Published version: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6981075

Improvements in Sub-Character HMM Model Based Arabic Text Recognition

Irfan Ahmad [1,2], Gernot A. Fink [2], and Sabri A. Mahmoud [1]

[1] Information and Computer Science Department
KFUPM, Dhahran Saudi Arabia
{irfanics, smasaad}@kfupm.edu.sa

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

***Abstract*—Sub-character HMM models for Arabic text recognition allow sharing of common patterns between different position-dependent shape forms of an Arabic character as well as between different characters. The number of HMMs gets reduced considerably while still capturing the variations in shape patterns. This results in a compact, efficient, and robust recognizer with reduced model set. In the current paper we are presenting our recent improvements in sub-character HMM modeling for Arabic text recognition where we use special '*connector*' and '*space*' models. Additionally we investigated contextual sub-characters HMMs for text recognition. We also present multi-stream contextual sub-character HMMs where the features calculated from a sliding window frame form one stream and its derivative features are part of the second stream. We report state-of-the-art results on the IFN/ENIT (benchmark) database of handwritten Arabic text and the recognition rate of 85.12% on set–*s* outperforms previously published results.**

***Keywords*—Sub-character HMM, contextual-HMM, multi-stream HMM, space modeling, Arabic text recognition.**

I. INTRODUCTION AND RELATED WORK

Handwritten text recognition is an active area of pattern recognition research. Researchers have tried various approaches for text recognition employing various techniques for preprocessing, feature extraction, and classifiers. Hidden Markov models (HMMs) have proven to be one of the most successful and widely used classifiers in the area of text recognition [1]. There are many reasons for success of HMMs in text recognition including avoidance of the need to explicitly segment the text into recognition units (like characters or strokes). In addition, HMMs have sound mathematical and theoretical foundations.

Most researchers adapted their existing HMM text recognizers with minimal changes (like selection of basic HMM units and setting up a right-to-left HMM) to work for the Arabic script (e.g. [2]). Although this has its advantages like script flexibility, nevertheless, it leaves an important area less explored i.e. investigating the script peculiarities and using them to improve the recognizer in terms of recognition accuracy, efficiency etc. Among the possible areas of improvement is the selection of the basic HMM units for Arabic script. Interested readers can refer to [3], [4] for background on Arabic handwritten text recognition.

An Arabic character can have different shapes and appearances based on its position in the word or Parts of Arabic Words (PAW). Some characters can take up to four different shapes while other characters might have two different position-dependent shapes. Moreover, some of the characters in Arabic script share part of the appearance with other characters. This property of the script was investigated in our previous research where we proposed sub-character HMMs for Arabic text recognition [5]. The technique exploits the similar patterns between different characters and their position-dependent shapes for Arabic script. This leads to significant reduction in number of basic HMM units as compared to commonly used character or character-shape HMM models for Arabic. Reduced HMM set allows for more sharing (leading to robust training), and a compact and efficient recognizer. The result reported for printed Arabic text recognition task was better than the baseline system. But for handwritten text recognition task, the reported results were worse than the baseline system.

Use of sub-character HMM has been reported for online text recognition particularly for East-Asian scripts like Kanji and Hiragana. Nakai et al. presented sub-stroke HMM based recognition system for Kanji characters [6]. Several motivations were stated for using sub-stroke HMMs as opposed to whole character HMMs including; compact system with less number of models and dictionary size, faster recognition due to efficient sub-stroke network search, and less training data requirement. A set of 25 sub-strokes were identified and modeled. It was stated that the presented 25 sub-strokes can represent different Kanji characters using a dictionary defining the character structures. A hierarchical dictionary was defined where the elementary units are the sub-strokes defining the strokes which in turn define the sub-Kanji characters finally defining the Kanji characters. Automatic generation of this dictionary was presented in [7]. Tokuno et al. presented sub-stroke HMM based online recognition of cursive Kanji and Hiragana characters [8]. They mentioned that any Kanji character can be modeled by concatenating 25 proposed sub-stroke HMMs. Further they experimented with context-dependent sub-stroke models to capture the variation of sub-strokes due to its context (its adjacent sub-strokes). They employed Successive State Splitting (SSS) algorithm to reduce the number of models by sharing similar states of the context-dependent sub-strokes. This lead to increase in recognition accuracy as compared to character HMM system and context-independent sub-stroke HMM system. Hu et al. presented sub-character HMM models for online handwriting recognition of isolated digits, characters, and isolated words [9]. A character (or digit) was constructed by concatenating sub-character models based on a lexicon. The main motivation stated for using sub-character models instead of character models was the reduction of the model set and requirement of fewer training samples. However they mentioned that although sub-character HMM model based recognizer will be efficient, the recognition results will not necessarily be higher and, in some cases (where large amount of training data is available), may end up even lower.

