentation process [6]. Base-area (instead of base-line) as shown in Figure (1) is used as a reference for defining ascending, descending and embedded characters of a word, so that these characters can be easily segmented. Ascending characters are expected to be in the area between the upper limit of base-area and the line delineating the top of the word. Descending characters are expected in between the lower limit of base-area and the line delineating the bottom-most area of the word. Base-Area for each text line of the input paragraph is estimated, empirically, it has been found that the height of base-area is almost equal to 40% of the text-line height. It was also found that the spans of upper and lower reference lines from the base-line are respectively equal to 63% and 37% of the base-area width.

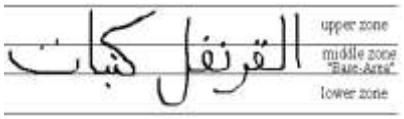

Figure 1 base-area used todetect ascending, aescending and embedded characters

Table 1 illustrates the three segmentation steps along with their accuracies when they are applied on different datasets.

Table 1 Segmentation algorithms vs. their accuracies for different datasets

Segmentation type	Selected neat hand-writing dataset	Dataset of various handwriting styles
Text-lines segmentation	97-100 %	90-93 %
Text-line to words/subwords	95-99 %	80-87 %
cWord/subword to characters	73-75 %	54-59 %

Text thinning

An efficient thinning algorithm was designed and tested on different datasets of complete text-line, word/subword and isolated characters. Very clean, smooth and mid-line character skeletons are obtained as an output of thinning process, skeletons having these merits have improved the performances of segmentation and features extraction processes.

The algorithm was tested on different Arabic handwritten in both cases discrete and cursive scripts. A preserved smooth skeleton was obtained. Figure 2 shows an example of test carried out on an Arabic handwriting word "محمد" images along with their output skeletons. Moreover, the figure clearly shows how a skeleton of this word has a shape reserved, smooth, intermediate and one pixel width line of the original image when they are superimposed.

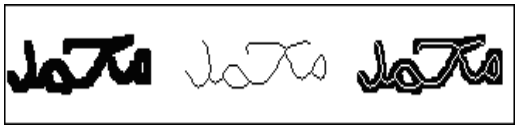

FIGURE 2 Samples of original Arabic handwritten images and their skeletons

Features Extraction

Features extraction is one of key modules in the proposed Arabic OCR system. This module was carefully designed so that the features collected are fully and efficiently representing the character under process. This module produces a data compatible with the requirements of the classifier input as shall be seen in the next section. The collected core features are based on pixels orientations according to Freeman chain code. The input to this module is number of Arabic characters-basic-shapes CBSs. The output data file of features extraction module is shown in Figure3

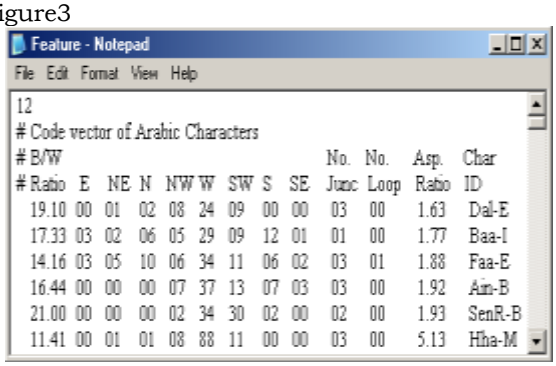

FIGURE 3 Sample of data file contains features extracted from different Arabic characters

Classification

In this module, Learning Vector Quantization LVQ algorithm for neural network is first time introduced as a classifier for Arabic handwriting. Classification has been performed in two different strategies, in first strategy, we use one classifier for all 53 Arabic CBSs in training and testing phases, in second strategy we use three classifiers and three subsets of 53 Arabic CBSs, the three subsets of Arabic CBSs are; ascending CBSs, descending CBSs and embedded CBSs. Three training algorithms; OLVQ1, LVQ2 and LVQ3 were examined, and OLVQ1 found as the best learning algorithm.

