Arabic Handwriting Synthesis

Yousef S. Elarian, Husni A. Al-Muhtaseb, and Lahouari M. Ghouti
King Fahd University of Petroleum & Minerals
{yarian, muhtaseb, lahouari}@kfupm.edu.sa

Abstract

Training and testing data for optical character recognition are cumbersome to obtain. If large amounts of data can be produced from small amounts, much time and effort can be saved. This paper presents an approach to synthesize Arabic handwriting. We segment word images into labeled characters and then use these in synthesizing arbitrary words. The synthesized text should look natural; hence, we define some criteria to decide on what is acceptable as natural-looking.

The text that is synthesized by using the natural-looking constrain is compared to text that is synthesized without using the natural-looking constrain for evaluation.

1. Introduction

Training and testing data are essential to the development and evaluation of Optical Character Recognition systems (OCRs). Holistic OCRs, for instance, need to be exposed to one sample at least of each entry the system is to recognize. The main advantage of holistic OCRs resides in the simplification or avoidance of character segmentation [1]. Character segmentation of Arabic scripts is considered hard and error-prone [2].

In general, training and benchmarking data are beneficial, but need much time and effort for gathering, scanning and labeling them. Researchers in Arabic OCRs have stated that the absence of standard testing databases is a main cause for the lagging-behind of research in the field [3-8]. Ad-hoc data are frequently used by individual researchers; which inhibit direct comparisons of researchers' results. A standard way of generating training and testing databases can be very beneficial.

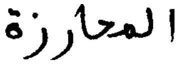
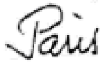
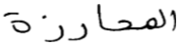
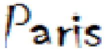
The rest of this paper is organized as follows: Section 2 provides necessary background on the Arabic scripting system. Section 3 presents a literature survey on handwriting synthesis. Section 4 presents the methodology of our work. Section 5 is devoted to show and discuss results. Section 6 addresses conclusions and future work.

2. Background on Arabic script

Arabic script is cursive in both its handwritten and its printed forms. Besides, Arabic script cursiveness obeys well-defined rules: some letters of the alphabet are never connected to their successors while others link to their within-word successors by a horizontal connection line called *Kashidah* [3]. One letter, *Hamzah*, prevents the previous letter from connections.

Because Arabic script follows clear rules in the connection and separation of characters, it is relieved from some ambiguities present in other scripts. Table 1 exemplifies the ambiguities resulting from the arbitrary connection of letters in Latin script and shows how Arabic printing and handwriting match in the connection of character. This feature makes it attractive to synthesize Arabic handwriting through simple concatenation.

Table 1: Examples of Arabic and Latin printed and handwritten words.

	Arabic	Latin
Typed word	المحارزة	Paris
Handwriting 1	 المحارزة	 Paris
Handwriting 2	 المحارزة	 Paris

Another attractive feature is that *Kashidahs* tend to lie horizontally around the baseline of the text. Hence, *Kashidahs* are easy to locate and recognize. Note that Arabic script goes from right to left. Consequently, the

Kashidah preceding a glyph is the right *Kashidah* (RK) and the one following it is referred to as the left *Kashidah* (LK).

In Arabic script, letters can take up to four shapes. The shape that a letter is to take depends on the connectability of it and of the neighboring letters, and on whether the letter is in border position of the word. The four shapes are referred to as: isolated (A), beginning (B), middle (M), and ending (E) [20].

We claim that the glyph shape of the beginning character often resembles the glyph shape of the middle character, except for a small leading extremity. Similarly, the glyph shape of an isolated character often resembles the glyph shape of the ending character, except for the small leading extremity. Figure 1 exemplifies some shapes that obey and don't obey this claim. Utilizing this claim, we can define a relaxed 2-Shape model for most characters. Figure 2 depicts the special segmentation method in this case in comparison to the 4-model method.

Beginning and Middle	يـ ـيـ	نـ ـنـ	مـ ـمـ	كـ ـكـ	صـ ـصـ
Isolated and Ending	ي ـي	ن ـن	م ـم	ك ـك	ص ـص

(a)

Beginning and Middle	هـ ـهـ	عـ ـعـ
Isolated and Ending	ه ـه	ع ـع

(b)

Figure 1: Glyphs in shapes that illustrate their (a) similarity (b) exceptions of similarity cases.

