Towards Arabic Handwritten Word Recognition via Probabilistic Graphical Models

Akram Khémiri, Afef Kacem
University of Tunis, LaTICE-ESSTT
Tunis, Tunisia
akramkhemiri@gmail.com
afef.kacem@esstt.rnu.tn

Abdel Belaïd
University of Lorraine, LORIA
Nancy, France
abdel.belaid@loria.fr

***Abstract*—In this work, we propose a novel system for the recognition of handwritten Arabic words. It is evolved based on horizontal-vertical Hidden Markov Model and Dynamic Bayesian Network Model. Our strategy consists of looking for various HMM architectures and selecting those which provide the best recognition performance. Experiments on handwritten Arabic words from IFN/ENIT strongly support the feasibility of the proposed approach. The recognition rates achieve 92.19% with horizontal-vertical Hidden Markov Model and 88.82% with a Dynamic Bayesian Network.**

Keywords— Hidden Markov Model; Dynamic Bayesian Network; Feature extraction; Pattern recognition

I. INTRODUCTION

The objective of this research is to investigate the use of Probabilistic Graphical Models (PGMs) for off-line recognition of Arabic handwritten words. Arabic script is naturally both cursive and unconstrained. Its recognition is a difficult task due to the high variability and uncertainty of human writing. Arabic contains dots and other small marks that can change the meaning of a word. These diacritic signs are needed to be taken into account by any computerized recognition system. Along with the dots and other marks representing vowels, this makes the effective size of the alphabet about 160 characters [5].

As defined by [17], PGMs are diagrammatic representations of probability distribution. They use a graph-based representation as the basis for compactly encoding a complex distribution over a high-dimensional space. In this graphical representation, nodes represent random variables and links express probabilistic relationships between variables. The attractive feature of graph-based methods is that, once the problem at hand has been abstracted to a relational structure, techniques from linear algebra, statistics, probability theory and spectral theory can be used for purposes of analysis.

A well known graphical modeling tools include stochastic models especially Hidden Markov Models (HMMs). Many variations of HMMs have been adapted and used in script recognition. Discrete, continuous and semi-continuous types were used with various topologies ranging from ergodic to left-to-right models with no state skipping. HMM-based algorithms were designed to handle letters, words, stroke or pseudo-characters using one-dimensional, two-dimensional or planar HMMs [3], [4], [6], [12], etc.

HMM-based systems received most of the attention, but other techniques were also used and proved to have satisfying results. This is the case of Bayesian Networks (BNs) which represent a set of random variables and their conditional dependencies via a directed acyclic graph (DAG). BNs allow representing probability models in an efficient and intuitive way [9], [16]. A Dynamic Bayesian Network (DBN) is a BN which relates variables to each other over adjacent time steps. The temporal extension of BN, towards DBN, has been recently applied to a range of different domains [14], [15].

Another kind of application exploits the ability of DBNs to be trained to detect patterns. For that, the observed information used as an input for the DBN is made of pre-extracted features. It is possible to use low-level data such as image pixels, as shown for instance by the application of DBNs to character recognition [10]. In [11], multiple models of BNs are applied for off-line recognition of Arabic handwritten Tunisian city names, extracted from IFN/ENIT database. Notice that a HMM can be considered the simplest DBN where there is only one observation stream and one state sequence. In fact, the main difference between a HMM and a DBN, as it will be explained later, is that in a DBN the hidden states are represented by a set of random variables whereas in a HMM, the state space consists of a single random variable.

The rest of the paper is organized as follows. In section II, we briefly present the state of the art in the field of BNs and writing recognition. In section III, we describe our proposed system based on many types of HMMs and Dynamic Bayesian Networks and show how our work makes it suitable for handwritten Arabic word recognition. In section IV, we display and discuss some experimental results. Conclusions and prospects are drawn in section V.

