Published version: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7333862

Multi-stage HMM based Arabic text recognition with rescoring

Irfan Ahmad [1,2]
[1] Information and Computer Science Department
KFUPM, Dhahran Saudi Arabia
irfanics@kfupm.edu.sa

Gernot A. Fink [2]
[2] Department of Computer Science
TU Dortmund University, Germany
gernot.fink@tu-dortmund.de

***Abstract* — In this paper, we present a multi-stage approach to handwritten Arabic text recognition using HMM where we separate the Arabic text image into core components and diacritics and recognize them separately using two separate HMM recognition systems. In the next stage, we combine the scores from both recognizers to make a final word hypothesis. This approach leads to huge reduction in the number of HMM models that need to be trained. Experiments conducted on a word recognition task using a publicly available benchmark database show the effectiveness of the technique. We achieve state-of-the-art results in addition to a compact model set for the recognition system.**

Keywords* — *Handwritten text recognition, hidden Markov models, multi-stage recognition, rescoring, Arabic text recognition, Model set reduction.

I. Introduction

Handwritten text recognition is an important area of research. Hidden Markov Model (HMM) is the classifier of choice in the area of text recognition [1]. One of the notable benefits of using HMM is that the text line image need not be segmented explicitly into recognition units like characters or strokes. Explicit segmentation of text is even more difficult in Arabic due to its cursive nature and thus the use of HMM in Arabic text recognition tasks is widely favored.

There are 28 basic characters in Arabic script. An Arabic character has a core-shape (also termed as *Rasm*) and may have diacritics (mainly dots) above or below the core-shape. Many characters share the same core-shape and differ only in diacritics. Additionally, most of the characters in Arabic script can take up to four different position-dependent visual shapes. Due to this reason, the most common approach is to model each character-shape, instead of character, as a separate HMM. This leads to a huge model set (typically more than 100 HMMs) in the recognizer. Having a large number of HMMs results in the need for large amounts of training data to adequately train them. Moreover, some character-shapes are infrequent in Arabic text and as such we end up having not enough samples to train them adequately.

In this paper we present an approach to HMM modeling for Arabic script where we separate the core-shapes from the diacritics. We model the core-shapes separately from the diacritics. This leads to a huge reduction in the number of basic HMM units thereby making the recognizer more efficient and robust, and can potentially perform relatively well under situations where limited training data is available. Our idea is a multi-stage approach where we recognize a test word image using features extracted from its core–shapes and, as a next stage, incorporate the information from the diacritics to make a final word hypothesis.

Menasri et al. [2] presented a hybrid word recognition system (using HMMs and neural networks) for Arabic script. Arabic letters were represented as, what they termed as, *letter-bodies*. Letters having same core-shape, and differing only in diacritics, were mapped into one class. They performed explicit grapheme segmentation and then extracted features from them. This was followed by iterative training of the hybrid system. Neural network was used to calculate the observation probability distribution for the HMM models. The system trained on letter bodies was used to recognize words using "letter-body to word" lookup dictionary. Recognition was made solely on the information from letter-bodies and the information from dots and other diacritics was not used. The authors stated that an approach needs to be devised to combine the information coming from core-shapes and the diacritics in order to benefit from the diacritics information.

Al Hajj et al. [3] presented an Arabic word recognition system using multiple HMMs. Two slanted windows, one to the left and the other to right, were used in addition to the vertical window for feature extraction. Slanted windows were proposed to address problems like writing inclination, and shifted position of diacritical marks. Features from each window orientation were used to train separate HMM recognizer. At the recognition stage, the results from the three recognizers were combined to make a final word hypothesis. Cheriet et al. [4] presented an HMM based system for recognition of handwritten Arabic legal amounts from bank checks. They proposed a two-step approach. As a first step, the words are recognized by combining the sub words (also known as Parts of Arabic Words or PAW). In the next step it tries to find the correct legal amount from a list of all the possible words using information from the previous step.

In our previous research [5], [6], we presented Arabic sub-characters for HMM based text recognition. A character is split into sub-characters based on the similar patterns between different characters and their position-dependent forms. The sub-character patterns are then used to reconstruct the characters during recognition. This leads to a huge reduction in the number of HMMs. Experiment conducted for a word recognition task showed the effectiveness of the approach. A comprehensive survey on handwritten Arabic text recognition can be found in [7]. A more recent survey was presented in [8].

