Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Performance of hidden Markov model and dynamic Bayesian network classifiers on handwritten Arabic word recognition

Jawad H. AlKhateeb [a,*], Olivier Pauplin [b], Jinchang Ren [c], Jianmin Jiang [b]

[a] Faculty of Science and Information Technology, Al-Zaytoona University of Jordan, Amman, Jordan
[b] School of Informatics, University of Bradford, Bradford BD7 1DP, United Kingdom
[c] Centre for excellence in Signal and Image Processing, Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, United Kingdom

ARTICLE INFO

Article history:
Received 6 May 2010
Received in revised form 26 January 2011
Accepted 9 February 2011
Available online 15 February 2011

Keywords:
Off-line handwritten recognition
Hidden Markov model (HMM)
Dynamic Bayesian network (DBN)
Performance evaluation
IFN/ENIT database

ABSTRACT

This paper presents a comparative study of two machine learning techniques for recognizing handwritten Arabic words, where hidden Markov models (HMMs) and dynamic Bayesian networks (DBNs) were evaluated. The work proposed is divided into three stages, namely preprocessing, feature extraction and classification. Preprocessing includes baseline estimation and normalization as well as segmentation. In the second stage, features are extracted from each of the normalized words, where a set of new features for handwritten Arabic words is proposed, based on a sliding window approach moving across the mirrored word image. The third stage is for classification and recognition, where machine learning is applied using HMMs and DBNs. In order to validate the techniques, extensive experiments were conducted using the IFN/ENIT database which contains 32,492 Arabic words. Experimental results and quantitative evaluations showed that HMM outperforms DBN in terms of higher recognition rate and lower complexity.

1. Introduction

Handwriting recognition (HWR) is a mechanism for transforming the written text into a symbolic representation, which plays an essential role in many human–computer interaction applications including cheque verification, automatic mail sorting, office automation as well as natural human–computer interaction [1]. HWR for Latin languages has been conducted and significant achievements have been made. However, there has been less work in Arabic handwriting recognition. This is due to the complexity of the Arabic language and lack of public Arabic handwriting databases. In general, HWR can be categorized into two distinct types: online and off-line based systems. Recognition in online systems uses the dynamics of writing by following the pen movement. Recognition in off-line based systems is based solely on an image of the written text. Online recognition is easier because it can make use of the additional information not available to the off-line systems such as the strength and sequential order of the writing [2]. However, online recognition is not possible in many applications so in this paper, we focus on the off-line recognition of handwritten Arabic text.

The recognition of handwritten Arabic scripts can be divided into segmentation based or segmentation free approaches. The former segments words into characters or letters for recognition and can be regarded as an analytical approach. The latter, which can be regarded as a global approach, takes the whole word image for recognition and therefore needs no segmentation. Although the global approach makes the recognition process simpler, it requires a larger input vocabulary than analytical approach [3].

This paper focuses on the Arabic handwritten word recognition phase and introduces new methods for extracting features. Several experiments have been conducted using the IFN/ENIT benchmark database [4] and our algorithm showed the best recognition rate among the existing work reported using the same database. The remainder of this paper is structured as follows: Section 2 presents the literature review while Sections 3 and 4 describe the proposed method in terms of pre-processing and feature extraction; Section 5 describes the HMM classification process in details; experimental results are presented in Section 6. The paper ends with conclusions and suggestions for further work.

2. Literature review

Khorsheed and Clocksin [5] presented a technique for the word can be recognized as single unit which depends on a predefined lexicon. Using the skeleton of the word based on the Stentiford's algorithm [6], all segments were extracted for recognition into feature vector. The extracted the structural features from Arabic cursive text in three consecutive steps: segment extraction, loop extraction and segment transformation. A 8-dimensional feature

* Corresponding author.
E-mail addresses: jawadalkhatib@yahoo.com, jawad.kh@alzaytoonah.edu.jo (J.H. AlKhateeb), o.pauplin@bradford.ac.uk (O. Pauplin), Jinchang.ren@strath.ac.uk (J. Ren), j.jiang1@bradford.ac.uk (J. Jiang).

0950-7051/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2011.02.008

vector was created for each segment. Using vector quantization (VQ) [7], each vector was mapped to the nearest symbol in the codebook resulting in a sequence of observation which is fed into HMM. The Viterbi algorithm [8] is used to form a codebook of 76 symbols by apportioning the training samples into several classes. The technique was tested with a lexicon of 294 words acquired from a different text sources using the HMM. Recognition rates of up to 97% were achieved.

Khorsheed [9] presented another holistic recognition system for recognizing Arabic handwritten words. Pre-processing tasks performed included using the Zhang-Suen thinning algorithm [10] to generate the skeleton graph. Structural features for the handwritten script were extracted after skeletonization by decomposing the word skeleton into a sequence of links with an order similar to the word writing order. Using the line approximation [6], each line was broken into small line segments, which were transferred into a sequence of discrete symbols using VQ [7]. Then an HMM recognizer was applied with image skeletonization to the recognition of an old Arabic manuscript which can be found in [11]. One HMM was performed using 296 states on the 32 character models. Each model is left to right HMM with no restriction jump margin. The system was tested on 12,960 recognition tests associated with 405 character samples of a single font extracted from the single manuscript. The recognition rates achieved was 72% without spelling check and 87% with spelling check.

Pechwitz and Maergner [12] presented an off-line recognition system for the Arabic isolated handwritten words. They validated their system using the IFN/ENIT benchmark database [4] which consists of four sets (*a*, *b*, *c*, and *d*). They used a sliding window based on the image representation of the word image using pixel values as main features. The sliding window is shifted across the word image from right to left and generates the feature vector. The word image is a gray normalized image. The Karhunen Loeve Transformation (KLT) is performed in order to reduce the feature vector dimension. They used Semi Continuous HMMs (SCHMM) classifier for recognition. They used sets *a*, *b*, and *c* for training and set *d* for testing their system. The recognition rate achieved was 89%.