In this paper we present improvements to the Arabic sub-character HMM modeling which was originally presented in [5]. We present special '*connector*' and '*space*' models. Additionally we investigate contextual HMMs for sub-character modelling. We also experimented with multi-stream HMMs by splitting the feature stream into two such that features calculated from a sliding window frame form one stream and its derivative features become part of the second stream. These improvements lead to significant improvement in the recognition results over the baseline system for the handwritten text recognition task. We are reporting state-of-the-art results on the IFN/ENIT database using our improved sub-character HMM system.

The rest of the paper is organized as follows: In Section II we briefly present the idea behind Arabic sub-character HMMs followed by details on our improvement in sub-character HMM recognition system for Arabic. In Section III we present experimental results and discussions on sub-character HMM based Arabic text recognition. Finally in Section IV, we present our conclusions.

II. SUB-CHARACTER HMM SYSTEM FOR ARABIC TEXT RECOGNITION

A. *Sub-Character Modeling*

The main idea behind sub-character modeling for Arabic text recognition was presented in [5]. A character is split into sub-characters exploiting the similar patterns between different characters and their different position-dependent shapes. The sub-character patterns can then be used to reconstruct the characters. This leads to huge reduction in number of basic HMMs which is important for many reasons like robust training with limited data (especially when using continuous HMMs), and a more compact recognizer. Fig. 1 illustrates the idea with some example characters. The sub-character HMM models, as proposed here, do not need any explicit segmentation of characters into sub-characters as is the case with other segmentation based approaches like the ones presented in [3][10]. Instead, the HMM models learn the patterns automatically from the data as long as they are defined adequately in the dictionary using the domain knowledge of the script. Once we list all the unique patterns (at sub-character level) from the Arabic script, creating the complete HMM structure is relatively straightforward. Character models can be constructed from sub-character models by concatenation. The structure can be extended hierarchically to perform lexicon based recognition. For more details on the background ideas related to Arabic sub-character HMM models, the reader can refer to [5]. In the following sub-sections, we present our recent improvements in sub-character modelling and its use in Arabic text recognition.

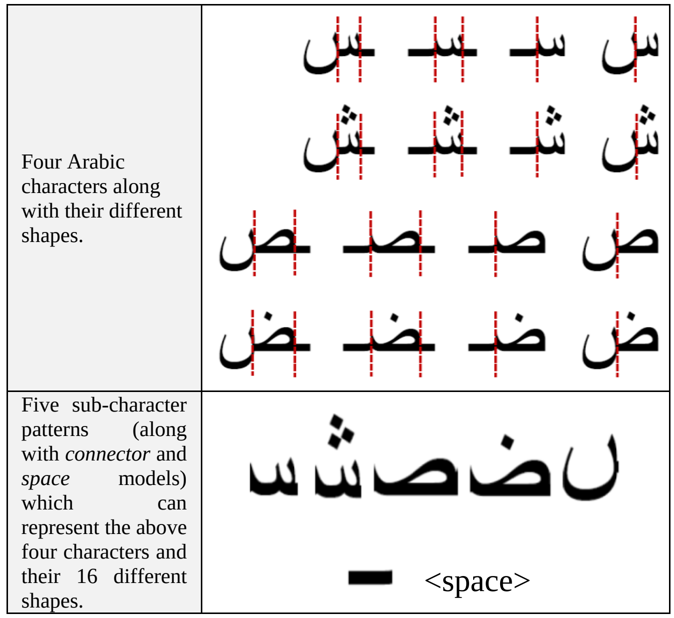


Fig. 1. Examples to illustrate sub-character modeling for Arabic script which can reduce the number of basic HMM models [5].