First strategy

In this strategy, as we mentioned earlier, all 53 Arabic CBSs are used as one sets for training and testing phases for one LVQ network. A total of 21 samples of each CBS are used in learning and testing steps, 11 samples of each CBS are taken as training set whereas the rest 10 samples of each CBS are used as testing set. Each CBS of training and testing sets are translated into code-vector by features extraction module and stored into two different data file (like the one shown in Figure 3), one data file contains code-vectors of all CBSs belong to training set, the other data file contains code-vectors of all CBSs belong to testing set. The same classification steps are followed [10]; we start by initialization of codebook-vectors, then the medians of the shortest distances between the initial codebook vectors of each class are computed (balancing), then we trained the LVQ network by specifying different combinations of learning parameters like; input and output files, learning rate $\alpha_{i(t)}$, number of codebook vectors, code-vectors for each class (CBS), number of iterations and number of k-NN. Finally we tested the network and monitor the recognition accuracies for each individual class (CBS) as well as the classifier overall recognition rate. The overall recognition accuracy of this classifier is 83.2 %. Graphical illustration of the performance of this classifier is shown in Figure 4.

Second strategy

This strategy is based on numbers of hypothesis and notions we started with. First; the number of Arabic characters (in four forms) is about 102 characters, and even if the basic shapes of these characters (53 CBSs) are used, the possibility of finding two or more of these basic shapes being morphologically close to each other is not ignorable. Second, the more the number of classes to be classified the more the complication of network design and programming are claimed. Third, tasks subdivision facilitates parallel programming, fast debugging and system development.

Based on what have been mentioned above, it has been decided to use three classifiers instead of one, the 53 CBSs were divided into three groups based on how each CBS located in a word with respect to base-area in the text-line, the three groups are ascenders, descenders and embedded, and accordingly the three classifiers are; ascenders classifier, descender classifier and embedded classifier. The same training and testing procedures - followed in case of first strategy - is applied for each individual classifier in this strategy.

In this stage, in particular, CBSs are sequentially retrieved from word/subword-to-characters module and fed into features extraction module first, then according to knowledge source CP-KS, shown in Table 2, it is fed into either ascenders, descenders or embedded classifier. The classified CBS is then fed to the next stage (composer) for further processing. Each individual classifier is trained and tested with its correspondent dataset; the detailed result of each classifier is given in Appendix A. Charts illustrate the recognition accuracy of each classifier are given in Figure 5, Figure 6 and Figure 7.

The overall recognition accuracies of ascenders, descenders and embedded classifier are 92.21 %, 88.24 % and 83.2 % respectively. The average recognition rate of the three classifiers is 89.1 % which is higher than the recognition rate of the first strategy classifier (which is 83.83 %). From other prospective, by noticing the three charts given in Figures 5, 6 and 7, one can conclude that as the number of CBS classes' increases the recognition accuracy decreases, and this emphasize our presumption of dividing the CBSs set into three categories and using three classifiers, one classifier for each category.

Word Composition

The final stage of the proposed Arabic OCR system is word composition. Reading can be interpreted as systematic execution of a number of processing steps, such as word encoding, lexical access, assigning semantic roles and relating the information in a given sentence to previous sentence and previous knowledge [2], Several studies have shown that prior knowledge plays a major role in reading text. This lead to a fact that humans use a number of knowledge sources in reading comprehension. In this research similar mechanism was followed to identify various "***knowledge sources***" (KS) for a handwritten Arabic character recognition system. In this stage the main program (Start_Recog()) uses different knowledge sources shown in Table (2) to compose back the discrete CBSs received from the classification stage into a Unicode words and text-lines.

Table 2 Knowledge Sources and their codes

Knowledge Source	KS-Code
Height of Text Lines	HT-KS
Word Coordinates	WC-KS
Word-End characters	WE-KS
Estimated Character Width	CW-KS
Char position w.r.t Base-Area	CP-KS
Statistical Distribution of Chars	SC-KS
Script Composition Rules	CR-KS
Pen thickness estimation	PT-KS
Gaps in text-line	GP-KS
Diacritic Position	DP-KS
Character Coordinates	CC-KS
Base-Area Coordinates	BC-KS

Knowledge Sources

In addition, each CBS's name (e.g. Mem-B) consists of two parts, the first part is the name of one of 28 Arabic Alphabet, whereas the second part is a character comes after the dash which shows whether that character is in the beginning (B), middle (M), or at the end (E) of a word. The following example illustrates how numbers of CBSs and knowledge sources are combined by the composer to form word/s in a text-line.