The rest of the paper is organized as follows: In Section II, we present some of the peculiarities of the Arabic script and we also present some discussions on how these peculiarities can be exploited for HMM based text recognition. In Section III, we present the details of our multi-stage HMM recognition system. In Section IV, we present the experimental results and discussions. Finally in Section V, we present the conclusions of our work.

II. Peculiarities Of Arabic Script – Core Components And Diacritics

Arabic script is cursive both in handwritten as well as machine printed forms. It does not have different upper case and lower case letters. Some character combinations (like *Lam-Alif* 'لا') lead to ligatures which look visually different than a simple concatenation of constituent characters. 22 of the 28 characters can appear in four different visual forms depending on its position (***beginning, middle, ending,*** and ***isolated***) in a word. The remaining six characters can take only two visual forms (***ending*** and ***isolated***) as no characters can connect after them. Many of the Arabic characters have the same core-shape i.e. the primary component (known as *Rasm*) and they differ only on secondary components i.e. position and number of dots and other diacritics. Dots are mandatory diacritics as opposed to optional diacritics like *Hamza* and *Shadda.* Fig. 1 shows character groups which share the same core-shape but are different due to the presence or absence of diacritics.

Isolated character shapes	Core shapes	Beginning character shapes	Core shapes	Middle character shapes	Core shapes	Ending character shapes	Core shapes
ا أ إ آ	ا					ـا ـأ ـإ ـآ	ـا
ب ت ث	ب	بـ تـ ثـ يـ ئـ نـ	ـبـ	بـ تـ ثـ يـ ئـ نـ	ـبـ	ـب ـت ـث	ـب
ى ي ئ	ى					ـى ـي ـئ	ـى
ن	ن					ـن	ـن
ج ح خ	ح	جـ حـ خـ	حـ	جـ حـ خـ	حـ	ـج ـح ـخ	ـح
د ذ	د					ـد ـذ	ـد
ر ز	ر					ـر ـز	ـر
س ش	س	سـ شـ	سـ	سـ شـ	سـ	ـس ـش	ـس
ص ض	ص	صـ ضـ	صـ	صـ ضـ	صـ	ـص ـض	ـص
ط ظ	ط	طـ ظـ	طـ	طـ ظـ	طـ	ـط ـظ	ـط
ع غ	ع	عـ غـ	عـ	عـ غـ	عـ	ـع ـغ	ـع
ف	ف	فـ قـ	فـ	فـ قـ	فـ	ـف	ـف
ق	ق					ـق	ـق
ك	ل	كـ	كـ	كـ	كـ	ـك	ـل
ل	ل					ـل	ـل
م	م	مـ	مـ	مـ	مـ	ـم	ـم
ه ة	ه	هـ	هـ	هـ	هـ	ـه ـة	ـه
و ؤ	و					ـو ـؤ	ـو
لا لأ لإ	لا					ـلا ـلأ ـلإ	ـلا

Fig. 1. Different Arabic character shapes and their mapping to representative core-shapes.

When modeling the Arabic characters and their different position-dependent shapes using HMMs, two most common approaches are used: (i) Modeling each character as an HMM. This leads to different position-dependent shapes, which are often visually quite different, sharing the same model parameters. Consequently, this is not a very effective modeling approach, (ii) Modeling each position-dependent character shape with a separate HMM. Although the second approach is better and is commonly used, it has its own issues. Firstly, it leads to a four-fold increase in number of HMMs which, in turn, means need for more training data. Secondly, some of these character shapes are not very frequent in Arabic text. This leads to inadequate training for their respective models as they have very few samples in the training set. In order to address these two major modeling issues related to Arabic script, we propose a multi-stage approach where we separate the diacritics from the core components. The core-shapes and the diacritics are modeled and trained separately resulting in two separate systems. We later combine the results from the two separate systems to make a final word hypothesis. This approach allows us to model all the character shapes and their variations and at the same time keep the number of HMM models low (less than half the number of HMMs as compared to HMM systems using character-shapes as basic units). The details on our multi-stage Arabic text recognition system are presented in the next section.

