Electronic Letters on Computer Vision and Image Analysis 13(3):41-62, 2014

A PGM-based System for Arabic Handwritten Word Recognition

Afef Kacem Echi*, Akram Khémiri* and Abdel Belaïd+

* *University of Tunis, ENSIT-LaTICE, 5 Avenue Taha Husseïn, BP 56 Babmnara 1008 Tunis, Tunisia*

+ *University of Lorraine, LORIA, Campus scientifique. BP 239 54506 Vandoeuvre-lès-Nancy, France*

Received 13th Dec 2013; accepted 20th Sep 2014

Abstract

This paper describes a system for off-line recognition of handwritten Arabic words. It uses simple and easily extractable features to construct feature vectors for words in the vocabulary. Some of these features are statistical, based on pixel distributions and local pixel configurations. Others are structural, based on the presence of ascenders, descenders and diacritic points. The system is evolved based on vertical and horizontal Hidden Markov Models and Dynamic Bayesian Network. Our strategy consists of looking for various architectures and selecting those which provide the best recognition performance. Experiments on handwritten Arabic words from IFN/ENIT database and ancient manuscripts strongly support the feasibility of the proposed system. The recognition rates achieve 91.89% (IFN/ENIT) and 94.61% (ancient manuscripts).

Key Words: Bayesian networks, Hidden Markov Models, Arabic handwritten recognition.

1 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 is a complex text language, because it has bidirectional script. It is written right to left, except for numbers. 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. Often the diacritic marks representing vowels are left out, and the word must be identified from its context. Many Arabic letters change their form depending on whether they appear alone, at the beginning, middle or end of the word. Along with the dots and other marks representing vowels, this makes the effective size of the alphabet about 160 characters [13].

PGMs are being exercised for writing recognition, showing promising results. As defined by [12], PGMs are diagrammatic representations of probability distribution. A well known graphical modelling tools include stochastic models especially Hidden Markov Models (HMMs). Many variations of HMMs have been adapted and used in script recognition research [18], [20], [21] and [22] . Discrete, continuous and semi-continuous types were used with various topologies ranging from ergodic to left-to-right models with no state

Correspondence to: <afef.kacem@esstt.rnu.tn>

Recommended for acceptance by <Xavier Otazu>
ELCVIA ISSN:1577-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

skipping. HMM-based algorithms were designed to handle letters, words, stroke or pseudo-characters using one-dimensional, two-dimensional or planar HMMs. Results were very encouraging in the handwritten case and appear to handle the cursiveness well as reported in [18]. Some works focused on the use of HMMs for the recognition of isolated forms of Arabic letters or digits only [8], [9] and [10]. HMM success can be attributed to the probabilistic nature of HMM models, which can perform a robust modeling of the handwriting signal with huge variability and sometimes corrupted by noise. HMMs have been already applied to handwritten Arabic word recognition [14], [15], [16], etc.

HMM-based systems received most of the attention, but other techniques were also used and proved to have satisfying results. Other graphical models are 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 [3], [4]. 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 [5], [6], have been recently applied to a range of different domains. In fact, DBNs have been used in speech recognition [7] as a flexible and efficient extension of HMMs. 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 [8]. In [17], multiple models of BNs are applied to 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 a single random variable.

This paper presents a comparative study of two machine learning techniques for recognizing handwritten Arabic words, where hidden HMMs and DBNs were evaluated. In section 2, we briefly present the state of the art in the field of BNs and writing recognition. In section 3, we describe our proposed system based on HMMs and on a DBN and show how the dynamic character of DBN makes it suitable for handwritten Arabic word recognition. In section 4, we display and discuss some experimental results. Conclusions and prospects are drawn in section 5.

2 An overview of DBNs for writing recognition

Before discussing PGM-based systems in general and DBNs in particular, the basic foundation of HMMs, BNs and DBNs are outlined below. The HMM is a finite set of states (N), each of which is associated with a probability distribution. Transitions among the states are governed by a set of probabilities called transition probabilities [11].

Generally, HMMs are denoted by λ which is defined by three sets of parameters: λ=(Π, A, B) where A, B, and Π represent the following parameters.

- **Matrix of transition probabilities (A):**

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \tag{1}$$

$$A = \{a_{i,j} | a_{i,j} = P(S_t = j | S_{t-1} = i)\} \tag{2}$$

$$a_{m,n} = P(S_n | S_m); m, n = 1, 2 \tag{3}$$

Where a_{mn} is the probability that the current state is S_n given that the previous state is S_m. This is calculated as the expected number of transitions from state S_m to state S_n divided by the expected number of transitions out of state S_m.

- **Matrix of emission probabilities (B):**

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \tag{4}$$

$$B = \{b_j(o_k)|b_j(o_k) = P(O_t = O_k|S_t = j)\} \tag{5}$$

$$b_{np} = b_n(p) = P(O_p|S_n); n = 1, 2; p = 1, 2, 3 \tag{6}$$

Where $b_n(p)$ is the probability that the current observation is O_p given that the current state is S_n. It can be calculated as the expected number of times where O_p observed with S_n divided by the expected number of times in state S_n.

- **Initial states probabilities (Π):**

$$\pi = \begin{bmatrix} \Pi_1 \\ \Pi_2 \end{bmatrix} \tag{7}$$

$$\pi = \{\pi_i|\pi_i = P(S_1 = i)\} \tag{8}$$

$$\pi_m = P(S_m); m = 1, 2 \tag{9}$$

Here π_m is the expected number of times being in state S_m at the start time.

While using the HMMs, there are three main problems associated with:
- The evaluation problem: Calculating the probability that a model λ=(Π, A, B) created a given sequence of observations.
- The decoding problem: Finding the most likely sequence of hidden states, in a given model λ=(Π, A, B), that is created by a given sequence of observations.
- The learning problem: Estimating the model parameters λ=(Π, A, B) so that they best fit a given training sequences of observations.

As defined by [8], 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 i.e., a Direct Acyclic Graph (DAG) 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. The distributions are represented either by a Conditional Probability Table (CPT) when a node and its parents represent discrete variables, or by a Conditional Probability Distribution (CPD) when a node represents a continuous variable. Each CPD usually follows a Gaussian probability density function (pdf). A key property of BN is that the joint probability distribution factors as follows where $Pa(X_i)$ denotes the parents of X_i and N refers to nodes number (variables). Product terms are conditional distributions of each node conditioned on variables corresponding to parents of that node in the graph.

$$P(X_1, X_2, X_3, ..., X_N) = \prod_{i=1}^{N} P(X_i|Pa(X_i)) \tag{10}$$

This property is central in the development of fast inference algorithms. So a BN is described by two ways: 1) Qualitative description of dependencies between variables (causal graph) and 2) Quantitative description of these dependencies.

DBN are an extension of static BN to temporal processes occurring at discrete times [8]:
$t \geq 1$. In the following, we consider DBN models which have two observation streams. We will use indices $i = 1, 2$ to denote the two streams. The variables X_i and Y_i denote the respective hidden state and observation attributes in stream i. X_t^i and Y_t^i are the random variables (nodes) for X_i and Y_i at time t.