CBSs sequence received from the classifier:
Alif, Lam-B, Raa-E, Hha-B, Mem-M, Non-E, Alif, Lam-B, Raa-E, Hha-B, Baa-M and Mem-E

Knowledge sources provided by main program Start_Recog() to composer module are:
DP-KS, WE-KS, CC-KS, GP-KS and CR-KS
The Unicode text output of composer module is:

الرحمن الرحيم

System Evaluation

The proficiency of any OCR system is measured mainly by two factors; the character recognition accuracy and the speed at which the OCR system can recognize characters per second [8], [9], our goal in this section is to evaluate the performance of the recognition system. Given a test page of Arabic characters, we define the following:

- Total number of characters in the page (n).
- Number of correctly recognized characters (n_{rec}).
- Number of misclassified characters (n_{mis}).
- Page Time cost (t): It is the total time consumed to process the given page.

Different performance measures are used to evaluate the recognition system. The following is a description of each of these measures:
The Recognition Rate, R_{rate}: It is called also Recognition Accuracy, defined as the ratio of the number of correctly recognized characters to the total number of characters. So, R_{rate} can be expressed in equation (1) as:

$$R_{rate} = \frac{n_{rec}}{n} \times 100 \quad (1)$$

The Recognition Error, R_{eror} : It is the ratio of the number of misclassified characters to the total number of characters. R_{eror} can be expressed in equation (2) as

$$R_{eror} = \frac{n_{mis}}{n} \times 100 \quad (2)$$

The Recognition Throughput, R_{to}: It is the number of processed characters per second (CPS). A more conservative definition excludes misclassified characters. The following throughput function which reports throughput while penalizing for errors is used:

$$R_{to} = \frac{n\text{-}P(n_{mis})}{t}$$

where P represents the penalty assigned to misclassified characters. When P = 0, then, R_{to} represents the raw throughput in terms of characters per second. When P = 1, R_{to} represents correct classified characters per second.

System validation

In order to validate the concept of our modular system as well as to show its robustness, experiments were ran on number of handwritten samples forms available in the used database. For all reported results, the aforementioned definitions of the recognition rate, error rate and throughput were used.
Handwritten paragraphs have been taken from almost all handwritten sample forms of our database; those forms which have paragraphs that have been very badly written - that even human cannot read them fluently - were excluded. The selected paragraphs are further divided into two groups; very neat written paragraph and bad written paragraph. Each paragraph is stored in separate image file (binary bitmap file). All preprocessing steps (in section () above) are applied on each paragraph image, and prepare it to be ready for further processing. Start_Recog() program receives one paragraph image as its input and starts applying the steps mentioned in section (). The recognized text drawn from each paragraph is analyzed, segmentation steps accuracies as well as the recognition accuracy for each individual handwritten paragraph are calculated. Among all handwritten paragraph, averages of different accuracies are computed and displayed in Table 3. The system attained a maximum of 525.7 characters per second during the classification phase. In processing of a paragraph of 176 characters, the time taken to perform other processes like; segmentation steps, features extraction, sequentially retrieving of CBSs and knowledge sources and CBSs-to-text-line/s composition is equal to 14.45 seconds. Now according to the speed of classifier 176 characters need 0.334 second for classification, hence the overall system throughput is 176/(14.45+0.334)=11.9 characters per second.

Conclusion

Number of Arabic handwritten samples are chosen for the test of our Arabic OCR system, each sample contains number of text-lines i.e. paragraph/s. The selected paragraphs were divided into two groups; very neat written paragraph and bad written paragraph. Each paragraph is stored in separate image file. Numbers of experiments were run on these paragraphs where steps like preprocessing, segmentation, features extraction and classification are involved. Recognition rates of 86.95 % and 54.55 % are obtained for very neat written paragraphs and bad written paragraphs respectively, the average recognition rate of our Arabic OCR system is 70.75 %.