El Abed and Margner [13] presented an Arabic isolated handwritten word recognition based on HMM. They used the sliding window approach for extracting the pixel features. They used the skeleton direction based feature extraction technique where each word image was splitting into uniform vertical frames and each word image was split into five horizontal zones with equal height. The lengths of all lines in each zone frame were calculated in four directions to form a 20 dimensional feature. They used HMM for recognition and they validated their system using the IFN/ENIT benchmark database [4]. They used sets *a*, *b*, and *c* for training and set *d* for testing their system. The recognition rate achieved was 89.1% for top 1 and 96.4% for top 10.

In [14], a similar sliding window is used in such a window, the word image is divided into vertical overlapping frames with a constant width and variable heights. They validated their system using the IFN/ENIT benchmark database [4]. The sliding window is shifted from right to left and a feature vector is calculated for each frame. For each frame, 24 features are extracted using foreground pixel densities and concavity features. In addition there are 15 baseline independent features. They used HMMs classifier for recognition based on character modeling. Each character has a left right topology. Their HMM model had four states for each character model resulting 159 character models in total. They used sets *a*, *b*, and *c* for training and set *d* for testing their system. The recognition rate achieved was 75.41%.

Al-Hajj et al. [15] presented a two stage system for recognizing Arabic handwritten words. The first stage system was based on three HMM based classifiers which used pixel features in [14]. Each HMM classifier produces the best ten candidates (top 10) based on the likelihood. The second stage combined the three HMM classifiers by fusing the candidates provided by the HMM classifiers. They used three schemes for combining the classifiers: the sum rule, the majority vote rule, and the neural network based combining classifier. Different combinations were experimented using the IFN/ENIT benchmark database [4] and the recognition rate achieved was 90.96%.

Al-Hajj Mohamad et al. [16] presented an off-line recognition system for handwritten Arabic words of city names using the IFN/ENIT benchmark database [4] based on HMM. They used the sliding approach for extracting the features [14]. Their system relies on combining three homogeneous HMM classifiers in order to increase the system performance. They used the same three schemes for combining the classifiers in [15]. The recognition rate achieved was higher than 90% depends on the candidates. It is 90.26% for top 1, 94.71 for top 2, and 95.68% for top 3. It is important to mention that this system used sets *a*, *b*, and *c* for training and set *d* for testing.

Benouareth et al. [17,18] presented an off-line unconstrained handwritten Arabic word recognition based on semi-continuous hidden Markov models (SCHMMs) with explicit state duration. Statistical and structural features were utilized on the basis of the adopted segmentation in which implicit word segmentation is used to divide images into vertical frames of constant and variable width for feature extraction. Based on maxima and minima analysis of the vertical projection histogram, morphological complexity of the Arabic handwritten characters is further considered. They used SCHMM for recognition and they validated their system using the IFN/ENIT benchmark database [4]. They used sets *a*, *b*, and *c* for training and set *d* for testing their system. The recognition rate achieved with uniform segmentation was 81.02.1% for top 1 and 91.74% for top 10. The recognition rate achieved with non uniform segmentation was 83.791% for top 1 and 92.12% for top 10.

Likforman-Sulem and Sigelle [19,20] presented a new approach for off-line printed character recognition based on DBN. Their model consists of coupling two HMMs in various DBN architectures. The image rows and image columns of the coupled HMMs were used as the main observations. Their system has been evaluated using various DBN architectures and achieved a recognition rate of 98.3% with the vertical HMM, and 93.7% with the horizontal HMM. However, when testing degraded letters the recognition rate went down such as 93.8% with the vertical HMM and 88.1% with the horizontal HMM.

In this paper, we proposed an off-line recognition system for the handwritten Arabic cursive using HMM. We split the description of the system into three stages: preprocessing, feature extraction, and classification.

3. Preprocessing

The main aim of the preprocessing is to enhance the inputted signal and to represent it in a way which can be measured consistently for robust recognition. Here the preprocessing stage involves scanning the paper document, removing noise, binarizing the images, segmenting lines and words and estimating baselines. These steps are strongly dependent on the quality of the paper document. As samples of words provided in the IFN/ENIT database have been manually separated and binarized during the development stage [4], the only processes still needed are estimation of the baseline and normalization. Although not needed here, we have investigated how to generally segment words and this and our technique to estimate the baseline can be found in [21].

In an ideal handwriting model, the word has to be written in a horizontal way with both ascenders and descenders aligned along the vertical direction. These conditions in real data are rarely

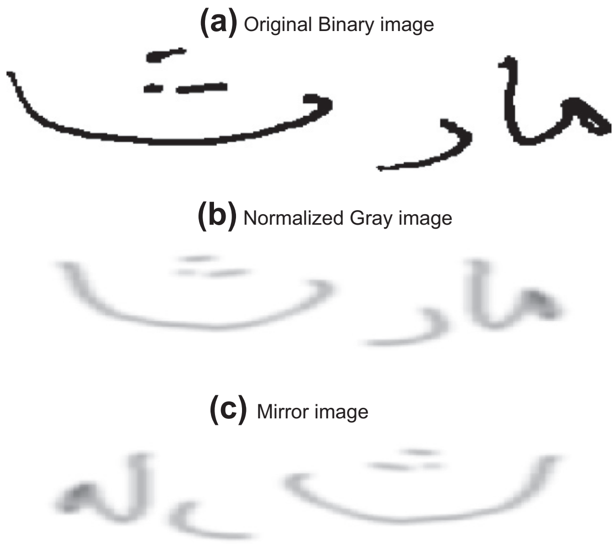

Fig. 1. An example of one typical data sample with original binary image (a), normalized gray image (b), and mirrored image (c).

found. So normalization is essential task to remove the variation in the handwritten images for consistent analysis and robust recognition. Among the many algorithms proposed for this purpose, the skeletonization technique is the most popular and likewise the normalization algorithm in [12] has been employed in this research. A sample image in binary format is shown in Fig. 1(a), along with its normalized counterpart in Fig. 1(b).