We assume that the process modeled by DBN is first-order Markovian and stationary. In practice, this means that the parents of any variable 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*. Parameters are thus tied and a DBN can be represented by the first two time slices as illustrated in Figure 6. For each observation sequence, the network is repeated as many times as necessary. Figure 6 shows an example of unrolled DBN for an observation sequence of length $T = 3$: the initial network is repeated T times. To fit the two observation sequences Y_1 and Y_2 of length T=3, the DBN is unrolled and represented on 3 time slices (see Figure 1 on the right). Parameters for this model are given by CPTs and CPDs:

- The three CPTs are: The initial state distribution encoding $P(X_1^1)$, the conditional state distribution $P(X_t^2|X_t^1)$ and finally the state transition distribution $P(X_t^2|X_{t-1}^2)$.
- The two CPDs are the Gaussian pdfs $P(Y_t^i|X_t^i), i = 1, 2$.

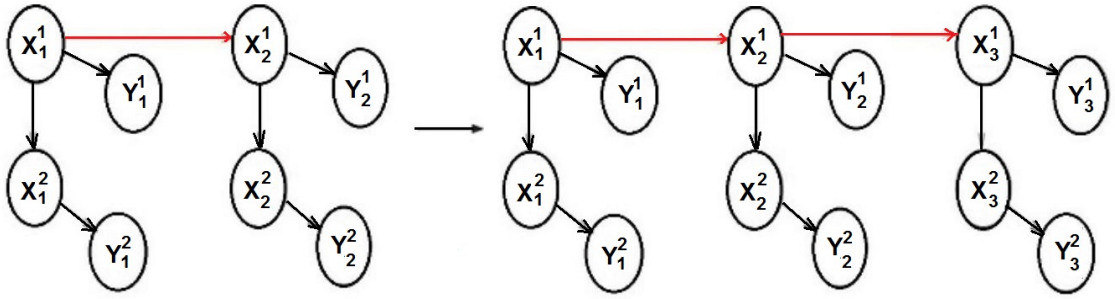


Figure 1: Unrolled DBN for an observation sequence of length T=3 [8]

In [8], authors investigate the application of DBN for the off-line recognition of degraded Latin handwritten digits. Digits are represented by coupling two HMM architectures into a single DBN model. The interacting HMMs are a vertical HMM and a horizontal HMM whose observations for the vertical (respectively horizontal) consist of columns (respectively rows) of pixels obtained from scanning the character image from left to right (respectively top to bottom) as shown in Figure 2 and Figure 3.

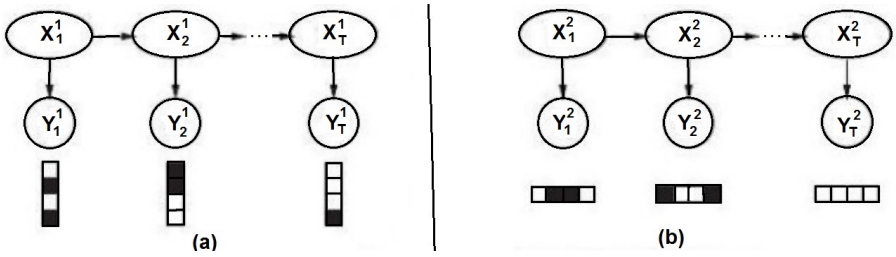


Figure 2: Independent HMM represented as DBN: (a) Vertical HMM (b) Horizontal HMM

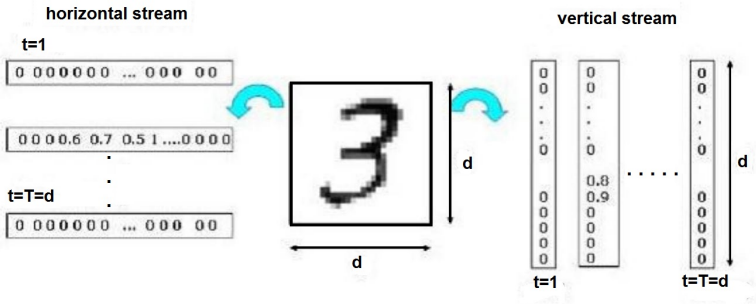


Figure 3: Horizontal and vertical observation sequences obtained by scanning digit 3 from top to bottom and from left to right, respectively. Digit images are normalized to size d: Length of observation sequences is $T = d$, length of observation vectors is also d.

Independent models also various coupling models (State-Coupled: STCPL, General-Coupled: GNLCPL and Auto-Regressive Coupled: ARCPL) are proposed in [8] where interactions are achieved through the causal influence between state variable (see Figures 4, 5, 6, 7). The STCPL model is obtained by adding the directed edges between the hidden state nodes of both vertical and horizontal HMMs. The GNLCPL model is obtained by adding an edge from hidden states in the horizontal stream X_t^2 to the observation variables in the vertical stream Y_t^1. The ARCPL model is obtained by coupling both vertical and horizontal streams. ARCPL model was chosen to be used since it is superior to other coupled models, having achieved the highest recognition rate.

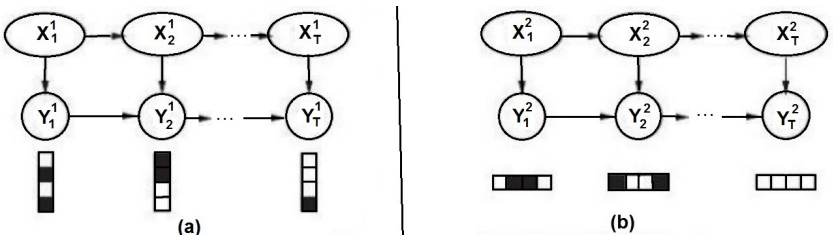


Figure 4: Independent auto-regressive AR-HMM represented as DBN: (a) AR-vertical and (b) AR-horizontal

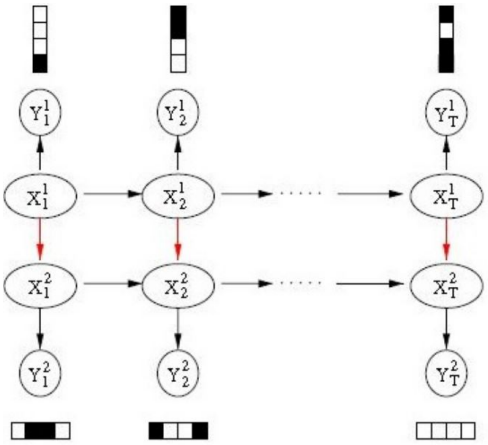


Figure 5: Coupled models: State-Coupled: STCPL.

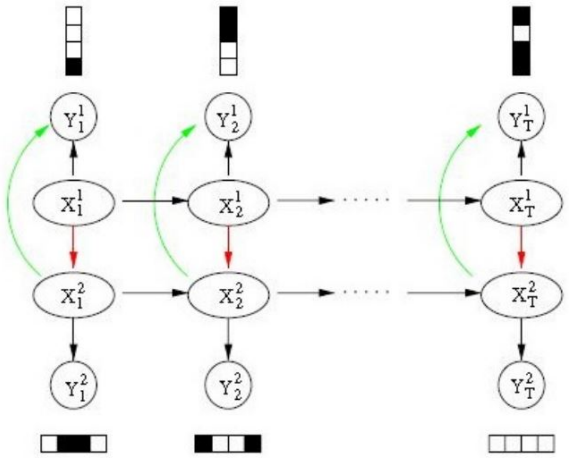


Figure 6: Coupled models: General-Coupled: GNLCPL.

