

Bayes Classification of Online Arabic Characters by Gibbs Modeling of Class Conditional Densities

Neila Mezghani, *Student Member, IEEE*, Amar Mitiche, *Member, IEEE*, and Mohamed Cheriet, *Senior Member, IEEE*

Abstract—This study investigates Bayes classification of online Arabic characters using histograms of tangent differences and Gibbs modeling of the class-conditional probability density functions. The parameters of these Gibbs density functions are estimated following the Zhu et al. constrained maximum entropy formalism, originally introduced for image and shape synthesis. We investigate two partition function estimation methods: one uses the training sample, and the other draws from a reference distribution. The efficiency of the corresponding Bayes decision methods, and of a combination of these, is shown in experiments using a database of 9,504 freely written samples by 22 writers. Comparisons to the nearest neighbor rule method and a Kohonen neural network method are provided.

Index Terms—Bayes classification, Gibbs density parameter estimation, histograms, online handwritten Arabic character recognition.

1 INTRODUCTION

PROBABILITY models are commonly used in pattern classification. In image classification, for instance, a parametric probability model can be used to obtain an optimal image domain partition and classify the contents of each region of the partition [21]. Probability models can also be used to synthesize image patterns by sampling and classify pattern shapes by the optimal Bayes decision [13], [56]. However, models must be representative and accurately estimated to be useful. This is particularly true in Bayes classification of pattern shapes [13], which this study applies to character recognition.

It is not unusual that pattern shape classification involves tens of pattern categories described by characteristic vectors of tens of entries. This is basically what makes estimation of the class conditional probability models particularly difficult, a serious impediment called the “curse of dimensionality” by Duda et al. [13]. However, the constrained maximum entropy Gibbs density parameters estimation, proposed by Zhu et al. [55], [56], affords a powerful means to learn these models and, therefore, to apply the optimal Bayes classification. The formalism was developed in the context of learning the universal statistics of natural images, that is, a generic prior model of these images, and was later applied to shape

synthesis [57]. This generic model duplicates the observed image statistics that serve to estimate it. The statistics used are empirical distributions (histograms) of filtered images. With statistics that capture the visual relevance of the images of interest, there is no difference between the estimated and true distribution as far as these statistics are concerned. The theory has shown remarkable results for texture synthesis, where complex textures could be synthesized from a few example patterns, sometimes from as little as a single example. The synthesis, by Gibbs sampling of the estimated probability density function, does not require the partition function. However, the partition functions are required for pattern classification, at least up a common scale factor. Estimation of partition functions is, in general, a difficult problem [43]. Here, however, it can be estimated up to a common scale factor, which is sufficient for pattern classification, from the learned Gibbs density parameters and the training sample or draws from a reference distribution.

The purpose of this study is to bring the formalism by Zhu et al. for Gibbs density parameters estimation [56] to bear on pattern classification, more specifically on Bayes classification of online Arabic characters. Characters are represented by empirical histograms of tangent differences measured at regularly sampled points on the character signal [34]. Although such a description is not, in general, sufficient to synthesize characters, it is adequate for classification. Using this representation and a training set of characters, the parameters of the class conditional Gibbs density functions are estimated [56]. These densities reproduce the empirical distributions observed in the training set, which means that they are indistinguishable from the underlying true densities as far as the chosen characteristics of representation are concerned. They also are as neutral as possible in the sense that they do not embody any other information on the shapes they describe.

The partition function is not required in image or pattern synthesis. Therefore, its estimation was not addressed in [56], [57]. Bayes classification, however, requires the class-conditional partition functions. We address the problem of

- N. Mezghani is with the Laboratoire de Recherche en Imagerie et Orthopédie (LIO), Centre de Recherche du CHUM, Pavillon J.A. de Sève, Hôpital Notre-Dame 1560, rue Sherbrooke Est, local Y-1615, Montréal (Québec), Canada, H2L 4M1. E-mail: neila.mezghani@etsmtl.ca.
- A. Mitiche is with the Institut National de la Recherche Scientifique (INRS-EMT), Place Bonaventure, 800 de la Gauchetière Ouest, suite 6900, Montréal (Québec), Canada, H5A 1K6. E-mail: mitiche@emt.inrs.ca.
- M. Cheriet is with the École de Technologie Supérieure (ETS), 1100, rue Notre-Dame Ouest, Montréal (Québec), Canada, H3C 1K3. E-mail: mohamed.cheriet@etsmtl.ca.

Manuscript received 10 July 2006; revised 7 Feb. 2007; accepted 25 June 2007; published online 3 Aug. 2007.

Recommended for acceptance by D. Lopresti.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0508-0706. Digital Object Identifier no. 10.1109/TPAMI.2007.70753.

estimating these partition functions and investigate two methods: One uses the training data set directly and the other draws from a reference distribution. The performance of the corresponding Bayes decision methods, and of a combination of these, are shown in experiments using a database of 9,504 freely written characters by 22 scriptors. Comparisons to a Kohonen neural network method and to the nearest neighbor (NN) rule classifier are provided. Before describing the representation and the algorithms, we review the state of the art in online Arabic character recognition.

2 ARABIC CHARACTER RECOGNITION

The document analysis literature contains an impressive number of studies on character recognition. We note several reviews on character recognition at large [12], [17], [18], [22], [25], [42], [51], [52] and more specifically on Arabic character recognition [1], [5], [6], [7], [23], [27]. The great majority of these studies deal with offline data from scanned paper documents and other such digital images. Significantly, fewer investigations [25], [26], [50], [51] pertain to online data because practicable applications of online character recognition are relatively recent. The basic concepts of pattern representation and classification remain the same for both input modalities, the main differences being the presence of a temporal dimension in online data and the real-time response, which online character recognition applications generally require.

Within the context of online handwritten character recognition, studies dealing with Arabic characters are scarce. However, advancements in telecommunications and the international diffusion of information technologies have opened up applications opportunities of online recognition of handwritten Arabic text. Applications include hand-held computers, digital notebooks, laptop screens (tablet PCs), and advanced mobile telephony.

Pattern representation has not been the main issue in most online Arabic character recognition studies, as these concentrated more on classification algorithms. A variety of character representations have been used, such as a decomposition into characteristic strokes [2], [30], global shape descriptors such as Fourier coefficients [29], [32], [33], and local geometric descriptors such as tangents [31], [32] and difference of tangents [34]. For other online script, the $x - y$ coordinate string of the input signal [46] has also been used, and there have been efforts to model pen-tip movements to extract time-dependent representation features such as curvilinear and angular velocities [40], [41]. All of these representations of shape describe reasonably well Arabic characters and have allowed focusing on the development of classification algorithms.

Most Arabic online character recognition methods implement algorithms of the conventional pattern classification paradigms. The operations underlying these algorithms include template matching [14], decision trees [2], [15], fuzzy logic reasoning [3], [4], [10], neural network mapping [3], [4], [31], [32], [33], [34], and hidden Markov modeling (HMM) [9], [28]. Combinations of different classifiers to improve recognition have also been investigated [3], [4], [31]. Decision trees provide a hierarchical classification by dividing the set of classes into subsets using features that characterize the subsets. In [2], [15], for instance, stroke matching follows a rule-based hierarchical division of the set of classes. It is

generally acknowledged that the division of data into subclasses is sensitive to the acquisition noise and distortions such as those usually present in multiwriters online data. This is also the case for template matching. Fuzzy logic borrows from rule-based reasoning and probability theory to translate a representation into classification rules. In general, it is quite difficult to produce discriminant classification rules. The success of HMM in speech recognition suggests that it can also serve well character recognition [28]. Learning class models require a large sample of representative data that is not available for online Arabic characters. Although preprocessing was done to minimize the training data, the use of discrete HMM and smoothing of symbols helped to reduce the training data required to estimate the models; still, too few samples are available for Arabic Characters. Neural networks, such as Kohonen memories, have been used because of their relatively short time of development and their good classification behavior [31], [32], [33], [34].

Although reasonable recognition rates have been reported for online Arabic characters, these remain lower than those for other scripts. Also, the samples used for both training and testing are quite small. This motivated us to collect a database of about 10,000 characters written without constraints by several scriptors. Although such a database is much smaller than the Latin character databases NIST (offline) and UNIPEN (online), it is reasonably sized for meaningful experiments with our method.

Arabic script is cursive. Therefore, characters are connected within a word/pseudoword. An important difficulty in Arabic character recognition is the segmentation of words/pseudowords into characters [8], [38], [44], [53], [54]. In this study, we do not address segmentation and assume isolated characters.

The remainder of this paper is organized as follows: Section 3 describes the representation. Section 4 summarizes the constrained maximum entropy formalism to estimate class-conditional Gibbs distribution parameters. Section 5 describes methods for partition function estimation. Section 6 contains experimental results and Section 7 a conclusion.