B. *Special 'Connector' and 'Space' Models*

We have two special models, the '*connector*' model and the '*space*' model, in our sub-character model set. These two were modelled and trained differently than the other models. The *space* model captures space (if present) between two sub-characters. In principle every sub-character that represents the last part of character, which has either the isolated or the ending position, should be followed by some space which is modeled by the *space* model. The *connector* models the stroke between two connected sub-characters. As these models are special and represent only a very specific pattern, they are modelled as single-state models. Furthermore, although a sub-character should be followed by a connector joining the next sub-character (or followed by space), virtually it is not always the case in handwritten texts. Sometimes strokes are written so closely that there is no room to capture the connectors or spaces using the sliding window.

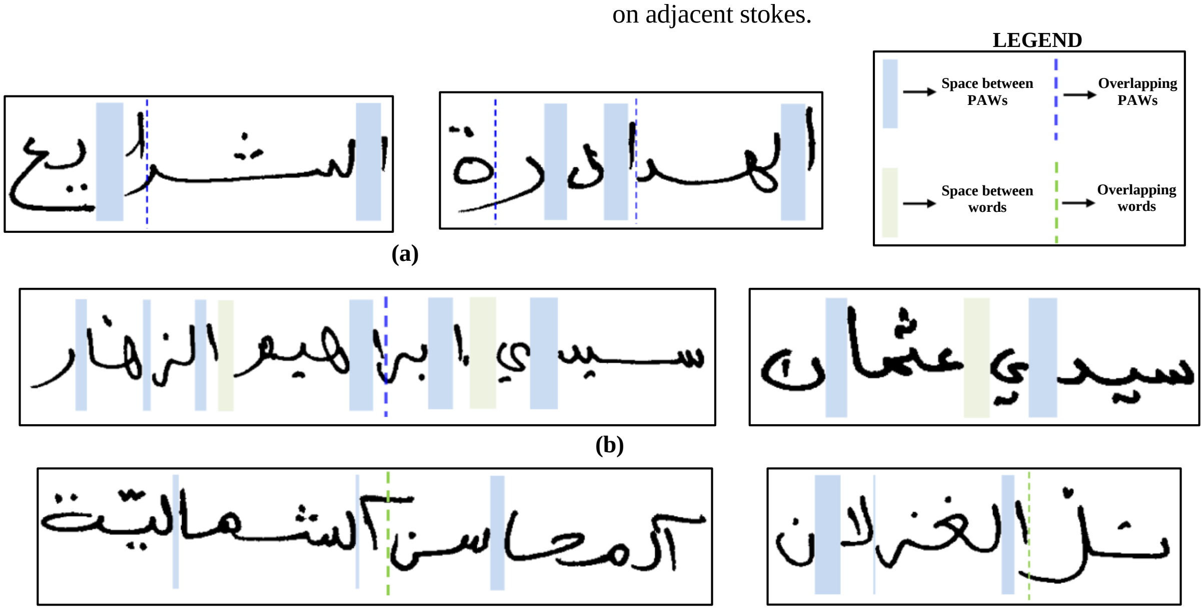
Dreuw et al. [11] presented white space modeling for Arabic text recognition. They proposed using special white-space models to explicitly model the spaces between words and within words (between PAWs). But the fundamental problem still remains when PAWs are overlapping in horizontal direction (as illustrated in Fig. 2 (a)) and in such situations, the space cannot be modelled properly. One possible approach to the problem (as proposed in [11]) is to use different writing variants in the lexicon where the recognition system can select the most suitable writing variant. Still a hard decision needs to be made on selecting variants which includes space between words or variants which proposes spaces between PAWs in addition to the words. In a real handwriting scenario, the presence of space between words and within words is not that regular. We can easily find samples where space is present between some PAWs but absent between other PAWs in the same text line as illustrated in Fig. 2 (a). Moreover, sometimes the width of space between PAWs may be larger (or similar) as compared to the width of space between words as illustrated in Fig. 2 (b). To add to the problem, in some cases it is difficult even to find space between words even though there might exist space between PAWs as illustrated in Fig. 2 (c). Thus we need better ways to deal with this problem especially when dealing with sub-character modelling as they rely on space models much more than the traditional character/character-shape modeling approaches.