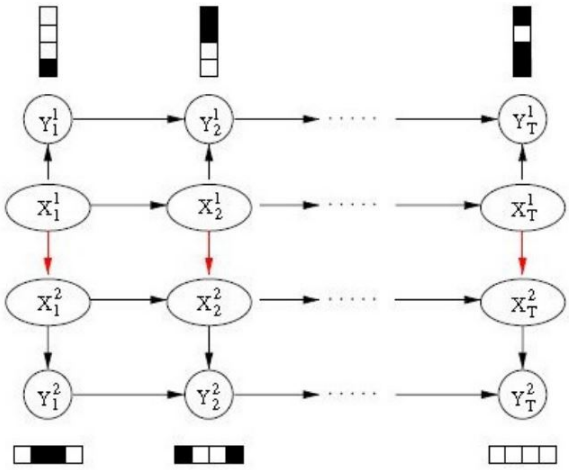


Figure 7: Coupled models: Auto-Regressive Coupled: ARCPL.

In the same work, authors use independent models and types of coupled models for the recognition of real degraded old printed characters extracted from the British Library's collection of digitized Renaissance festival books.

In [9], authors present four DBN models trained for classification of Latin handwritten digits. The structure of these models is partly inferred from the training data of each class of digit before performing parameter learning. The DBN models have their structure partly based on inter-slice links learned from data. Model_1 is a DBN made of evidence nodes only, with inter-slice links learned from the data (columns of pixels). Model_2 has the same evidence nodes as in Model_1, with one hidden node per slice, connected to every evidence nodes of the slice. Model_3 is a DBN observing columns and lines of pixels coupled through hidden nodes. Model_4 is same as Model_2, completed with intra-slice links learned from the data (see Figure 8).

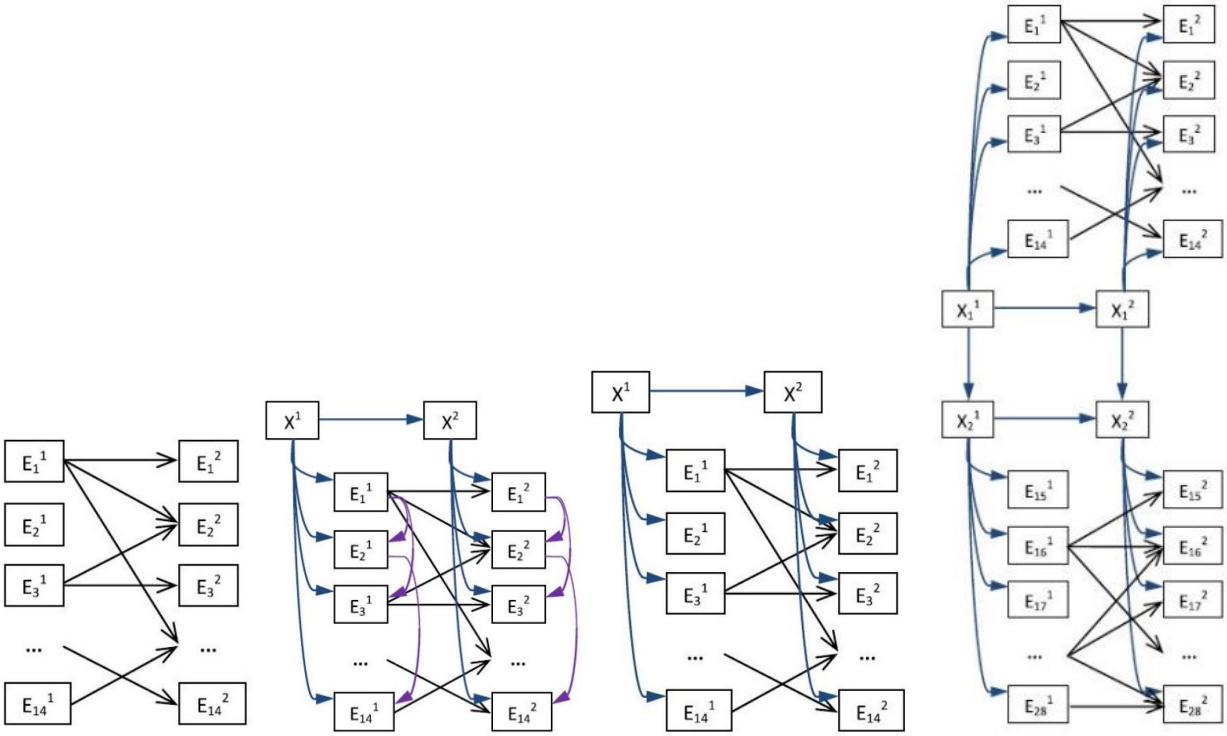


Figure 8: Model_1, Model_2, Model_3 and Model_4 (from left to right).

Like work reported in [8], authors in [10] use independent models (auto-regressive horizontal and vertical HMMs). To evaluate the performance of their recognition system, experiments are conducted on the ADBase database. Currently, only a small set of 500 digits are used in experiments. Note that the system relies on the DBN classification and density features. The main drawback is the long training time required for some applications.

To recognize Arabic handwritten words with DBN models, authors in [11] divide their work into three stages, namely pre-processing, feature extraction and classification. Pre-processing includes baseline estimation and normalization as well as segmentation. Pixel features are then extracted from each of the normalized words based on a sliding window approach. More precisely, 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). The DBN parameters are learned using the EM algorithm. The IFN/ENIT database is used for training and testing. We think that since DBN is working based on time slice, this is consistent with features extracted from sliding windows.

In [17], multiple models of BNs are applied to recognition of Arabic handwritten city names. First, 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 low-level features: Zernike and Hu moments. 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 (Nave Bayes, Tree Augmented Nave Bayes (TAN), Forest Augmented Nave Bayes (FAN) and DBN to classify the whole word image. Figure 9 shows the DBN model represented as coupled HMMs.

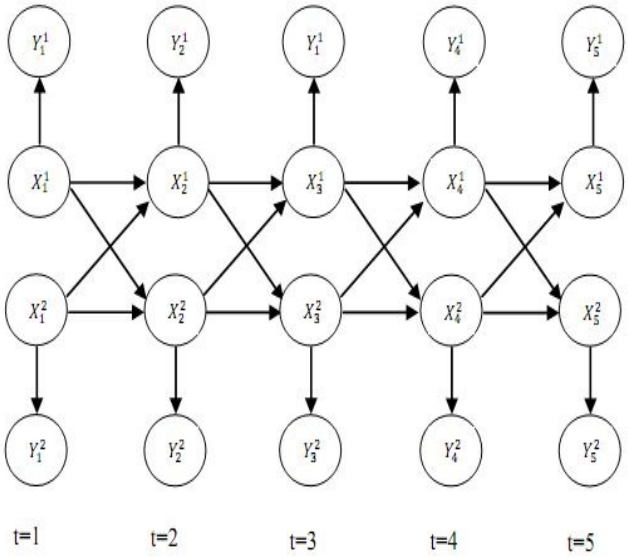


Figure 9: DBN model represented as coupled HMMs.

In [19], authors propose to consider planar HMM (PHMM) based architecture is adopted. A PHMM 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). In Figure 10, the architecture of PHMM for the word "الرقوبة" (Alragoba) is presented.