3 REPRESENTATION

A good representation is important because it contributes to high recognition performance. It uses measurements, called features, whose values are similar for patterns in the same category, and different for patterns in different categories [13]. The features of representation must also be invariant to relevant transformations such as translation, rotation, and scaling. In this study, we use a representation based on the empirical distribution (histograms) of features, referred to as feature *statistics* in [55], [56], [57]. As features, we use tangent differences at regularly sampled points on the characters signal [34]. Difference of tangents at points distant by α is a discrete approximation of curvature on the signal subsampled by α . The difference of tangents is expected to be decorrelated beyond some distance. As empirical marginal distributions, histograms of features preserve, in general, more information about shape than scalar functions of the features.

3.1 Features

Let $\Gamma(s)$ be a parametric representation of the curve of a character and consider N consecutive points $\{s_k\}_0^{N-1}$,

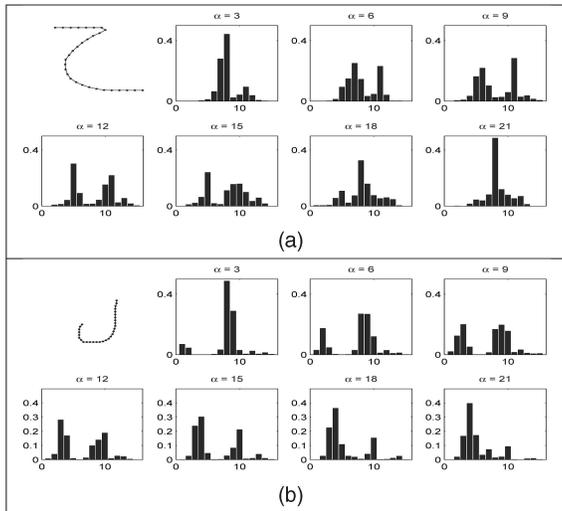


Fig. 1. Tangent differences H histogram for α multiple of 3.

with corresponding coordinates $\{(x_k, y_k)_0^{N-1}\}$. We represent a curve by

$$\Gamma = ((x_0, y_0), (x_1, y_1), \dots, (x_{N-1}, y_{N-1})), \quad \Gamma \in \Omega_\Gamma \subset \mathbb{R}^{2N},$$

where Ω_Γ is the space of character shapes satisfying the constraint $x_i \in [0, a]$ and $y_i \in [0, b]$, $a \times b$ being, in this application, the size of the acquisition tablet.

Let $\theta_k, k = 0, \dots, N-1$ be the tangent angles computed by $\theta_k = \arctan\left(\frac{y_{k+1} - y_k}{x_{k+1} - x_k}\right)$, where indices are modulo N . For $\alpha \in \mathbb{N}$, $1 \leq \alpha \leq N-1$, let $\phi^{(\alpha)}$ be the feature corresponding to the tangent angle differences defined by $\phi^{(\alpha)}(s_k) = \theta_{k+\alpha} - \theta_k$. Therefore, for each shape Γ , we have a set of features $\Phi = \{\phi^{(\alpha)}\}$, $\alpha = 1, 2, \dots, N$. Note that these features are invariant under translation and rotation.

3.2 Feature Statistics

Characters are represented by statistics of features. Such statistics are powerful in representing images, as demonstrated by the studies in [55] and [56] on texture and shape synthesis. The histograms are defined by

$$H^{(\alpha)}(\Gamma, z) = \frac{1}{N} \sum_{k=0}^{N-1} \delta(z - \phi^{(\alpha)}(s_k)), \quad (1)$$

where δ is the Dirac delta function with unit mass at zero, and $\delta(x) = 0$, for $x \neq 0$. $H^{(\alpha)}(\Gamma, z)$ is the number of points on the shape where the discretized feature is equal to z . In practice, histograms are discretized into m bins, as illustrated in Fig. 1. Therefore, for each feature $\phi^{(\alpha)}$, the histogram is an m -dimensional vector

$$H^{(\alpha)}(\Gamma) = (H_1^{(\alpha)}, H_2^{(\alpha)}, \dots, H_m^{(\alpha)}).$$

For a fixed number of bins, the histograms of tangent angle differences are invariant to character scaling. In addition, they are invariant to translation and rotation because the features are invariant to these transformations. Invariance to rotation is not as important as invariance to translation and scale. Also, although normalization with respect to rotation might be counterproductive for certain letters (*Dal* and *Nun*, for instance) written by particular scriptors, it helps for most of the characters.

4 ESTIMATION OF CLASS-CONDITIONAL GIBBS DENSITY PARAMETERS

The purpose of this section is to summarize the constrained maximum entropy formalism to estimate class-conditional Gibbs distribution parameters [55], [56], [57], originally used for texture and shape synthesis. The parameters of these distributions are estimated from samples of training characters, or *observed* characters. We will assume that the observed characters in a class i do not contribute information about the distribution of the statistics in classes $j \neq i$. Therefore, we treat each class separately and forgo class indices to simplify notation.

Let $\{\Gamma_i^{obs}, i = 1, \dots, M\}$ be a set of M observed characters of a class, that is, the training set of the class. Given features $\Phi^{obs} = \{\phi^{obs(\beta)}, \beta = 1, \dots, K\}$ and their corresponding statistics $\mathcal{H}^{obs} = \{H^{obs(\beta)}, \beta = 1, \dots, K\}$, we compute the statistics average histograms

$$\mu^{obs(\beta)} = \frac{1}{M} \sum_{i=1}^M H^{(\beta)}(\Gamma_i^{obs}) \quad \beta = 0, 1, \dots, K, \quad (2)$$

where β designates the feature index.

For large samples, the sample averages $\mu^{obs(\beta)}$, $\beta = 0, 1, \dots, K$, are good estimates of the expectations $E_f[H^{(\beta)}(\Gamma)]$, where E_f denotes the expectation with respect to the underlain true density $f(\Gamma)$. To approximate $f(\Gamma)$, a probability model $p(\Gamma)$ is constrained to reproduce the observed statistics, that is,

$$E_p[H^{(\beta)}(\Gamma)] = \mu^{obs(\beta)} \quad \beta = 1, 2, \dots, K. \quad (3)$$

Let Ω_p be the set of distributions that reproduce the observed statistics

$$\Omega_p = \left\{ p(\Gamma) \mid E_p[H^{(\beta)}(\Gamma)] = \mu^{obs(\beta)}, \beta = 1, 2, \dots, K \right\}.$$

Following the maximum entropy principle [19], we determine an estimate p^* of f such that [56]:

$$p^*(\Gamma) = \arg \max_p \left\{ - \int p(\Gamma) \log p(\Gamma) d\Gamma \right\} \quad (4)$$

subject to constraints

$$\int p(\Gamma) d\Gamma = 1, \quad (5)$$

$$E_p[H^{(\beta)}(\Gamma)] = \int H^{(\beta)}(\Gamma) p(\Gamma) d\Gamma = \mu^{obs(\beta)} \quad \beta = 1, 2, \dots, K. \quad (6)$$

Basically, we want to determine the estimate p^* so that it reproduces the observed statistics and is as neutral as possible in the sense that it does not embody any other information. The solution, by Lagrange multipliers, of this constrained optimization problem is a Gibbs density of the form [56]

$$p(\Gamma; \Lambda) = \frac{1}{Z} e^{-\sum_{\beta=1}^K \lambda^{(\beta)} \langle \lambda^{(\beta)}, H^{(\beta)}(\Gamma) \rangle}, \quad (7)$$

where $\langle \cdot, \cdot \rangle$ denotes inner product; Z is the partition function; $\Lambda = (\lambda^{(1)}, \lambda^{(2)}, \dots, \lambda^{(K)})$ are Lagrange multipliers, or *potential functions*. This density function depends only on

the potential functions, a maximum likelihood estimate of which can be computed by gradient descent [56]

$$\frac{d\lambda^{(\beta)}}{dt} = E_{p(\Gamma;\Lambda)}[H^{(\beta)}(\Gamma)] - \mu^{obs(\beta)}, \quad \beta = 1, 2, \dots, K. \quad (8)$$

The expected values $E_{p(\Gamma;\Lambda)}[H^{(\beta)}(\Gamma)]$ are estimated as follows [16], [47]: Synthesize samples Γ^{sym} from $p(\Gamma; \Lambda)$ and use the averages $\mu^{sym(\beta)}$ as estimates. Therefore, descent equation (8) becomes

$$\frac{d\lambda^{(\beta)}}{dt} = \mu^{sym(\beta)} - \mu^{obs(\beta)}, \quad \beta = 1, 2, \dots, K. \quad (9)$$

The corresponding algorithms are given in the appendices.

5 PARTITION FUNCTION ESTIMATION

The partition function is a normalization constant necessary to calculate the probability estimate $p^*(\Gamma; \Lambda)$. The partition function is not used in image synthesis [56]. However, in this case of pattern classification, it is necessary. In general, estimation of a partition function is a difficult problem [43]. In our case, however, the difficulty is lessened because Bayes pattern classification requires the class-conditional partition functions only up to a common scale factor. We have investigated two methods. One uses the training characters (the *direct* method) and the other uses patterns drawn from a reference distribution (the *indirect* method).