We propose a special structure for space and connector models such that these models can be used in some instances while in other instances, where spaces or connectors are missing in the text, the models can be skipped. The idea is adapted from the concept of '*tee*' models used in speech recognition to model *silence* and *short pauses* [12] [13]. We allow transitions from the entry state to the exit state in the *space* and *connector* models thereby allowing a possibility to skip the emitting state. But an issue faced in this setup was that the *space* model skipped most of the time during the training and, as such, was not modeling the spaces robustly. It was somehow hard to force what to model when the possibility of skipping was possible. To avoid this problem we created a special *second-space* model to model blank spaces at the beginning and end of the text line images. This is a single-state model with very rigid transition possibilities such that the model consumes only the first and the last frame (the first and last sliding window frame contains only the white background and no text). The state of this *second-space* model was tied with the emitting state of the skipping *space* model. Using this setup, the model seems to train well. The idea of space modeling is illustrated in Fig. 3. Examples from forced-alignment on training samples confirmed that the *space* model captures the space-frames and skips at other times to model space more robustly.

C. *Contextual sub-character HMMs*

Context-dependent HMMs have been used successfully in speech recognition and online text recognition with significant improvements in recognition results [14], [15]. However, the use of context-dependent HMMs for offline text recognition has not been extensively reported. Probably the first such attempt was made by Fink and Plötz for Roman script [16]. Prasad et al. tried them for the task of printed Arabic OCR [17]. Limited improvements were reported in both the works. Inability to satisfactorily describe the nature of contextual influence appearing in handwritings is one possible explanation to the low improvements [16]. For printed Arabic text recognition, the use of separate HMMs for each "presentation" form of a character may be enough to capture the contextual variations [17]. The use of context-dependent HMMs was also reported recently on handwritten Arabic text recognition task but it is not clear if it lead to improvements (if any) as compared to the context-independent HMM system [18].

The sub-character models are smaller than the character or character-shape models and its shape has high variability depending on many factors like writing style and its adjacent sub-characters. Thus it seems important to model the contextual forms of sub-characters. Tokuno et al. [8] presented their work on contextual sub-character modeling for online handwriting recognition of Kanji and Hiragana text. They reported that the contextual sub-character models lead to significant improvement in recognition results for both the Kanji and the Hiragana handwriting recognition tasks even though the non-contextual sub-stroke approach led to deterioration of recognition accuracy when compared to the whole-character HMM models based recognition. Our motivation for using contextual sub-character models also increases when looking at the nature of the Arabic script where the same strokes may be written differently depending on adjacent stokes.



(a)

(b)

(c)

Fig. 2. Example text images from the IFN/ENIT database [19] illustrating different issues related to *space* modeling.