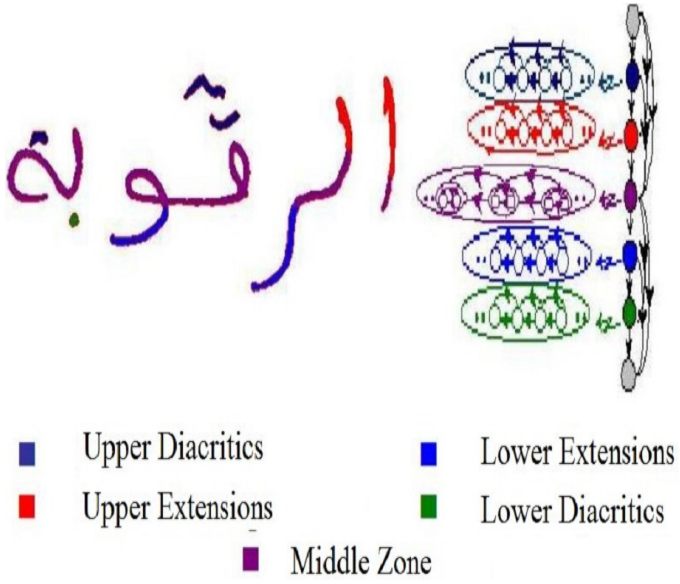


Figure 10: Architecture of PHMM for the word "الرقوبة" (Alragoba).

Below is a recapitulative table which summarizes some existing works for off-line pattern recognition based on PGM models.

Table 1: Off-line recognition.

Ref.	Pattern	Method	Database	Recognition Rate (%)
[8]	Latin Artificially degraded handwritten digits	AR, STCPL, GNLCPL, ARCPL	MNIST	AR vertical=93.2 AR Horizontal=87.7, STCPL=92.4, GNLCPL=93.4, ARCPL=94.9, ARCPL=94.9
[8]	Real Latin degraded old printed characters	AR, STCPL, GNLCPL, ARCPL	British Library's collection of digitized Renaissance festival book	AR vertical=94.5, AR Horizontal=91.2, STCPL=95.5, GNLCPL=94, ARCPL=96
[9]	Latin handwritten digits	4 models of learning DBN structure	MNIST	$Model_1$=67.6, $Model_2$=69.9, $Model_3$=74.8, $Model_4$=71.2
[10]	Arabic handwritten digits	AR, STCPL, GNLCPL, ARCPL	ADBase	95.26
[11]	Arabic handwritten words	AR, STCPL, GNLCPL, ARCPL	IFN/ENIT	66.56
[17]	Arabic handwritten words	Naïve BN, TAN, FAN, DBN	IFN/ENIT	Naïve BN=73, TAN=80, FAN=82.56, DBN=83.7
[19]	Arabic handwritten words	Plannar HMM	IFN/ENIT	PHMM=88.7

In sum, we can conclude that most of PGM-based systems, proposed for off-line handwritten digit, character or word recognition, are able to recognize the writing from limited and noisy (artificially or real degraded) observations, and to make good decisions under uncertainty. The main drawbacks are the long training time and the large input vocabulary required for some applications. Also and because of the complexity of the Arabic language, there has been less work in Arabic than Latin handwriting recognition based on PGMs.

3 Proposed PGM-based System

This section describes different independent discrete HMMs (horizontal HMM, vertical HMM and vertical-horizontal HMM) and a DBN (coupled vertical and horizontal HMMs) that we conceived for the off-line recognition of Arabic handwriting. As PGM-based systems need observation sequences as input, we used discrete values which are extracted from word images. In fact, because of the huge variability of the handwriting style and the noise affecting the data, it is almost impossible to directly recognize handwritten word from its bitmap representation. Thus, the need of features extraction method that allows extracting a feature set from the word image is obvious for classification. Feature extraction methods are generally based on two types of features: statistical and structural. Major statistical features, used for word representation, are derived from distribution of points: Zoning, projections and profiles, crossing and distances. Words can be represented by structural features with high tolerance to distortions and style variations. This type of representation may also encode some knowledge about word structure or may provide some knowledge as to what sort of components make up that word. Structural features are based on topological and geometrical properties of the word, such as aspect ratio, cross points, loops, branch points, strokes and their directions, inflection between two points, horizontal curves at top or bottom, etc. In this work, we explored various types of features which are popular for Arabic cursive handwriting recognition. Some of these features are statistical, based on pixel distributions or local pixel configurations. Others are structural, based on the presence of ascenders, descenders and diacritical marks. We believe that the use of multiple sources of information represents one of the advisable orientations in pattern recognition. We show how these features can be efficient within PGM-based system.

3.1 Horizontal HMM (H-HMM)

In off-line recognition systems based on HMMs, the main concept is to transform the word image into a sequence of observations. So, we divide 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 Figure 11 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 consider local pixel configurations as statistical feature at pixel level. We compute 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 nine obtained blocks: (R_1, C_1), (R_2, C_1), (R_3, C_1), (R_1, C_2), (R_2, C_2), (R_3, C_2), (R_1, C_3), (R_2, C_3), (R_3, C_3) reflect a local description of the word image "الحاج"

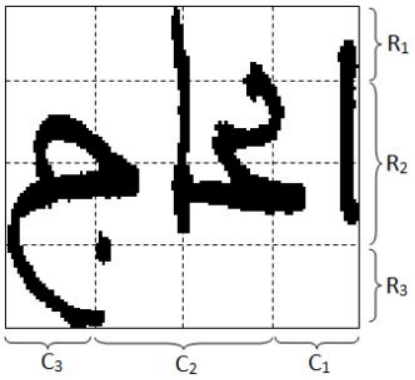


Figure 11: Example of word image "الحاج" (*Alhaj*) divided into lines and columns.

As shown in Figure 13, 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 five pixel transitions in (R_2,C_2) block and no transitions in the remaining blocks. We noted that the number of white to black pixel transitions can vary from zero to five transitions (6 possible values: 0, 1, 2, 3, 4, 5+) per row and column. So for regions that contain 5 or more transitions (which is increasingly rare), we associate the same coding. Thus, 54 values (16 per row), are computed. Figure 12 shows the H-HMM structure where each row presents a node.

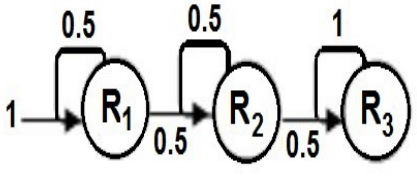


Figure 12: Structure of H-HMM (initialization step).

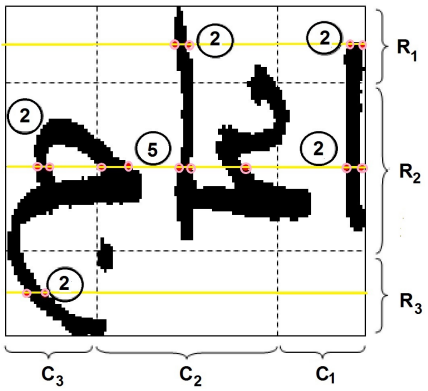


Figure 13: Number of white/black and black/white pixel transitions of the word "الحاج" (*Alhaj*).

3.2 Vertical HMM (V-HMM)

Here some global structural features (ascenders, descenders and diacritic points) are extracted considering their positions in the word. Dividing word into three columns from right to left, as shown in Figure 11, serves to look if extracted 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 as a sequence of structural features gathered from each column. Next, we will explain how to extract ascenders (also called stems), descenders (also called legs) and diacritic points based on their position according the central band. We will also clarify how to distinguish between different shapes of ascenders and descenders.

The central band is delimited by horizontal projection after baseline location which generally corresponds to the major accumulation of black pixels in a line (see Figure 14). Notice that baseline is quite tricky to locate in Arabic word because of letter extensions or horizontal ligatures. The obtained elementary building strips respectively contain ascending, central and descending components in the upper band, the central band and the lower band.

