# Combining Slanted-Frame Classifiers for Improved HMM-Based Arabic Handwriting Recognition

Ramy Al-Hajj Mohamad, Laurence Likforman-Sulem, *Member*, *IEEE*, and
Chafic Mokbel, *Senior Member*, *IEEE*

**Abstract**—The problem addressed in this study is the offline recognition of handwritten Arabic city names. The names are assumed to belong to a fixed lexicon of about 1,000 entries. A state-of-the-art classical right-left hidden Markov model (HMM)-based recognizer (reference system) using the sliding window approach is developed. The feature set includes both baseline-independent and baseline-dependent features. The analysis of the errors made by the recognizer shows that the inclination, overlap, and shifted positions of diacritical marks are major sources of errors. In this paper, we propose coping with these problems. Our approach relies on the combination of three homogeneous HMM-based classifiers. All classifiers have the same topology as the reference system and differ only in the orientation of the sliding window. We compare three combination schemes of these classifiers at the decision level. Our reported results on the benchmark IFN/ENIT database of Arabic Tunisian city names give a recognition rate higher than 90 percent accuracy and demonstrate the superiority of the neural network-based combination. Our results also show that the combination of classifiers performs better than a single classifier dealing with slant-corrected images and that the approach is robust for a wide range of orientation angles.

**Index Terms**—Arabic handwriting, word recognition, feature extraction, IFN/ENIT database, hidden Markov models, HMM, neural network, multilayer perceptron, classifier combination.

## 1 Introduction

The recognition of Arabic writing has many applications such as mail sorting, bank check reading, and, more recently, the recognition of historical manuscripts. Arabic writing is naturally cursive and challenging for offline recognition systems [3], [29]. Different approaches have been proposed for dealing with isolated characters [13], [15], [4], words [2], [14], [32], [16], [28], and printed text lines [6]. For recognition of isolated characters, structural, neural network-based, or statistical methods were reported to be efficient. For word recognition, there are two basic strategies: analytical and holistic. In the analytical strategy, a word is first segmented into the set of its compound letters (or smaller units), and then characters are recognized. A word model is built from the concatenation of character models. On the other hand, the holistic strategy considers word images as a whole and does not attempt to segment words into characters or any other units. Word models are built from word images without segmentation. The advantage of the holistic strategy is that it avoids segmenting words into characters. In the case of Arabic writing, this segmentation is hard to obtain and an incorrect segmentation at an early stage of the process may generally result in recognition errors. The advantage of the analytical strategy is that the size of the lexicon may be large, even without limits. Moreover, adding new words to the lexicon is more convenient with the analytical approach because words can be described through their compound letters, thus, there is no need for providing images of the words themselves. However, manual labeling of character images is often necessary.

The ability of hidden Markov models (HMMs) to cope with variable length observation sequences and with nonlinear distortions make them suitable for cursive handwriting such as Arabic, Latin, and Korean scripts [24], [33], [5], [36], [42]. Most HMM implementations can be considered as hybrid methods, with advantages from both holistic and analytical strategies. First, the character-based representation of words, which qualifies the analytical strategy, is convenient for enlarging the vocabulary. In some cases, the vocabulary can be open such as in the work of Khorsheed [23], where a single HMM models any character sequence. Second, recognition is holistic as it can be performed without presegmentation into characters. However, implicit character segmentation occurs during decoding [9]. Third, character models may be trained from word images without manual segmentation into characters in contrast to basic analytical methods. The so-called *cross training* refines character models through decoding iterations. It is to be noted that a small set of word images hand

- *R. Al-Hajj Mohamad is with the Lebanese International University, Mazraa/Salim-Slam, Beirut, Lebanon. E-mail: al-hajj@enst.fr.*
- *L. Likforman-Sulem is with TELECOM ParisTech/TSI 46 rue Barrault, 75013 Paris, France. E-mail: likforman@telecom-paristech.fr.*
- *C. Mokbel is with the Faculty of Engineering, University of Balamand, El-Koura, PO Box 100 Tripoli, Lebanon.*
  *E-mail: chafic.mokbel@balamand.edu.lb.*

*Manuscript received 20 Feb. 2007; revised 10 Oct. 2007; accepted 22 Apr. 2008; published online 21 May 2008.*
*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-018-0207.*
*Digital Object Identifier no. 10.1109/TPAMI.2008.136.*

0162-8828/09/$25.00 © 2009 IEEE Published by the IEEE Computer Society

TABLE 1
Sample Arabic Characters and Their Correspondence with Characters from the Original Phoenician Alphabet and Other Derived Alphabets

| Phoenician | Arabic | Hebrew | Greek | Latin |
|---|---|---|---|---|
| 𐤀 | ا | א | A | A |
| 𐤁 | ب | ב | B | B |
| 𐤂 | ج | ג | Γ | C G |
| 𐤃 | ذ د | ד | Δ | D |
| 𐤄 | هـ | ה | E | E |

segmented into characters may be necessary to initialize the cross training process.

In this paper, a new method for recognizing Arabic handwritten city names is developed. We believe that this method is robust to variation in the writing inclinations, overlap, and shifted positions of diacritical marks, which were identified as major sources of errors in a preliminary study we had conducted. The method relies on the combination of a reference HMM-based classifier with two other classifiers which are the modified versions of the reference classifier presented in [11]. The reference classifier is a right-left HMM that includes a set of baseline-dependent and baseline-independent features which is considered to be efficient for this task [30], [11]. This HMM classifier benefits from both analytical and holistic strategies as word models are concatenations of character models and as words do not need to be segmented. The two remaining HMMs have the same topology as the reference classifier, but the sliding windows are slanted to cope with slanted writing. The three classifiers are fused at the decision level, where we compare three combination schemes: the sum rule, the majority vote rule, and an original neural network-based combination whose decision function is learned through candidate words' scores. This system thus includes major novelty (slanted windows, combination), compared to the reference system [11].

This paper is organized as follows: Section 2 reviews some characteristics of Arabic handwriting. Section 3 describes the set of features extracted within each window. The feature set presented here is a more accurate and extended version of the feature set presented previously in [11]. In Section 4, the topology of character models, the training, and recognition phases are presented as well as the combination schemes. Section 5 reports the experiments carried out on the IFN/ENIT database of handwritten city names. Our results show that considering the writing inclination within the system architecture yields a significant improvement of recognition performance. This improvement is higher than the one obtained by using a single classifier dealing with slant-corrected images. In Section 6, some conclusions and perspectives are discussed.

## 2 Characteristics of Arabic Handwriting

The Arabic alphabet shares a common origin with Hebrew, Latin, and Greek alphabets (see Table 1). All of them originate from Phoenician and proto-Canaan consonantal handwriting. Around the 14th century BC, the Phoenician handwriting developed into the Aramaic alphabet, from which Semitic writings such as Arabic and Hebrew evolved [43].

TABLE 2
Set of Arabic Characters

| character | isolated | beginning | middle | end |
|---|---|---|---|---|
| Alef | ا | - | - | ـا |
| Beh | ب | بـ | ـبـ | ـب |
| Teh | ت | تـ | ـتـ | ـت |
| Theh | ث | ثـ | ـثـ | ـث |
| Jeem | ج | جـ | ـجـ | ـج |
| Hah | ح | حـ | ـحـ | ـح |
| Khah | خ | خـ | ـخـ | ـخ |
| Dal | د | - | - | ـد |
| Thal | ذ | - | - | ـذ |
| Reh | ر | - | - | ـر |
| Zain | ز | - | - | ـز |
| Seen | س | سـ | ـسـ | ـس |
| Sheen | ش | شـ | ـشـ | ـش |
| Sad | ص | صـ | ـصـ | ـص |
| Dad | ض | ضـ | ـضـ | ـض |
| Tah | ط | طـ | ـطـ | ـط |
| Zah | ظ | ظـ | ـظـ | ـظ |
| Ain | ع | عـ | ـعـ | ـع |
| Ghain | غ | غـ | ـغـ | ـغ |
| Feh | ف | فـ | ـفـ | ـف |
| Quaf | ق | قـ | ـقـ | ـق |
| Kaf | ك | كـ | ـكـ | ـك |
| Lam | ل | لـ | ـلـ | ـل |
| Meem | م | مـ | ـمـ | ـم |
| Noon | ن | نـ | ـنـ | ـن |
| Heh | ه | هـ | ـهـ | ـه |
| Waw | و | - | - | ـو |
| Yeh | ي | يـ | ـيـ | ـي |

The Arabic alphabet has 28 characters. Letter shapes are context sensitive according to their position within the word (beginning, middle, and end) or when the character is isolated, resulting in 100 different shapes. The set of Arabic characters is shown in Table 2. Characters are named according to the Unicode standard [40]. Additional marks (hamza, shadda, etc.), dots, and diacritical marks change the letter and the word meaning or indicate vowels (see Table 3). Additional characters (see Table 4) and character ligatures produce additional character shapes: We model some of them using HMM modeling (see Section 4.1).