4. Feature extraction

The main goal for feature extraction is to remove the redundancy from the data and gain a more effective representation of the word image by a set of numerical characteristics. Feature extractions deals with extracting most of the essential information from image raw data. Depending on the problem to be solved and its data, different techniques can be applied to extract the features. Features are then mapped into a classifier in order to separate the input words into classes since the features have to be invariant to the variations of the members of each class. Based on [22] features used in off-line recognition are classified into high level ones which are extracted from the whole word image, medium level ones extracted from the letters, and low level ones extracted from sub letters.

Features can be also classified into structural and statistical ones. Structural features describe the topological and geometrical characteristics of a pattern; these include strokes, endpoints, loops, dots and their position related to the baseline. While statistical features are derived from the statistical distribution of pixels and describing the characteristic measurements of a pattern; these include zoning, density distribution of pixels that counts the ones and zeros, moments [23].

This paper implements the HMMs to recognize an unknown Arabic handwritten word. Ideally, this implies that the feature vectors extracted from the Arabic handwritten word is computed as a function of independent variable similar to the feature extracted in speech recognition [24]. Many researchers used the sliding windows/frames technique for extracting the feature vectors in off-line Arabic text recognition from right to left based on the Arabic writing direction [24–26]. In this paper, the sliding window technique used in speech recognition has been applied [25]. In order to speed both training and testing process, each image has been reversed using the mirror tool as shown in Fig. 1(c). It is worth mention that all the word images were normalized to have a height 45 pixels. Starting from the first pixel of the word, a sliding window is applied to the mirror word image to estimate the number of black pixels. In this phase the feature vectors for each word mirror image is performed by applying a horizontal sliding window having the same height of the word image, three pixels in width and one pixel overlap. The word mirror image is divided into fifteen horizontal uniform frames; the sliding window is shifted across the word mirror image from left to right as shown in Fig. 2, and the feature vector is computed for each window strip. Here, each window is divided into fifteen uniform areas. Each sliding window has 30 features and the first fifteen features (F_1–F_{15}) are estimated by averaging the pixels in each region, i.e.

$$F_i = (\text{Avg of pixels in the } i\text{th vertical area}) | i \in [1, 15] \tag{1}$$

The 16th feature F_{16} is the average of all the first 15 features as follows:

$$F_{16} = \text{Avg}\left(\sum_{i=1}^{15} F_i\right) \tag{2}$$

Then, 14 additional features (F_{17}–F_{30}) can be determined as follows:

$$F_{i+16} = \text{Avg}\{(F_i + F_{i+1}) | i \in [1, 14]\} \tag{3}$$

The features extracted by the sliding window can be summarized as follows:

```
Algorithm FEATEXTWINDOWSLIDE
for k = 1 to number of images
Img_in = Read the normalized image
Img_in = resize the Img_in into $45\times270$
Mirror_image = flip_left_right (Img_in)
Divide the Mirror_image into 15 horizontal frame (width = 3 pixels)
Apply Sliding window width = 3 and overlap = 1 from right to left
  featX(1:15,:) = featX(1:15,:)/18;
  featX(16,:) = featX(16,:)/270;
  featX(17:30,:) = featX(17:30,:)/36;
end
```

5. Classifiers

5.1. Hidden Markov models

There are several techniques for classifying the text; among these techniques is the hidden Markov models (HMMs) which is used for recognizing character, words, and lines. HMMs are widely used in the field of text recognition [25,27]. The HMMs are statistical models which originally used for speech recognition effectively. Due to the success of the HMMs in speech recognition and due to the similarities between the recognition of speech and cursive handwriting, HMMs were extended for online and off-line handwriting recognition [3,28].

In off-line recognition systems based on HMMs, the main concept is to transform the word image into a sequence of observations. Several researchers in speech recognition computed the feature vector as a function of independent variable from the speech signal with respect to time by dividing it into frames to simulate the HMMs using sliding windows/frames [24–27,29]. This technique was used in off-line text recognition where the feature vector is computed as a function of independent variable where the horizontal position along the text line is the independent

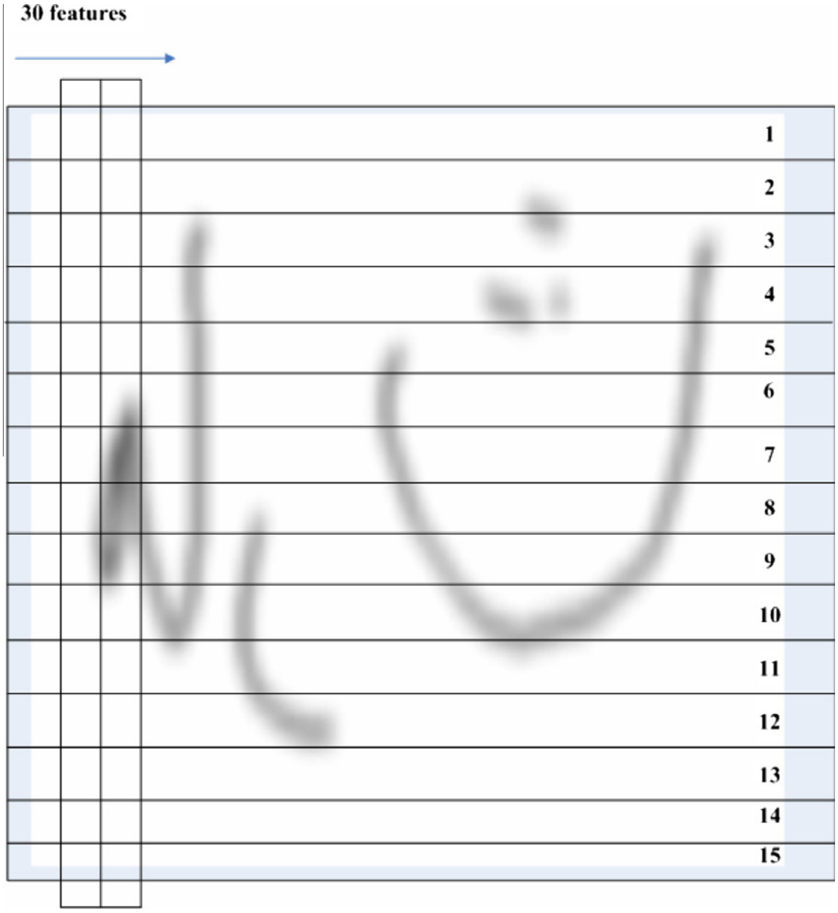

Fig. 2. Regions used for feature extraction and sliding window.

variable. In this paper, a sliding window is moving across the normalized word image to compute the features based on the pixels of the whole Arabic word.