References

[1]- Lutfieh S. Alhomed and Kamal M. Jambi, (2018)A Survey on the Existing Arabic Optical Character Recognition and Future Trends, Inte rnational Journal of Advanced Research in Computer and Communication Engineering, Vol. 7, Issue 3, March 2018, pp. 78-88
[2]- Amara, M.; Zidi, K.; Ghedira, K., (2017) An efficient and flexible Knowledge-based Arabic text segmentation approach. Int. J. Comput. Sci. Inf. Secur. Pp. 25–35.
[3]- Radwan, M.A., Khalil. M.I., Abbas, H.M, Neural networks pipeline for offline machine printed Arabic OCR.Neural Process. Lett. 2017, 1–19
[4]- El rube', I.A.; El Sonni, M.T.; Saleh, S.S. Printed Arabic sub-word recognition using moments. World Acad.Sci. Eng. Technol. 2010,4, pp.610–617
[5]- Mohamed A. Fadeel, An Efficient Segmentation Algorithm for Arabic Handwritten Characters Recognition System. (2016) (IEEE) Third Inter. Conference on Mathematics and Computers in Sciences and in Industry, pp. 172-177
[6]- Mohamed A. Ali, "Base-Area Detection and Slant Correction Techniques Applied for Arabic Handwritten Characters Recognition Systems", International Conference on Artificial

Intelligence and Pattern Recognition (AIPR-09), Orlando, USA, (2009) pp. 133-138

[7]- Mansoor Alghamdi, William Teahan, (2017) "Experimental evaluation of Arabic OCR systems", PSU Research Review: An International Journal., pp. 229-241.

[8]- Saber, S., Ahmed, A. and Hadhoud, M., "Robust metrics for evaluating Arabic OCR systems", First Inter. Image Processing, Applications and Systems Conference, IEEE, (2014), pp. 1-6.

[9]- Shimaa Saber ; Ali Ahmed ; Mohy Hadhoud, Robust metrics for evaluating Arabic OCR systems, IEEE International Image Processing, Applications and Systems Conference (2014), Egypt (pp.449-459)

[10]- A. Ali, A. Shaout, and M. Elhafiz, (2015), Two stage classifier for Arabic Handwritten Character Recognition, International Journal of Advanced Research in Computer and Communication Engineering, pp. 646-650,

[11]- Duda, R. & Hart, P.. Pattern Classification and Scene Analysis, John Wiley and Sons, New York. 1973, pp.