Arabic writing is set on a baseline, where character connections occur and from where descending and ascending characters extend. Arabic handwriting may be more difficult to recognize than Latin because of the presence of

TABLE 3
Set of Arabic Diacritical Marks

| Hamza | Damma | Shadda | Fatha/ Kasra | Maddah | Sukun |
|---|---|---|---|---|---|
| ء | ُ | ّ | َ | ~ | ْ |

loops of variable size and shapes, of horizontal character ligatures of variable lengths, of vertical ligatures which create new shapes, and because of variable positions of diacritical marks and dots [7].

## 3 Word Recognition System

### 3.1 System Overview

An overview of the recognition system is shown in Fig. 1. First, baselines are searched from word images. Language-independent features are then extracted within three sliding windows of different orientations. Each orientation is associated to one of the three HMM-based classifiers: The reference classifier uses a vertical window (frame), while the other two observe the image through slanted frames. Within each frame, vertical or slanted, a set of 28 features is extracted. There are two types of features: distribution features based on foreground (black) pixel densities and concavity features that reflect local concavity and stroke directions. Within distribution and concavity features, a subset is baseline dependent to emphasize the presence of ascenders and descenders. Finally, each classifier produces a list of word candidates with their scores. Those candidate lists are then fused at the decision level.

### 3.2 Baseline Extraction

In Latin as well as Arabic script, ascending and descending characters are salient characteristics for recognition. Our feature set thus includes both baseline-dependent and independent features which emphasize the presence of ascenders and descenders. Hence, two baselines are extracted on each word at a preprocessing stage: the lower baseline and the upper baseline. These baselines divide the image into a core zone without ascenders and descenders and two other zones, one including ascenders and the other including descenders. Fig. 2 provides an example of automatically extracted baselines.

TABLE 4
Additional Characters

| character | isolated | end |
|---|---|---|
| Alef Maksura | ى | ـى |
| Teh Marbuta | ة | ـة |

Our approach to baseline extraction uses the algorithm described in [8] with few alterations. It is based on the vertical projection profile obtained by summing pixel values along the horizontal axis for each y word image value. First, the peak corresponding to the maximum of the projection profile curve is determined: The position of the maximum identifies the lower baseline. It is justified by the fact that, in Arabic handwriting, most letters have many pixels on the lower baseline. Then, the algorithm scans the image from top to bottom to find the upper baseline. The position of the upper-baseline corresponds to the position of the first line with a projection value greater than the average row density. Inaccurate baselines may be found in the case of very short words because of the greater influence of diacritical points. It is worth noting that in the IFN/ENIT database most of the words are horizontal: the average orientation of ground-truth baselines is only about 1.36°. Otherwise, words and subwords orientation may be detected using algorithms such as in [28], [19], [39], [17].

### 3.3 Vertical and Slanted Frames

Features are extracted from vertical and slanted windows of fixed width. The sliding window mechanism is applied in

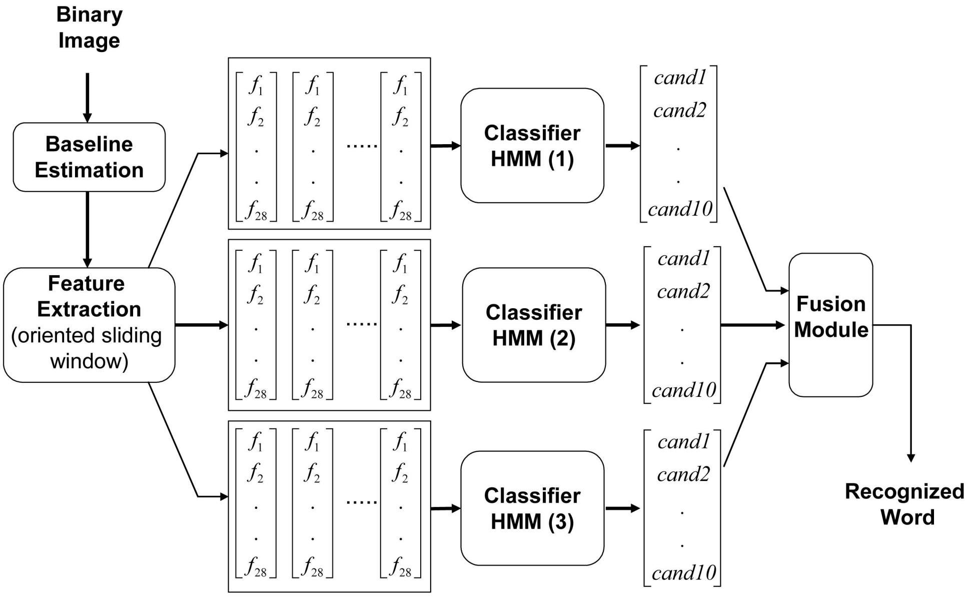


Fig. 1. System overview.

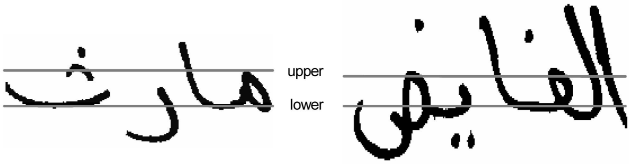


Fig. 2. Upper and lower baselines on sample words.

order to extract features over the whole word image. For the vertical reference system, the image is divided into vertical overlapping windows (or frames) of width $w$ and with overlap parameter $\delta$. The window height depends on each word image and is therefore variable. In order to obtain a fixed vertical dimension, the vertical frame is divided into $\mathrm{n_c}$ cells.

The slanted frames are obtained by slanting the vertical window by an angle $\alpha$, as shown in Fig. 3. Angle $\alpha$ is considered with respect to the vertical axis (Fig. 4). In practice, the image is slanted by the angle $\alpha$ and the vertical windowing above is applied to get the cells in slanted frames. Slanting the image is performed by shifting each image row r by $\Delta_\mathrm{r}$, as shown in Fig. 4. The word image of height H is thus enlarged by $\mathrm{H} * \mathrm{tg}(\alpha)$ to the left or to the right according to the sign of $\alpha$. Finally, different slanting angles have been studied (see Section 5).

### 3.4 Distribution Features

The set of distribution features consists of 16 features that characterize the density of foreground pixels within frames and frame cells. Let $H$ be the height of the frame in an image, $h$ be the variable height of a cell, $w$ be the width of a frame, and $n_c$ be the number of cells in a frame.

Feature $f_1$ is the density of foreground pixels within the frame. Feature $f_2$ is the number of black/white transitions between two consecutive frame cells:

$$f_2 = \sum_{i=2}^{n_c} |b(i) - b(i-1)|,$$

where $b(i)$ is the *density level* of cell $i$. $b(i)$ is equal to one if the cell contains a least one foreground pixel and is equal to zero otherwise.

Feature $f_3$ is a derivative feature defined as the difference between the $y$-coordinate $g$ of the center of gravity of foreground pixels of two consecutive frames $t$ and $t-1$. $g$ is given by

$$g = \frac{\sum_{j=1}^{H} j.r(j)}{\sum_{j=1}^{H} r(j)},$$

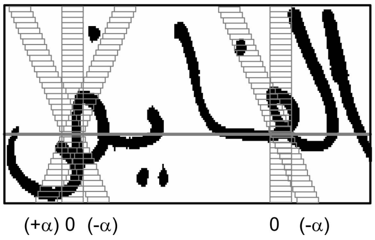


Fig. 3. Word image divided into vertical and slanted frames. The gray line is the lower baseline.