Let C_1, C_2, \dots, C_c be c character classes and $\Lambda_{C_j} = \{\lambda_{C_j}^{(1)}, \lambda_{C_j}^{(2)}, \dots, \lambda_{C_j}^{(K)}\}$ the corresponding potential functions. We have

$$p(\Gamma; \Lambda|C_j) = \frac{1}{Z_{C_j}} e^{-\sum_{\beta=1}^K \langle \lambda_{C_j}^{(\beta)}, H^{(\beta)}(\Gamma) \rangle}, \quad (10)$$

$$j = 1, 2, \dots, c \quad \beta = 0, 1, \dots, K,$$

where Z_{C_j} is the partition function for class C_j :

$$Z_{C_j} = \int e^{-\sum_{\beta=1}^K \langle \lambda_{C_j}^{(\beta)}, H^{(\beta)}(\Gamma) \rangle} d\Gamma. \quad (11)$$

5.1 Direct Method

Note that, for each class C_j , (11) is the volume under the function in the integrand. The direct method take as estimates of this volume the average of this function over the M training samples of class C_j times the measure of the set of nonzero probability shapes.

Since we need the partition function only up to a scale factor, we use the following estimate:

$$Z_{C_j} = \sum_{i=1}^M e^{-\sum_{\beta=1}^K \langle \lambda_{C_j}^{(\beta)}, H^{(\beta)}(\Gamma) \rangle} \quad (12)$$

assuming that the measure of the set of nonzero probability shapes is the same for all classes. In the case of a small number of training samples, this can be a crude estimate. However, we will see that the recognition error profile using this method is different from the one using the indirect method (described next), which gives us the opportunity to combine both methods.

5.2 Indirect Method

This general method uses samples from a reference density χ [11], [20]. The main issue is the choice of the reference distribution, a choice that may vary with the application. A good reference density is one that overlaps sufficiently the density for which we want to estimate the partition function. In our case, a good reference density overlaps sufficiently all the class-conditional densities. We chose the following reference density:

$$\chi(\Gamma) = \frac{1}{\tilde{Z}} \frac{1}{c} \sum_{i=1}^c e^{-\sum_{\beta=1}^K \langle \lambda_{C_i}^{(\beta)}, H^{(\beta)}(\Gamma) \rangle}. \quad (13)$$

This density can be seen as a weighed average density over classes. It is, a priori, a reasonable reference because, theoretically, it overlaps each of the densities it averages. We will estimate the class-conditional partition functions up to a division by \tilde{Z} , that is, Z_{C_j}/\tilde{Z} for all classes

$$Z_{C_j} = \int e^{-\sum_{\beta=1}^K \langle \lambda_{C_j}^{(\beta)}, H^{(\beta)}(\Gamma) \rangle} d\Gamma \quad (14)$$

and

$$\tilde{Z} = \frac{1}{c} \int \sum_{i=1}^c e^{-\sum_{\beta=1}^K \langle \lambda_{C_i}^{(\beta)}, H^{(\beta)}(\Gamma) \rangle} d\Gamma, \quad (15)$$

where c is the number of classes. Therefore,

$$\begin{aligned} \frac{Z_{C_j}}{\tilde{Z}} &= \frac{\int e^{-\sum_{\beta=1}^K \langle \lambda_{C_j}^{(\beta)}, H^{(\beta)}(\Gamma) \rangle} d\Gamma}{\frac{1}{c} \int \sum_{i=1}^c e^{-\sum_{\beta=1}^K \langle \lambda_{C_i}^{(\beta)}, H^{(\beta)}(\Gamma) \rangle} d\Gamma} \\ &= \int \frac{e^{-\sum_{\beta=1}^K \langle \lambda_{C_j}^{(\beta)}, H^{(\beta)}(\Gamma) \rangle}}{\frac{1}{c} \int \sum_{i=1}^c e^{-\sum_{\beta=1}^K \langle \lambda_{C_i}^{(\beta)}, H^{(\beta)}(\Gamma) \rangle} d\Gamma} d\Gamma \\ &= \int \frac{e^{-\sum_{\beta=1}^K \langle \lambda_{C_j}^{(\beta)}, H^{(\beta)}(\Gamma) \rangle}}{\frac{1}{c} \sum_{i=1}^c e^{-\sum_{\beta=1}^K \langle \lambda_{C_i}^{(\beta)}, H^{(\beta)}(\Gamma) \rangle}} \cdot \\ &\quad \frac{\frac{1}{c} \sum_{i=1}^c e^{-\sum_{\beta=1}^K \langle \lambda_{C_i}^{(\beta)}, H^{(\beta)}(\Gamma) \rangle}}{\frac{1}{c} \int \sum_{i=1}^c e^{-\sum_{\beta=1}^K \langle \lambda_{C_i}^{(\beta)}, H^{(\beta)}(\Gamma) \rangle} d\Gamma} d\Gamma \\ &= \int \frac{1}{\frac{1}{c} \sum_{i=1}^c e^{-\sum_{\beta=1}^K \langle \lambda_{C_i}^{(\beta)} - \lambda_{C_j}^{(\beta)}, H^{(\beta)}(\Gamma) \rangle}} \chi(\Gamma) d\Gamma. \end{aligned}$$

which yields

$$\frac{Z_{C_j}}{\tilde{Z}} = E_{\chi} \left[\frac{1}{\frac{1}{c} \sum_{i=1}^c e^{-\sum_{\beta=1}^K \langle \lambda_{C_i}^{(\beta)} - \lambda_{C_j}^{(\beta)}, H^{(\beta)}(\Gamma) \rangle}} \right], \quad (16)$$

where E_{χ} is the reference distribution expectation. For the purpose of classification, factor $1/c$ can be omitted from (13) and (16). Estimation of the partition functions by this indirect method consists basically of sampling the reference χ , which does not require the partition function (sampling is done as in [57]; see algorithm Table 8), and computing ensemble averages. Given the potential functions and a good reference density (which overlaps sufficiently all the class-conditional densities), the advantage of this indirect method is that a large number of draws from the reference distribution can be used to have a good estimate (up to a scale factor) of the partition functions.

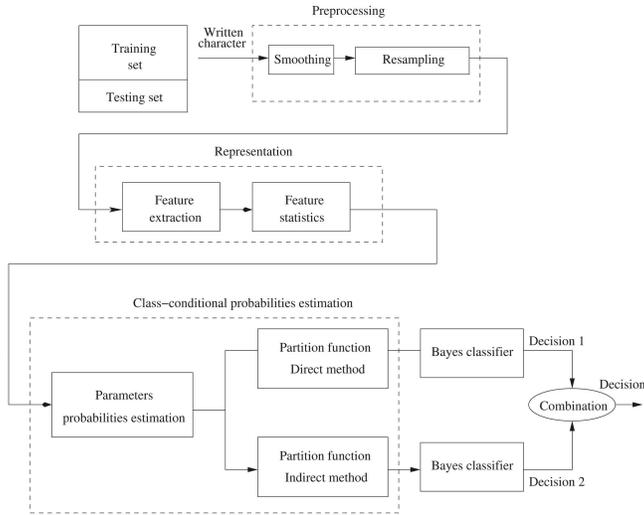


Fig. 2. Diagram block of the proposed system.

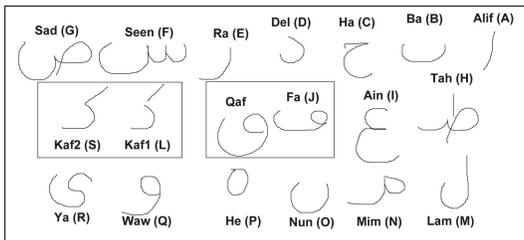


Fig. 3. The 18 shapes of Arabic isolated characters and their assigned labels.

6 EXPERIMENTAL RESULTS

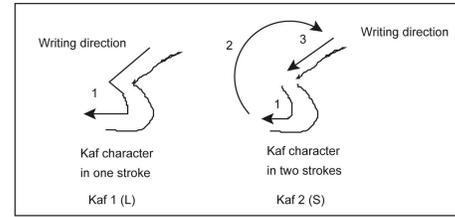
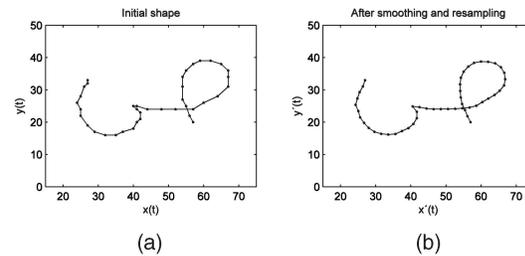
Fig. 2 shows the functional modules of the complete recognition system we implemented, as well as the flow of processing. The representation was described in Section 3 and class-conditional probability estimation in Sections 4 and 5. The next two sections describe the data used in our experiments (Section 6.1) and the preprocessing (Section 6.2). Results are detailed in Section 6.3. Comparisons are also provided (Section 6.4).