III. HMM Based Multi-Stage Text Recognition

The main idea is to train the HMMs for the core-shapes and the diacritics separately. Because many characters and their position-dependent shapes share the primary components with other characters, removing the diacritics allows us to reduce the model set significantly. In fact, if we consider only the 28 basic Arabic characters and their position-dependent shapes, we end up with more than 100 different visual forms. This does not include the additional models needed for characters that may have optional diacritics like *Hamza* and *Shadda*. If we remove the diacritics and model the primary components using separate HMMs, the model set gets reduced to less than 60 HMMs in addition to six HMMs that are needed to model the diacritics. The six HMMs that need to be trained for the diacritics use the same training data (with the core-shapes removed from the image and with the labels modified to preserve only the diacritics information) and so does not elicit the need for extra training data.

The overall idea of the multi-stage text recognition system is illustrated in Fig. 2. For training, a word image is split into two separate images; one for the core-shapes and the other for the diacritics. The core-shape images along with their labels are used to train their corresponding HMM models. The original labels are modified to map characters-shapes to their representative core-shapes based on the rules as illustrated in Fig. 1**.** Similarly, the diacritics images and their associated labels (original labels are modified to include only diacritics information) are used to train the HMMs for them. During recognition, an input image is split into core-shape image and diacritics image as was done during training. Features are extracted from each of these images. The features extracted from the core-shape image are fed to the word recognition system trained on core-shape images. The core-shape HMM system serves as the primary recognition engine. It generates an N-best word-list utilizing a dictionary which has entry pairs for word and its corresponding core-shape definition. Features from the diacritics image are fed to the HMM word recognition system which was trained on diacritics. This system only rescores the original N-best word list generated by the core-shape system. It utilizes a dictionary which has entry pairs for word and its corresponding diacritic definition. Finally, as a last step, the scores from both systems are added together and the word having the highest score is hypothesized as the candidate word for the input image.

The algorithm for separating the core-shapes and the diacritics, along with two illustrative examples, is presented in Fig. 3. It is a simple rule-based algorithm. The main idea is to compute the average component size for an input image. Any component whose size is less than the average component size is included in the list of possible diacritics which undergoes a second stage of analysis and the remaining components are assigned to the core-shape list. During the second stage of analysis, if a component is long (its length is more than a threshold) and narrow (its length is at least twice its width), it is

removed from the diacritics list and added to the core-shape list as they most probably are character *Alifs* or are fragmented segments of characters which have a long vertical stroke (like *Lam* 'ل' *and Taa* 'ط'). Moreover, if a component is wide (i.e. its width is more than a threshold) and it is located around the baseline, it is also removed from the diacritics list and added to the core-shape list as they most likely are fragmented parts of the core writing or they are some of the isolated characters (like *Noon* 'ن' and *Daal* 'د'). Once the list of the core-shape components and the diacritics are finalized based on the algorithm, they are saved as separate images.