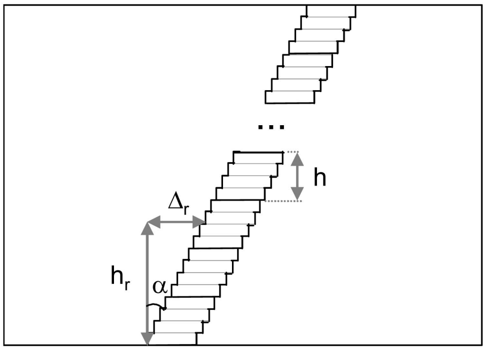


Fig. 4. Angle $\alpha$ and shift $\Delta_\mathrm{r}$ used for building slanted frames and cells.

where $r(j)$ is the number of foreground pixels in the $j\mathrm{th}$ row of a frame.

The eight features $f_4$ to $f_{11}$ represent the densities of black (foreground) pixels for each vertical column of pixels in each frame (in our case, the width of the frame is 8 pixels).

Let $L$ be the position of the lower baseline. Feature $f_{12}$ is the vertical distance from the lower baseline of the center of gravity of foreground pixels, normalized by the height of the frame:

$$f_{12} = \frac{g - L}{H}.$$

Feature $f_{13}$ (respectively, $f_{14}$) represents the density of foreground pixels over (respectively, under) the lower baseline:

$$f_{13} = \frac{\sum_{j=L+1}^{H} r(j)}{H.w}, \quad f_{14} = \frac{\sum_{j=1}^{L-1} r(j)}{H.w}.$$

Feature $f_{15}$ is the number of transitions between two consecutive cells of different density levels above the lower baseline:

$$f_{15} = \sum_{i=k}^{n_c} |b(i) - b(i-1)|,$$

where $k$ is the cell that contains the lower baseline.

Feature $f_{16}$ represents the zone to which the gravity center of black pixels belongs, with respect to the upper and lower baselines. Actually, the two baselines divide a frame into three zones: above the upper baseline ($f_{16} = 1$), a core zone ($f_{16} = 2$), and below the lower baseline ($f_{16} = 3$).

### 3.5 Local Concavity Features

Concavity features are features that provide local concavity information and stroke direction within each frame. Each concavity feature $f_{17}$ to $f_{22}$ represents the (normalized) number of white pixels (background) that belong to six types of concavity configurations. They are explored by using a $3 \times 3$ window, as shown in Fig. 5.

The concavity features are calculated as follows: Let $N_{lu}$, (respectively, $N_{ur}$, $N_{rd}$, $N_{dl}$, $N_v$, and $N_h$) be the number of background pixels that have neighboring black pixels in the following directions: left and up (respectively, up-right, right-down, down-left, vertical, and horizontal). In our implementation, image border pixels are excluded. Thus, the six normalized concavity features are defined as

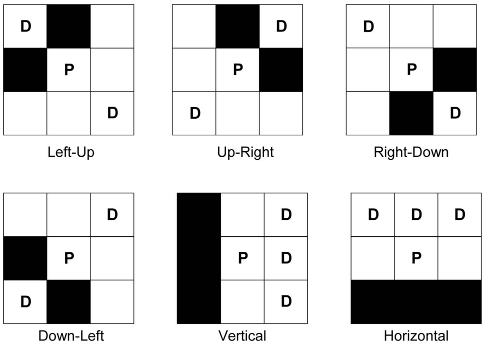


Fig. 5. Six types of concavity configurations for a background pixel P. Pixels marked by D (Don't care) could be either black or white.

$$f_{17} = \frac{\mathrm{N_{lu}}}{H} \quad f_{18} = \frac{\mathrm{N_{ur}}}{H} \quad f_{19} = \frac{\mathrm{N_{rd}}}{H}$$
$$f_{20} = \frac{\mathrm{N_{dl}}}{H} \quad f_{21} = \frac{\mathrm{N_{v}}}{H} \quad f_{22} = \frac{\mathrm{N_{h}}}{H}.$$

By using the information coming from the detection of the two baselines (upper and lower), we generate six new features $f_{23}$-$f_{28}$, describing the concavities in the core zone of a word, that is, the zone bounded by the two upper and lower baselines. Let $CNZ_{lu}$ (respectively, $CNZ_{ur}$, $CNZ_{rd}$, $CNZ_{dl}$, $CNZ_{v}$, and $CNZ_{h}$) be the number of background pixels in the core zone that have neighboring black pixels in the configuration left-up (respectively, up-right, right-down, down-left, vertical, and horizontal).

Thus, the six additional and baseline-dependent concavity features related to the core zone are defined as

$$f_{23} = \frac{\mathrm{CNZ_{lu}}}{d} \quad f_{24} = \frac{\mathrm{CNZ_{ur}}}{d} \quad f_{25} = \frac{\mathrm{CNZ_{rd}}}{d}$$
$$f_{26} = \frac{\mathrm{CNZ_{dl}}}{d} \quad f_{27} = \frac{\mathrm{CNZ_{v}}}{d} \quad f_{28} = \frac{\mathrm{CNZ_{h}}}{d}$$

with $d$ being the distance between the two baselines (upper and lower).

This results in a 28-feature vector per frame, 17 of them are baseline independent ($f_1 - f_{11}, f_{17} - f_{22}$), whereas the 11 remaining ones are calculated with respect to baseline positions. It is to be noted that those features are convenient to any script that can be decomposed into three zones (core, ascending, and descending zones), such as the Latin cursive script.

## 4 HMM-Based Recognition and Classifier Combination

### 4.1 Individual HMM Classifiers

HMM framework is used for word modeling and recognition. Word models are built by concatenating character models, and there is no need to presegment words into characters in both training and classification stages. Each character model has a right-left topology as shown in Fig. 6.

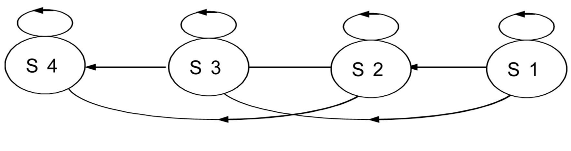


Fig. 6. Right-left HMM (character model).

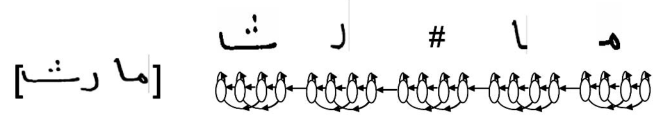


Fig. 7. Word model for Arabic word.

All models share the following characteristics: four states and three transitions per state when possible (a self-transition and two transitions to the next states). The observation probability in each state is described by a mixture of three Gaussian distributions with diagonal covariance matrices. Those characteristics were determined experimentally (see Section 5.1).

Character models are built in a way that takes into account the different shapes of characters according to their position in the word (at beginning, middle, and end word positions). This results in about 100 models. Then, 43 models are built for characters with additional marks such as chadda (ّ). There are also models for digits, additional shapes, and ligature characters, and there is one space model. This results in a total of 167 character/space models. Each entry of the lexicon is modeled by concatenating the appropriate character/space models. A space model, represented by symbol "#", is always inserted between the different words composing a city name. Arabic words may also be composed of several subwords (often called PAWs: Pieces of Arabic Words). When a PAW ends with character Alef, a space model is added between the PAW and the following one as we have noticed that writers often include such space after an Alef in final form (see the Arabic word مارث in Fig. 7). All other spaces are not modeled by a space model.

The HCM toolkit [34] compiles models and performs training and decoding. The training module estimates character-HMM parameters on the training set thanks to the segmental k-means algorithm [37]. During the expectation phase, observations are assigned to the most likely states using the Viterbi algorithm. Then, parameters are reestimated during the Maximization phase. At initialization, observations are assigned to states linearly and parameters are estimated. The training of character models requires no segmentation and is performed through cross training from word images and their transcription (see Fig. 8 for cross-training of Arabic character "ـبـ"). In the recognition phase, the feature sequence extracted from the image is passed to a network of lexicon entries formed of character models. The character sequence providing the maximum likelihood identifies the recognized entry. To determine the most likely sequence of characters, the Viterbi algorithm is used. All HMM-based classifiers in the combined system share the same topology and use the same training and recognition algorithms. However, feature sequences differ for each classifier as frames are of different orientations. This yields different parameters for each classifier and different log-likelihoods are provided for a word under test.