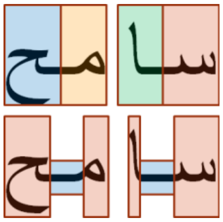


Figure 2: 4-Shape model segmentation (above) and 2-Shape model segmentation (below).

Ligatures might be formed when some Arabic letters come in sequence. Ligatures are symbols replacing a set of letters with a different shape from that of the mere concatenation of the letters. One ligature is obligatory *Lam-Alef*. There are many optional ligatures in Arabic script depending on the handwriting style in handwritten text and the used font in printed text. Figure 3 shows examples of some of these ligatures.

لـ آ	لـ أ
لآ ✗	لأ ✗
لآ ✓	لأ ✓

صـ ر	لـ مـ	بـ ج
صر	لمـ	بج
صر	لمـ	بج

Figure 3: Obligatory (left) and optional (right) ligatures.

3. Literature Survey

Synthesis of handwriting data has been proposed in the literature for several applications ranging from forensics [12-14] to the mere aesthetical touch of human writing [15]. Synthesis of handwriting is often reported in two senses: perturbing data [16-18], and concatenating glyphs [15,19]. We are not aware of any previous work on automatic offline handwriting synthesis for Arabic. Synthesis has been addressed for the levels of characters [21] cursive words [22] and complete texts [23]. Targets have been online as well as printed [25] and handwritten [26] images.

Rao [27] used parametric representations to concatenate letters into cursive words and sub-words. Later, he concatenated primitive elements into characters in a similar way. Although his method is originally designed for online data, it can be extended to treat offline data, as well.

Another parametric model for online character synthesis views the character as the impulse response of a signal [26]. The authors believe that further research is open to enable the concatenation of models of single letters into a single connected word.

A very straightforward, but successful, approach to synthesize connected glyphs makes use of special groups of letters collected from writers on online tablets. The letter group lexicon is selected based on the frequency counts in a linguistic corpus. Input text is parsed into letter groups found in the image lexicon. Images of the corresponding connected components are aligned in juxtaposition to appear as arbitrarily cursive words. This simple approach works well for subjective tests but some limitations cannot be hidden from the trained eye: abrupt pen lifts may appear between glyphs, repetitions of exactly the same glyphs are also possible; and inking may seem too regular. Occasionally, geometric transformations were used to reduce regularity of inking [15].

Varga and Bunke [25] studied the effect of adding perturbed data to the training set of HMM-based OCR. Their method can be applied on the levels of characters, connected components, words, or on complete text-lines. Their results show that synthetic data does achieve improvements.

Miyao and Maruyama [21] also studied the impact of adding perturbed data to the training set of a Hiragna script OCR system. They used models extracted from online data to generate offline data. They tested their work on SVM-OCRs and got improvements on the performance of OCR. Another combination for the treatment of Hiragna letters, performed by Dolinsky and Takagi [29], is the naturalness learning approach. They use printed reference shapes to model the deviations that handwriting samples manifest from it.

Style-preserving English handwriting synthesis from online characters to cursive writing combines several ideas in the field [23]. They gathered data in a special user interface and computed features (borrowed from forensic sciences) of the style of a writer. They use glyph sampling similarly to the work done by Guyon [15], but injecting more shape perturbation and pressure assignment to it. They also add the possibility of connecting neighboring components with polynomial interpolation, in a similar way to the work of [27].

A pretty different goal is approached in [22]. Handwriting that is legible to humans but not easy for machine reading is aimed at. They perform character perturbation, auto-scaling, automatic baseline determination, ligature endpoint detection and concatenation. Their work defines lookup tables and several thresholds. Tests on two OCR systems proved its illegibility to the machine.

As for printed character synthesis, some work has been done [30,31] to substitute costly manpowered ground-truthing of real “printed and scanned” documents by data directly synthesized from its ASCII-code ground truth. Surprisingly, the results of [33] suggest that the cleanest synthesized images were not necessarily recognized most accurately. One justification is that such data may have thin strokes that are one or two pixels width.

Although the work of Margner and Pechwitz [31] is on transferring ASCII codes into printed characters, it gains more importance from several facts: they work on the Arabic language, they implemented IFN/ENIT [32], a popular handwritten dataset, and they conduct biannual competitions and reviews on recognizing IFN/ENIT [30].