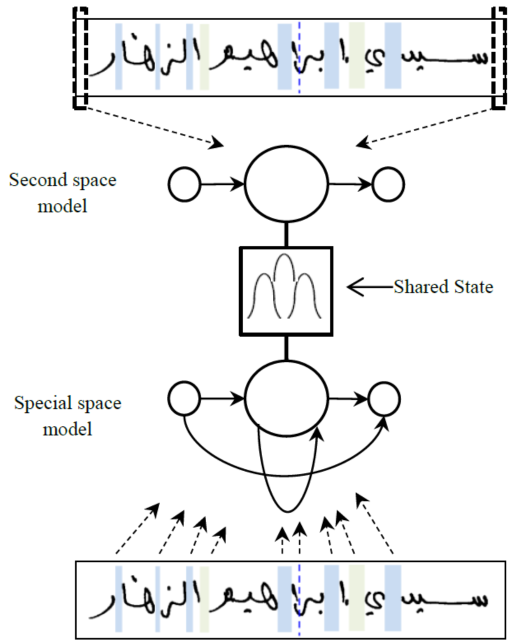


Fig. 3. Space modelling illustrated with an example.

We investigated the data driven approach to contextual sub-character HMMs for Arabic text recognition. First all the different contextual forms are generated from the non-contextual forms using the training transcription. Some contextual forms occur very few times in the training data and as such robustly estimating their parameters is difficult. Due to this reason, adequate level of tying is performed such that the tied parameters share the same pool of data during training. As a first step, we tie the transition probabilities of all the contextual forms of a sub-character. Next, we perform the training for the contextual forms using few iterations of the Baum-Welch training algorithm. As a next step, we perform state tying between different contextual forms of the sub-character preserving the state sequence i.e. corresponding states in different contextual forms of a sub-character are tied if the distance between two states is within a threshold. The distance $d(x,y)$ between two states, x and y, is computed using the following equation:

$$d(x,y) = -\frac{1}{M}\sum_{m=1}^{M} log\left[b_y(\mu_{xm})\right] + log\left[b_x(\mu_{ym})\right]$$

where M is the number of mixture components, μ_{xm} is the mean vector for the m^{th} mixture component of state x, and $b_y(o)$ is the probability of generating observation o by state y which is given by:

$$b_y(o) = \sum_{m=1}^{M} c_{ym}\, N\left(o|\mu_{ym}, \Sigma_{ym}\right),$$

where c_{ym} is the weight of the m^{th} mixture component of state y, and $N(o|\mu,\Sigma)$ is a multivariate Gaussian with mean vector μ and covariance matrix $\sum$, which is:

$$N(o|\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^n|\Sigma|}}\, e^{-\frac{1}{2}(o-\mu)'\Sigma^{-1}(o-\mu)}$$

where n is the dimensionality of the vector o.

State tying is an important step, especially when using continuous HMMs, so that the parameters get trained robustly with availability of few training samples. After performing the state tying we estimate the HMM parameters using a few iterations of Baum-Welch training. The steps followed for contextual HMMs are similar to what was described by Young and Woodland for continuous speech recognition [13], [14].

D. *Multi stream HMMs with stream splitting*

Multi-stream HMMs have been successfully used for speech recognition. The most common setup is to separate features from two different sources (audio and visual) into two separate streams in audio-visual automatic speech recognition [20], [21]. Kessentini et al. presented multi-stream HMM based offline word recognition in [22]. Four different features, two density-features using different sliding window configurations and the upper and lower contours of text image, were treated as four separate streams and the results were compared to feature-combination and classifier-combination approaches. The authors reported that the multi-stream approach performed better than feature-combination and classifier-combination. Although researchers have used multi-stream approaches on completely different feature sets, we investigated multi-stream HMMs in a seemingly interesting setup. From our set of 18 features per frame, nine are calculated directly from the image whereas the other nine are the derivative features. We split our features into two streams such that the computed nine features constitute one stream whereas the derivative features are part of the second stream. Based on several experiments we conducted, we found that using multi-stream HMMs in the present setup leads to improved accuracy as can be seen from the experimental results presented in Table I.

III. Experiments And Results

We evaluated our sub-character HMM based recognizer on the benchmark IFN/ENIT database [19]. The database consists of 32,492 handwritten word images of Tunisian cities and towns divided into sets *a* – *e*. In addition, there are two extra sets '*f*' and'*s*' which were added later. In order to compare the results of sub-character HMMs with character-shape HMMs, we first built our baseline system using the character-shape HMMs as basic modeling unit. Our text recognition system is a continuous HMM system using the HTK tools [13]. We extracted nine statistical features from the word images. These features are adapted from [23] and [24]. We appended nine derivative features to the original features such that the dimension of the feature vector is 18. It is worth noting that we extracted the features directly from the raw images without any preprocessing (like slant correction, text normalization) as our focus was on modeling.