The reference HMM classifier reached state-of-the-art results for Arabic handwritten word recognition. A thorough analysis identified writing inclination as a major

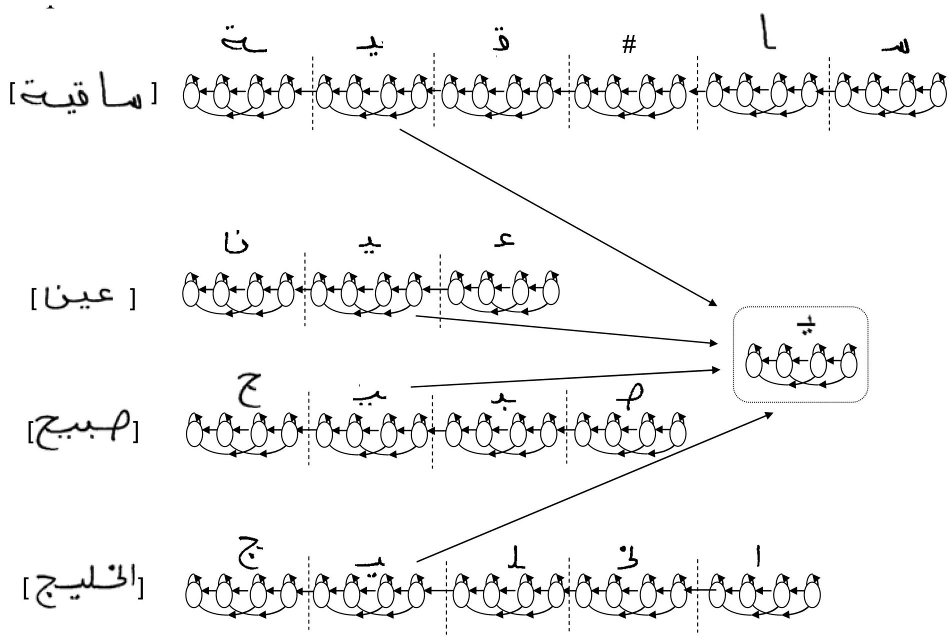


Fig. 8. Cross training of Arabic character through different word images.

source of errors. Overlapping ascenders and descenders, shifted positions of diacritical marks, and horizontal and vertical ligatures were also major reasons for errors. Traditionally, orientation in writing is compensated in a preprocessing stage [19], [17], [25]. In this work, we propose to compensate for slant by the choice of the orientation of the analysis windows. The use of slanted analysis windows can also compensate for the overlapping of ascenders and descenders. In order to avoid the explicit detection of the orientation angle, we propose to perform the modeling using different orientations and to combine the output of the different slanted classifiers in a fusion scheme.

## 4.2 Combination

Combining classifiers is known to be a useful means for increasing performance. There are various combination schemes which are characterized by different architectures (parallel, sequential, and hierarchical) and by the classifiers involved which may use different training algorithms, training samples, feature sets, or initializations [26], [10]. Combination may use scores or class rankings [20], [38] and ends up reranking or rescoring classes.

Here, we have chosen to combine the three HMM-based classifiers presented above at the decision level. This ensemble of classifiers uses different feature sequences, obtained by varying frame orientation as described in Section 3.3. Frame orientation to the left and the right may be symmetric or asymmetrical. Although different, the feature vectors are still homogeneous. The scored lists of candidate words provided by the individual classifiers are used to produce a new combined list.

Below we compare three combination schemes. The first two are the sum rule and the majority vote rule, commonly used for combining classifiers. The third one consists of a neural network-based combining classifier which dynamically selects the correct classifier among the three HMMs. The classifier is trained to output, for each test word, the index of the classifier which best estimates its class.

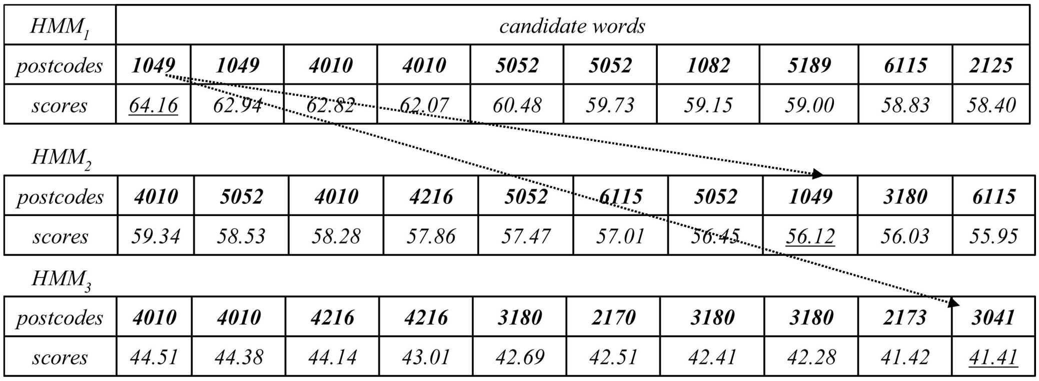
### 4.2.1 Sum and Majority Vote Rules

When a word image is classified by an individual HMM classifier $i$, each candidate word $word\_lex_j$ of the lexicon is given a score $loglik_i(word\_lex_j)$ which reflects the confidence that the candidate word is the word ground-truth class. Scores are the normalized log-likelihoods of the observation sequence extracted on the word image for all models (candidate words). Log-likelihoods are provided by Viterbi decoding, and the normalization factor is the difference between the maximum and minimum log-likelihoods of the training set. Table 5 shows the top-4 candidate words and the top-10th word provided by the individual HMM classifiers for the test word (طبّابة).

The sum rule combination scheme assigns a new score to each candidate word $word\_lex_j$ of the lexicon by summing

TABLE 5
Candidate Word Lists and Scores Provided by Each HMM Classifier for a Test Word

| $HMM_1$ | | $HMM_2$ | | $HMM_3$ | |
|---|---|---|---|---|---|
| candidates | scores | candidates | scores | candidates | scores |
| الدخانيّة | 15.83 | كثّانة | 17.13 | طبّابة | 18.68 |
| طبّابة | 15.51 | طبّابة | 16.96 | كثّا نة | 17.74 |
| الشّابة | 15.29 | الشّابة | 16.87 | الكبّاريّة | 17.22 |
| كثّانة | 14.71 | الدخّانيّة | 16.45 | الشّا بة | 17.01 |
| .<br>. | .<br>. | .<br>. | .<br>. | .<br>. | .<br>. |
| السّاقية | 14.25 | السبّالة | 15.23 | الكبّارة | 15.03 |



| $HMM_1$ | candidate words | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| postcodes | **1049** | **1049** | **4010** | **4010** | **5052** | **5052** | **1082** | **5189** | **6115** | **2125** |
| scores | 64.16 | 62.94 | 62.82 | 62.07 | 60.48 | 59.73 | 59.15 | 59.00 | 58.83 | 58.40 |

$HMM_2$

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| postcodes | **4010** | **5052** | **4010** | **4216** | **5052** | **6115** | **5052** | **1049** | **3180** | **6115** |
| scores | 59.34 | 58.53 | 58.28 | 57.86 | 57.47 | 57.01 | 56.45 | 56.12 | 56.03 | 55.95 |

$HMM_3$

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| postcodes | **4010** | **4010** | **4216** | **4216** | **3180** | **2170** | **3180** | **3180** | **2173** | **3041** |
| scores | 44.51 | 44.38 | 44.14 | 43.01 | 42.69 | 42.51 | 42.41 | 42.28 | 41.42 | 41.41 |

Fig. 9. First three inputs $e_1$-$e_3$ (underlined) for a test word (postcode 4010), obtained from the candidate scored lists output by the three HMM classifiers.

the three scores provided by each classifier. The sum score is defined as

$$score_{SUM}(word\_lex_j) = \sum_{i=1}^{3} \log lik_i(word\_lex_j).$$

In the majority vote rule, we consider the class which is the most represented within the three lists for estimating the top-1 class of a test word.