6.1 Database

Data collection was done using a digital Wacom Graphire tablet with a resolution accuracy of 23 points/cm and a sampling frequency of 100 points/seconds. The recorded data consisted exclusively of the coordinates sequence of each character written on the tablet.¹

Arabic has an alphabet of 28 isolated letters constructed from 18 distinct shapes (Fig. 3) augmented by diacritical points above or below. We consider classification of these 18 distinct shapes, rather than of the 28 letters. Note that diacritical points could possibly be used for recognition, in a postprocessing step, for instance. However, this study does not address the subject.

Because letters *Fa* and *Qaf* have the same shape except for their position with respect to the baseline, both letters are considered of the same class (Fig. 3). Finally, because letter *Kaf* can be written in one or two strokes and that the

Fig. 4. The *Kaf* character subclasses.Fig. 5. Samples of letter *Sad* and letter *Mim*.Fig. 6. (a) An acquired character *Sad* and (b) after smoothing and resampling (50 equidistant points).

direction of writing depends on the number of strokes (Fig. 4), we considered two classes for this letter (*Kaf1* and *Kaf2*). Thus, we have a total of 18 classes. Note that the *Kafs* in Fig. 4 are those that appear in pseudowords and not the Arabic isolated *Kaf*, which consists of two symbols, the character shape and a *Hamza*.

The database contains 528 characters of each letter from each of 22 writers for a total of 9,504 characters. This is by far the largest database of online Arabic characters we know of. Samples in other studies are simply too small for us to use and draw meaningful conclusions.

Writing of the characters was not constrained, leading to a wide variety of size and orientation. Also, the database contains both clearly written characters and roughly written ones. Because the acquired signal is a set of coordinates corresponding to the pen position on the tablet, the size of this set is writer speed dependent as illustrated by the samples of letters *Sad* and *Mim* in Fig. 5. In Section 6.2, we describe preprocessing operations to smooth the character signals and normalize them with respect to writer speed.

6.2 Preprocessing: Smoothing and Resampling

We smooth the online character signals to remove noise and resample to normalize with respect to writer speed (Fig. 6). We use a weighted 3-point averaging for smoothing, a simple scheme that is sufficient for this application:

$$\begin{cases} x_{si} = \frac{1}{4}x_{i-1} + \frac{1}{2}x_i + \frac{1}{4}x_{i+1}, \\ y_{si} = \frac{1}{4}y_{i-1} + \frac{1}{2}y_i + \frac{1}{4}y_{i+1}, \end{cases} \quad (17)$$

where x_i, y_i designate pen position on the tablet.

1. <http://webd.etsmtl.ca/zone2/recherche/labo/lio/fiche/neilamezghani/>.

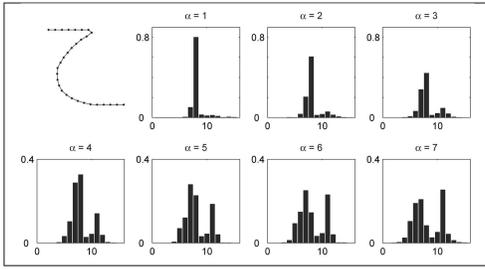


Fig. 7. Tangent differences H histogram for α multiple of 1.

Resampling is implemented in almost every online handwritten system. Points captured during writing are, generally, equidistant in time but not in space. Hence, the number of captured points varies according to writing speed. It also varies from character to character. To normalize the size of the acquired string of coordinates, the sequence of captured points is replaced with a sequence of a fixed number of equidistant points. Basically, this is done by sampling a fixed number of points on the polygonal line of the character, with the distance between adjacent points equal to the total length of the polygonal line divided by the number of intervals (within the resolution of the acquisition tablet). This normalization removes the variability due to the varying writer speed and, therefore, eases the burden of recognition. In our experiments, we used 30 equidistant points. This is about the average number of points in the database characters.

6.3 Results

The database is divided into two distinct sets. The training set contains 6,336 samples and the testing set 3,168 samples (which corresponds to 352 samples of each character for training and 176 other samples of each character for testing). The database is organized as a *sequence* of scriptor data, that is, all the data of a scriptor follows that of another. The division of this sequence results in scriptors being represented in the training data but not in the test data. This simulates writer independent testing.

The first task is training to learn the class-conditional densities from the training samples. This consists of computing the representation features and their histograms (Section 3), determining the potential functions for all classes (Section 4 and Appendix), and their partition functions (Section 5). Training involves the choice of a set of feature indices (α s) and the number of bins in the feature histograms. We determined these experimentally as described subsequently.

6.3.1 Feature Indices and Number of Bins

Recall that a histogram is an m -dimensional vector:

$$H^{(\alpha)}(\Gamma) = \left(H_1^{(\alpha)}, H_2^{(\alpha)}, \dots, H_m^{(\alpha)} \right).$$

The use of all statistics $H^{(\alpha)}$, $\alpha \in \{0, 1, \dots, N-1\}$ can result in a representation of significantly high dimension. For $N = 50$ and $m = 24$, for example, we have a representation of 1,200 entries. Moreover, such a representation contains a significant amount of redundant information because features corresponding to close values of α have close histograms, which, therefore, contain similar information. This is illustrated in Fig. 7, which shows histograms of tangent differences for α equal to multiples of 1. We note that

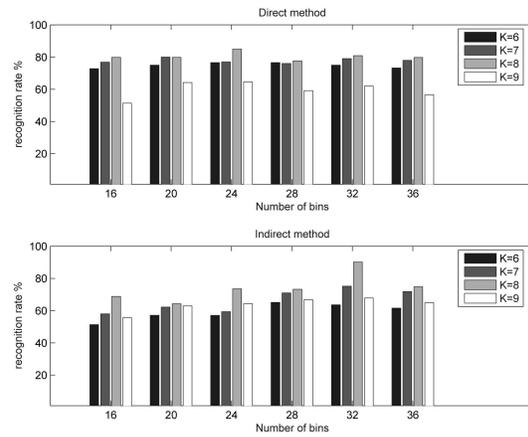


Fig. 8. Recognition rate (percent) versus the number of bins for different values of the number of features.

TABLE 1
Retained Parameter for the Design of the Bayesian Classifiers

| Parameters | Direct method | Indirect method |
|--------------------|---------------|-----------------|
| Number of bins | 24 | 32 |
| Number of features | 8 | 8 |
| Recognition rate | 84.85 % | 90.19 % |

the histograms corresponding to $\alpha = 2, 3$ are very close, as are histograms for $\alpha = 5, 6, 7$. Therefore, these histograms contain redundant information that can be omitted from the representation. By contrast, histograms for α multiples of 3 in Fig. 1 are quite different from each other and, therefore, contain complementary information.

Selection of the most relevant features from a set of features is an important problem because exhaustive, combinatorial, experimentation is, in general, computationally prohibitive. Feature selection has been the subject of several studies ([37], [39], [48], [57], for instance). Since the focus of our study is to show that Zhu et al.'s theory of Gibbs density estimation can be useful in pattern classification, isolated Arabic characters specifically, we did not address fully the problem of "optimizing" with respect to the feature indices and the number of histogram bins. For our stated purpose, it was sufficient to have a reasonable set of characteristics to be able to concentrate on classification proper.

For α , we consider multiples of 3. For example, with $K = 4$ (four statistics), the corresponding histograms are $H^{(3)}$, $H^{(6)}$, $H^{(9)}$, and $H^{(12)}$.

We experimented with α equal to multiples of 2, 3, and 4, and with the number of features equal to 6, 7, 8, and 9. The number of histogram bins was varied as a multiple of 4 between 16 and 36.

The best results were obtained with $\alpha = 3$. Fig. 8 shows the recognition rate versus the number of bins when varying the number of features (K) for $\alpha = 3$. The recognition rates have a typical behavior. They tend to increase with number of features and the number of bins to attain a maximum and then fall. Table 1 lists the results, the recognition rate being higher for the indirect method (90.19 percent compared to 84.85 percent for the direct method).

TABLE 2
Three-Fold Cross Validation Recognition Rates

| Fold | Direct method | Indirect method |
|------|---------------|-----------------|
| 1 | 84.85 % | 90.16 % |
| 2 | 84.56 % | 90.27 % |
| 3 | 82.29 % | 89.2 % |
| Mean | 84.01% | 89.87 % |

6.3.2 Cross Validation Results

The average of the recognition rates by L -fold cross validation is more representative of performance because recognition rates can vary from one data test set to another due to statistical peculiarities that can be present in individual data test sets. We used $L = 3$. Therefore, the database is divided into three disjoint parts of (approximately) equal size. Two parts are used for training, and the other part is used for testing. By varying the role of each part, we make three separate experiments. If we assume that the number of samples in the database is a multiple of L , then, as pointed out in [36], $L = 3$ gives the least number of training samples strictly greater than the number of test samples. This value reflects the concern of training a classifier on as much data as possible and test it on as much distinct data as possible under the constraint that the training sample is strictly larger than the test sample. In this case, recognition rates can be viewed as worst case rates compared to those with $L > 3$. Table 2 gives the recognition rates. For both methods (direct and indirect), the rates are close, which indicates that the classifiers are stable.