II. An Overview of PGMs for Arabic Writing Recognition

Table I summarizes some PGM-based recognition systems to which, we will compare our proposed system. In [2], a feature vector for each word mirror image is performed by applying a horizontal sliding window having the same height of the word image. Words are finally recognized using HMM (a left to right Bakis topology) and DBNs (conceived as several coupled HMMs architectures: state coupled model, general coupled model and auto regressive coupled model). Since DBN is working based on time slice, this is consistent with features extracted from sliding windows.

In [11], authors divide the word image into three elementary building blocks reflecting its local description. For each block (composed of a character or a part of the word), they compute a vector of descriptors which include moment invariants of Zernike and Hu descriptors. As these descriptors provide signatures of continuous values and BN requires discrete variables, a discretization method, based on K-means, is used to transform the variables with continuous values into variables with discrete value. Finally, they apply four variants of Bayesian networks classifiers (Naïve Bayes, Tree Augmented Naïve Bayes (TAN), Forest Augmented Naïve Bayes (FAN) and DBN to classify the whole word image.

In [13], authors propose to consider planar HMM (PHMM) based architecture is adopted. A PHAMM is a HMM whose emission probabilities are also modeled by HMMs. The retained PHMM architecture has a vertical principal model composed of seven super-states: beginning, end and five intermediate super-states associated to the different logical bands (median zone, upper/lower extensions and diacritics zones).

TABLE I. Some Related Works

System	*Database*	*Recognition Rate (%)*
[2]	IFN/ENIT	HMM:82, DBN:66
[11]	IFN/ENIT: 18 words, 3600 samples	FAN:82.56, TAN:80 Naïve BN:73, DBN:83.7
[13]	IFN/ENIT: 25 words, 2347 samples	Planar HMM:88.7

III. Proposed System

The recognition of Arabic handwritten word is achieved into three principal steps: baseline estimation, feature extraction and word classification.

A. Baseline Estimation

Due to letter extensions (ascending or descending letters), words are not usually written in a single baseline and position of ascenders and descenders varies according to the writer's style. For that reason our system tries not to extract a single baseline for the entire word, but a sequence of sub-baselines for an accurate structural feature extraction. It starts by extracting the connected components of the Arabic word, and then filtering diacritics. An approximate baseline by word horizontal projection is obtained (see the purple line in Fig. 1). Let's only consider the commune zone between PAWs (Parts of Arabic Word) and a variation range (an approximate central band) of ±10 pixels relative to the approximate baseline (see Fig. 1). This commune zone includes graphemes which correspond to characters or portions of characters. For each PAW, the system detects its proper baseline as follows:

- If the PAW is wide enough (involves more than one character), the baseline is extracted via horizontal projection of the commune zone (see Fig. 2(a)),
- If the PAW is reduced to one character (its width is relatively small compared to that of the word), then
  - If the PAW looks like the character 'Alif' (ا) which is an ascending character with an aspect ratio much greater than one, then the baseline will be extracted at the commune zone bottom (see Fig. 2(b)),
  - If the PAW concerns characters such as 'Raa' (ر), 'Noun' (ن), 'Aïn' (ع) or 'Haa'(ح) which can be assimilated to descending characters having an aspect ratio generally lower than one, then the baseline will be extracted at the commune zone top (see Fig. 2(c)),
  - If the PAW has the form of the character 'Dal' (د) seen as a central letter (without extensions) with an aspect ratio close to one, then the baseline will be extracted at the commune zone bottom (see Fig. 2(d)),
  - If the PAW has the form of an isolated loop like 'Ta' (ة) and 'Ha' (ه), then the baseline will be also extracted at the bottom of the commune zone (see Fig. 2 (e)).

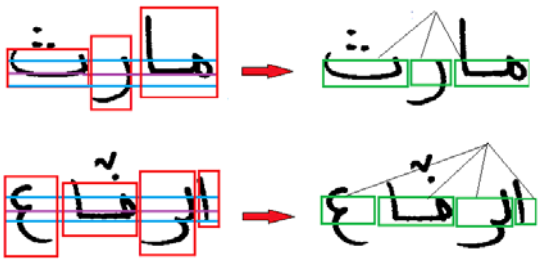

Fig. 1. Commune zone between PAWs and approximate central band.