### 4.2.2 Multilayer Perceptron Combination

Neural networks, in particular multilayer perceptrons, have been used successfully in combination schemes [18], [10], [27], [21]. In this work, we propose a novel way of using multilayer perceptrons in a fusion scheme, where a training set is used to build the neural network-based classifier used for combination. The output scores from the three individual classifiers are used as inputs for this combining classifier. The combining classifier is trained to select the classifier giving the correct class (candidate word) as top-1 choice. Thus, our assumption is that at least one individual classifier gives the correct answer.

The combining classifier is a 9-6-3 network. The nine inputs are scores selected from the outputs of the three individual classifiers as described below. Each of the three network outputs corresponds to one individual classifier.

Let $\mathrm{HMM}_i$, $i = 1, 2, 3$ be the individual HMM classifiers. Let $cand_j^i$ be the candidate word provided by classifier $\mathrm{HMM}_i$ at rank $j$, and let $score_j^i$ be the normalized log-likelihood of the test observation sequence for this candidate word. Let $e$ be the network input vector $e = [e_1\ e_2\ e_3\ \dots\ e_9]$.

The first three network input values are

- $e_1 = score_1^1$,
- $e_2 = score_j^2$ if $\ni j \in \{1, \dots, 10\}$ $cand_j^2 = cand_1^1$ *else* $e_2 = score_{10}^2$, and
- $e_3 = score_j^3$ if $\ni j \in \{1, \dots, 10\}$ $cand_j^3 = cand_1^1$ *else* $e_3 = score_{10}^3$.

This means that $e_1$ is the score of the top-1 candidate word provided by the first classifier ($\mathrm{HMM}_1$). This word may also be recognized by the remaining individual classifiers, at the same or at different ranks. The scores, output by classifiers $\mathrm{HMM}_2$ and $\mathrm{HMM}_3$ for this word candidate, are the inputs $e_2$ and $e_3$, respectively. In case the candidate word is not recognized by the other classifiers within their top-10 candidate word list, the input ($e_2$ or $e_3$) is given a very low value: the least score in the top-10 candidates.

It can be noted that a candidate word (or, equivalently, its postcode) may appear several times within the candidate top-10 list. It is the case when a word is represented in the lexicon by several character shape sequences: The highest ranked candidate word is then considered.

Symmetrically, inputs $e_4-e_6$ (respectively, $e_7-e_9$) follow the same principle. However, we consider the top-1 candidate word of classifier $\mathrm{HMM}_2$ (respectively, $\mathrm{HMM}_3$) instead of the candidate word of classifier $\mathrm{HMM}_1$.

The desired output vector is $d = [d_1\ d_2\ d_3]$ with $d_i = 1$ if $cand_1^i$, $i = 1 \dots 3$ is the ground-truth candidate word, else $d_i = 0$.

Consider the following example: City names are represented in the IFN/ENIT database by their corresponding postcodes. The scores (normalized log-likelihoods) output by each individual classifier for the test word whose postcode is *4010* are shown in Fig. 9.

The input vector $e$ for this test word is the following:

$$e = [\overset{e_1}{64.16}\ \overset{e_2}{56.12}\ \overset{e_3}{41.41}\ \overset{e_4}{59.34}\ \overset{e_5}{62.82}\ \overset{e_6}{44.51}\ \overset{e_7}{44.51}\ \overset{e_8}{62.82}\ \overset{e_9}{59.34}]$$

and the desired output vector $d$ is

$$d = [0\ 1\ 1].$$

$d_2$ and $d_3$ are equal to one since 4010 is the top-1 choice of $\mathrm{HMM}_2$ and $\mathrm{HMM}_3$ and this postcode corresponds to the ground truth.

The training set consists of scores provided by the classifiers on sets *a*, *b*, and *c*. Network convergence is obtained after 120 epochs with the back-propagation algorithm and we use a logistic sigmoid activation function.

## 5 Experiments

We tested all of the classifiers on the benchmark IFN/ENIT database of Arabic city names [35]. This database is the first database of Arabic handwritten words available to the scientific community. It was produced by the Institute for Communications Technology at the Technical University of Braunschweig (IFN) and the "Ecole Nationale d'Ingenieurs de Tunis." The total number of binary images of handwritten Tunisian town/village names is 26,459. Those names were written by 411 writers, and they were labeled according to 946 name classes. Ground truth information is

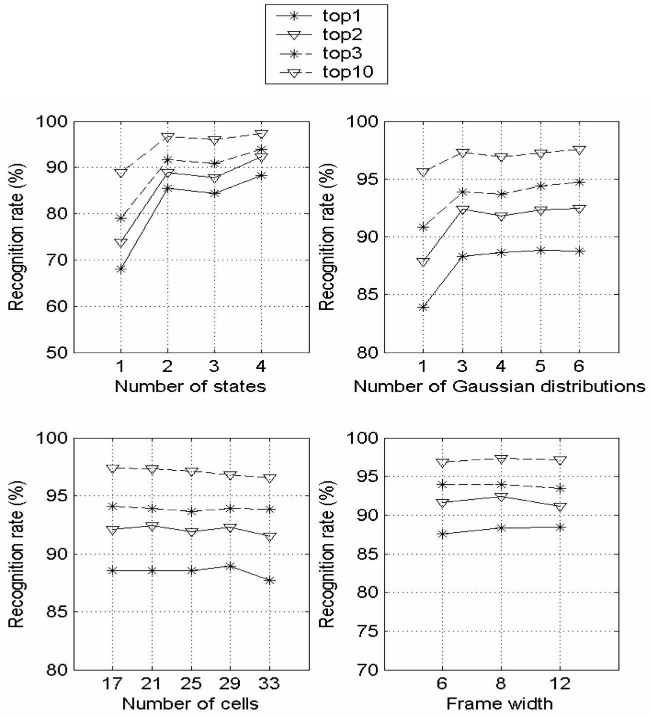


Fig. 10. Recognition rates versus the number of states, the number of Gaussian distributions, the number of cells, and frame width.

added to each entry of the database including character shape sequence and corresponding postcode. More details about ground truth data can be found in [31]. The database is separated into four sets, a, b, c, and d, in order to perform fourfold cross validation experiments.

### 5.1 Effect of System Parameters

We study the effect of varying system parameters such as the number of states, the number of Gaussian distributions associated with each state, the number of cells in a frame, and the width (in pixels) of the frame.

From Fig. 10, it is clear that the system is not very sensitive to the number of Gaussian distributions in a range from 3 to 6, to the number of cells in the range from 17 to 30, and to the frame width between 6 and 12. In the following, we set the number of states for character models to 4, the number of Gaussian distributions to 3, frame width to 8 pixels, the overlap parameter $\delta$ to 4 pixels, and the number of cells to $\mathrm{n_c} = 21$.

### 5.2 Influence of Features

We study the influence of baseline-dependent features on performance. The system considered here is the UOB system [11] presented at the ICDAR '05 competition. The UOB system is the first version of the reference classifier $\mathrm{HMM_1}$: It includes the 24 features, $f_1$-$f_{20}$ and $f_{23}$-$f_{26}$, described in Sections 3.4 and 3.5. The whole set of 24 features is called the $\mathrm{F_w}$ set. The set of 15 baseline-independent features ($f_1$-$f_{11}$ and $f_{17}$-$f_{20}$) is the $\mathrm{F_{bi}}$ set. The UOB system is tested on a restricted number of classes, 450 classes, which include more than eight samples in the training database. Recognition results according to feature sets $\mathrm{F_{bi}}$ and $\mathrm{F_w}$ are given in Table 6. Those results show that the use of both baseline-dependent and independent features (set $\mathrm{F_w}$) increases performance. Actually, the feature set is larger and the baseline-dependent features bring significant additional information.

TABLE 6
Recognition Accuracy (%) of the UOB System Using All Features (Set $\mathrm{F_w}$) or Baseline-Independent Features Only (Set $\mathrm{F_{bi}}$)

| Test set | Feature set | |
|---|---|---|
| | $\mathrm{F_w}$ | $\mathrm{F_{bi}}$ |
| A | 86.10 | 74.80 |
| B | 86.88 | 75.33 |
| C | 85.45 | 74.40 |
| D | 87.20 | 75.41 |

*Lexicon size is 450 (from [11]).*

### 5.3 Comparison with Other Systems

The IFN/ENIT database is available to the scientific community and this makes system comparison possible. The first international Arabic handwritten word competition was held in 2005 in Seoul and was organized by IFN [30]. Five competing groups sent their systems to IFN and systems were tested on the recognition of an unknown set $e$ of 6,033 word images of the IFN/ENIT database.