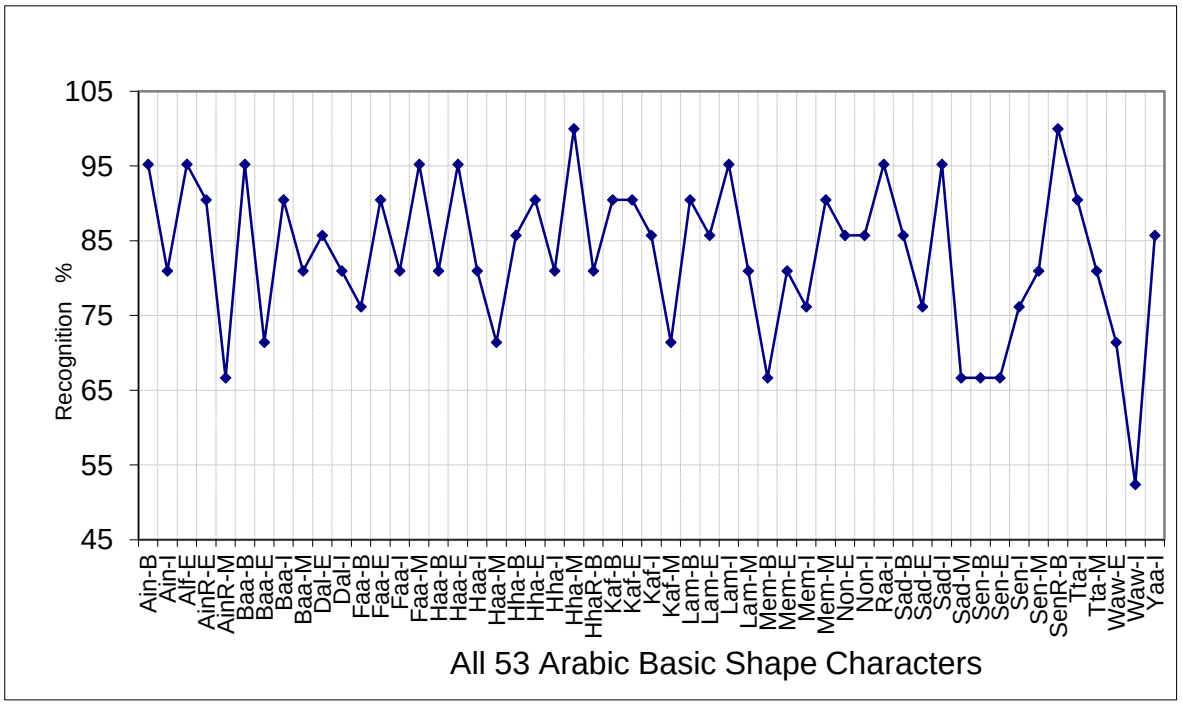

FIGURE 4 Classification output using one classifier for all 53 Arabic Characters Basic Shapes CBSs,