Another work [34] for Arabic concatenates online characters into sub-words. They report that the OCR behavior with synthesized data is comparable to that with the corresponding real sub-words, and conclude that the concatenation process works properly.

4. Methodology

We can view the system as to consist of training and synthesis processes. The training process encompasses segmentation, ground truth alignment, and feature extraction. The synthesis process encompasses matching (classification) and concatenation.

Figure 4 shows the block diagram of the image synthesis system. The following subsections detail the implementation of Figure 4. The method needs labeled image samples as input, and produces images as output. However, the number of outcomes it provides can be greater by orders than the number of inputs it takes. This is analogues to recognition systems that need text inscribed samples for their training.

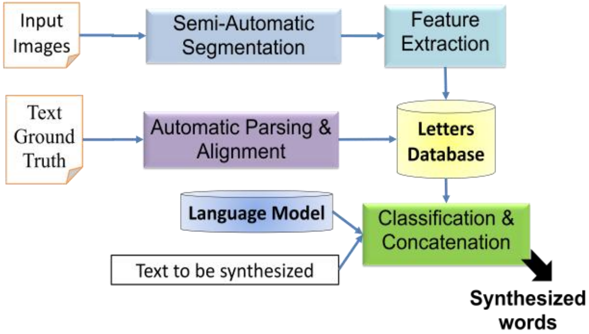


Figure 4. Block diagram showing the steps to synthesize words.

4.1 Data segmentation and labeling

We use the IFN/ENIT database of handwritten Arabic words [32] as a source of input script images. The database has Tunisian town/village names written by different writers. Ground truth information on the Arabic letters, their shapes, the writer ID and the baseline positions is available. Unfortunately, the database doesn't provide images of segmented character.

To obtain segmented characters, we introduce an interactive GUI tool that eases the segmentation of words into characters. The segmentation and labeling block receives images of words and displays them on a GUI tool. The tool accepts hints from the user on the segmentation cut-points and on the *Kashidah* position. The tool automatically associates the character ground truth (taken from the IFN/ENIT ground-truth files) to the segmented images. Finally, the segmented images are saved, along with their labels to be used in subsequent steps.

We can define two models for segmentation/ synthesis. The 4-Shape model implies that segmentation should occur so that *Kashidahs* are parts of characters. The 2-Shape model cuts characters at their borders and considers *Kashidah* as an independent character. The 2-Shape model is more flexible for our work and gives chance for more combinations of styles. In tradeoff, it needs more hints from the user. Since the segmentation step is not yet fully automatic, the less demanding 4-Shape model was chosen for this work.

4.2 Feature Extraction

Upon segmentation, features are extracted from both sides of every *Kashidah* cut (i.e. to the right and to the left of the cut (hereon abbreviated as RC and LC, respectively) within a predefined window size *n*.)

Two kinds of features are used: the width feature (W-feature) and the direction feature (D-feature). The W-feature simply provides the thickness of the cut *Kashidah* in a specific column (in number of pixels). The D-feature finds the difference between the y-coordinates of the centre of gravity (COG) of the *Kashidah* at two consecutive pixel columns.

Each letter sample is associated with a feature vector containing the features for whatever RK or LK is present. Each of these *Kashidahs* has a RC and a LC, as depicted in Figure 5.

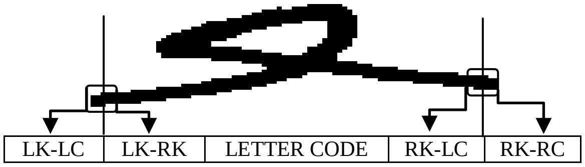


Figure 5. Feature vectors of both sides of the cut *Kashidah* cut.

4.3 Matching and concatenation

The matching step decides on the best sample of a letter to fit in the word being synthesized. The concatenation step forms connected components by aligning character images on their *Kashidahs'* COGs.

The nearest "Euclidean distance" neighbor is used for matching. The character sample with RK matching best the LK of the previous character is chosen. Figure 6 shows the parts that are compared in image and feature domains. Alignment occurs on cut points.

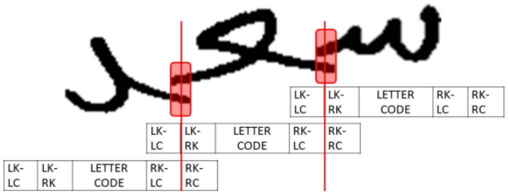


Figure 6. Sequential matching of the word سعد.