The hidden Markov model 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. In order to develop a word recognition system based on the HMMs, the following procedures must be completed: (i) choose the number of states and observation, (ii) choose the HMM topology, (iii) select the training and the samples, (iv) train the system using the training data, and (v) test the system using the testing data. In this paper, we use the HMM classifier which was implemented on the HMM Toolkit (HTK) for speech recognition [30]. In the literature, many different model topologies were proposed using HMMs. In this paper, a left to right Bakis topology is implemented for the handwriting Arabic word recognition. Fig. 3 shows the case of seven states HMM allowing the transition to the same state, next state, and to the following states only. Each state has three different paths the transitions to the same state, next state, and to the state after the next state [31]. The sequence of state transition in training and testing the model depends on the feature observation of the Arabic word.

Generally, HMMs are denoted by λ and it is defined by three sets of parameters $\lambda = (\pi, A, B)$where A, B, and π represent the following parameters.

5.1.1. Matrix of transition probabilities (A)

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

$$\mathbf{A} = \{a_{ij} | a_{ij} = P(S_t = j | S_{t-1} = i)\} \tag{5}$$

$$a_{mn} = P(S_n | S_m); \quad m, n = 1, 2 \tag{6}$$

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.

5.1.2. Matrix of emission probabilities (B)

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix} \tag{7}$$

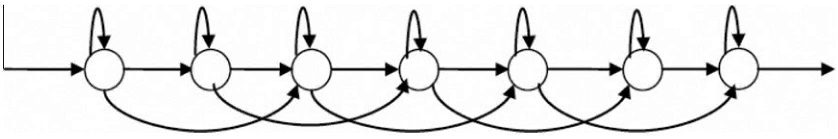

Fig. 3. A 7-state hidden Markov model (HMM).

$$\mathbf{B} = \{b_j(o_k)|b_j(o_k) = P(O_t = o_k|S_t = j)\} \tag{8}$$

$$b_{np} = b_n(p) = P(O_p|S_n); \quad n = 1, 2; \; p = 1, 2, 3 \tag{9}$$

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.

5.1.3. Initial states probabilities (π)

$$\pi = \begin{bmatrix} \pi_1 \\ \pi_2 \end{bmatrix} \tag{10}$$

$$\pi = \{\pi_i|\pi_i = P(S_1 = i)\} \tag{11}$$

$$\pi_m = P(S_m); \quad m = 1, 2 \tag{12}$$

Due to the several advantages of the HMMs, HMMs have been used in recognizing the Arabic text by many researchers. Among these advantages,

(1) There is no need for segmenting the Arabic text.
(2) HMMs are capable to tolerate the writing variation due to its resistant to noise.
(3) The HMM tools are available freely [25], where 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:

1. *The evaluation problem:* Calculating the probability that a model $\lambda = (\pi, A, B)$ created a given sequence of observations.
2. *The decoding problem:* Finding the most likely sequence of hidden states, in a given model $\lambda = (\pi, A, B)$, that is created by a given sequence of observations.
3. *The learning problem:* Estimating the model parameters $\lambda = (\pi, A, B)$ so that they best fit a given training sequences of observations.

More details about these problems are discussed in [8,31]. In addition, a detailed tutorial on the use of HMMs in Speech Recognition is provided in [31]. HMMs have been used for recognizing Arabic handwritten words [1,12], off-line Arabic handwritten digits [26,27,32] and Arabic characters [26]. In this research, we use the HMM classifier which was implemented on the HMM Toolkit (HTK) for speech recognition [30].

In the training phase, the model needs to be optimized using the training data. This is done by applying an iterative optimization method. The Baum–Welch algorithm iterative optimization method is applied to maximize the observation sequence probability $P(O|\lambda)$ of the chosen model $\lambda = (\pi, A, B)$ The Baum–Welch algorithm is a variant of the Expectation Maximization (EM) algorithm used to optimize an HMM model parameters π, A, and B based on the training data. Here, all the model parameters are re-estimated in order to improve the model quality.

The Baum–Welch algorithm is used to adjust the model parameters $\lambda = (\pi, A, B)$ to best fit the observed data. If we have a training dataset of L observation sequences $V = V_1V_2\cdots V_L$ and a known values for the number of hidden states (N) and the number of possible observations (M), then we aim to maximize the term $P(V|\lambda)$.

The set of hidden states is $S = \{S_1, S_2, \ldots, S_N\}$ with the sequence $Q = q_1q_2\cdots q_t$ representing a sequence of hidden states up to time t. In addition, an observed sequence from the set of possible observations $\{O_1, O_2, \ldots, O_M\}$ can be represented by $O = o_1o_2\cdots o_T$ which is a sequence of T observations.

According to [31], the following variables need to be defined:

$$\alpha_t(m) = P(o_1 o_1 \cdots o_t, q_t = S_m|\lambda) \tag{13}$$

which is the joint probability of the partial observation sequence up to time t and that the hidden state at time t is S_m given λ.