The five systems in competition were: the ICRA system [1] based on PAWs (pieces of Arabic words) and words in a hierarchical neural network-based combination, the SHOCRAN system from Egypt, the TH-OCR analytical system from [22] of Tsinghua University, the UOB system, and the REAM system [41]. REAM was tested on a reduced set of 3,000 images due to a failure on a full set of images. Results are given in Table 7 for the unknown set $e$ and for the 946 classes. The HMM-based ARAB-IFN system [36] did not participate but results were also given (Table 7). The UOB system reached state-of-the-art performance on this database, slightly higher than the HMM-based IFN system.

### 5.4 Classifier Combination

The third experiment compares the three individual classifiers and the effect of combination. Preliminary results for classifier combination were presented in [12]. We present here complete results including new experiments such as testing a wider range of angles, testing asymmetrical combination, and comparing the combined system

TABLE 7
Results from ICDAR '05 Competition

| | Recognition rate (%) | | |
|---|---|---|---|
| **System** | **Top- 1** | **Top-5** | **Top-10** |
| ICRA | 65.74 | 83.95 | 87.75 |
| SHOCRAN | 35.70 | 51.62 | 51.62 |
| TH-OCR | 29.62 | 43.96 | 50.14 |
| **UOB** | **75.93** | **87.99** | **90.88** |
| REAM | 15.35 | 18.52 | 19.86 |
| ARAB-IFN | 74.69 | 87.07 | 89.77 |

*Test set is $e$ (from [30]).*

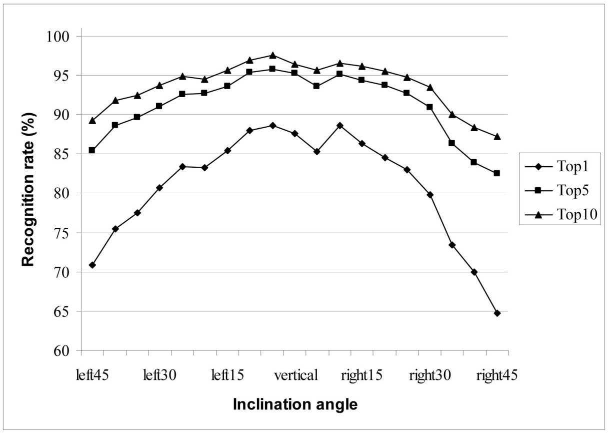


Fig. 11. Recognition rates for individual classifiers with different inclination angles.

with the approach consisting of deslanting words during preprocessing (see Section 5.5). In Fig. 11, the individual performances for individual classifiers working with angles ranging from $-45°$ to $+45°$, including the vertical classifier are shown. In these experiments, we consider all 946 name classes. Training of word and character models are performed on sets $a$, $b$, and $c$, and test on the $d$ set of IFN/ENIT. Comparing the individual classifiers, the reference classifier ($\alpha = 0°$) and the classifiers with weak oriented angles ($5°$ to $10°$) are the best classifiers, which indicates that observing the word image through vertical frames is better than through oriented frames.

In addition, we note that the accuracy of the reference classifier 87.6 percent is improved compared to the accuracy of the UOB system [11] which is only 83.31 percent on set $d$ (considering 946 name classes) because of the added features $(f_{21}, f_{22}, f_{27}, f_{28})$ related to horizontal and vertical configurations.

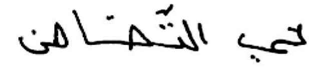
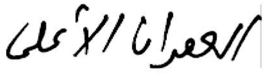
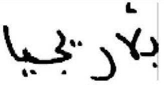
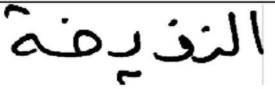
Table 8 provides results in terms of error rates for words and at character level. It is clear that the character error rate is of same order as the word error rate. Most of the errors at the character level are substitution errors followed by insertion errors. The number of omission errors is lower, and omission errors do not increase largely when the inclination angle reaches $45°$.

Individual classifiers using oriented frames are less powerful than the reference classifier, but any combination of the reference classifier with those classifiers brings significant improvement. Table 9 shows words as examples for which the correct class could be estimated by the combination of the three classifiers and not by the reference classifier as an individual classifier. For those words, the correct class is found by at least the classifier oriented in the

TABLE 8
Error Rates for Individual Classifiers Together with Character Error Rates (s: Substitution, o: Omission, i: Insertion)

| | L40° | L30° | L20° | L10° | V0° | R10° | R20° | R30° | R40° |
|---|---|---|---|---|---|---|---|---|---|
| Top1 | 24.5% | 19.3% | 16.8% | 12.0% | 12.4% | 11.4% | 15.5% | 20.2% | 30.1% |
| Top5 | 11.4% | 8.9% | 7.2% | 4.6% | 4.8% | 4.9% | 6.3% | 9.2% | 16.1% |
| Top10 | 8.2% | 6.3% | 5.5% | 3.1% | 3.6% | 3.5% | 4.5% | 6.6% | 11.7% |
| Character Error rate | 26.8% [s 17.9, o 2.1, i 6.8] | 20.7% [s 14.0, o 1.9, i 4.8] | 17.7% [s 12.0, o 1.6, i 4.1] | 12.2% [s 8.4, o 1.2, i 2.6] | 12.4% [s 8.7, o 1.6, i 2.0] | 11.7% [s 8.1, o 1.5, i 2.1] | 16.1% [s 10.7, o 1.2, i 4.2] | 21.7% [s 14.6, o 1.8, i 5.3] | 32.9% [s 22.2, o 2.8, i 7.8] |

TABLE 9
Oriented Words and Results Provided by Individual Classifiers

| oriented words | inclination | HMM$_1$ ($\alpha$=0) | HMM$_2$ ($\alpha$<0) | HMM$_3$ ($\alpha$>0) |
|---|---|---|---|---|
|  | left | incorrect | **correct** | incorrect |
|  | right | incorrect | incorrect | **correct** |
|  | left | incorrect | **correct** | **correct** |
|  | left | incorrect | **correct** | incorrect |

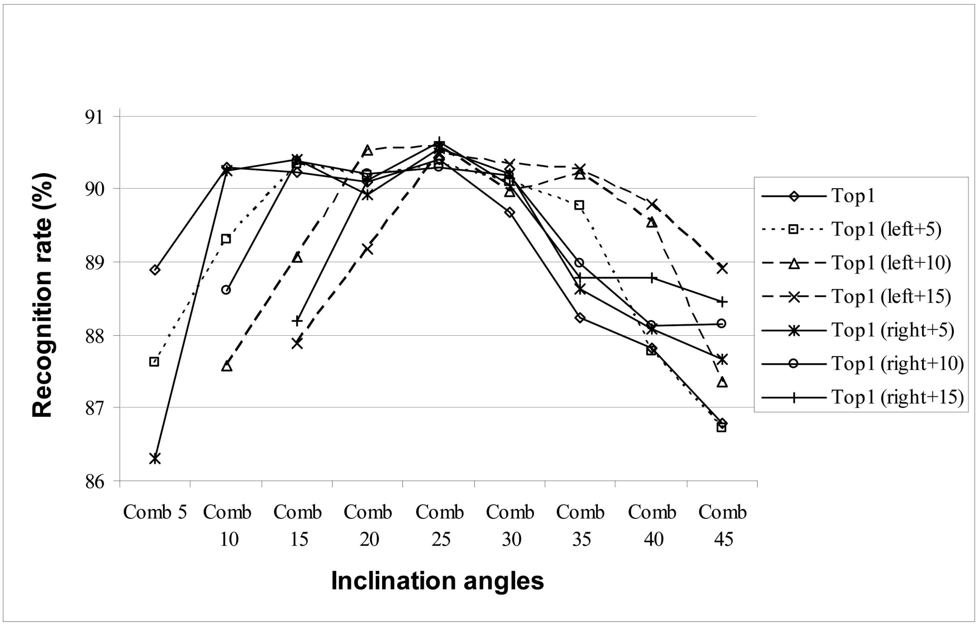


Fig. 12. Recognition rates obtained by combining classifiers with different inclination angles; Top1 $[\alpha - 0 - \alpha]$, Top1 $(\mathrm{left} + \mathrm{x})$ $[\alpha + \mathrm{x} - 0 - \alpha]$, and Top1 $(\mathrm{right} + \mathrm{x})$ $[\alpha - 0 - \alpha + \mathrm{x}]$. Only the Top1 recognition rate is shown.