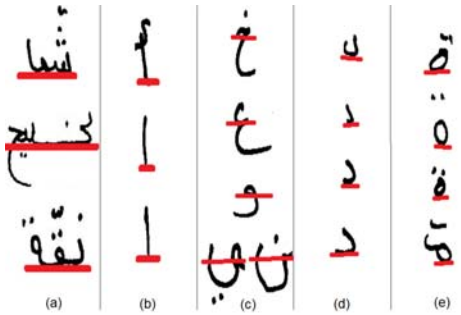

Fig. 2. Examples of PAW's baseline.

The entire word baseline is formed by the juxtaposition of its PAW baselines as shown in Fig. 3 (a). A comparison with the word horizontal projection is made (see Fig. 3(a) and (b)). As it can be seen, the extracted baselines fit better to word

support lines than those obtained by word horizontal projection.

Fig. 3. (a) Extracted baseline, (b) baseline by horizontal projection

B. Feature Extraction

For more effective representation of the word image, we extracted structural features (ascenders, descenders, loops, diacritic points and their position related to the baseline) which describe the topological and geometrical characteristics of the word and statistical features at pixel level (number of black to white or white to black pixel transitions) which describe the characteristic measurements of the word. Here, we only focused on extraction of structural features. For that, the effective word central band must be determined given baseline estimation. As Arabic letters without extensions (such as 'ة', 'ب', 'ج', 'ف', etc.), in central band, protrude more above than below the baseline (see Fig. 3), the upper resp. the lower band will be respectively and empirically fixed to baseline -25 pixels and +10 pixels (see Fig. 4). In what follows, we will explain how to extract some structural features such as diacritic points, ascenders, descenders and loops using upper, central and lower bands of the word.

Fig. 4. Upper, central and lower bands extraction based on baseline ordinate.

Diacritic points may occur in the upper and/or the lower bands of words, at the beginning, in the middle and/or at their end. The number of diacritic points varies from one to three. Notice that diacritic points do not cross the baseline and they are reduced in area (height*width) and have high density (number of black pixels/area). Sometimes, two or three diacritic points can be attached and then considered as only diacritic point (see Fig. 5).

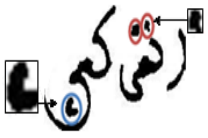

Fig. 5. Diacritic points extraction from the word التركي.

Ascenders and descenders are respectively located in the upper and lower bands of words. Ascenders can be of two types: 'Stem-Alif' (ا) and 'Stem-Kef' (ك) (see Fig. 6) while descenders can be classified as 'Leg-Noun' (ن), 'Leg-Raa' (ر), 'Leg-Haa' (ح) (see Fig. 7). Stem classification is based on aspect ratio, density of their connected compounds and the number of pixel transitions (white to black or black to white) along a vertical axis (colored in yellow) which divides the image in the middle. It is clear that 'Stem-Alif' has higher aspect ratio and density than 'Stem-Kef' but a lower number of pixel transitions (2 and 4 pixel transitions resp. for 'Stem-Alif' and 'Stem-Kef') as explained in Fig. 6.

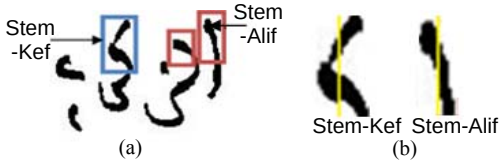

Fig. 6. (a) Ascender extraction (b) Ascender classification based on number of white to black or black to white pixel transitions.

Descender classification is based on the aspect ratio, density of their connected compounds and the number and position of their contact points with the lower line of the central band. As Fig. 7 shows, in 'Leg-Noun' there is generally two contact points with the lower line of the central band whereas in both 'Leg-Raa' and 'Leg-Haa' there is only one contact point. To distinguish between 'Leg-Raa' and 'Leg-Haa', we check if there is a discontinuity on the left or on the right of the connected component. In fact, 'Leg-Raa' is discontinuous on the left which is not the case in 'Leg-Haa'. More details about structural features extraction have be done in a previous work [7], [8]. Loops may be at the beginning, middle and/or at the end of the word (see Fig. 8).