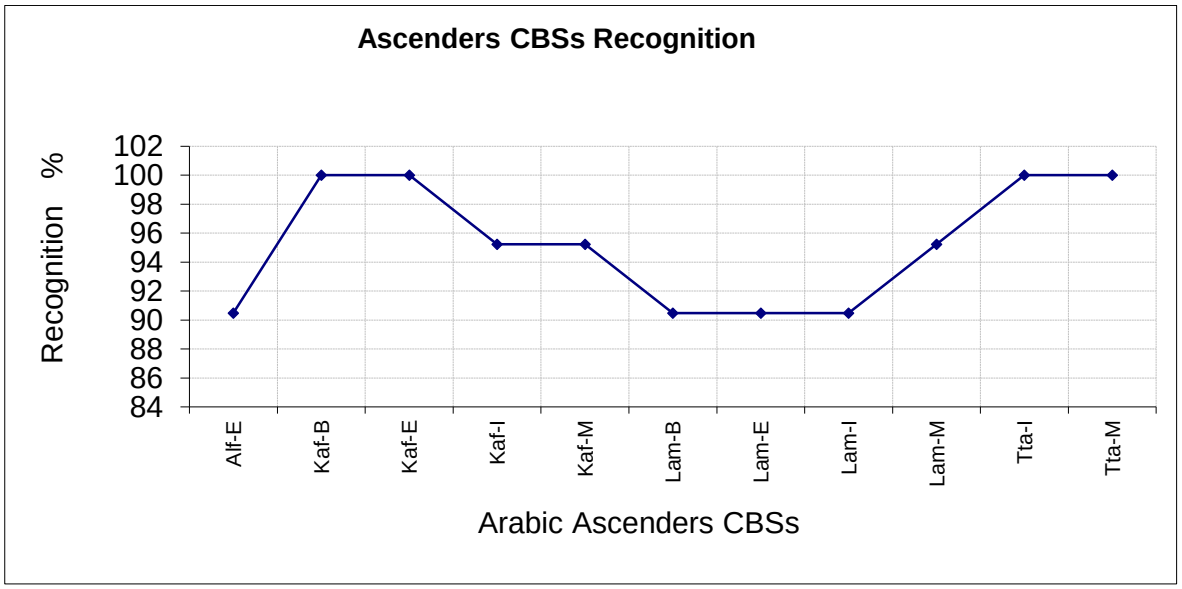

FIGURE 5 Accuracy of Ascenders CBSs using second strategy

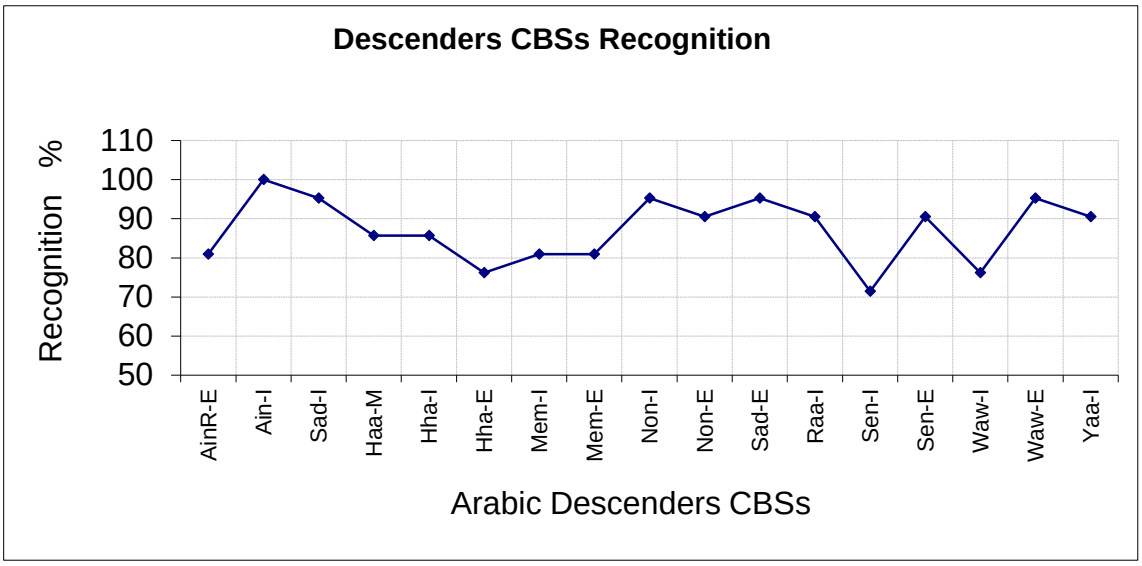

FIGURE 6 Accuracy of Descenders CBSs using second strategy

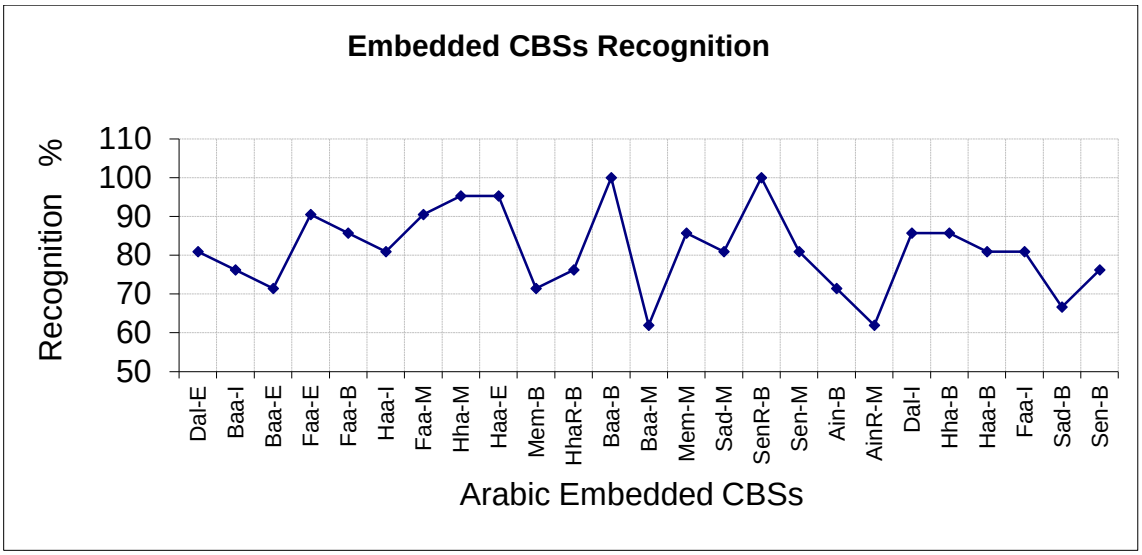

FIGURE 7 Accuracy of Embedded CBSs using second strategy

Table 3 Different accuracies obtained in the validation stage

Accuracy of individual process	11 very neat written samples	3 Bad written samples
Lines segmentation	98-100 %	90-93 %
Line-to-words/subwords segm.	95-99 %	80-87 %
Word-to-characters segm.	73-75 %	54-59 %
Recognition/specific samples	86.95 %	54.55 %
System recognition	70.75	