6.3.3 Combination of Classifiers

For a given test sample, the entry (i, j) of the confusion matrix is the number of times the classifier identifies a test character i as a character j . Each column of the matrix corresponds to the classifier output and each row to the input. The confusion matrices (Table 4 for the direct method and Table 5 for the indirect method) show that many of the errors occur with some letters and that the error profiles of the two classifiers are different. For example, the recognition rate for *Alif* is 79.0 percent with the direct method and 98.3 percent with the indirect method. However, the direct method performs better with the letter *Mim* (95.5 percent) than the indirect method (82.4 percent). The recognition rate r_i for letter i can be seen as a confidence rating of the classifier when its decision is i : When its output is i , it is r_i percent of the time right. Therefore, we combined the two classifiers using the following simple decision rule: the combination classifier output is the output of the classifier with the highest confidence rating. This simple combination was able to improve the recognition rate to 92.61 percent, up from 84.85 percent for one classifier and 90.19 percent for the other. The confusion matrix for the combination is shown in Table 6.

Fig. 9 shows misclassified characters. Some, such as the third sample of *Kaf* and the first sample of *Mim*, are quite distorted. Others, such as the fourth sample of *Seen* and the fourth sample of *Mim* are legible but misclassified, most likely because they have an appearance rare among this letter's training examples.

6.4 Comparisons

We trained a Kohonen neural network classifier [24] with histograms of tangent differences. Kohonen networks has

TABLE 3
Comparative Summary of the Results

| Classifier | τ |
|------------------------------------|---------|
| Bayes classifier (direct method) | 84.85 % |
| Bayes classifier (indirect method) | 90.19 % |
| Combination of Bayes classifiers | 92.61 % |
| Kohonen neural network | 90 % |
| NN | 94.00 % |

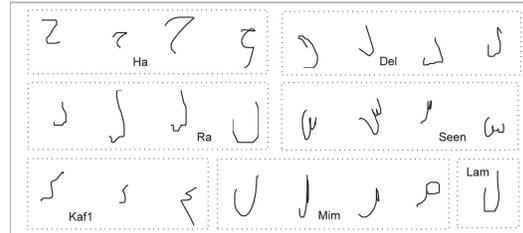


Fig. 9. Some misclassified samples.

been used in other studies of Arabic character recognition using Fourier coefficients [32], tangents [31], and histograms of tangents and tangent differences [34].

The Kohonen neural network (also called Kohonen self-organizing map, or SOM) runs an unsupervised clustering algorithm. It has attractive properties such as topological ordering and good generalization, which make it a potent classifier.

We also ran the NN classifier to provide a benchmark comparison [35], [45]. The classifier functions as follows: Let S be a set of labeled samples (called reference patterns). The NN rule assigns an observation X to the class of the sample in S nearest X . Its asymptotic error rate (as a function of the number of reference patterns) is bounded by twice the optimal Bayes classifier rate. It is an excellent performer in practice, with recognition rates better than other classifiers such as neural networks [35]. However, its asymptotic throughput is zero (as a function of the number of reference patterns). Generally, NN affords a practical upper bound on the recognition rate [45].

Table 3 gives a summary of the results. The Bayes classifiers (lines 1, 2, and 3), the Kohonen neural network (line 4), and NN (line 5), all use the same training/test data. They also used the same representation by histograms of tangent differences, which were determined using the same selection procedure, described in Section 6.3.1. These results clearly support the conclusion that the Zhu et al. formalism can estimate class conditional Gibbs density functions accurately enough to serve Bayes classification, notwithstanding small training samples. In this application, the Bayes classifiers and the Kohonen neural network, generally a good performer, have comparable performance, and combining the Bayes classifiers draws closer to NN. If c is the number of classes, n the number nodes in the Kohonen network, and r the number of NN reference patterns, the Bayes classifiers require c slots of memory to store the Gibbs density parameters and the partition function of each class of characters, compared to n for the Kohonen memory to store the nodes feature histograms, and r for NN to store the reference patterns feature histograms. In this application, $c = 18$, $n = 400$, and $r = 6, 336$. Therefore, the response time for a classification request can be several times shorter for the Bayes classifiers.

TABLE 4
Confusion Matrix—First Bayesian Classifier (Direct Method)

| | Alif | Ba | Ha | Dal | Ra | Seen | Sad | Tah | Ain | Fa | Kaf1 | Lam | Mim | Nun | He | Waw | Ya | Kaf2 | $\tau(\%)$ |
|-------|------|----|-----|-----|-----|------|-----|-----|-----|-----|------|-----|-----|-----|-----|-----|-----|------|------------|
| Alif | 139 | 0 | 1 | 0 | 15 | 0 | 0 | 0 | 0 | 5 | 11 | 0 | 1 | 0 | 0 | 4 | 0 | 0 | 79.0 |
| Ba | 0 | 92 | 0 | 32 | 1 | 8 | 0 | 0 | 0 | 1 | 0 | 39 | 1 | 0 | 0 | 0 | 2 | 0 | 52.3 |
| Ha | 0 | 0 | 172 | 0 | 0 | 0 | 0 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 97.7 |
| Dal | 0 | 1 | 0 | 154 | 2 | 0 | 0 | 0 | 0 | 3 | 0 | 6 | 5 | 0 | 0 | 5 | 0 | 0 | 87.5 |
| Ra | 0 | 0 | 0 | 19 | 149 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 0 | 0 | 0 | 0 | 0 | 84.7 |
| Seen | 0 | 1 | 0 | 2 | 0 | 156 | 0 | 0 | 0 | 6 | 0 | 0 | 10 | 0 | 0 | 0 | 1 | 0 | 88.6 |
| Sad | 0 | 0 | 0 | 0 | 0 | 0 | 160 | 0 | 0 | 9 | 0 | 0 | 6 | 0 | 0 | 0 | 1 | 0 | 90.9 |
| Tah | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 163 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 8 | 92.6 |
| Ain | 0 | 0 | 9 | 0 | 0 | 0 | 0 | 0 | 166 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 94.3 |
| Fa | 0 | 1 | 0 | 2 | 0 | 1 | 0 | 0 | 0 | 166 | 0 | 0 | 4 | 0 | 0 | 1 | 1 | 0 | 94.3 |
| Kaf1 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 173 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 98.3 |
| Lam | 0 | 2 | 0 | 24 | 5 | 0 | 9 | 0 | 0 | 2 | 0 | 123 | 1 | 0 | 1 | 0 | 5 | 4 | 69.9 |
| Mim | 0 | 0 | 5 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 168 | 0 | 0 | 0 | 0 | 0 | 95.5 |
| Nun | 0 | 8 | 0 | 29 | 0 | 0 | 3 | 0 | 0 | 4 | 0 | 16 | 0 | 83 | 16 | 0 | 16 | 1 | 47.2 |
| Ha | 0 | 0 | 6 | 0 | 0 | 0 | 1 | 1 | 0 | 6 | 5 | 0 | 2 | 0 | 153 | 1 | 1 | 0 | 86.9 |
| Waw | 0 | 0 | 2 | 0 | 1 | 0 | 1 | 0 | 0 | 8 | 5 | 0 | 15 | 0 | 0 | 144 | 0 | 0 | 81.8 |
| Ya | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 169 | 1 | 96.0 |
| Kaf2 | 0 | 3 | 2 | 3 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 4 | 0 | 4 | 0 | 0 | 158 | 89.8 |
| Error | 0 | 16 | 27 | 113 | 24 | 9 | 17 | 2 | 3 | 48 | 23 | 61 | 63 | 0 | 21 | 11 | 27 | 15 | |