$$\beta_t(m) + P(o_{t+1} o_{t+1} \cdots o_T|q_t = S_m, \lambda) \tag{14}$$

which is the probability of the partial observation sequence from time $t + 1$ till T given λ and that the hidden state at time t is S_m

$$\begin{aligned}\xi_t(m, t) &= P(q_t = S_m, q_{t+1} = S_n|o_1 o_2 \cdots o_T, \lambda) \\ &= \frac{P(q_t = S_m, q_{t+1} = S_n|o_1 o_2 \cdots o_T)}{P(o_1 o_2 \cdots o_T, \lambda)} \\ &\quad + \frac{\alpha_t(m) a_{mn} b_n(O_{t+1}) \beta_{t+1}(n)}{\sum_{m=1}^{N} \sum_{n=1}^{N} \alpha_t(m) a_{mn} b_n(O_{t+1}) \beta_{t+1}(n)}\end{aligned} \tag{15}$$

which is the probability that the hidden state at time t is S_m and at time $t + 1$ is S_n given the observation sequence and λ

$$\begin{aligned}\gamma_t(m) &= P(q_t = S_m|o_1 o_1 \cdots o_T, \lambda) = \frac{P(q_t = S_m, o_1 o_2 \cdots o_T)}{P(o_1 o_2 \cdots o_T|\lambda})\\ &= \frac{\alpha(m)\beta_t(m)}{\sum_{m=1}^{N} \alpha_t(m)\beta_t(m)} = \sum_{n=1}^{N} \xi_t(m, n)\end{aligned} \tag{16}$$

which is the probability that the hidden state at time t is S_m given the observation sequence and λ.

As explained in [33], the Baum–Welch algorithm and the iterative Expectation Maximization (EM) algorithm are identical (have the same solution) for the current problem. Hence, the adjustment process for the parameters $\lambda = (A, B, \pi)$ is started as follows:

1. Initialize the parameters $\lambda = (A, B, \pi)$ randomly: a_{mn} is initialized to $1/N$, b_{mp} is initialized to $1/M$, and π_m is initialized to $1/N$.
2. From the Eqs. (13)–(16), calculate the parameters $\alpha_t(m)$, $\beta_t(m)$, $\xi_t(m,n)$ and $\gamma_t(m)$.
3. Calculate the new parameters of the model $\lambda^* = (A^*, B^*, \pi^*)$ according to the values calculated in step 2 as follows:

$$a^*_{mn} = \frac{\sum_{t=1}^{T} \xi_t(m, n)}{\sum_{t=1}^{T} \gamma_t(m)}, \quad b^*_n(p) = \frac{\sum^{T}_{\substack{t=1 \\ o_t = O_p}} \gamma_t(n)}{\sum_{t=1}^{T} \gamma_t(n)}, \quad \pi^*_m = \gamma_1(m) \tag{17}$$

4. Calculate $P(V|\lambda^*)$. While the probability $P(V|\lambda^*)$ is increasing repeat steps 2 and 3.

After the model parameters converge to some values, these parameters will be describing a model that best fits the training observation sequences.

In the testing phase, the modified Viterbi algorithm is used for recognition. Given the HMM parameter as $\lambda = (\pi, A, B)$ and the observation sequence $O = \{o_1, o_2, \ldots, o_N\}$ was fed into the HTK. The HTK models the observation (feature vector) with a mixture of Gaussian, and it uses the Viterbi algorithm in recognition phase which searches for the highest model probability of a word given the input feature vector $P(O|\lambda)$ as

$$Q = \arg\max P(O|\lambda) \tag{18}$$

5.2. Dynamic Bayesian networks

Before discussing dynamic Bayesian networks (DBNs) the basic foundation of Bayesian networks (BN) is outlined below. BNs are directed probabilistic graphical models. The random variables are represented by nodes, and the conditional dependences among the variables are represented by the arcs between the nodes [34].

BNs are graphical structures that represent the probabilistic relationships among large number of variables. Formally, a Bayesian network for a set of variables $X = \{X_1, X_2, \ldots, X_n\}$ is a pair $B = (G, \theta)$ [35]. The first parameter, G, is the BN structure, i.e. a directed acyclic graph (DAG) whose nodes correspond to the variables $X_i \in X$ and whose edges present their conditional dependencies. For example, if there is an edge from node X_1 to node X_2, then X_1 is parent of X_2. Hence, the value of X_2 is conditionally dependent on the value of X_1. The second parameter (θ) indicates the set of parameters encoding the conditional probabilities of each node variable X_i given its parents $\text{Pa}(X_i)$. These distributions are represented either by conditional probability tables (CPT) or by conditional probability distributions (CPDs). The CPT is represented when a node and it parents represent discrete variables. In contrast, the CPDs are represented for Gaussian continuous variables [19].

In this paper, the application of DBNs is investigated to the handwritten Arabic word recognition. To our knowledge, this is the first time that the DBN is created to carry out Arabic recognition. The coupled HMMs architectures to be represented as a single DBN [19].

Several coupled HMMs architectures can be constructed by adding directed edges between the two streams within the same time slice [19]. In order to enhance the influence of the vertical stream, the edges are directed from the vertical stream to the horizontal one. Experimentally, it has been proved that the vertical HMM is more reliable than the horizontal one [19]. Due to the fact that both streams are synchronized at each time slice, it is required that both observation sequences in the proposed coupled HMMs architectures have the same length. Therefore, all the normalized word images are resized to be 45 × 270.

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 value X_t^1. The observation dependences are expressed by the dependences between the horizontal and vertical states.

Fig. 4 shows three main coupled architectures; the state coupled model (ST_CPL), the general coupled model (GNL_CPL), and the auto regressive coupled model (AR_CPL). Those models were suggested by Likforman-Sulem and Sigelle [19]. The (ST_CPL) is obtained by adding the directed edges between the hidden state nodes of both vertical and horizontal HMMs as shown in Fig. 4(a). The GNL_CPL 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 as shown in Fig. 4(b). The AR_CPL is obtained by coupling both vertical and horizontal streams as shown in Fig. 4(c). More details about these three models are found in [19].

In this research, the AR_CPL model is chosen to be used since it is superior and it has achieved the highest recognition rate compared to other coupled models [19].