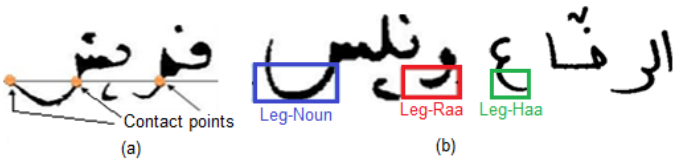

Fig. 7. (a) Contact points, (b) Descender classification.

Fig. 8. Loops in different positions in the word.

C. *Word Classification based on PGMs*

We proposed various HMM models as shown below. For all proposed HMM models, we used the Baum–Welch algorithm, to maximize the observation sequence P=(O|λ) of the HMM model λ=(π, A, B) and so to optimize the HMM model parameters based on the training data. In the testing phase, we used the Viterbi algorithm for word recognition. Hereafter, we describe architectures of the proposed HMM models.

- **Horizontal HMM**: We divided the word image into 3 rows: R_1, R_2 and R_3 and three columns: C_1, C_2 and C_3 as shown in Fig. 9 where R_1 is the higher quarter, R_2 is the central half and R_3 is the lower quarter of word image. For each row, from right to left, we considered local pixel configurations as statistical feature at pixel level. We computed the number of pixel transitions (white to black or black to white) along an horizontal axis which divides the word image rows in the middle, considering their position in the word: in the beginning (C_1, the rightist quarter), in the middle (C_2, the middle half) or at the end of the word (C_3, the leftist quarter). Note that C_2 is taken twice that of C_1 and C_3 to consider the elongation aspect, often seen in Arabic words. The 9 obtained blocks $(R_i, C_j)_{i=1..3, j=1..3}$ reflect a local description of the word image. As shown in Fig. 11, there are two pixel transitions in (R_1, C_1), (R_1, C_2), (R_2, C_1), (R_2, C_3) and (R_3, C_3) blocks. There are five pixel transitions in (R_2, C_2) block and no transitions in the remaining blocks. We noted that the number of pixel transitions can vary from zero to five transitions (6 possible values) per row and column. So 54 values (18 per row), are computed. Fig. 10 shows the used Horizontal HMM (H-HMM) structure where each row presents a node.

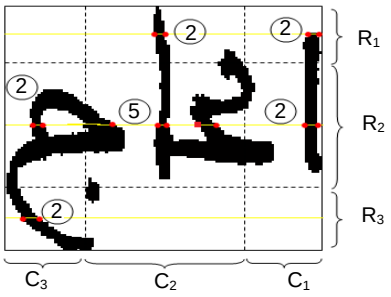

Fig. 9. Number of white/black and black/white pixel transitions.

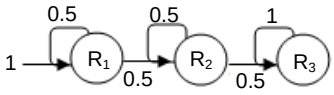

Fig. 10. H-HMM structure (initialization step).

- **Vertical HMM**: Dividing word into three columns from right to left, as shown in Fig. 9, serves to look if extracted structural features are in the beginning: C_1, in the middle: C_2 or at the end: C_3 of the word. Word description is then performed from right to left, based on the Arabic writing direction, as a sequence of structural features gathered from each column. From the image word 'الحاج', we respectively extracted a 'Stem-Alif', two 'Stem-Alif', a diacritic point and a 'Leg-Haa' in the beginning, middle and end of the word as shown in Fig. 11.

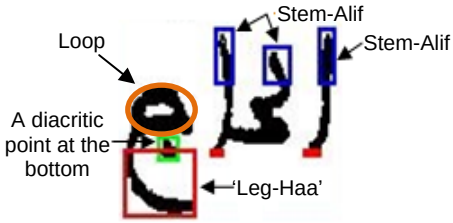

Fig. 11. Structural feature extracted from the word image: الحاج.

Fig. 12 shows the structure of the vertical HMM (V-HMM) where each column represents a node. Both of H-HMM and V-HMM are discrete, one-dimensional and left-to-right HMM without state skipping. We selected this basic topology because it has been effectively used in handwriting recognition.