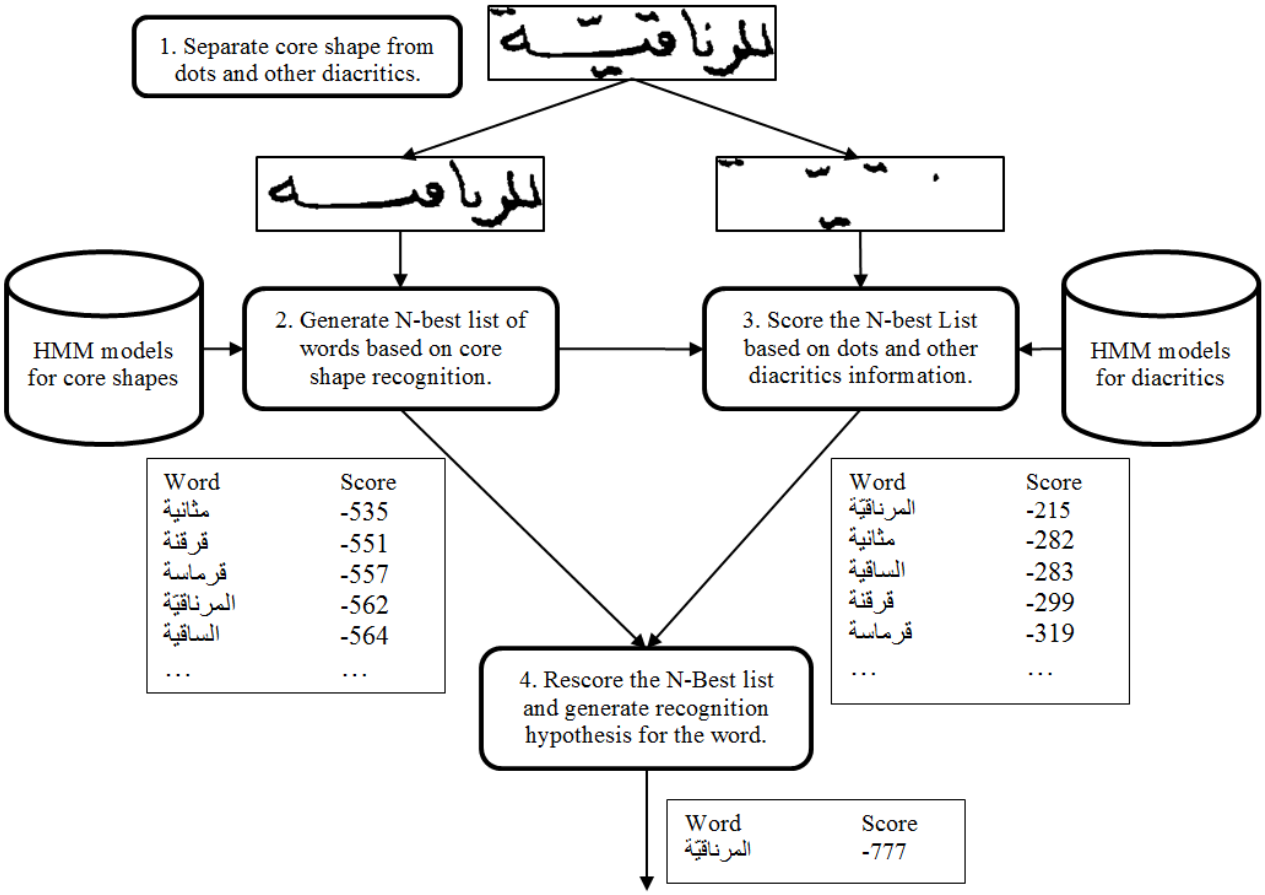

Fig. 2. HMM based multi-stage text recognition system illustrated with an example.

Algorithm	*Illustration Example 1*	*Illustration Example 2*

Input: Original image (Img)

Output: Image having core shapes (imgCore), and image having diacritics (ImgDia)

1. **Img**: The original image
2. ImgDia: includes all the components from the original image whose size is less than the average size of components in the original image
3. Remove from imgDia component 'c' if:
   - the length of 'c' is > thresh1 (30 pixels), AND
   - the length is at least twice the width

(**For Alifs and characters having vertical long stroke**)

4. Remove from imgDia component 'c' if:
   - the width of 'c' is > thresh2 (50 pixels), AND
   - it is within the core shape region i.e., it is not too far from the baseline

(**For fragmented core shapes or small isolated characters like Noon**)