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

The Recognition System	***Train–Test Configuration*** *(Statistical Significance)*			
	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
Sub-character HMM system	95.90	91.55	89.74	82.14
Contextual sub-character HMM system	96.67	92.91	91.57	84.49
Multi-stream contextual sub-character HMM system	97.22	93.52	92.15	85.12

There are a total of 178 unique character-shapes in the IFN/ENIT dataset. Each of these is modeled as a separate HMM. We replaced few models (mainly representing shapes with diacritic '*shadda'* over them) whose frequency in training data (set *a*, *b*, and *c*) was very low. These were replaced with models representing the same character-shapes without the *shadda* over them. Thus the original 178 HMM models got reduced to 157 models in our baseline system. Each character-shape HMM was modelled with the same number of states. The optimal number of states was decided based on the uniform initialization (flat start) results on experiment configuration ac–d (i.e. the sets *a* to *c* were used for training and the set–*d* was used for evaluation). The optimal number of Gaussian mixture per state was also decided based on the same experimental configuration. As a first step, a uniform initialization (flat start) was done using the training data. In the next step, the alignment information from the training data was used to train individual HMMs using the Viterbi training followed by a number of iterations of Baum-Welch retraining. Finally the word hypothesis was made using the Viterbi decoding. The recognition results of the baseline system is comparable to the current state-of-the-art (cf. Table V, [25]).

For the sub-character HMM system, the sub-character representation uses 97 HMMs to model all the characters and their shape variations. This by itself is a great improvement in terms of system compactness and efficiency. Parameters like the optimal number of states per HMM (the *shape* and *connector* models were treated differently as discussed in Section II.B) were decided based on experiment configuration *abc–d*. Apart from that, most of the experimental setup was the same as the baseline system. We extended our experiments by conducting contextual sub-character experimentation as discussed in Section II.C. Finally, multi-stream contextual sub-character HMM based text recognition was performed, the details of which are presented in Section II.D. Table I summarizes the results of the experiments on following training–test configurations: *abc–d*, *abcd–e*, *abcde–f*, and *abcde–s*. The results are shown in terms of *Word Recognition Rate* (*WRR*). We also report the statistical significance at 95% confidence level. From the table it can be seen that there are significant improvements in the recognition results as compared to the baseline systems for all the three enhancements proposed in this work.

In Table II, 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. To the best of our knowledge, our recognition rate of 85.12% on set–*s* outperforms all the previously published results on the same set. From the table we can see that the recognition rates of our system on other evaluation sets are comparable to the other top systems.

IV. CONCLUSIONS

Sub-character HMMs for Arabic text recognition allow the sharing of common patterns between different shape forms of an Arabic character as well as between different characters. The number of HMMs gets reduced considerably while still capturing the variations in shape patterns. This results in a compact, efficient, and robust recognizer with reduced model set. In this paper we present our improvements to the sub-character modeling which includes, special '*connector*' and '*space*' models. The *space* model provides flexibility in modeling spaces between words and PAWs and it addresses situations where the writing is overlapping and/or compact. We investigated the data-driven contextual sub-character HMM system for Arabic text recognition. Parameter tying was done at different levels (state tying and transition probability tying) to estimate the parameters robustly in the presence of limited data. Additionally we investigated multi-stream contextual sub-character HMMs where features calculated from a sliding window frame forms one stream and its derivative features forms the second stream. Experiments were conducted with different train-test configurations on the IFN/ENIT database of handwritten Arabic text. We achieved state-of-the-art results on the database and our results on set–*s* outperformed all the previously reported results on the benchmark database.