Finding the optimal matching needs a number of comparisons that explodes exponentially with the number of letters in the connected component to be synthesized, as indicated by Equation 1. Let n_i be the number of samples we have for the i^{th} letter in a connected component. Then, an exhaustive test that will assure a global minimum sum of matching distances requires a number of comparisons in the order of:

$$\prod_{i=1}^{\text{length of CC}} n_i \qquad \textbf{Equation 1}$$

Instead, we use a suboptimal greedy algorithm. The greedy algorithm starts by finding the best-matching pair of samples for the first two characters of a connected component. Then, the sample of the required character shape that matches best the last sample of the already formed chain is chosen. Once a sample is chosen, it never changes. The number of comparisons in the greedy algorithm reduces to the order of:

$$n_1 \times n_2 + \sum_{i=3}^{\text{length of CC}} n_i \qquad \textbf{Equation 2}$$

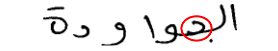
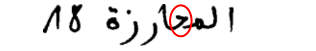
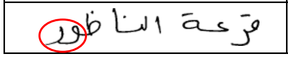
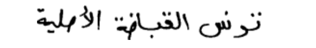
The concatenation step aligns the chosen segmented characters on their *Kashidahs'* cut point COGs. It also adds small white space after non-connectable characters and save the new images in files.

5. Experimentation and discussions

The first step of image segmentation is not fully automated. For that reason, we need to limit the size of our test-bed. Within the IFN/ENIT database, we select 2 writers, each of which contributes with 60 city names (which is the maximum number of city names by a writer in IFN/ENIT). This training serves for the proof of concept rather than being comprehensive.

Examples of the entries used for each writer are shown in Table 2. The circles indicate noisy samples in the inputs. The cases of Writer 1 have extra spikes. The case of Writer 2 shows an incomplete connection of letters (cut *Kashidah*). We refer to these two cases as black and white spikes, respectively.

Table 2. Examples of the handwriting from the two writers contributing in the experiments.

Writer 1	Writer 2
	
	

Examples of the outputs of the program are shown in Figure 7. The width of the space between words is a parameter chosen by the user in the segmentation GUI. Figure 8 depicts an interesting case in which ligatures are needed. The word “سلام” needs the obligatory ligature *Lam-Alef* that the concatenation model cannot reproduce.

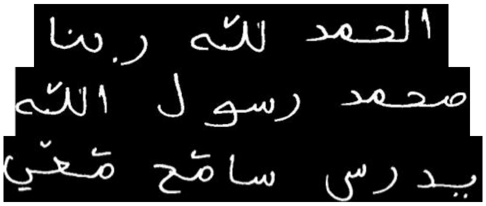


Figure 7. Examples of the output of the program.

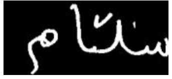
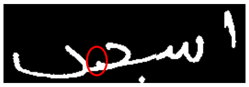
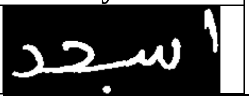
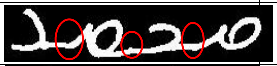
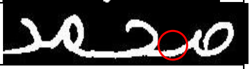
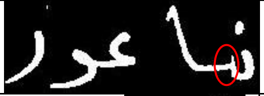
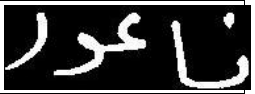
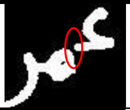
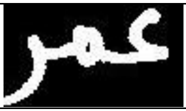
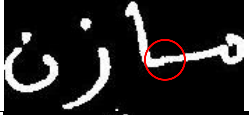
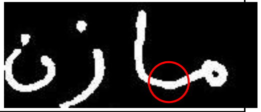
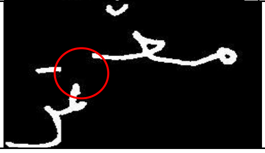
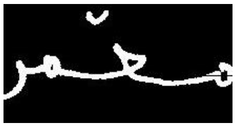


Figure 8. An example in which the system needs the Lam-Alef ligature.

In order to ease the evaluation of the synthesis criteria, we generate the *worst-matching* words (having maximum Euclidean distances). Table 3 shows the best and worst synthesis results side to side. See Section 6 for future work on objective OCR tests.