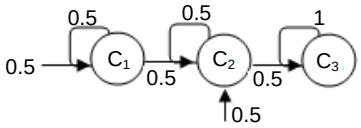

Fig. 12. V-HMM structure (initialization step).

- **Horizontal and Vertical HMM**: So far, we considered a single (vertical or horizontal) HMM to model word images. Here, we conceived an independent two-dimensional HMM which considers features extracted from both rows (54 features) and columns (30 features) of the word. Fig. 13 shows the structure of the proposed horizontal and vertical HMM (HV-HMM).

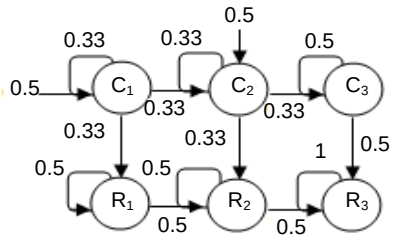

Fig. 13. HV-HMM structure (initialization step).

- **Dynamic Bayesian Network**: In this work, we also implemented a DBN to recognize an unknown Arabic handwritten word. Before discussing DBNs, the basic foundation of BNs is outlined below. As defined by [10] a static BN associated with a set of random variables $X=(X_1, X_2, ..., X_N)$ is a pair: $B=(G, \theta)$ where G is the structure of the BN (a Directed Acyclic Graph whose nodes correspond to the variables $X_i \in X$

and whose edges represent their conditional dependencies) and θ represents the set of parameters encoding the conditional probabilities of each node variable given its parents. A key property of BNs is that the joint probability distribution factors as:

$$P(X_1, X_2, ..., X_N) = \prod_{i=1}^{N} P(X_i | P_a(X))$$

where $P_a(X_i)$ denotes the parents of X_i and N refers to nodes number. This property is central in the development of fast inference algorithms. DBNs are an extension of static BNs to temporal processes occurring at discrete times ($t \geq 1$). They are defined in [18] as a repetition of conventional networks in which we add a causal link (representing the time dependencies) from one time slice to another. As underlined by [10], several coupled HMMs architectures can be represented as a single DBN. In this work, we constructed a coupled H-HMM and V-HMM by adding directed edges between the horizontal and vertical streams within the same time slice. In the coupled models, there are two states: vertical and horizontal states. The vertical states correspond to the column observations, while the horizontal states correspond to the row observations respectively. Similar to the classic left right HMMs, a transition to the vertical state X_t^1 is depending only on the preceding state value X_{t-1}^1. However, a transition to the horizontal state X_t^2 is depending on both the preceding state value X_{t-1}^2 and the current vertical state X_t^1. The observation dependences are expressed by the dependences between the horizontal and vertical states. As shown in Fig. 14, the proposed DBN, conceived as a state coupled model, is obtained by adding the directed edges between the hidden state nodes of both vertical and horizontal HMMs. Coupled H-HMM and V-HMM are divided into 3 times slices (t_1, t_2 and t_3). The DBN, in each time slice, contains a number of random variables representing observations and hidden states of the process. The dependencies between H-HMM and V-HMM, modeling both horizontally and vertically stream, are performed by the relations between states. A state of a HMM is connected to the adjacent state in the same slice of the other HMM. The DBN is composed of a sequence of $t=3$ hidden state variables. Note that a hidden state in the DBN is represented by a set of hidden state variables (here is represented by two hidden state variables). Let use indices $i=1, 2$ and 3 to denote the 3 times slices. The variables X_i and Y_i denote the respective hidden state and observation attributes. The process modeled by the proposed DBN is first-order Markovian and stationary since the parents of any variables X_t^i or Y_t^i belong to the time slice t or $t-1$ only and that model parameters are independent of t.