TABLE 5
Confusion Matrix—Second Bayesian Classifier (Indirect Method)

| | Alif | Ba | Ha | Dal | Ra | Seen | Sad | Tah | Ain | Fa | Kaf1 | Lam | Mim | Nun | He | Waw | Ya | Kaf2 | $\tau(\%)$ |
|-------|------|-----|-----|-----|-----|------|-----|-----|-----|-----|------|-----|-----|-----|-----|-----|-----|------|------------|
| Alif | 173 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 98.3 |
| Ba | 0 | 164 | 0 | 4 | 2 | 2 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 93.2 |
| Ha | 0 | 0 | 161 | 0 | 3 | 0 | 0 | 0 | 9 | 0 | 0 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 91.5 |
| Dal | 1 | 6 | 0 | 146 | 12 | 0 | 0 | 0 | 0 | 1 | 4 | 0 | 5 | 0 | 1 | 0 | 0 | 0 | 83.0 |
| Ra | 6 | 1 | 0 | 1 | 168 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 95.5 |
| Seen | 0 | 4 | 0 | 0 | 1 | 162 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 0 | 0 | 0 | 6 | 0 | 92.0 |
| Sad | 0 | 0 | 0 | 5 | 0 | 0 | 157 | 0 | 0 | 4 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 7 | 89.2 |
| Tah | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 174 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 98.9 |
| Ain | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 173 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 98.3 |
| Fa | 0 | 3 | 0 | 9 | 0 | 2 | 1 | 0 | 0 | 150 | 0 | 0 | 0 | 1 | 1 | 4 | 5 | 0 | 85.2 |
| Kaf1 | 2 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 165 | 0 | 0 | 0 | 0 | 6 | 0 | 0 | 93.8 |
| Lam | 0 | 1 | 0 | 15 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 135 | 0 | 9 | 0 | 0 | 8 | 0 | 76.7 |
| Mim | 7 | 0 | 9 | 1 | 1 | 1 | 0 | 3 | 0 | 3 | 1 | 0 | 145 | 0 | 0 | 3 | 0 | 2 | 82.4 |
| Nun | 0 | 12 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 146 | 3 | 0 | 10 | 0 | 83.0 |
| He | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 2 | 0 | 1 | 168 | 1 | 0 | 0 | 95.5 |
| Waw | 0 | 0 | 0 | 10 | 3 | 0 | 0 | 0 | 0 | 1 | 3 | 0 | 0 | 0 | 0 | 159 | 0 | 0 | 90.3 |
| Ya | 0 | 4 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 2 | 0 | 1 | 5 | 1 | 156 | 4 | 88.6 |
| Kaf2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 11 | 0 | 0 | 9 | 0 | 0 | 153 | 86.9 |
| Error | 18 | 31 | 12 | 50 | 33 | 5 | 3 | 5 | 9 | 11 | 10 | 20 | 5 | 19 | 21 | 17 | 30 | 14 | |

TABLE 6
Confusion Matrix—Classifier Combination

| | Alif | Ba | Ha | Dal | Ra | Seen | Sad | Tah | Ain | Fa | Kaf1 | Lam | Mim | Nun | He | Waw | Ya | Kaf2 | $\tau(\%)$ |
|-------|------|-----|-----|-----|-----|------|-----|-----|-----|-----|------|-----|-----|-----|-----|-----|-----|------|------------|
| Alif | 173 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 98.3 |
| Ba | 0 | 164 | 0 | 4 | 2 | 2 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 93.2 |
| Ha | 0 | 0 | 161 | 0 | 3 | 0 | 0 | 0 | 9 | 0 | 0 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 91.5 |
| Del | 1 | 6 | 0 | 141 | 12 | 0 | 0 | 0 | 0 | 3 | 1 | 6 | 0 | 5 | 0 | 1 | 0 | 0 | 80.1 |
| Ra | 6 | 1 | 0 | 1 | 168 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 95.5 |
| Seen | 0 | 4 | 0 | 0 | 1 | 162 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 0 | 0 | 0 | 6 | 0 | 92.0 |
| Sad | 0 | 0 | 0 | 1 | 0 | 0 | 161 | 0 | 0 | 4 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 7 | 91.5 |
| Tah | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 174 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 98.9 |
| Ain | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 173 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 98.3 |
| Fa | 0 | 1 | 0 | 2 | 0 | 2 | 1 | 0 | 0 | 164 | 0 | 0 | 0 | 1 | 0 | 4 | 1 | 0 | 93.2 |
| Kaf1 | 2 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 170 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 96.6 |
| Lam | 0 | 1 | 0 | 8 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 138 | 0 | 9 | 0 | 0 | 8 | 4 | 78.4 |
| Mim | 7 | 0 | 1 | 1 | 1 | 1 | 0 | 3 | 0 | 3 | 1 | 0 | 153 | 0 | 0 | 3 | 0 | 2 | 86.9 |
| Nun | 0 | 7 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 151 | 2 | 0 | 9 | 1 | 85.8 |
| He | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 2 | 0 | 1 | 167 | 1 | 1 | 0 | 94.9 |
| Waw | 0 | 0 | 0 | 9 | 3 | 0 | 0 | 0 | 0 | 2 | 4 | 0 | 0 | 0 | 0 | 158 | 0 | 0 | 89.8 |
| Ya | 0 | 3 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 2 | 1 | 2 | 0 | 1 | 0 | 0 | 161 | 4 | 91.5 |
| Kaf2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 6 | 0 | 0 | 2 | 0 | 0 | 165 | 93.8 |
| Error | 18 | 23 | 3 | 31 | 33 | 5 | 3 | 5 | 9 | 17 | 12 | 17 | 6 | 19 | 7 | 11 | 26 | 19 | |

The previous studies of online Arabic character recognition that used Kohonen neural network classifiers reported classification rates of 89 percent (Fourier coefficients) [32], 85 percent (tangents) [31], and 94 percent (tangents and tangent differences) [34]. However, one should be cautious about comparing with these results because the studies used a different training/test database (an earlier version of the

database in this study, about two thirds the size). Also, the representations used 100 points on the character shape, rather than 50, as in this study. Finally, the focus of that in [34] was on representation and, therefore, feature selection was more extensive than in this study, which focuses on classification. Nevertheless, the good performance in [34] indicates that a more extensive investigation of feature statistics can improve

TABLE 7
Potential Functions Estimation Algorithm

Given a set of observed (training) characters Γ_i^{obs} , $i = 1, \dots, M$.

- Compute a set of features $\{\phi_i^{obs(\beta)}, \beta = 1, \dots, K\}$.
- Compute a set of their corresponding statistics $\{H_i^{obs(\beta)}, \beta = 1, \dots, K\}$.
- Compute the mean of statistics $\mu^{obs(\beta)}, \beta = 1, 2, \dots, K$.
- Initialize $\lambda^{(\beta)} \leftarrow 0, \beta = 1, \dots, K$.
- Initialize synthesized shapes $\Gamma_j^{syn}, j=1,2,\dots,M'$.
- Repeat
 - Compute $\{H_j^{syn(\beta)}, \beta = 1, 2, \dots, K; j = 1, 2, \dots, M'\}$ for $\{\Gamma_j^{syn}, j = 1, 2, \dots, M'\}$.
 - Compute $\{\mu^{syn(\beta)}, \beta = 1, 2, \dots, K\}$ corresponding to shapes $\{\Gamma_j^{syn}, j = 1, 2, \dots, M'\}$
 - Update $\lambda^{(\beta)}$ by equation (9) (so that $p(\Gamma; \Lambda)$ is updated).
 $\lambda^{(\beta)}(n+1) = \lambda^{(\beta)}(n) + (\mu^{syn(\beta)} - \mu^{obs(\beta)}), \beta = 1, 2, \dots, K$.
 - Sample from $p(\Gamma; \Lambda)$ (Table VIII) to synthesize $\Gamma_j^{syn}, j = 1, 2, \dots, M'$ in s sweeps.

Until $|\mu^{obs(\beta)} - \mu^{syn(\beta)}| \leq \epsilon, \beta = 1, 2, \dots, K$

the class conditional density function estimates and, therefore, the recognition rate of the Bayes methods.

7 CONCLUSION

The purpose of this study was to bring Gibbs density parameters estimation formalism by Zhu et al. to bear on pattern classification. More specifically, we investigated Bayes classification of online Arabic characters using an efficient representation by empirical distributions of tangent differences and Gibbs modeling of the class-conditional density functions. The parameters of these density functions were estimated following the Zhu et al. constrained maximum entropy formalism, which was originally used for image synthesis. We also addressed the problem of estimating the class conditional partition functions, which, unlike image and pattern synthesis, Bayes classification requires. We investigated two partition function estimation methods: one uses the training sample, and the other draws from a reference distribution. The efficiency of the corresponding Bayes decision methods, and of a combination of these, was evaluated in experiments using a database of about 10,000 freely written samples by a number of scriptors. Comparisons to the NN rule method and a recent Kohonen neural network method were made.

APPENDIX A

POTENTIAL FUNCTIONS ESTIMATION ALGORITHM

We recall that the potential functions are estimated iteratively by the descent equation:

$$\frac{d\lambda^{(\beta)}}{dt} = \mu^{syn(\beta)} - \mu^{obs(\beta)}, \quad \beta = 1, 2, \dots, K. \quad (18)$$

This aims at determining potential functions so as to match the mean statistics (feature histograms) of the synthesized shapes with the corresponding mean observed statistics. The algorithm, given in Table 7, is basically as in [57] (the differences reside in details of implementation as described below).