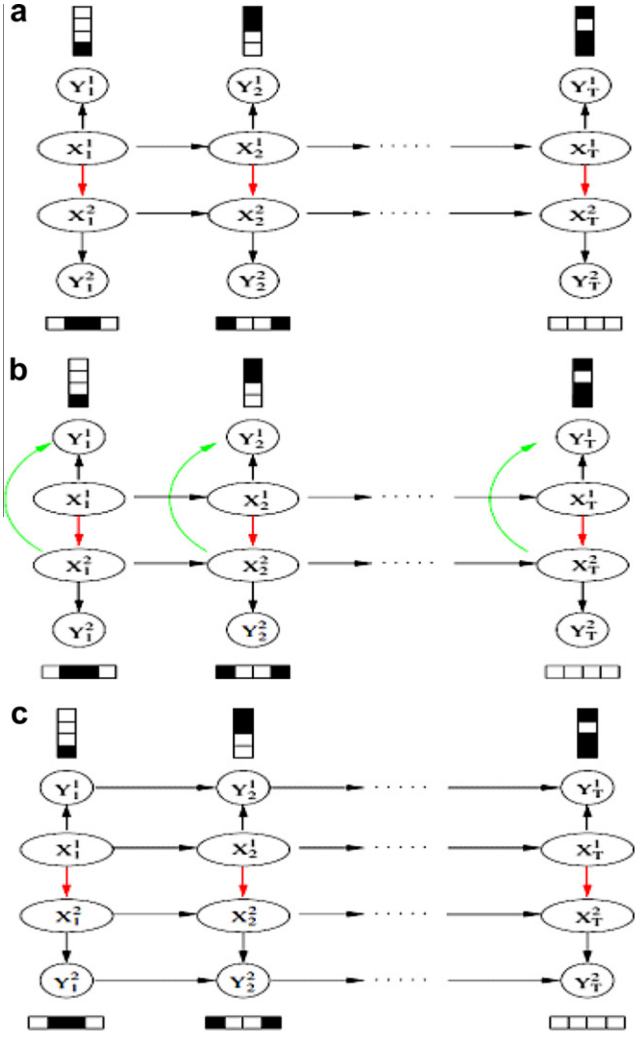

Fig. 4. Coupled architectures, each representing a single DBN.

6. Experimental results

Any recognition system needs a large database to train and test the system. Real data from banks or the post code are confidential and inaccessible for non commercial research. Although some work was conducted in Arabic handwritten words, but generally they had small databases of their own or the presented results on databases which were unavailable to the public. Consequently, there was no benchmark to compare the results obtained by researches. The work for Arabic script recognition has started more than three decades ago. There was no standard database till 2002 when the IFN/ENIT database (www.ifnenit.com) became available free for non commercial research [4]. This database is very important in this context as it has been used as a standard test database in such a context [4]. In total more than 1000 different people were selected as writers to put their names. In addition, each writer was asked to fill one or more than one form with handwritten pre-selected names of Tunisian town/villages with the corresponding postcode. All the forms were scanned with 300 dpi and converted to binary images. The images are divided into five sets so that researches can use some of them for training or testing, respectively.

In order to evaluate the performance of our recognition system, several experiments are conducted on the IFN/ENIT database [4] which contains 32,492 Arabic words handwritten by more than 1000 different writers and divided into five distinct sets *a, b, c, d,* and *e*. In our experiments, we use cross validation to verify the performance of our neural classifier. Each time 80% of the samples in the database (sets *A, B, C,* and *D*) are used for training and the remaining 20% (set *E*) for testing. The HMM classifier was trained and then tested.

6.1. HMM experiments

In off-line recognition systems based on HMMs, the main concept is to transform the word image into a sequence of observations. In speech recognition, several researchers computed the feature vector as a function of independent variable from the speech signal with respect to time by dividing it into frames to simulate the HMMs using sliding windows/frames [24–27,29]. This technique was used in off-line text recognition where the feature vector was computed as a function of the horizontal position along the text line as the independent variable. In this HMM research, different techniques have been used to extract the features of the Arabic word as a whole rather than a sliding window, which computes the features based on the DCT coefficients or the mean values of the overlapping blocks of the whole Arabic word.

In this paper, a left to right HMM was used for the Arabic handwritten word. Each word is represented by its feature vector, and each word requires a number of observations for training and testing the HMM. In quantizing the data, experiments were conducted using four codebook size parameter values: 8, 16, 32, and 64. Fig. 5 shows the result for the different codebook size values. This suggests that an increase in codebook size yields to a better recognition rate, but it increases the training and testing times. Several numbers of states were tested and the best performance was found using 25 states to represent the word.

Different number of states could be assigned to different words, but in fact the same number of states was chosen for all words. It has been noted that the recognition rate improves as the number of states increases till the HTK reaches the maximum possible state for specific feature set. This makes the training data independent of the testing data, and hence avoids over fitting the classifier to test the data. To test the effectiveness of different features, two groups of features were mapped separately into the HMM classifier, and the results are reported in Fig. 6. Several experiments were carried out based on 80% cross validation using the HMM and the average recognition rate was recorded.

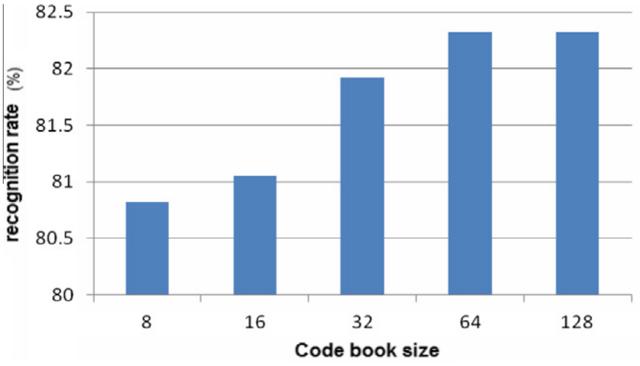

Fig. 5. Recognition rate vs. codebook size in HMM.

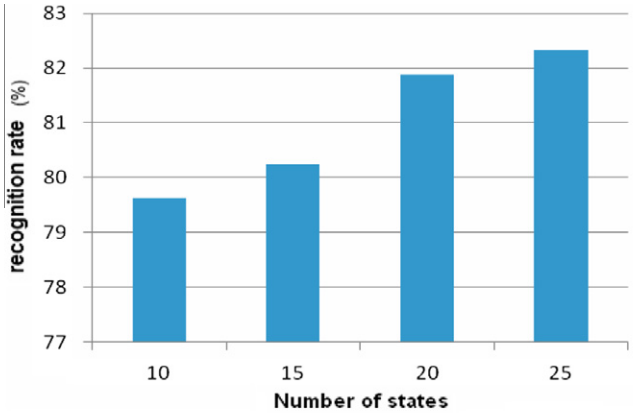

Fig. 6. Recognition rate vs. number of states in HMM.