TABLE II. 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 [26]	95.20	**93.90**	92.20	84.62

RWTH-OCR [25], [27]	96.53	92.74	92.20	84.55
Azeem and Ahmed [28]	**97.70**	93.44	**93.10**	84.80
Su et al. [29]	96.81	93.55	-	-
Present Work	97.22	93.52	92.15	**85.12**

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, 2009.

[2] M. P. Schambach, J. Rottland, and T. Alary, "How to convert a Latin handwriting recognition system to Arabic," *Proc. ICFHR*, pp. 265–270, 2008.

[3] 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.

[4] 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.

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

[6] M. Nakai, N. Akira, H. Shimodaira, and S. Sagayama, "Substroke approach to HMM-based on-line Kanji handwriting recognition," in *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*, 2001, pp. 491–495.

[7] M. Nakai, H. Shimodaira, and S. Sagayama, "Generation of hierarchical dictionary for stroke-order free Kanji handwriting recognition based on substroke HMM," in *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, 2003, pp. 514 – 518 vol.1.

[8] J. Tokuno, N. Inami, S. Matsuda, M. Nakai, H. Shimodaira, and S. Sagayama, "Context-dependent substroke model for HMM-based on-line handwriting recognition," in *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*, 2002, pp. 78–83.

[9] J. Hu, S. Gek Lim, and M. K. Brown, "Writer independent on-line handwriting recognition using an HMM approach," *Pattern Recognit.*, vol. 33, no. 1, pp. 133–147, 2000.

[10] C. B. Bose and S.-S. Kuo, "Connected and degraded text recognition using hidden Markov model," *Pattern Recognit.*, vol. 27, no. 10, pp. 1345–1363, Oct. 1994.

[11] P. Dreuw, S. Jonas, and H. Ney, "White-space models for offline Arabic handwriting recognition," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 2008, pp. 1–4.

[12] H.-H. Shih, S. S. Narayanan, and C.-C. J. Kuo, "A statistical multidimensional humming transcription using phone level hidden Markov models for query by humming systems," *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, vol. 1. pp. I–61–4 vol.1, 2003.

[13] S. J. Young, G. Evermann, M. J. F. Gales, D. Kershaw, G. Moore, J. J. Odell, D. G. Ollason, D. Povey, V. Valtchev, and P. C. Woodland, "The HTK book version 3.4," 2006.

[14] S. J. Young and P. C. Woodland, "State clustering in hidden Markov model-based continuous speech recognition," *Comput. Speech Lang.*, vol. 8, no. 4, pp. 369–383, Oct. 1994.

[15] A. Kosmala, J. Rottland, and G. Rigoll, "Improved on-line handwriting recognition using context dependent hidden Markov models," in *Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on*, 1997, vol. 2, pp. 641–644 vol.2.

[16] G. A. Fink and T. Plotz, "On the Use of Context-Dependent Modeling Units for HMM-Based Offline Handwriting Recognition," in *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, 2007, vol. 2, pp. 729–733.

[17] R. Prasad, S. Saleem, M. Kamali, R. Meermeier, and P. Natarajan, "Improvements in hidden Markov model based Arabic OCR," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 2008, pp. 1–4.

[18] M. Hamdani, A. E.-D. Mousa, and H. Ney, "Open Vocabulary Arabic Handwriting Recognition Using Morphological Decomposition," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, 2013, pp. 280–284.

[19] 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.

[20] J. Luettin, G. Potamianos, and C. Neti, "Asynchronous stream modeling for large vocabulary audio-visual speech recognition," *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, vol. 1. pp. 169–172 vol.1, 2001.

[21] H. Manabe and Z. Zhang, "Multi-stream HMM for EMG-based speech recognition," *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE*, vol. 2. pp. 4389–4392, 2004.

[22] Y. Kessentini, T. Paquet, and A. Hamadou, "Off-line handwritten word recognition using multi-stream hidden Markov models," *Pattern Recognit. Lett.*, vol. 31, no. 1, pp. 60–70, 2010.

[23] 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.

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

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.