TABLE 8
Shape Sampling Algorithm

Given a shape $\Gamma = ((x_1, y_1), (x_2, y_2), \dots, (x_{n_{point}}, y_{n_{point}}))$

- For $s = 1 \rightarrow s_{max}$, with s_{max} being the maximum sweep number
 - For $p = 1 \rightarrow n_{point}$, with n_{point} being the point number on shapes
 - For $d = 1 \rightarrow n_{direction}$, with $n_{direction}$ being the number of directions
 - move A_l according to d directions to obtain a set of new shapes $\Gamma_d, l = 1, 2, \dots, d$
 - Compute $p(\Gamma_d; \Lambda)$ by equation (7).
 - Keep $\hat{\Gamma}$ which maximizes $p(\Gamma_d; \Lambda)$ and affect it to Γ^{syn} .
 - Compute $\eta(\Gamma \rightarrow \Gamma^{syn}) = \min\left(\frac{p(\Gamma^{syn}; \Lambda)}{p(\Gamma; \Lambda)}, 1\right)$.
 - Draw a random number $r \in [0, 1]$ from a uniform distribution
 - If $r < \eta$, replace Γ by Γ^{syn} .

It starts by initializing to zero the parameters $\lambda^{(\beta)}, \beta = 1 \dots, K$ and by initializing the forms Γ^{syn} . At each iteration t , the update uses the difference between $\mu^{syn(\beta)}$ and $\mu^{obs(\beta)}$. At each iteration, the calculation of $\mu^{syn(\beta)}$ can be done by averaging M' forms $\Gamma_j^{syn}, j = 1, 2 \dots M'$ sampled (synthesized) from the current distribution $p(\Gamma; \Lambda)$, using a stochastic sampler MCMC (Markov Chain Monte Carlo).

APPENDIX B

MCMC SHAPE SAMPLING ALGORITHM

The sampling process [49] starts with a given character shape

$$\Gamma = ((x_1, y_1), (x_2, y_2), \dots, (x_{n_{point}}, y_{n_{point}})).$$

At each step n , it considers a point $A_l(x_l, y_l)$ on the shape Γ and proposes to move it to a candidate point $A'_l(x'_l, y'_l)$ in a local neighborhood of A_l (Table 8). The proposal A'_l is accepted with probability

$$\eta(\Gamma \rightarrow \Gamma^{syn}) = \min\left(\frac{p(\Gamma^{syn}; \Lambda)}{p(\Gamma; \Lambda)}, 1\right). \quad (19)$$

If the ratio $r = \frac{p(\Gamma^{syn}; \Lambda)}{p(\Gamma; \Lambda)}$ is higher than 1, then the point candidate $A'_l(x'_l, y'_l)$ is accepted. Else, the point is accepted with a probability r . This algorithm makes it possible to synthesize e shapes Γ^{syn} used to calculate potential functions Λ .

In order to improve the sampling process, we chose the candidate point A'_l among points $A_l^d(x_l^d, y_l^d)$ by moving the point A_l according to the directions d

$$x_l^d = x_l + \tau \sin(\kappa_d), \quad y_l^d = y_l + \tau \cos(\kappa_d), \quad (20)$$

where κ_d indicates the angle corresponding to direction d .

$$\kappa_d = d \frac{2\pi}{n_{direction}} \quad (21)$$

and τ is a scale factor calculated according to $\tau = \frac{ds}{constant}$, ds being the length of the segment between two consecutive points of the form Γ . Note that all the points of a form Γ are equidistant following resampling (Section 6.2), and consequently, the distance between two consecutive points on Γ is constant. We chose a code with eight directions ($n_{direction} = 8$), and a *constant* equal to 5 for the calculation of the scale factor τ .

ACKNOWLEDGMENTS

N. Mezghani was with the Institut National de la Recherche Scientifique, Center for Énergie, Matériaux et Télécommunications (INRS-EMT). This research was supported in part under grant NSERC OGP0004234.

REFERENCES

- [1] B. Al-Badr and S.A. Mahmoud, "Survey and Bibliography of Arabic Optical Text Recognition," *Signal Processing*, vol. 41, no. 1, pp. 49-77, 1995.
- [2] T.S. Al-Sheikh and S.G. El-Taweel, "Proc. Real-Time Arabic Handwritten Character Recognition," *Pattern Recognition*, vol. 23, no. 12, pp. 1323-1332, 1990.
- [3] A.M. Alimi, "An Evolutionary Neuro-Fuzzy Approach to Recognize On-Line Arabic Handwriting," *Proc. Int'l Conf. Document Analysis and Recognition*, vol. 1, pp. 382-386, Aug. 1997.
- [4] A.M. Alimi, "A Neuro-Fuzzy Approach to Recognize On-Line Arabic Handwriting," *Proc. Int'l Conf. Neural Networks*, vol. 3, pp. 1397-1400, Aug. 1997.
- [5] N. Ben Amara and F. Bouslama, "Classification of Arabic Script Using Multiple Sources of Information: State of the Art and Perspectives," *Int'l J. Document Analysis and Recognition*, vol. 5, no. 4, pp. 195-212, July 2003.
- [6] A. Amin, "Off Line Arabic Character Recognition—A Survey," *Proc. Int'l Conf. Document Analysis and Recognition*, vol. 1, p. 596, Aug. 1997.
- [7] A. Amin, "Off-Line Arabic Character-Recognition: The State of the Art," *Pattern Recognition*, vol. 31, no. 5, pp. 517-530, May 1998.
- [8] G.R. Ball, S.N. Srihari, and H. Srinivasan, "Segmentation-Based and Segmentation-Free Methods for Spotting Handwritten Arabic Words," *Proc. Int'l Workshop Frontiers of Handwriting Recognition*, Oct. 2006.
- [9] F. Biadys, J. El-Sana, and N. Habash, "Online Arabic Handwriting Recognition Using Hidden Markov Models," *Proc. Int'l Workshop Frontiers of Handwriting Recognition*, Oct. 2006.
- [10] F. Bouslama and A. Amin, "Pen-Based Recognition System of Arabic Character Utilizing Structural and Fuzzy Techniques," *Proc. Second Int'l Conf. Knowledge-Based Intelligent Electronic Systems*, pp. 76-85, 1998.
- [11] X. Descombes, R. Morris, J. Zerubia, and M. Berthod, "Maximum Likelihood Estimation of Markov Random Field Parameters Using Markov Chain Monte Carlo Algorithms," *Proc. Int'l Workshop Energy Minimization Methods*, May 1997.
- [12] D. Doermann, "The Indexing and Retrieval of Document Images: A Survey," *Computer Vision and Image Processing*, vol. 70, no. 3, pp. 287-298, June 1998.
- [13] P.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification*, 2001.
- [14] M.S. El-wakil and A.A. Shoukry, "On-Line Recognition of Handwritten Arabic Character Recognition," *Pattern Recognition*, vol. 22, no. 2, pp. 97-105, 1989.
- [15] S. Al Emami and M. Usher, "On-Line Recognition of Handwritten Arabic Characters," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 704-710, July 1990.
- [16] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Recognition and Machine Intelligence*, vol. 6, no. 6, pp. 721-741, 1984.
- [17] V.K. Govindan and A.P. Shivaprasad, "Character Recognition—A Review," *Pattern Recognition*, vol. 23, no. 7, pp. 671-683, 1990.
- [18] A.H. Hassin, X.L. Tang, J.F. Liu, and W. Zhao, "Printed Arabic Character Recognition Using HMM," *J. Computer Science Technology*, vol. 19, no. 4, pp. 538-543, 2004.
- [19] E.T. Jaynes, "Information Theory and Statistical Mechanics I," *Physical Rev.*, vol. 106, pp. 620-630, 1957.
- [20] T. Jebara and T. Jaakkola, "Feature Selection and Dualities in Maximum Entropy Discrimination," *Uncertainty in Artificial Intelligence*, 2000.
- [21] S. Jehan-Besson, A. Herbulot, M. Barlaud, and G. Aubert Shape Gradient for Image and Video Segmentation, *Math. Models in Computer Vision: The Handbook*, 2005.
- [22] M. Jianchang, "Neural Networks in Off-Line Text Recognition: A Review," *Proc. Int'l Joint Conf. Neural Networks*, vol. 4, pp. 2934-2939, July 1999.
- [23] M.S. Khorsheed, "Off-Line Arabic Character Recognition—A Review," *Pattern Analysis and Applications*, vol. 5, no. 1, pp. 31-45, 2002.
- [24] T. Kohonen, *Self-Organizing Maps*. Springer, 1995.
- [25] C.L. Liu, S. Jaeger, and M. Nakagawa, "Online Recognition of Chinese Characters: The State-of-the-Art," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 198-213, Feb. 2004.
- [26] G. Lorette, "Handwriting Recognition or Reading? What Is the Situation at the Dawn of the 3rd Millenium," *Int'l J. Document Analysis and Recognition*, vol. 2, no. 1, pp. 2-12, 1999.
- [27] L.M. Lorigo and V. Govindaraju, "Offline Arabic Handwriting Recognition: A Survey," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 712-724, May 2006.
- [28] M.A. Mahjoub, "Apport de la modélisation de la durée d'état dans la reconnaissance en-ligne des caractères arabes par HMMs," *Proc. 17th Journée Tunisienne en Électrotechnique et Automatique*, pp. 341-348, 1997.
- [29] S.A. Mahmoud, "Arabic Character-Recognition Using Fourier Descriptors and Character Contour Encoding," *Pattern Recognition*, vol. 27, no. 6, pp. 815-824, June 1994.
- [30] S. Marukatat, R. Sicard, T. Artières, and P. Gallinari, "A Flexible Recognition Engine for Complex On-Line Handwritten Character Recognition," *Proc. Seventh Int'l Conf. Document Analysis and Recognition*, vol. 2, pp. 1048-1051, Aug. 2003.
- [31] N. Mezghani, M. Cheriet, and A. Mitiche, "Combination of Pruned Kohonen Maps for On-Line Arabic Characters Recognition," *Proc. Seventh Int'l Conf. Document Analysis and Recognition*, vol. 2, pp. 900-905, Aug. 2003.
- [32] N. Mezghani, A. Mitiche, and M. Cheriet, "On-Line Recognition of Handwritten Arabic Characters Using a Kohonen Neural Network," *Proc. Eighth Int'l Workshop Frontiers in Handwriting Recognition*, pp. 490-495, 2002.
- [33] N. Mezghani, A. Mitiche, and M. Cheriet, "Reconnaissance En-Ligne de Caractères Arabes Manuscrits par un Réseau de Kohonen," *Proc. Vision Interface*, pp. 186-191, 2002.
- [34] N. Mezghani, A. Mitiche, and M. Cheriet, "A New Representation of Shape and Its Use for Superior Performance in On-Line Arabic Character Recognition by an Associative Memory," *Int'l J. Document Analysis and Recognition*, vol. 7, no. 4, pp. 201-210, 2005.
- [35] A. Mitiche and J.K. Aggarwal, "Pattern Category Assignment by Neural Networks and the Nearest Neighbors Rule," *Int'l J. Pattern Recognition and Artificial Intelligence*, vol. 10, pp. 393-408, 1996.
- [36] A. Mitiche and M. Lebidoff, "Pattern Classification by a Condensed Neural Networks," *Neural Network*, vol. 14, no. 4, pp. 575-580, May 2001.
- [37] I.S. Oh, J.S. Lee, and B.R. Moon, "Hybrid Genetic Algorithms for Feature Selection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1424-1437, Nov. 2004.
- [38] G. Olivier, H. Miled, K. Romeo, and Y. Lecourtier, "Segmentation and Coding of Arabic Handwritten Words," *Proc. Int'l Conf. Pattern Recognition*, vol. 3, pp. 264-268, 1996.
- [39] V. Onnia, M. Tico, and J. Saarinen, "Feature Selection Method Using Neural Network," *Proc. Int'l Conf. Image Processing*, vol. 1, pp. 513-516, 2001.
- [40] R. Plamondon, "A Model-Based Segmentation Framework for Computer Processing of Handwriting," *Proc. 11th IAPR Int'l Conf. Pattern Recognition*, pp. 303-307, 1992.
- [41] R. Plamondon, "Handwriting Generation: The Delta Lognormal Theory," *Proc. Fourth Int'l Workshop Frontiers in Handwriting Recognition*, pp. 1-10, 1994.
- [42] R. Plamondon and S.N. Srihari, "On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 63-84, Jan. 2000.
- [43] G. Potamianos and J. Goutsias, "Stochastic Approximation Algorithms for Partition Function Estimation of Gibbs Random Fields," *IEEE Trans. Information Theory*, vol. 43, no. 6, pp. 1948-1965, 1997.
- [44] K.R. Pakker, A. Ameer, C. Olivier, and Y. Lecourtier, "Structural-Analysis of Arabic Handwriting: Segmentation and Recognition," *Machine Vision and Applications*, vol. 8, no. 4, pp. 232-240, 1995.
- [45] M. Sabourin, A. Mitiche, D. Thomas, and G. Nagy, "NN 1 or Hand-Printed Digit Recognition Using Nearest Neighbors," *Proc. Second Ann. Symp. Document Analysis and Information Retrieval*, pp. 397-412, Apr. 1993.