In order to have a complete specification of our DBN, we needed to define: transition probability between states ($X_t|X_{t-1}$) the conditional probability of hidden states given an observation $P(Y_t|X_t)$ and the initial state probability $P(X_1)$. The first two parameters should be given at each time. To learn DBN parameters, a model is developed for each class. Models of all classes share a single DBN structure, but their parameters change from one class to another. Learning the model parameters is performed independently model by model, using the Expectation Maximization (EM) algorithm which is an iterative approach of maximum likelihood estimators. To recognize a word, its features are extracted. Then, the likelihood of each model relative to the sample is calculated using an exact inference algorithm: a junction tree algorithm. The word is assigned to the class that gives the maximum likelihood.

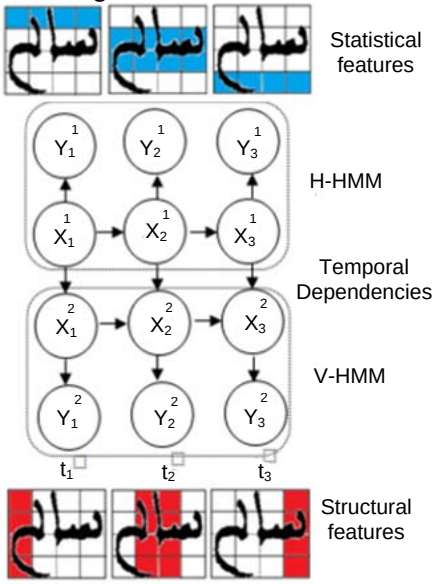

Fig. 14. DBN model.

IV. EXPERIMENTATIONS

Our system has been applied to the well-known IFN/ENIT database containing handwriting words written by different writers. From experiments, we note that HMMs generally outperform DBN in terms of higher recognition rate and lower complexity. In fact, the highest rate was reached when using an independent HV-HMM (see table II). It is surprising to note that DBN performs much worse than HMM, although HMM is regarded as much simplified version of DBN. As explained in [2], DBN is generally capable of modeling more complicated cases like spatial and temporal structure, even in multi-resolution. On the contrast, HMM is suitable for modeling linear cases such as speech. As a result, DBN has the potential to deal well with handwritten recognition tasks as images of handwritten words are in 2D. However, extracted statistical features at pixel level and

structural features, scanning the word binary images from right to left might have simplified handwritten recognition to a linear case, hence HMM works more effectively than DBN. How to extract useful features and fully make use of the potential of DBN needs further investigation. Compared to some related works (see Table I), our system achieved an average recognition rate of 92.19% which is clearly better. Note that we used a larger sample of words: 5279 samples of 21 words compared to those works.

TABLE II. RECOGNITION RATES

Word	*samples*	*Training*	*Test*	*DBN (%)*	*HV-HMM (%)*
الرضاع	374	249	125	94.4	97.6
الخليج	345	230	115	83.49	88.69
نقة	343	229	114	93.86	95.61
شعال	338	225	113	92.03	95.57
مارث	338	225	113	84.07	91.15
شماخ	322	215	107	90.65	90.65
زنوش	319	213	106	92.45	93.40
الدخانية	318	212	106	89.62	92.45
الفايض	312	208	104	85.58	87.5
اكودة	298	199	99	86.87	89.90
سبعة أبار	293	195	98	93.88	94.90
سيدي ابراهيم الزهار	290	193	97	90.72	90.72
المرناقية 20 مارس	274	183	91	93.41	94.50
شتاوة صحراوي	245	163	82	96.34	96.34
الفكة	171	114	57	91.23	92.98
اوتيك	138	92	46	82.61	91.11
الفحص	138	92	46	93.48	91.30
الشرايع	134	89	45	82.22	95.55
حي الانطلاقة	120	80	40	80	90
شواط	109	73	36	83.33	86.11
حي التضامن	60	40	20	85	90
Average	**5279**	**3519**	**1760**	**88.82**	**92.19**

V. CONCLUSION AND FUTURE DIRECTIONS

Our objective is to conceive and carry out an automatic off-line recognition system of Arabic handwritten words based on PGMs. The proposed system is divided into three stages, namely preprocessing, feature extraction and word classification. Preprocessing includes baseline estimation. Structural features (ascendants, descendants, loops and diacritic points) and statistical features at pixel level (pixel density distributions and local pixel configurations) are then extracted from word images. Notice that the use of multiple sources of information represents one of the advisable orientations in pattern recognition. Words are finally recognized using a variety of PGMs, including traditional Markovian models (H-HMM, V-HMM and HV-HMM) and DBN. Experiments have been conducted on IFN/ENIT database. Our system showed the best recognition rate among the existing work reported using the same database. The superior performance of HV-HMM can be attributed to the fact it better represents the perceptually relevant aspects of the Arabic handwritten word and it considers the different morphological variation specific to Arabic script. In the future, we plan to look for best features for Arabic handwritten word description and also for best PGM architectures for its classification.