5. **ImgCore**: Img - ImgDia

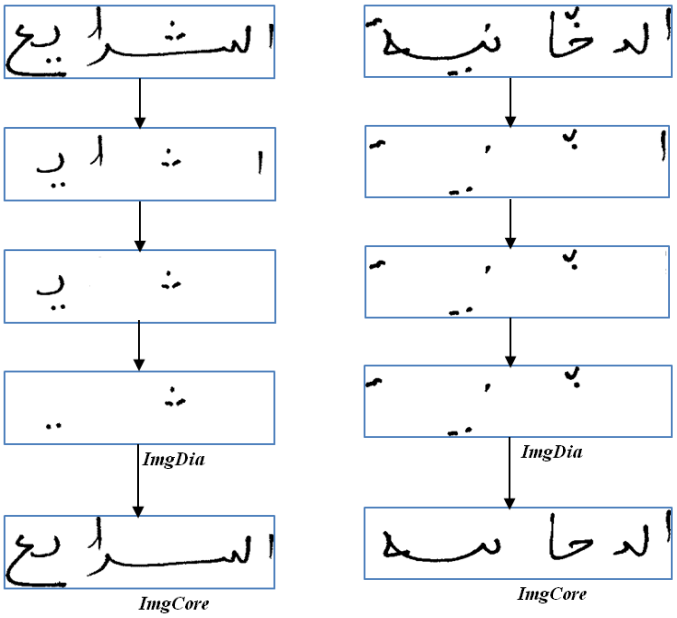

Fig. 3. A rule-based algorithm for separating core-shapes and diacritics.

IV. EXPERIMENTS AND RESULTS

In this section we present the experimental details and the results we obtained for an Arabic word recognition task using our multi-stage recognition system.

A. Recognition task

Our task is offline handwritten Arabic word recognition using the IFN/ENIT database [9]. The database consists of handwritten word images of Tunisian cities and towns divided into seven sets *a – f*, and *s*. The lexicon size is 937 names but some names have two or more variations (mainly due to *ligature* models and optional *shadda* diacritic). We experimented with the most common *train – test* configurations reported in the literature including the competitions using the IFN/ENIT database.

B. Experimental details and discussions

Our recognizer is a continuous HMM system built using the HTK tools [10]. Basically, we built two separate HMM recognizers; the primary one for the core-shapes and the other one for the diacritics. The core-shape system has 71 HMMs to model all the characters and their shape variations. The diacritics system has 17 HMMs to model the dots and the other optional diacritics like *Shadda* and *Hamza*. In addition to the multi-stage system, we also developed and evaluated a traditional character-shape based system in order to provide a baseline for our recognition results. A comparison based on the number of HMMs in the system of the commonly used character-shape system and our present system is presented in Table I. As we can see from the table, the number of HMMs in the new system gets reduced by more than half. This by itself is a major achievement as long as the recognition rates of the new system are comparable if not better than the character-shape based system. Fewer HMMs lead to compact and robust system and can work reasonably well in situations where few training samples are available. In fact, as can be seen from the table, the number of HMMs in the current system is even smaller than the sub-character HMM system proposed recently by the authors [5], [6].

Bakis topology was used for all the HMMs. We extracted nine geometrical features (the average number of ink pixels, the number of black-white transitions, the distance of the upper contour, the lower contour, and the center-of-gravity of the ink-pixels from the writing line, the orientation of the upper contour, the lower contour, and the center-of-gravity of the ink pixels) from each frame sliding over the word images. These features are adapted from [11] and [12]. We appended nine derivative features to the original features so the dimension of the feature vector is 18. It is worth noting that no preprocessing was done on the images prior to feature extraction. In the baseline system, we replaced few models (having diacritic *Shadda* over them) whose frequency in the training set was very low. These were replaced with models representing the same character-shapes without the *shadda* over them. The 178 HMMs models got reduced to 157 models in our baseline system. System parameters like number of states per HMM, number of mixtures in each state were optimally configured based on experiment configuration *abc – d* used for validation i.e., the system was trained using sets *a*, *b*, and *c* and evaluated on set – *d*. As a first step, a uniform initialization was done using the training data. In the next step, information from forced alignment of the training data was used to initialize individual HMMs using Viterbi initialization. This was followed by a number of iterations of Baum-Welch retraining. Finally the word hypothesis was made using Viterbi decoding. The evaluation results are shown in Table II on the following training – test configurations: *abc – d*, *abcd – e*, *abcde – f*, and *abcde – s*. The significance intervals at 95% confidence level are also provided in Table II. The results are shown in terms of *Word Recognition Rate* (*WRR*). It can be seen from the table (first row) that the recognition results of the baseline system are comparable to the state-of-the-art systems evaluated on the same database (c.f. Table III). To understand the effectiveness of the core-shape system, we carried out an experiment where we only used the core-shape HMM system to recognize word images without using any information from the diacritics. The results are tabulated in Table II (second row). It can be seen from the table that the performance of the core-system is quite good by itself.