- [46] M. Schenkel, I. Guyon, and D. Henderson, "On-Line Script Recognition Using Time Delay Neural Networks and Hidden Markov Models," *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, vol. 2, pp. 637-640, 1994.
- [47] J. Shah, "Minimax Entropy and Learning by Diffusion," *Proc. Int'l Conf. Computer Vision*, pp. 92-97, 1998.
- [48] K. Shima, M. Todoriki, and A. Suzuki, "SVM-Based Feature Selection of Latent Semantic Features," *Pattern Recognition Letters*, vol. 25, no. 9, pp. 1051-1057, July 2004.
- [49] J.C. Spall, "Estimation via Markov Chain Monte Carlo," *IEEE Control Systems Magazine*, vol. 23, pp. 34-45, 2003.
- [50] C.C. Tappert, C.Y. Suen, and T. Wakahara, "Online Handwriting Recognition—A Survey," *Proc. Int'l Conf. Pattern Recognition*, vol. 2, pp. 1123-1132, Nov. 1988.
- [51] C.C. Tappert, C.Y. Suen, and T. Wakahara, "The State of the Art in On-Line Handwriting Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 8, pp. 787-808, Aug. 1990.
- [52] O. Trier, A. Jain, and T. Taxt, "Feature Extraction Methods for Character Recognition—A Survey," *Pattern Recognition*, vol. 29, pp. 641-662, 1996.
- [53] A.M. Zeki, "The Segmentation Problem in Arabic Character Recognition the State of the Art," *Proc. First Int'l Conf. Information and Communication Technologies*, pp. 11-26, Aug. 2005.
- [54] L. Zheng, A.H. Hassin, and X. Tang, "A New Algorithm for Machine Printed Arabic Character Segmentation," *IEEE Pattern Recognition Letters*, vol. 25, no. 15, pp. 1723-1729, Nov. 2004.
- [55] S.C. Zhu, Y. Wu, and D. Mumford, "Minimax Entropy Principle and Its Application to Texture Modeling," *Neural Computer*, vol. 9, no. 9, pp. 1627-1660, 1997.
- [56] S.C. Zhu, Y. Wu, and D. Mumford, "Filters, Random Fields and Maximum Entropy (FRAME): Towards a Unified Theory for Texture Modeling," *Int'l J. Computer Vision*, vol. 27, no. 2, pp. 107-126, 1998.
- [57] S.C. Zhu, "Embedding Gestalt Laws in Markov Random Fields," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1170-1187, Nov. 1999.



Amar Mitiche received the Licence Ès Sciences degree in mathematics from the University of Algiers and the PhD degree in computer science from the University of Texas at Austin. He is currently a professor at the Institut National de la Recherche Scientifique (INRS), Department of Telecommunications (INRS-EMT), Montreal. His research is in computer vision and pattern recognition. His current interests include image segmentation, motion analysis in monocular and stereoscopic image sequences (detection, estimation, segmentation, tracking, 3D interpretation), focusing on level set methods, and character recognition, focusing on neural networks and Bayes methods. He is a member of the IEEE and the IEEE Computer Society.



Mohamed Cheriet received the BEng degree from the University of Science and Technology Houari Boumediene (USTHB), Algiers, in 1984 and the MSc and PhD degrees in computer sciences from the University of Pierre et Marie Curie (Paris VI) in 1985 and 1988, respectively. Since 1992, he has been a professor in the Automation Engineering Department, École de Technologie Supérieure (University of Quebec), Montreal, where he was appointed as a full professor since 1998. He was the director of the Laboratory for Imagery, Vision, and Artificial Intelligence (LIVIA), from 2000 to 2006 and is the director of SYNCHROMEDIA Consortium since 1998. His current research interests include document image analysis, optical character recognition (OCR), mathematical models for image processing, pattern classification models, learning algorithms, and perception in computer vision. He is a senior member of the IEEE and the IEEE Computer Society.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.



Neila Mezghani received an engineering degree from École Supérieure des Télécommunications, Tunis, Tunisia, in 1996, the diplôme d'études approfondies (DEA) in signal processing from École Nationale des Ingénieurs de Tunis, Tunisia, in 1999, the diplôme d'études supérieures spécialisées in information theory from École Supérieure des Télécommunications, Tunis, Tunisia, in 2001, and the PhD degree from the Institut National de la Recherche Scientifique, Center for Énergie, Matériaux et Télécommunications (INRS-EMT), Montreal, in 2005. Currently, she is a postdoctoral researcher in the Laboratoire de Recherche en Imagerie et Orthopédie (LIO), Centre de Recherche du CHUM, Montreal. Her research interests include online character recognition, neural networks, probability density estimation, and biomedical image and signal analysis and classification. She is a student member of the IEEE and the IEEE Computer Society.

Her research interests include online character recognition, neural networks, probability density estimation, and biomedical image and signal analysis and classification. She is a student member of the IEEE and the IEEE Computer Society.