6.2. DBN experiments

The DBN experiments have been conducted with the BayesNet Toolbox for Matlab [37], which provides source code to perform several operations on BNs and DBNs. The DBN parameters are learned using the EM algorithm. Again the first four set (a–d) of the IFN/ENIT database are used for training and the remaining one (set e) is used for testing.

Since DBN is working based on time slice, this is consistent with our features extracted from sliding windows. To test the effectiveness of the DBN, the pixel features extracted using the sliding window were mapped into the DBN. The DBN requires a balanced database for training and testing which is not the case in the IFN/ENIT database. To overcome this problem, the training and testing samples from the IFN/ENIT were randomly selected and used in our experiments. This process repeated five times, and an average recognition rate is then obtained. Table 1 summarizes the recognition results, where in general the results from the five tests are quite similar to each other. An average recognition rate of abut 66% is achieved.

6.3. Comparing with other systems

We compared our results with other systems tested in the same data set and conditions in ICDAR 2005 Arabic handwriting [36]. Fig. 7 summarizes this comparison. As seen from Fig. 7, our algorithm performs better in classifying the word image with an improved recognition rate of about 7%. In addition, it is found that HMM yields much better results than DBN in terms of recognition rate, despite of complicated modeling used in DBN.

It is worth noting that in the reference systems three of them have produced good results with recognition rate over 65%, i.e. the first, the fourth and the sixth. In fact, a two-tier neural classifier is employed in the first system, where modeling of subwords and lexicon aided classification are utilized. In the fourth system, baseline dependent features are extracted and fed into a 1-D HMM, and

Table 1
DBN recognition using AR_CPL model.

Exp.	Recognition rate (%)
1	65.46
2	67.86
3	65.32
4	66.27
5	67.86
Average	66.56

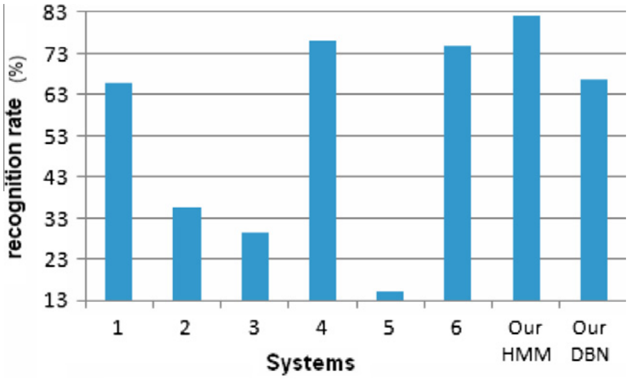

Fig. 7. Results of our HMM and DBN classifiers in comparison with six others in ICDAR 2005.

character model is learnt with support of language knowledge modeling for effective word recognition. In the sixth system [12], semi-continuous 1-D HMM is employed for word recognition, using pixel value features collected from a sliding window approach. To reduce writing variability, several normalization steps are adopted involving normalized height, length and baseline skew. Using HMM classifier, the fourth and the sixth produce higher recognition rate of about 75%, i.e. 10% more than that of the first one using neural classifier.

The effectiveness of our proposed techniques and their potential applications are analyzed as follows. First of all, it is the proposed features which contribute more to the good performance of the proposed system. When the similar HMM classifier is used, our system has gained over 7% in terms of recognition rate. Unlike pure pixel value features used in [12], average pixel values and their cross-frame combinations in our system are useful to overcome variability in writing towards more robust recognition. Such features and concepts can be also applied in other pattern recognition and machine learning tasks where signals contain continuous nature, such as speech recognition, gesture recognition and recognition of other handwritten languages.

Secondly, it is worth noting that it seems HMM is among the best classifiers as it produces much higher recognition rate than those using neural network and DBN. Again, this is due to the fact that the probabilistic state transfer in HMM has the intrinsic capacity in modeling connected nature of Arabic cursive script [12,14,24]. On the other hand, it is surprising to find that DBN performs much worse than HMM, although HMM is regarded as a much simplified version of DBN [19]. The reason behind probably can be explained as follows. Generally, DBN is 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 2-D. However, features extracted using average pixel values from sliding window schemes 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.

7. Conclusions

In this paper, the performance of both HMMs and DBNs classifiers were compared in terms of recognizing the handwritten Arabic words. Both classifiers are superior in classifying handwritten and printed scripts. The performance of both HMMs and DBNs classifiers on handwritten Arabic word recognition are reported. Actually, HMM is an excellent in classifying Arabic handwritten words. The system has been applied to the well-known IFN/ENIT database containing handwriting words written by different writers. We have found that the pixel density features are effective in our classifiers, and good results of recognition rate have been achieved. In addition, this system can be applied to other patterns for recognition with slightly adaptation. The result obtained in this research show that the best performances are always reached by the HMMs. This shows the superiority of the HMMs over all various classifiers used. Regarding the speed, HMMs were faster in training and testing time.

References

[1] S. Alma'adeed, C. Higgins, D. Elliman, Off-line recognition of handwritten Arabic words using multiple hidden Markov models, Knowledge-Based Systems 17 (2004) 75–79.

[2] A. Amin, Off-line Arabic character recognition: the state of the art, Pattern Recognition 31 (1998) 517–530.

[3] M.S. Khorsheed, Off-line Arabic character recognition – a review, Pattern Analysis and Applications 5 (2002) 31–45.

[4] M. Pechwitz, S.S. Maddouri, V. Märgner, N. Ellouze, H. Amiri, IFN/ENIT – database of Arabic handwritten words, in: Colloque International Francophone sur l'Ecrit et le Document (CIFED), 2002, pp. 127–136.