TABLE 10
Comparative Recognition Results: Reference System $(\mathrm{HMM}_1)$ and Three Combination Schemes

| Classifier | Recognition rate (%) | | |
|---|---|---|---|
| | Top1 | Top2 | Top3 |
| HMM1 ( 0° ) | **87.60** | **91.42** | **93.76** |
| HMM2 (α= + 20 °) | 83.31 | 88.42 | 90.82 |
| HMM3 (α= - 20 °) | 84.59 | 89.28 | 91.76 |
| MLP | **90.96** | **92.95** | **94.44** |
| Sum | **90.61** | **94.89** | **95.87** |
| Majority vote | **90.26** | **94.71** | **95.68** |

*Test subset is $d$. Number of classes is 946.*

same direction, left or right, as the true word inclination. This illustrates the advantage of using oriented frames for slanted words.

Before comparing the combination strategies, a study is performed on the classifier angles to be considered in the combination. At first, we combine the left and right classifiers for a given angle with the vertical classifier. This leads to a combination with symmetric angles and the Top1 graph of Fig. 12. Then, a difference between the left classifier angle and the right classifier angle is applied. Recognition performance for the different scenarios is plotted in Fig. 12. The sum combination is used in this experiment.

From the results shown in Fig. 12, it is obvious that the combination of different classifiers significantly improves performance. For a given scenario, performance seems to be stable for a large interval of inclination angles. For example, for the basic scenario where two classifiers, left and right, use the same angle, the performance is stable for an interval ranging from 10° to 30°. For angle values less than 10°, the classifiers with oriented frames provide close information to the one provided by the classifier with vertical frame. For angle values higher than 30°, the classifiers with oriented frames perform poorly and therefore do not help a lot when combined with the classifier with vertical frames. When taking different angles for the left and right-oriented classifiers, the same behavior is observed. Good and stable performance is observed within a wide range of orientation angles. However, this range tends to be reduced when increasing the difference between the frame orientations of the two classifiers, which would be expected. Since the basic combination yields a good stability property, we maintain it for the following experiments.

Concerning the character error rate, we noticed that, for the combined systems, the character error rate is also consistent with the word error rate as seen for individual systems.

Comparing combination strategies, the MLP combination outperforms the sum rule and majority vote combinations for the Top1 choice as shown in Table 10. The accuracy for the Top2 and Top3 choices of the MLP combination is lower than that of the other combinations because the MLP is trained to select only one classifier and its Top1 answer. The sum rule performs better than the majority vote combination.

The system based on MLP combination yields the highest accuracy among all the other systems which indicates that observing word images through both the vertical and the oriented frames is the most efficient. This is

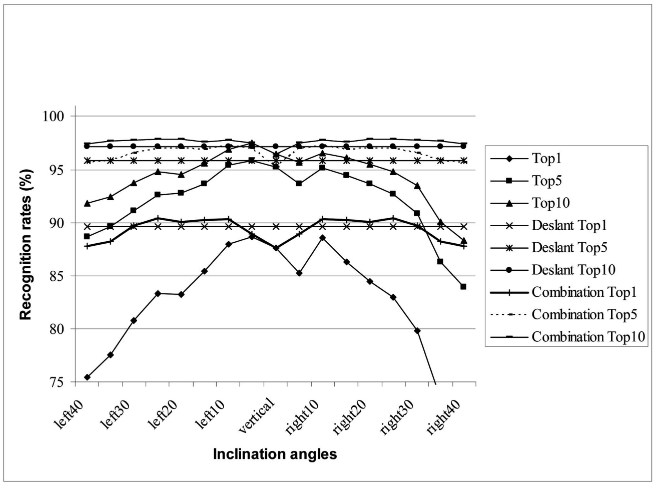


Fig. 13. Recognition rates for different frame inclination angles, for the individual classifiers, the combined classifiers, and the reference classifier after word-slant correction.

due to the fact that the two systems with oriented frames $(+\alpha, -\alpha)$ can cope with slanted strokes and with shifted positions of diacritical marks and dots. Such shifted positions are very common in Arabic handwritten words. As a consequence, characters are modified or blank spaces are occluded.

To summarize, the use of baseline-dependent/independent features observed through both vertical and oriented frames in the context of the HMM framework is efficient for the recognition of Arabic city names and reaches state-of-the-art performance. The experiments conducted show that oriented classifiers with weak inclination angles provide similar performance to the reference system. For larger inclination angles, the classifiers offer complementary information that helps to improve performance significantly. Moreover, the improvement of performance seems to be stable over a large range of orientation angles and for different combination strategies. This proves that the approach is robust and may be generalized to other applications or data.

### 5.5 Comparison with Slant Correction

The final set of experiments aims at comparing the present approach with the one which consists of using a single classifier dealing with slant-corrected images as a preprocessing step. We use the KSC slant correction method proposed in [25]. The word contour is searched and the average slant is estimated through chain elements of direction 45°, 90°, and 135°.

The classifier using vertical frames and slant-corrected images is compared to the individual recognizers and to the sum combination strategy for different inclination angles. Results are given in Fig. 13. From those results, it is obvious that deslanting images during preprocessing yields improved performance for the reference system. However, the combined system outperforms the classifier using slant-corrected images for all the operational combination angles and for all angles if we consider the Top5 and Top10 results. This demonstrates the advantage of the proposed combined approach.

Another experiment has also been conducted consisting of selecting manually the correct answer from all individual classifiers. This leads to a recognition rate of 98.2 percent (and thus, an error rate of 1.8 percent). Therefore, improving the combination (inclination selection) strategy will certainly increase system performance even compared to the combined system.

## 6 Conclusion

In this paper, we have shown the efficiency of baseline-dependent and baseline-independent features observed through vertical and oriented frames for the recognition of Arabic handwritten city names. The method reported here is HMM-based and benefits from both holistic recognition and analytical word modeling. A thorough analysis of the errors of the reference system showed that writing orientations are major sources of errors. The overlapping ascenders and descenders, as well as the shifted positions of diacritical marks, are also sources of errors. Therefore, we have proposed coping with such errors by using a slanted window analysis. To avoid the estimation of the orientation angle, we propose to use different classifiers, each of them corresponding to a possible slanted angle analysis window.

Three individual classifiers are combined at the decision level. Each classifier observes the image with a given orientation. We show that combining those classifiers significantly increases accuracy and that this accuracy is higher than using a single classifier dealing with slant-corrected images. It is also clear from the experimental results that the proposed method is robust for a wide range of orientation angles. The best combination scheme is obtained with a novel combining MLP classifier which dynamically selects the correct classifier. The accuracy ratio reaches a recognition rate higher than 90 percent on the $d$ set of the IFN/ENIT benchmark database. This combination has substantially improved previous state-of-the-art results.

Future work will involve the use of a rejection rule to detect ambiguous words. For these words, finer analysis may be done including checking the number and position of dots, or searching for individual word baselines. Combining this system with other existing ones using a similar approach but different types of features may also increase accuracy. Finally, the upper bound for the performance that could be achieved by manually selecting the well-oriented individual classifier is higher than 98 percent. Therefore, optimizing the combination strategy is a major future goal.

## Acknowledgments

The authors wish to thank the reviewers for their fruitful comments. The authors also wish to acknowledge the SARIMA program (http://www.irisa.fr/sarima) for partially supporting this research.

## References

[1] A. Abdulkader, "Two Tier Approach for Arabic Offline Handwriting Recognition," *Proc. Int'l Workshop Frontiers in Handwriting Recognition,* pp. 42-47, 2006.

[2] H. Almuallim and S. Yamaguchi, "A Method of Recognition of Arabic Cursive Handwriting," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 9, pp. 715-722, 1987.

[3] A. Amin, "Off-Line Arabic Character Recognition: The State of the Art," *Pattern Recognition,* vol. 31, no. 5, pp. 517-530, 1998.