Table 3. Best and worst results for the Writer 1, Writer 2 and the combination of the two.

	Worst Synthesis	Best Synthesis
Writer 1		
		
Writer 2		
		
Combination		
		

The subjective evaluation reveals that the black and white spikes (circled) appear more and more annoyingly in the cases of the worst synthesis.

6. Conclusions and future work

In this work, we propose concatenating characters of Arabic script to form words and sentences. Such handwritten-like synthesis can be very useful in training and testing OCRs, as well as in forensics and other applications.

We train the system by providing it with labeled segmented characters. The process of segmenting characters is computer-aided. Features from the connection lines (*Kashidahs*) are extracted and stored.

In the synthesis process, *Kashidahs* of the samples of the required word are compared. Starting from the first character, a chain of matching is conducted to find fitting *Kashidahs* according to the minimum distance measure. The chosen images are then aligned into words and sentences.

We experiment with two single writers and for a combined mode. Subjective observation shows that the approach has promising results, although it is still in its infancy.

For future work, automatic segmentation is needed. Also ligatures need to be parsed in the training process. The IFN/ENIT dataset fortunately has ligatures encoded in it. Using the K-nearest neighbors (KNNs), instead of the current nearest neighbor, may increase the number of possible outputs. The parameter K needs to be studied to determine the values until which script can still be considered natural. Neural networks can be used as matchers. Randomized starting points (i.e. different from the best first two letters) can also improve the diversity in outputs. Finally, evaluation needs to be objective, rather than subjective. This goal can be achieved by using OCRs and forensic programs to evaluate the output.

Acknowledgment

We would like to acknowledge King Fahd University of Petroleum & Minerals (KFUPM) for supporting this research.

References

[1] Lavrenko V, Rath TM, Manmatha R. Holistic word recognition for handwritten historical documents. (Center for Intelligent Information Retrieval, University of Massachusetts Amherst); 2004.

[2] Vinciarrelli A. A survey on off-line cursive word recognition. Pattern recognition 2002; 35:1433-1446.I. Guyon. Handwriting synthesis from handwritten glyphs. In Proc. 5th Int. Workshop on Frontiers in Handwriting Recognition, pages 309–312, Essex, England, 1996.

[3] Amin A. Off-line Arabic character recognition: The state of the art. Pattern Recognition 98; 31(5): 517-30.

[4] Amin A. Recognition of printed Arabic text based on global features and decision tree learning techniques. Pattern recognition 2000; 33: 1309-23.

[5] Al-Badr B, Mahmoud SA. Survey and bibliography of Arabic optical text recognition. Signal Processing 1995; 41: 49-77.

[6] Al-Ohalia Y, Cheriet M, Suen C. Databases for recognition of handwritten Arabic cheques. Pattern Recognition 2003; 36: 111-121.

[7] Alshebeili SA, Nabawi AAF, Mahmoud SA. Arabic character recognition using 1-D slices of the character spectrum. Signal Processing 1997; 56: 59-75.

[8] Aissaoui A, Haouari A. Normalised Fourier coefficients for cursive Arabic script recognition. Applied Sig. Process. 1999; 6:115–22.

[9] Abandah G, Khedher M. Printed and handwritten Arabic optical character recognition –initial study. A report on research supported by the Higher Council of Science and Technology. Amman, Jordan, 2004, August.

[10] Khedher M, Abandah G. Arabic character recognition using approximate stroke sequence. Third Int’l Conf. on Language Resources and Evaluation (LREC 2002), Arabic Language Resources and Evaluation –status and prospects workshop; 2002, June.

[11] Elms AJ, Procter S, Illingworth J. The advantage of using an HMM-based approach for faxed word recognition. IJDAR 1998; 1: 18-36.

[12] Ballard, L., Lopresti, D., and Monrose, F. Evaluating the security of handwriting *biometrics. In The 10th International Workshop on the Foundations of Handwriting Recognition* (October 2006), pp. 461-466.

[13] Y. Yamazaki, A. Nakashima, K. Tasaka, and N. Komatsu. A study on vulnerability in on-line writer verification system. In Proceedings of the Eighth International Conference on Document Analysis and Recognition, pages 640–644, Seoul, South Korea, August-September 2005.