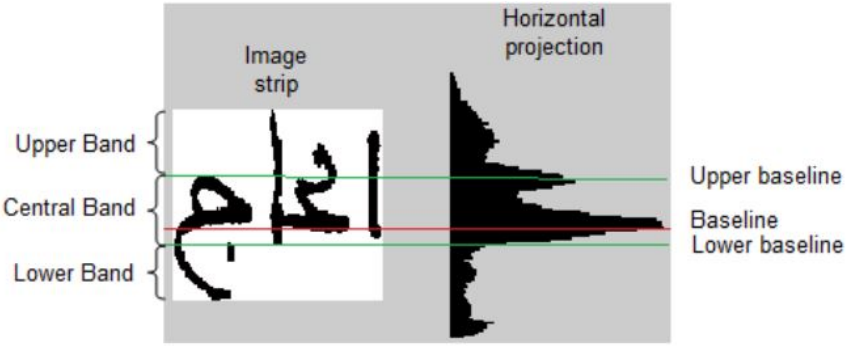


Figure 14: Central band and base line extracted from the word image "الحاج" (*Ahaj*).

Diacritic points may occur in the upper and/or the lower bands of words, at the beginning, in the middle and/or at the end of words. 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). The number of diacritic points depends on the aspect ratio (height/width) of their connected components because two or three diacritic points can be attached and then considered as only diacritic point (see Figure 15).

Figure 15: Diacritic point extraction from the word image "التُركي" (*Altorki*).

Ascenders and descenders are respectively located in the upper and lower bands of words. Ascenders can be of two types: "Stem-Alif" and "Stem-Kef" (See Figure 16) while descenders can be classified as "Leg-Noun", "Leg-Raa", "Leg-Haa" (See Figure 17). 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 which divides the image in the middle. It is clear that "Stem-Alif" has higher aspect ratio and density than "Stem-Kef" and a lower number of pixel transitions as explained in Figure 16. Leg classification is based on aspect ratio and density of their connected compounds, but also on the number and position of contact points with the lower line of the central band. As Figure 17 shows, in "Leg-Noun" there are two contact points with the lower line of the central band whereas in either "Leg-Raa" or "Leg-Haa" there is only one contact point. To distinguish between "Leg-Raa" and "Leg-Haa", we should 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 of "Leg-Haa". More details about structural features extraction and how are robust it their extraction are given in a previous work [1] and [2].

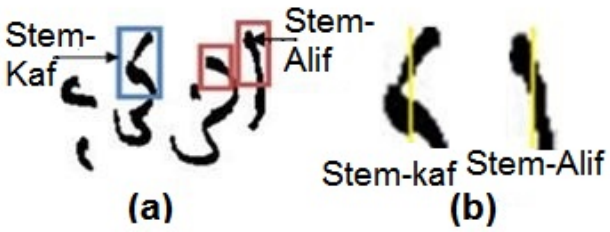


Figure 16: Stem extraction from the word image "الزكري" (Alzekri).

Figure 17: Leg extraction from the words "قريش" (Krich), "ونيس" (Ounis) and "الرضَاع" (*Al Radhah*).

In total, we extracted 27 structural features (9 features per column) as shown in Table 2.

Table 2: Set of extracted structural features

C_3	C_2	C_1
Stem-Alif at the end (SAE) Stem-Kef at the end (SKE)	Stem-Alif in the middle (SAM) Stem-Kef in the middle (SKM)	Stem-Alif in the beginning (SAB) Stem-Kef in the beginning (SKB)
1 Diacritic point to top at the end (1PUE) 2 Diacritic points up at the end (2PUE)	1 Diacritic point up in the middle (1PUM) 2 Diacritic points up in the middle (2PUM)	1 Diacritic point up in the beginning (1PUB) 2 Diacritic points up in the beginning (2PUB)
1 Diacritic point down at the end (1PDE) 2 Diacritic points down at the end (2PDE)	1 Diacritic point down in the middle (2PDM) 2 Diacritic points down in the middle(2PDM)	1 Diacritic point down in the beginning (2PDB) 2 Diacritic points down in the beginning (2PDB)
Leg-Raa at the end (LRE) Leg-Noun at the end (LRE) Leg-Haa at the end (LHE)	Leg-Raa in the middle (LRM) Leg-Noun in the middle (LRM) Leg-Haa in the middle (LHM)	Leg-Raa in the beginning (LRB) Leg-Noun in the beginning (LNB) Leg-Haa in the beginning (LHB)

The extracted features from the word image "الحاج" (*Alhaj*) are as shown in Figure 18.

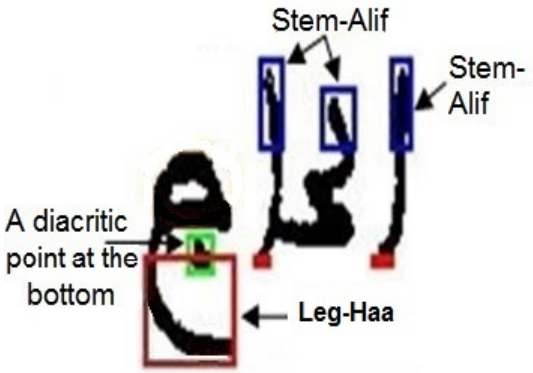


Figure 18: Feature extraction from the word "الحاج" (*Alhaj*).

Figure 19 shows the structure of V-HMM where each column represents a node.

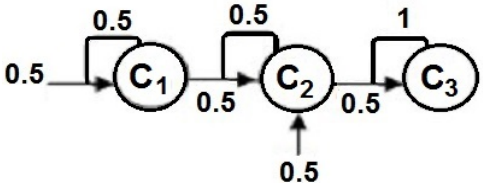
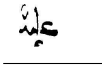
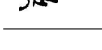
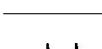


Figure 19: V-HMM structure (initialization step).

Due to the high variability in unconstrained handwritten script words, obtaining these features is a difficult task. To evaluate structural features extraction results, we compute the The edit-distance which is a string metric for measuring the amount of difference between two sequences. This distance is defined as the minimum number of edits needed to transform one sequence into the other, with the allowable edit operations being insertion (case of feature extracted in superfluous), deletion (case of not extracted feature), or substitution (case of incorrectly extracted feature) of a single feature. In the following table, E, T and D respectively refer to sequences of extracted features and truth description features and the edit distance.

Table 3: Examples of features extraction results

Word	E	T	D
	SAM, SAB, SAB, LM, LRE, 1PDE, 1PDE, 1PDM, 1PDB	SAM, SAB, SAB, LM, LRE, 1PDE, 1PDE, 1PDM, 1PDB	0
	LNB, 1PDM, 1PDB	LNB, 1PDM, 1PDB	0
	SAM, 1PUB, LHE	SAM, 1PUB, LHE	0
	SAM, 2PUE, 2PUE, 1PDM	SAM, 2PUE, 2PUE, 2PDM	1
	SAM, LRM, 2PUB, 2PUB	SAM, LRM, 2PUB, 2PUB	0
	SAM, LRE	SAM, LRE	0
	SKM, SKB, LRM, LRB, 2PUM, 2PDM, 1PUB	LRM, LRB, 2PUM, 2PDM, 1PUB	2
	SAB, LM, LNM, LRM, 1PUB	SAB, LM, LNM, LRM, 1PUB	0
	SAM, LBC, LRM, 2PDE, 2PUE, 2PDM	SAM, LBC, LRM, 2PDE, 2PUE, 2PDM	0
	SAM, SAB, LNE, 2PDM, 1PDM, 1PDB	SAM, SAB, LNE, 2PDM, 1PDM, 1PDB	0
	SAE, LNE, 1PUE, 1PDM, 2PUB	SAE, LNE, 1PUE, 1PDM, 2PUB	0
	SAM, LHE, 2PUB	SAM, LHE, 2PUB	0
	LRE, LRM, LRM	LRE, LRM, LRM	0

Notice that for the name "علية", although only one diacritic point was extracted, instead of two, but it was located in the right position. For the name "قريش", wanted features are correctly extracted but wrongly stems were detected in superfluous. Most of features extraction errors can be attributed to the writing style and the poor quality of some data samples. Table 3 displays evaluation results of structural feature extraction. Table 4 displays evaluation results of structural feature extraction using two databases: personal names, extracted from registers of the national archive of Tunisia, and Tunisian city names from the public database IFN-ENIT [23].