TABLE I. COMPARISON OF NUMBER OF HMM MODELS IN TRADITIONAL SYSTEM VS. THE PROPOSED SYSTEM.

System	Number of HMMs
Character-shape HMM system	178
Sub-character system as proposed in [6] [5].	97
Multi-stage system	71 (core-shape system), 17 (diacritics system)

For the multi-stage system, most of the system properties were similar to the baseline system with the obvious differences as presented in Section IV. In addition, we modeled space explicitly using special space models as presented in [6]. The thresholds for the diacritics separation algorithm were decided by manually inspecting the algorithms performance on set – *a*. Other system parameters were optimally calibrated based on the experiment configuration *abc – d*. The word recognition results are presented in Table II. We can see that the multi-stage system's performance is better in all the evaluation sets except set – *s* where the baseline system did better than the multi-stage system but the difference in not statistically significant at 95% confidence level. As an extension, we used multi-stream HMMs as presented in [6] where we split the features into two streams such that the computed nine features constitute one stream and the derivative features are in the second stream. As a further extension, we incorporated contextual-HMM modeling as presented in [6] which is similar to the concept of tri-phone HMMs in speech recognition. Results on all the evaluations sets are presented in Table II. We can see from the table that the results are quite good on the Arabic word recognition task. Improvements can be observed for all the enhancements (like multi-stream HMMs and contextual-HMM modeling). Considering the fact that the model set in the new system is less than half the model set of the traditional character-shape systems, the performance of the multi-stage system is really impressive.

In Table III, we present a comparison of the recognition rates of our system with other state-of-the-art HMM systems evaluated on the IFN/ENIT database. From the table we can see that the recognition rates of our system are comparable to the other top systems. We got slightly lower results on *set – s* but the difference is not statistically significant at 95% confidence level.

TABLE II. SUMMARY OF THE RESULTS (WRR) FOR HANDWRIITEN TEXT RECOGNIITON ON THE IFN/ENIT DATABASE.

The Recognition System	***Train–Test Configuration*** *(Statistical significance at 95% confidence level)*			
	abc–d *(±0.38)*	***abcd–e*** *(±0.56)*	***abcde–f*** *(±0.50)*	***abcde–s*** *(±1.56)*
Character-shape HMM system (*Baseline*)	95.38	90.48	89.40	80.69
Core-shape recognition system	93.68	88.50	86.96	75.60
Multi-Stage Recognition System	95.82	91.69	89.78	79.95
Multi-Stage Recognition System + multi-stream HMMs	97.85	94.71	92.07	83.46
Multi-Stage Recognition System + multi-stream HMMs + contextual-HMM modelling	98.08	94.93	92.30	84.55

TABLE III. COMPARISON WITH OTHER STATE-OF-THE-ART SYSTEMS EVALUTED ON IFN/ENIT DATABASE.

Systems	***Train–Test Configuration***			
	abc–d	***abcd–e***	***abcde–f***	***abcde–s***
UPV-PRHLT [13]	95.20	93.90	92.20	84.62
RWTH-OCR [14], [15]	96.53	92.74	92.20	84.55
Azeem and Ahmed [16]	97.70	93.44	**93.10**	84.80
Su et al. [17]	96.81	93.55	-	-
Ahmad et al. [6]	97.22	93.52	92.15	**85.12**
Present work	**98.08**	**94.93**	92.30	84.55

V. CONCLUSION

Text recognition is an interesting, as well as challenging, research area in the field of pattern recognition. HMMs are widely used classifier for text recognition. In this paper we presented a multi-stage HMM based text recognition system for Arabic script where the HMM units are core-shapes and diacritics instead of the commonly used character-shapes. This leads to reduced model set and results in a compact and efficient recognizer. Experiments conducted on IFN/ENIT database for word recognition task demonstrated that the system can perform very well and the results are comparable to the state-of-the-art results published in the literature using the same database. This is in addition to the fact that less than half the number of HMMs was used when compared to the commonly used approach of using character-shape as HMMs. It will be interesting to investigate the impact of using language models estimated using the core shapes. As the number of classes gets reduced, it maybe that, more robust language models can be estimated which might improve the recognition performance.