[5] M.S. Khorsheed, W.F. Clocksin, Structural features of cursive Arabic script, in: The 10th British Machine Vision Conference, The University of Nottingham, UK, 1999, pp. 422–431.

[6] J.R. Parker, Algorithms for Image Processing and Computer Vision, John Wiley and Sons, Inc., 1997.

[7] R.M. Gray, Vector quantization, IEEE Transactions on ASSP (1989) 4–29.

[8] L. Rabiner, B. Juang, An introduction to hidden Markov models, IEEE ASSP Magazine 3 (1986) 4–16 (see also IEEE Signal Processing Magazine).

[9] M.S. Khorsheed, Recognising handwritten Arabic manuscripts using a single hidden Markov model, Pattern Recognition Letters 24 (2003) 2235–2242.

[10] T.Y. Zhang, C.Y. Suen, A fast parallel algorithm for thinning digital patterns, Communications of the ACM 27 (1984) 236–239.

[11] M.S. Khorsheed, Automatic recognition of words in Arabic manuscripts, Computer Laboratory, vol. Ph.D., University of Cambridge, 2000, p. 220.

[12] M. Pechwitz, V. Maergner, HMM based approach for handwritten Arabic word recognition using the IFN/ENIT – database, in: Proceedings of the Seventh International Conference on Document Analysis and Recognition, 2003, pp. 890–894.

[13] H. El Abed, V. Margner, Comparison of different preprocessing and feature extraction methods for offline recognition of handwritten Arabic words, in: Ninth International Conference on Document Analysis and Recognition, 2007, ICDAR 2007, 2007, pp. 974–978.

[14] R. El-Hajj, L. Likforman-Sulem, C. Mokbel, Arabic handwriting recognition using baseline dependant features and hidden markov modeling, in: Eighth International Conference on Document Analysis and Recognition (ICDAR'05), vol. 2, 2005, pp. 893–897.

[15] R. Al-Hajj, C. Mokbel, L. Likforman-Sulem, Combination of HMM-based classifiers for the recognition of Arabic handwritten words, in: Ninth International Conference on Document Analysis and Recognition, 2007, ICDAR 2007, 2007, pp. 959–963.

[16] R. Al-Hajj Mohamad, L. Likforman-Sulem, C. Mokbel, Combining slanted-frame classifiers for improved HMM-based Arabic handwriting recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2009) 1165–1177.

[17] A. Benouareth, A. Ennaji, M. Sellami, HMMs with explicit state duration applied to handwritten Arabic word recognition, in: 18th International Conference on Pattern Recognition, ICPR 2006, 2006, pp. 897–900.

[18] A. Benouareth, A. Ennaji, M. Sellami, Semi-continuous HMMs with explicit state duration for unconstrained Arabic word modeling and recognition, Pattern Recognition Letters 29 (2008) 1742–1752.

[19] L. Likforman-Sulem, M. Sigelle, Recognition of degraded characters using dynamic Bayesian networks, Pattern Recognition 41 (2008) 3092–3103.

[20] L. Likforman-Sulem, M. Sigelle, Recognition of broken characters from historical printed books using dynamic Bayesian networks, in: Ninth International Conference on Document Analysis and Recognition, 2007, pp. 173–177.

[21] J.H. AlKhateeb, J. Ren, S.S. Ipson, J.A.J.J. Jiang, Knowledge-based baseline detection and optimal thresholding for words segmentation in efficient pre-processing of handwritten Arabic text, in: Fifth International Conference on Information Technology: New Generations, ITNG, 2008, pp. 1158–1159.

[22] S. Madhvanath, V. Govindaraju, The role of holistic paradigms in handwritten word recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001) 149–164.

[23] L.M. Lorigo, V. Govindaraju, Offline Arabic handwriting recognition: a survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (2006) 712–724.

[24] M.S. Khorsheed, Offline recognition of omnifont Arabic text using the HMM ToolKit (HTK), Pattern Recognition Letters 28 (2007) 1563–1571.

[25] A.A.-M. Husni, A.M. Sabri, S.Q. Rami, Recognition of off-line printed Arabic text using hidden Markov models, Signal Processing 88 (2008) 2902–2912.

[26] S. Mahmoud, Recognition of writer-independent off-line handwritten Arabic (Indian) numerals using hidden Markov models, Signal Processing 88 (2008) 844–857.

[27] I. Bazzi, R. Schwartz, J. Makhoul, An omnifont open-vocabulary OCR system for English and Arabic, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (1999) 495–504.

[28] S. Günter, H. Bunke, HMM-based handwritten word recognition: on the optimization of the number of states, training iterations and Gaussian components, Pattern Recognition 37 (2004) 2069–2079.

[29] M. Khorsheed, Mono-font cursive Arabic text recognition using speech recognition system, in: Structural Syntactic and Statistical Pattern Recognition, 2006, pp. 755–763.

[30] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, P. Woodland, The HTK Book, Cambridge University Engineering Department, 2001.

[31] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (1989) 257–286.

[32] S.M. Awaidaha, S.A. Mahmoud, A multiple feature/resolution scheme to Arabic (Indian) numerals recognition using hidden Markov models, Signal Processing 89 (2009) 1176–1184.

[33] A. Gellert, L. Vintan, Person movement prediction using hidden Markov models, Studies in Informatics and Control 15 (2006) 17.

[34] A. Dielmann, S. Renals, Automatic meeting segmentation using dynamic Bayesian networks, IEEE Transactions on Multimedia 9 (2007) 25–36.

[35] P.A.D. Castro, F.J. Von Zuben, Learning Bayesian networks to perform feature selection, in: International Joint Conference on Neural Networks, 2009, IJCNN 2009, 2009, pp. 467–473.

[36] V. Margner, M. Pechwitz, H.E. Abed, ICDAR 2005 Arabic handwriting recognition competition, in: Eighth International Conference on Document Analysis and Recognition, vol. 1, 2005, pp. 70–74.

[37] K.P. Murphy, BaysNet Toolbox for Matlab, 2007. Available on the web at: <http://people.cs.ubc.ca/~murphyk/Software/BNT/bnt.html>.