Table 4: Structural Feature Extraction Accuracy.

Data test	Recall	Precision	F-measure
Personal names (116)	0.89	0.89	0.89
IFN-ENIT (534)	0.78	0.82	0.80

It is worth to note that both of V-HMM and H-HMM are discrete, one-dimensional and left-to-right HMM without state skipping to model Arabic word. We selected this basic topology because it has been effectively used in handwriting recognition.

3.3 Vertical and Horizontal HMM (VH-HMM)

Here, we conceive an independent two-dimensional HMM which consider features extracted from both columns and lines of word image. We also used the zoning method to compute pixel density distributions which is a simple statistical feature at pixel level extracted from word image rows. For that, we divided each block into 16 cells and we considered cells having a density pixel over than 25 pixels. Thus, 144 features are extracted (48 per row) using pixel density distribution.

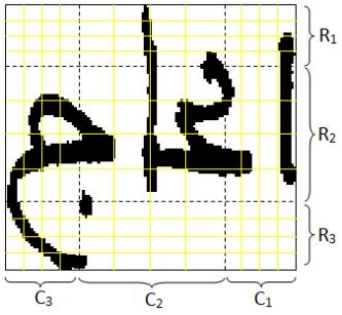


Figure 20: Pixel density distribution of the word image "الحاج" (*Alhaj*) .

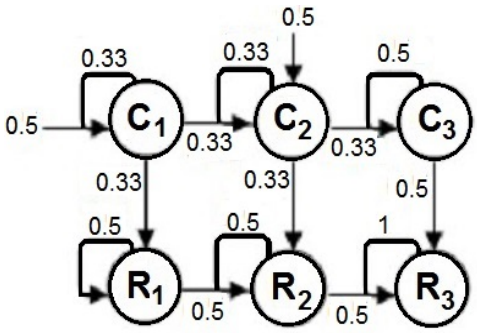


Figure 21: Vertical-Horizontal HMM structure (initialization step).

3.4 DBN

So far, we considered a single (vertical or horizontal) discrete HMM to model word images. To conceive a DBN, we thought about coupling the V-HMM and H-HMM by adding direct links between nodes in the graph to represent dependencies between state variables. Adding links requires learning graph structure from data

or fixing a DBN structure for all word images. In our case, we set a structure as illustrated in Figure 22: a DBN model based on coupling two hidden Markov chains in which we add a causal link (representing the time dependencies) from one time slice to another. The structure is completely known a priori and all variable are observable from the data. Coupled V-HMM and H-HMM are divided into 3 times slices. The DBN, in each time slice, contains a number of random variables representing observations and hidden states of the process. The dependencies between V-HMM and H-HMM modelling both vertically and horizontally data flow 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 our DBN is represented by a set of hidden state variables. Here is represented by a set of two hidden state variables.

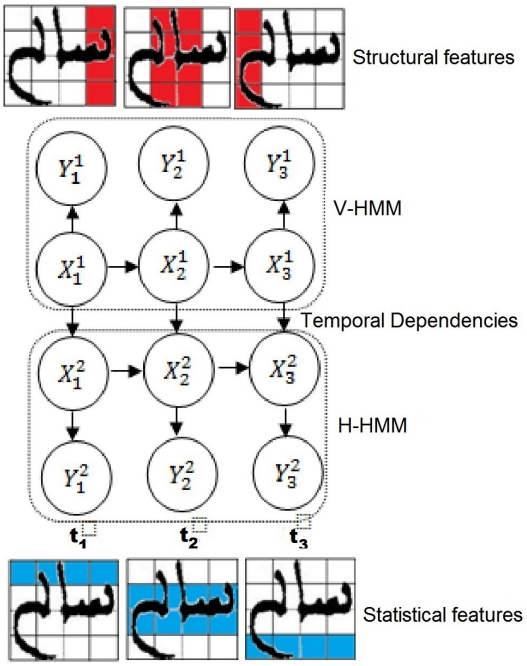


Figure 22: DBN model based on state-coupled V-HMM and H-HMM.

In the above model, the DBN has three observation streams. Let use indices $i = 1, 2, 3$ to denote the three streams. The variables X_i and Y_i denote the respective hidden state and observation attributes in stream $i = 1, 2, 3$. 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.

As shown in Figure 22, each image of handwritten word is described by two sequences of feature vectors that represent the input to the DBN: the first feature vector sequence models the flow of observations on the columns C_1, C_2 and C_3 whereas the second feature vector sequence models the flow of observations on the rows R_1, R_2 and R_3. Features, extracted by scanning vertically and horizontally image of the word, are respectively modelled by discrete V-HMM and H-HMM. To enhance the influence of the vertical stream, the edges are directed from the vertical stream to the horizontal one (see Figure 22). Experimentally, it has been proved that the vertical HMM (V-HMM) is more reliable than the horizontal one (H-HMM) as shown in Table 5. In order to have a complete specification of our DBN, we need 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 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 and the word is assigned to the class that gives the maximum likelihood.

4 Experimental Result Analysis

Table 5 gives results obtained with HMMs and DBN carried on words (350 samples of 7 Arab personal names) extracted from ancient Arabic manuscripts (see Figure 23). These experiments showed how robust is the proposed models since these old manuscripts are generally noisy and high degraded documents.