[14] Ballard, L., Lopresti, D., and Monrose, F. Forgery quality and its implications for behavioral biometric security. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Special Edition) 37*, 5 (October 2007), 1107-1118.

[15] I. Guyon. Handwriting synthesis from handwritten glyphs. In *Proc. 5th Int. Workshop on Frontiers in Handwriting Recognition*, pages 309–312, Essex, England, 1996.

[16] H. Baird. State of the art of document image degradation modeling. In *Proc. 4th IAPRWorkshop on Document Analysis Systems (DAS 2000)*, Invited plenary talk, Rio de Janeiro, Brasil, December 2000.

[17] J. Cano, J. Perez-Cortes, J. Arlandis, and R. Llobet. Training Set Expansion in Handwritten Character Recognition. In *Proc. 9th SSPR / 4th SPR*, pages 548–556, Windsor, Ontario, Canada, 2002.

[18] T. Ha and H. Bunke. Off-line handwritten numeral recognition by perturbation method. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(5):535–539, May 1997.

[19] M. Mori, A. Suzuki, A. Shio, and S. Ohtsuka. Generating new samples from handwritten numerals based on point correspondence. In *Proc. 7th Int. Workshop on Frontiers in Handwriting Recognition*, pages 281–90, Amsterdam, The Netherlands, 2000.

[20] Trenkle J, Gillies A, Erlandson E, Schlosser S, Cavin S. Advances in Arabic text recognition. Symposium on Document Image Understanding Technology; Columbia, MD; 2001.

[21] H. Miyao and M. Maruyama, “Virtual Example Synthesis Based on PCA for Off-Line Handwritten Character Recognition,” Document Analysis Systems VII, 2006, pp. 96-105.

[22] A.O. Thomas, A. Rusu, and V. Govindaraju, “Synthetic handwritten CAPTCHAs,” Pattern Recogn., vol. 42, 2009, pp. 3365-3373.

[23] Z. Lin and L. Wan, “Style-preserving English handwriting synthesis,” Pattern Recogn., vol. 40, 2007, pp. 2097-2109.

[24] V. Margner and M. Pechwitz, “Synthetic data for Arabic OCR system development,” Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on, 2001, pp. 1159-1163.

[25] T. Varga and H. Bunke, “Generation of synthetic training data for an HMM-based handwriting recognition system,” Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on, 2003, pp. 618-622 vol.1.

[26] O. Stettiner and D. Chazan, “A statistical parametric model for recognition and synthesis of handwriting,” Pattern Recognition, 1994. Vol. 2 - Conference B: Computer Vision & Image Processing., Proceedings of the 12th IAPR International. Conference on, 1994, pp. 34-38 vol.2.

[27] P. Rao, “Shape vectors: An efficient parametric representation for the synthesis and recognition of hand script characters,” Sadhana, vol. 18, Mar. 93, pp. 1-15.

[28] M. Helmers and H. Bunke, “Generation and Use of Synthetic Training Data in Cursive Handwriting Recognition,” Pattern Recognition and Image Analysis, 2003, pp. 336-345.

[29] J. Dolinsky and H. Takagi, “Synthesizing Handwritten Characters Using Naturalness Learning,” Computational Cybernetics, 2007. ICCC 2007. IEEE International Conference on, 2007, pp. 101-106.

[30] Fouad Slimane, Rolf Ingold, Slim Kanoun, Adel M. Alimi, and Jean Hennebert, "A New Arabic Printed Text Image Database and Evaluation Protocols," 10th International Conference on Document Analysis and Recognition (ICDAR2009), 2009, pp. 946-950.

[31] V. Maergner and M. Pechwitz. Synthetic data for Arabic OCR system development. In *6th Int. Conference on Document Analysis and Recognition*, 1159–63, 2001.

[32] IFN/ENIT database –Database of handwritten Arabic Words– available online at: http://www.ifnenit.com/

[33] F. Jenkins and J. Kanai, “The use of synthesized images to evaluate the performance of optical character recognition devices and algorithms,” Proceedings, Document Recognition, San Jose, CA: 1994, 194-203.

[34] R. Saabni and J. El-Sana, “Efficient Generation of Comprehensive Database for Online Arabic Script Recognition,” Document Analysis and Recognition, ICDAR 2009, pp. 1231-35.