REFERENCES

[1] J. H. AlKhateeb, "Offline Handwritten Arabic Digit Recognition Using Dynamic Bayesian Network," ICCIT, 2012, pp. 176-180.

[2] J. H. AlKhateeb, O. Pauplin, J. Ren, and J. Jiang, "Performance of hidden Markov model and dynamic Bayesian network classifiers on handwritten Arabic word recognition," Journal knowledge-based systems, vol. 24, issue 5, July 2011, pp. 680-688.

[3] A. Benouareth, A. Ennaji and M. Sellami, "Arabic Handwritten Word Recognition Using HMMs with Explicit State Duration," EURASIP, 2008.

[4] N. Ben Amara, A. Belaïd and N. Ellouze, "Utilisation des modèles markoviens en reconnaissance de l'écriture arabe état de l'art," CIFED, Lyon, juillet 2000.

[5] P. Burrow, Arabic Handwriting Recognition, PhD thesis, University of Edinburgh, 2004.

[6] M. El Yacoubi, R. Gilloux, C. Sabourin, and Y. Suen, "An HMM-based approach for off-line unconstrained handwritten word modeling and recognition," IEEE Transactions on PAMI, vol. 21, no 8, 1999, pp. 752-760.

[7] A. Kacem, N. Aouïti, and A. Belaïd, "Structural Features Extraction for Handwritten Arabic Personal Names Recognition," ICFHR, 2012.

[8] A. Kacem, A. Khémiri A, N. Aouïti, and N. Aouadi., "Système, à base de MMC, pour la reconnaissance de noms propres manuscrits Arabes", CIDE, 2012.

[9] D. L. Kelly, and C. L. Smith, "Bayesian inference in probabilistic risk assessment: The current state of the art," Reliability Engineering and System Safety 94, 2008.

[10] L. Likforman-Sulem, and M. Sigelle, "Recognition of degraded characters using dynamic Bayesian networks," Pattern Recognition, 41, 2008.

[11] M. A. Mahjoub, N Ghanmy, K. Jayech and I. Miled, "Multiple models of Bayesian networks applied to offline recognition of Arabic handwritten city names," CVPR, 2013.

[12] V. Märgner, H. Abed, and M. Pechwitz, "Offline Handwritten Arabic Word Recognition Using HMM: a Character Based Approach without Explicit Segmentation," CIFED, 2006, Fribourg, Swiss, pp. 18-21.

[13] S. Masmoudi Touj, N. Essoukri Ben Amara, and H. Amiri, "Arabic Handwritten Words Recognition Based on a Planar Hidden Markov Model," IAJIT, vol. , no 4, 2005, pp. 318-325.

[14] V. Mihajlovic, and M. Petkovic, "Dynamic Bayesian Networks, A State of the Art, Technical reports series," TR-CTIT-34, 2001.

[15] K. P. Murphy, "Dynamic Bayesian Networks, Representation, Inference and Learning," PhD dissertation, UC Berkeley, Computer Science Division, July 2002.

[16] O. Pauplin and J. Jiang, "A Dynamic Bayesian Network Based Structural Learning towards Automated Handwritten Digit Recognition," 2010.

[17] S. Srihari, www.cedar.buffalo.edu/~srihari /CSE574/, 2010.

[18] C. Y. Suen, M. Berthod, and S. Mori, "Automatic Recognition of Handprinted Characters: The State of The Art," Proceedings of the IEEE, vol. 68, no. 4, 1980, pp. 469-487.