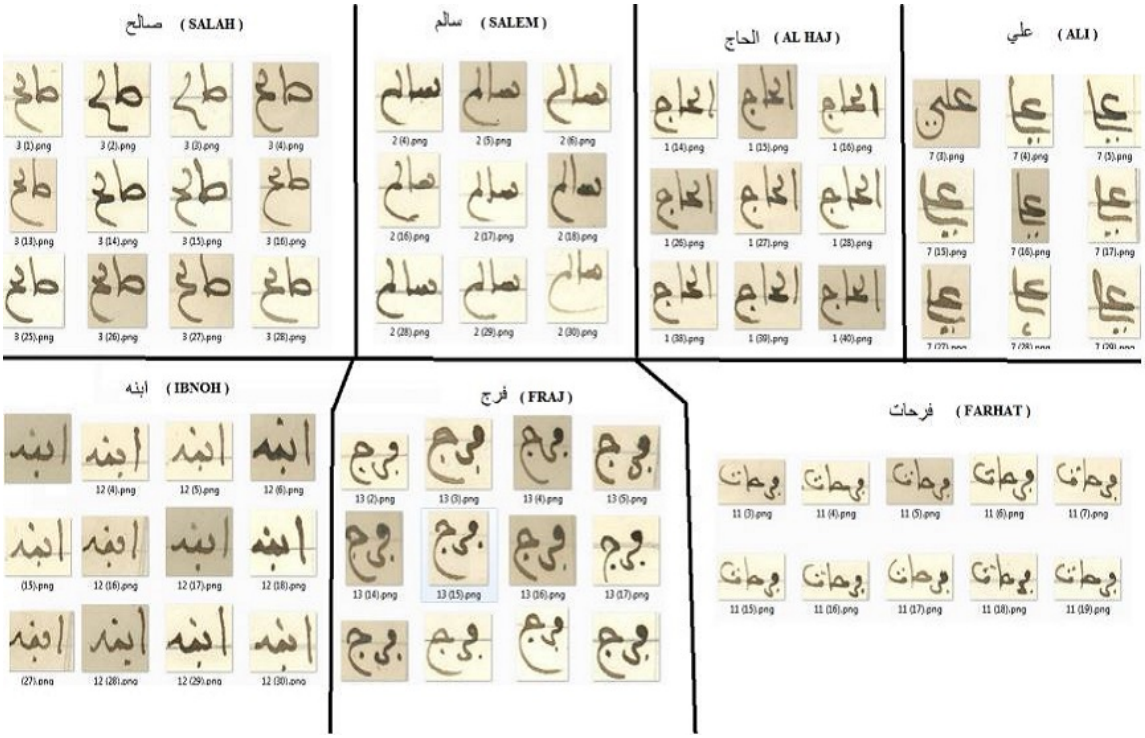


Figure 23: Examples of handled Arabic handwritten personal names.

Table 5: Recognition rates with HMM and DBN (Personal names).

Word	**H-HMM(%)**	**V-HMM(%)**	**DBN(%)**	**VH-HMM (zoning)(%)**	**VH-HMM(%)**
الحَاج (*Alhaj*)	94.64	96.43	92.28	94.64	96.43
سَالم (*Salem*)	71.15	92.31	98.1	96.15	96.15
صَالح(*Salah*)	71.87	40.62	84.37	81.25	81.25
علي (*Ali*)	83.82	91.18	95.59	95.59	92.65
فرحَات (*Farhat*)	96.15	100	70.37	96.15	100
إبنه (*Ibnoh*)	92.96	97.18	94.37	94.37	95.77
فرج (*Fraj*)	82.22	93.33	97.77	97.14	100
Average	84.69	87.29	90.4	93.61	94.61

The obtained results shows that VH-HMM has higher recognition rates than the remaining models. The average rate of recognition for DBN attempts 90.4% whereas VH-HMM recognition rate is 94.61%. The results show rates increased for the arabic words "فرج", "الحَاج", "سَالم", "فرحَات" and "إبنه". The rates are relatively acceptable for the word "صَالح" and "علي". Table 6 shows obtained results with the proposed DBN tested on words extracted from the IFN/ENIT database.

Table 6: Recognition rates of DBN and VH-HMM (IFN/ENIT).

Word	**Samples**	**Training**	**Test**	**DBN (%)**	**VH-HMM(%)**
الرضَاع (*Al Radhah*)	374	249	125	62.4	92
الخلِيج (*Al khalij*)	345	230	115	78.26	86.09
نقة (*Nakka*)	343	229	114	92.98	95.61
شعَال (*Chaâl*)	338	225	113	70.80	92.92
شعَال (*Chaâl*)	338	225	113	84.95	92.92
شمَاخ (*Chammakh*)	322	215	107	95.33	98.13
زنوش (*Zanouch*)	319	213	106	80.19	95.28
الدُخَانية (*Al Dokhania*)	318	212	106	84.90	94.34
الفَايض (*Al Fayadh*)	312	208	104	76.92	92.31
اكودة (*Akouda*)	298	199	99	83.84	88.88
سبعة ابَار (*Sabaat Abar*)	293	195	98	92.86	98.98
سيدي ابرَاهيم الزهَار (*Sidi Ibrahim Al Zahhar*)	290	193	97	97.94	94.84
المرنَاقية 20 مَارس (*Al Mornaguia 20 Mars*)	274	183	91	96.70	94.50
شتَاوة صحرَاوي (*Chtawah Sahraoui*)	245	163	82	93.90	90.24
الفكة (*Al Fakka*)	171	114	57	59.65	85.96
اوتيك (*Utique*)	138	92	46	65.22	89.13
الفحص (*Al Fahs*)	138	92	46	69.56	86.96
الشرَايع (*Al Charaaya*)	134	89	45	66.67	86.67
حي الَانطلَاقة (*Hay Al Intilaka*)	120	80	40	72.5	95
شوَاط (*Chawat*)	109	73	36	91.67	88.88
حي التضَامن (*Hay Attadhamon*)	60	40	20	85	90
Average	5279	3519	1760	81.06	91.89

We selected, for each class of words, those who are written differently to consider all possible variations in writing styles and thus to have the most representative set as possible of words. For some classes (like "الرضَاع": 347 samples, "الخلِيج": 345 samples) the writing style variations are numerous. That is why they include more samples. For other classes (such as "حي التضَامن": 60 samples) where the variations of the writing styles are few, the used number of samples is relatively small.

Our experiments show that independent VH-HMM and coupled V-HMM and H-HMM, represented as a DBN, cope better than basic V-HMM and H-HMM. This is because coupled architectures are able to predict missing information and may provide at least one uncorrupted stream within time slices. Experiments also show that DBN performs much worse than VH-HMM, although HMM in general is regarded as much simplified version of DBN. In fact, 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 VH-HMM works more effectively than DBN.

The superior performance of VH-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. But it would require large and full covariance matrices in order to take into account dependencies between the vertical and horizontal streams.

Table 7 shows results obtained by some related works. It should be noted that since the methods involved use different protocols and different subsets of the IFN/ENIT dataset, it is not possible to make a fair comparison. Actually, we only focused on some specific works which are closely related to our objective: PGM-based system for Arabic handwritten word recognition. So, we just considered works based on DBN or planar HMM which is a 2D-HMM and not works based on 1D-HMMs.

No details about the selected words are given in [11] and [19]. Only the number of samples is indicated. But in [17], Figure 24 specifies the set of used words.

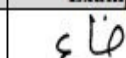
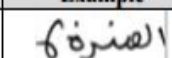
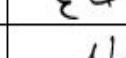
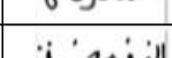
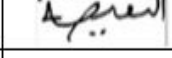
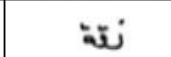
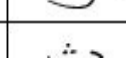
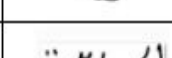
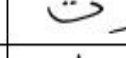
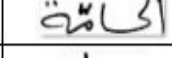
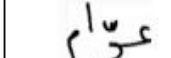
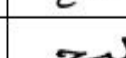
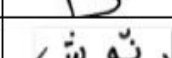
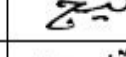
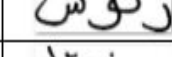
Class	City name	Images Example	Class	City name	Images Example
C1	الرضاع		C10	المنزه 6	
C2	شعال		C11	النفيضة	
C3	نحال		C12	نقة	
C4	مارث		C13	الحامة	
C5	شماخ		C14	عوام	
C6	الخليج		C15	رنوش	
C7	الرقاب		C16	بوزقام	
C8	الفايض		C17	خزندار	
C9	سيدي إبراهيم الزهار		C18	المنصورة	

Figure 24: Selected words from IFN-ENIT used in [17].

Table 7: Overview of some related work results.

System	Corpus	Recognition Rate (%)
[11]	IFN/ENIT	HMM=82%, DBN=66%
[19]	25 words, 2347 samples	Planar HMM=88.7%
[17]	18 words, 3600 samples	FAN=82.56%, TAN=80%, Naïve BN=73%, DBN=83.7%
Our system	21 words, 5279 samples	DBN=81.06%, VH-HMM=91.89%

Note that we proposed, as done by [11] a data fusion scheme which couples two data streams, image columns

and image rows into a single DBN classifier. More precisely, we constructed a state coupled model by adding the directed edges between the hidden state nodes of both vertical and horizontal HMMs. But, it differs from the proposed one in [11] where only pixel features are used for word representation. In fact, we proposed to use not only statistical features but also to combine them with structural ones. As it can be seen, the combined use of different types of features provides better results.