ACKNOWLEDGMENT

The authors would like to acknowledge the support provided by King Fahd University of Petroleum and Minerals (KFUPM) for funding this work through project number RG 1313-1/2.

REFERENCES

[1] T. Plötz and G. A. Fink, "Markov models for offline handwriting recognition: a survey," *Int. J. Doc. Anal. Recognit.*, vol. 12, no. 4, pp. 269–298, Oct. 2009.

[2] F. Menasri, N. Vincent, E. Augustin, and M. Cheriet, "Shape-based alphabet for off-line arabic handwriting recognition," in *Ninth International Conference on Document Analysis and Recognition, ICDAR*, 2007, vol. 2, pp. 969–973.

[3] R. Al-Hajj Mohamad, L. Likforman-Sulem, and C. Mokbel, "Combining slanted-frame classifiers for improved HMM-based Arabic handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 7, pp. 1165–1177, 2009.

[4] M. Cheriet, Y. Al-Ohali, N. Ayat, and C. Y. Suen, "Arabic Cheque Processing System: Issues and Future Trends," in *Digital Document Processing*, B. B. Chaudhuri, Ed. London: Springer London, 2007, pp. 213–234–234.

[5] I. Ahmad, L. Rothacker, G. A. Fink, and S. A. Mahmoud, "Novel Sub-character HMM Models for Arabic Text Recognition," in *12th International Conference on Document Analysis and Recognition*, 2013, pp. 658–662.

[6] I. Ahmad, G. A. Fink, and S. A. Mahmoud, "Improvements in Sub-Character HMM Model Based Arabic Text Recognition," in *14th International Conference on Frontiers in Handwriting Recognition*, 2014, pp. 537–542.

[7] L. M. Lorigo and V. Govindaraju, "Offline Arabic handwriting recognition: a survey.," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 712–24, May 2006.

[8] M. T. Parvez and S. A. Mahmoud, "Offline arabic handwritten text recognition: A Survey," *ACM Comput. Surv.*, vol. 45, no. 2, pp. 23–35, Mar. 2013.

[9] M. Pechwitz, S. S. Maddouri, V. Märgner, N. Ellouze, and H. Amiri, "IFN/ENIT - Database of Handwritten Arabic Words," in *7th Colloque International Francophone sur l'Ecrit et le Document , CIFED 2002*, 2002, pp. 129–136.

[10] S. J. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.2. 1)*. Cambridge University Engineering Department, 2002.

[11] M. Wienecke, G. A. Fink, and G. Sagerer, "Toward automatic video-based whiteboard reading," *Int. J. Doc. Anal. Recognit.*, vol. 7, no. 2, pp. 188–200, 2005.

[12] U.-V. Marti and H. Bunke, "Handwritten sentence recognition," in *Proc. of 15th International Conference on Pattern Recognition. ICPR-2000*, 2000, pp. 463–466.

[13] V. Margner and H. E. Abed, "ICFHR 2010 - Arabic Handwriting Recognition Competition," in *Frontiers in Handwriting Recognition (ICFHR), 2010 International Conference on*, 2010, pp. 709–714.

[14] P. Dreuw, D. Rybach, G. Heigold, and H. Ney, "RWTH OCR: A Large Vocabulary Optical Character Recognition System for Arabic Scripts," in *Guide to OCR for Arabic Scripts SE - 9*, V. Märgner and H. El Abed, Eds. Springer London, 2012, pp. 215–254.

[15] V. Margner and H. E. Abed, "ICDAR 2011 - Arabic Handwriting Recognition Competition," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pp. 1444–1448.

[16] S. Azeem and H. Ahmed, "Effective technique for the recognition of offline Arabic handwritten words using hidden Markov models," *Int. J. Doc. Anal. Recognit.*, vol. 16, no. 4, pp. 399–412, 2013.

[17] B. Su, X. Ding, L. Peng, and C. Liu, "A Novel Baseline-independent Feature Set for Arabic Handwriting Recognition," in *Document*

Analysis and Recognition (ICDAR), 2013 12th International Conference on, 2013, pp. 1250–1254.