From experiments, conducted using a subset from IFN/ENIT benchmark database, the recognition rates achieved by the proposed VH-HMM and DBN, in comparison to some related works, are among the best for the same task. Overall, we achieved very promising results.

5 Conclusion

The goal of this work is to conceive and carry out an automatic off-line recognition system of Arabic handwritten words based on PGMs. We build a variety of models, including traditional Markovian independent models(H-HMM, V-HMM, VH-HMM). We also we coupled data streams into single DBN classifier. This coupling is performed through a DBN architecture which combines two basic HMM: the V-HMM whose outputs are structural features extracted from word image columns and H-HMM whose outputs are statistical features extracted from word image rows. Both structural features (ascenders, descenders, diacritic points) and statistical features at pixel level such as pixel density distributions and local pixel configurations are extracted scanning the word binary images from left to right and top to bottom. In this model, the interactions are performed through states leading to efficient model in terms of model complexity.

In sum, this study investigated various PGM architectures, different types of features for word representation and their contributions to provide the best recognition performance for handwritten Arabic word recognition. Thus, Independent HMMs also a coupling HMMs architecture, represented as a single DBN, are proposed, evaluated and compared in order to select the best architecture for the task of handwritten Arabic word recognition. Our objective is to provide a useful comparison of PGM-based architectures for the task of word recognition.
- The observations for the proposed models are the word image rows and columns. This results in finer representations of word images and in improvement of the basic HMM framework.
- We used statistical (pixel transition number, pixel density) and structural(number, type and position in the word of stems or ascenders, legs or descenders, diacritics) features separately in H-HMM and V-HMM than we combined the two different features in VH-HMM to benefit from their advantages. Note that reported works used either structural or statistical features not both of them at the same time. We believe that the use of multiple sources of information represents one of the advisable orientations in pattern recognition.
- For word representation, we tried to respect Arabic word morphology. That is to extract structural features according to the word central band (stems at the top, legs below and diacritic points on both sides of the central band) and to their position in the word (at the beginning, in the middle or at the end). Statistical features are used to support structural features. For example, if one stem is detected at the beginning of the word, the number of black to white or white to black pixel transitions should be 2.
- No word segmentation is required for feature extraction. This is especially crucial in case of cursive script recognition because it is not obvious that the handwritten Arabic word will be correctly segmented.

We showed proposed models interest in off-line Arabic handwritten word recognition. First, experiments have been conducted using words extracted from ancient Arabic manuscripts, conserved in the national archives of Tunisia, to demonstrate how robust the proposed models are. Then, others experiments were conducted using a subset of the IFN/ENIT standard database. Experimental results and quantitative evaluations showed that a 2D-HMM outperforms DBN in terms of higher recognition rate and lower complexity. In fact, the highest rate was achieved when using an independent VH-HMM: 94.61% (ancient manuscripts) and 91.89% (IFN/ENIT). In the future, we plan to look for the best representation of the Arabic word image, respecting its morphology, and for other PGM-based architectures which provide the best recognition performance.

References

[1] A. Kacem, N. Aouïti, A. Belaïd, "Structural Features Extraction for Handwritten Arabic Personal Names Recognition", *Proc. of ICFHR*, Italy, 268-273, 2012.

[2] A. Kacem, A. Khémiri, N. Aouïti, N. Aouadi, "Système, à base de MMC, pour la reconnaissance de noms propres manuscrits Arabes", *Proc. of CIDE*, Tunisia, 2012.

[3] D. L. Kelly, C. L. Smith, "Bayesian inference in probabilistic risk assessment: The current state of the art", *Reliability Engineering and System Safety*, 94(2):628-643, 2009.

[4] S. Russell, P. Norvig, *Artificial Intelligence, A Modern Approach*, 2nd edn. Prentice Hall, Englewood Cliffs, 2003.

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

[6] V. Mihajlovic, M. Petkovic, *Dynamic Bayesian Networks, A State of the Art*, CTIT technical reports series, TR-CTIT-34, 2001.

[7] K. Daoudi, D. Fohr, C. Antoine, "Dynamic Bayesian networks for multi-band automatic speech recognition", *Computer Speech and Language*, 17(2,3):263-285, 2003.

[8] L. Likforman-Sulem, M. Sigelle, "Recognition of degraded characters using dynamic Bayesian networks", *Pattern Recognition*, 41(10):3092-3103, 2008.

[9] O. Pauplin, J. Jiang, "A Dynamic Bayesian Network Based Structural Learning towards Automated Handwritten Digit Recognition", *Proc. of HAIS the 5th international conference on Hybrid Artificial Intelligence Systems*, 120-127,2010.

[10] J. H. AlKhateeb, "Offline Handwritten Arabic Digit Recognition Using Dynamic Bayesian Network", *Proc. of ICCIT*, 176-180, 2012.

[11] J. H. AlKhateeb, O. Pauplin, J. Ren, J. Jiang, "Performance of hidden Markov model and dynamic Bayesian network classifiers on handwritten Arabic word recognition", *knowledge-based systems*, 24(5):680-688, July 2011.

[12] S. Srihari,, "Machine Learning and Probabilistic Graphical Models Course", www.cedar.buffalo.edu/ srihari/CSE574/, 2010.

[13] P. Burrow, *Arabic Handwriting Recognition*", PhD thesis, University of Edinburgh, 2004.

[14] M. El Yacoubi, R. Gilloux, C. Sabourin, Y. Suen, "An HMM-based approach for off-line unconstrained handwritten word modeling and recognition", *IEEE Transactions on PAMI*, 21(8):752-760, 1999.

[15] V. Margner, H. Abed, M. Pechwitz, "Offline Handwritten Arabic Word Recognition Using HMM: a Character Based Approach without Explicit Segmentation", *Proc. of CIFED*, Fribourg, Swiss, 2006.

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

[17] M. A. Mahjoub, N. Ghanmy, K. Jayech, I. Miled, "Multiple models of Bayesian networks applied to offline recognition of Arabic handwritten city names", *Computer Vision and Pattern Recognition*, 2013.

[18] F. Biadsy, R. Saabni, J. El-Sana, , "Segmentation-Free Online Arabic Handwriting Recognition", *IJPRAI*, 1009-1033, 2011.

[19] S. Masmoudi Touj, N. Essoukri Ben Amara, H. Amiri, "Arabic Handwritten Words Recognition Based on a Planar Hidden Markov Model", *IAJIT*, 2(4):318-325, 2005.

[20] S. Espana-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, J. and F. Zamora-Martinez, "Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models", *Pattern Analysis and Machine Intelligence*, 33(4):767779, 2011.

[21] V. Margner, H. Abed, M. Pechwitz, "Offline Handwritten Arabic Word Recognition Using HMM: a Character Based Approach without Explicit Segmentation", *Proc. of CIFED*, Fribourg, Swiss, 2006.

[22] L.Rothacker, S.Vajda, G. A. Fink, "Bag-of-Features Representations for Offline Handwriting Recognition Applied to Arabic Script", *Proc. of ICFHR*, 149-154, 2012.

[23] Website of IFN-ENIT database, http://www.ifnenit.com/, September 2006.