[4] A. Amin, H. Al-Sadoun, and S. Fischer, "Hand-Printed Arabic Character Recognition System Using an Artificial Network," *Pattern Recognition,* vol. 29, no. 4, pp. 663-675, 1996.

[5] N. Arica and F.T. Yarman-Vural, "Optical Character Recognition for Cursive Handwriting," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 6, pp. 801-813, June 2002.

[6] I. Bazzi, R. Schwartz, and J. Makhoul, "An Omnifont Open Vocabulary OCR System for English and Arabic," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 6, pp. 495-504, June 1999.

[7] N. Ben Amara and F. Bouslama, "Classification of Arabic Script Using Multiple Sources of Information: State of the Art and Perspective," *Int'l J. Document Analysis and Recognition,* vol. 5, pp. 195-212, 2003.

[8] M. Blumenstein, C.K. Cheng, and X.Y. Liu, "New Pre-Processing Techniques for Handwritten Word Recognition," *Proc. Second IASTED Conf. Visualization, Imaging and Image Processing,* pp. 480-484, 2002.

[9] M. Dehghan, K. Faez, M. Ahmadi, and M. Shridhar, "Handwritten Farsi (Arabic) Word Recognition: A Holistic Approach Using Discrete HMM," *Pattern Recognition,* vol. 34, no. 5, pp. 1057-1065, 2001.

[10] R. Duin, "The Combining Classifier: To Train or Not to Train," *Proc. 16th Int'l Conf. Pattern Recognition,* 2002.

[11] R. El-Hajj, L. Likforman-Sulem, and C. Mokbel, "Arabic Handwriting Recognition Using Baseline Dependent Features and Hidden Markov Modeling," *Proc. Eighth Int'l Conf. Document Analysis and Recognition,* pp. 893-897, 2005.

[12] R. El-Hajj, C. Mokbel, and L. Likforman-Sulem, "Combination of HMM-Based Classifiers for the Recognition of Arabic Handwritten Words," *Proc. Ninth Int'l Conf. Document Analysis and Recognition,* 2007.

[13] T.S. El-Sheikh and R.M. Guindi, "Recognition of Isolated Arabic Characters," *Signal Processing,* vol. 14, pp. 177-184, 1988.

[14] T.S. El-Sheikh and R.M. Guindi, "Computer Recognition of Arabic Cursive Scripts," *Pattern Recognition,* vol. 21, no. 4, pp. 293-302, 1988.

[15] M.S. El-Wakil and A. Shoukry, "On-Line Recognition of Handwritten Isolated Arabic Characters," *Pattern Recognition,* vol. 22, pp. 97-105, 1989.

[16] A. El-Yacoubi, M. Gilloux, R. Sabourin, and C.Y. Suen, "An HMM-Based Approach for Off-Line Unconstrained Handwritten Word Modeling and Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 8, pp. 750-752, Aug. 1999.

[17] A. El-Yacoubi, R. Sabourin, M. Gilloux, and C. Suen, "Off-Line Handwritten Word Recognition Using Hidden Markov Models," *Knowledge-Based Intelligent Techniques in Character Recognition,* L.C. Jain and B. Lazzerini, eds., pp. 191-229, CRC Press, 1999.

[18] N. Farah, M.T. Khadir, and M. Sellami, "Artificial Neural Network Fusion: Application to Arabic Words Recognition," *Proc. 13th European Symp. Artificial Neural Networks,* 2005.

[19] F. Farooq, V. Govindaraju, and M. Perrone, "Pre-Processing Methods for Handwritten Arabic Documents," *Proc. Eighth Int'l Conf. Document Analysis and Recognition,* pp. 267-271, 2005.

[20] T.K. Ho, J. Hull, and S.N. Srihari, "Decision Combination in Multiple Classifier Systems," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, no. 1, pp. 66-75, Jan. 1994.

[21] Y. Huang and C. Suen, "A Method of Combining Multiple Experts for the Recognition of Unconstrained Handwritten Numerals," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 1, pp. 90-94, Jan. 1995.

[22] J. Jin, H. Wang, X. Ding, and L. Peng, "Printed Arabic Document System," *Proc. IST/SPIE Electronic Imaging Conf.,* 2005.

[23] M.S. Khorsheed, "Recognizing Arabic Manuscripts Using a Single Hidden Markov Model," *Pattern Recognition Letters,* vol. 24, pp. 2235-2242, 2003.

[24] H.J. Kim, S.K. Kim, K.H. Kim, and J.K. Lee, "An HMM-Based Character Recognition Network Using Level Building," *Pattern Recognition,* vol. 30, no. 3, pp. 491-502, 1997.

[25] F. Kimura, M. Shridhar, and Z. Chen, "Improvements of a Lexicon Directed Algorithm for Recognition of Unconstrained Handwritten Words," *Proc. Second Int'l Conf. Document Analysis and Recognition,* pp. 18-22, 1993.

[26] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On Combining Classifiers," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 3, pp. 239-266, Mar. 1998.

[27] D.S. Lee and S. Srihari, "A Theory of Classifier Combination: The Neural Network Approach," *Proc. Third Int'l Conf. Document Analysis and Recognition,* pp. 42-45, 1995.

[28] L.M. Lorigo and V. Govindaraju, "Segmentation and Pre-Recognition of Arabic Handwriting," *Proc. Eighth Int'l Conf. Document Analysis and Recognition,* pp. 605-609, 2005.

[29] L.M. Lorigo and V. Govindaraju, "Off-Line Arabic Handwriting Recognition: A Survey," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 5, pp. 712-724, May 2006.

[30] V. Märgner, M. Pechwitz, and H. El-Abed, "Arabic Handwriting Word Recognition Competition," *Proc. Eighth Int'l Conf. Document Analysis and Recognition,* pp. 70-74, 2005.

[31] V. Märgner, H. El-Abed, and M. Pechwitz, "Offline Handwritten Arabic Word Recognition Using HMM: A Character-Based Approach without Explicit Segmentation," *Proc. Colloque Int'l Francophone sur l'Ecrit et le Document,* 2006.

[32] H. Miled, C. Olivier, M. Cheriet, and Y. Lecourtier, "Coupling Observation/Letter for a Markovian Modelization Applied to the Recognition of Arabic Handwriting," *Proc. Fourth Int'l Conf. Document Analysis and Recognition,* pp. 580-583, 1997.

[33] M.A. Mohamed and P.D. Gader, "Handwritten Word Recognition Using Segmentation-Free Hidden Markov Modeling and Segmentation-Based Dynamic Programming Techniques," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 5, pp. 548-554, May 1996.

[34] C. Mokbel, H. Abi Akl, and H. Greige, "Automatic Speech Recognition of Arabic Digits over the Telephone Network," *Proc. Research Trends in Science and Technology,* 2002.

[35] M. Pechwitz, S.S. Maddouri, V. Märgner, N. Ellouze, and H. Amiri, "IFN/ENIT-Database of Handwritten Arabic Words," *Proc. Colloque Int'l Francophone sur l'Ecrit et le Document,* pp. 129-136, 2002.

[36] M. Pechwitz and V. Märgner, "HMM-Based Approach for Handwritten Arabic Word Recognition Using the IFN/ENIT-Database," *Proc. Seventh Int'l Conf. Document Analysis and Recognition,* pp. 890-894, 2003.

[37] L.W. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE,* vol. 77, no. 2, pp. 257-285, 1989.

[38] A. Rahman and M. Fairhurst, "Multiple Classifier Decision Combination Strategies for Character Recognition: A Review," *Int'l J. Document Analysis and Recognition,* vol. 5, pp. 166-194, 2003.

[39] Y.H. Tay, P.M. Lallican, M. Khalid, C. Viard-Gaudin, and S. Knerr, "An Off-Line Cursive Handwritten Word Recognition System," *Proc. IEEE Region 10 Int'l Conf. Electrical and Electronic Technology,* 2001.

[40] *The Unicode Consortium,* The Unicode Standard, Version 5.0, Addison-Wesley, 2006.

[41] S.M. Touj and N. Ben Amara, "Arabic Handwritten Words Recognition Based on a Planar Hidden Markov Model," *Int'l Arab J. Information Technology,* vol. 2, no. 4, pp. 318-325, 2005.

[42] A. Vinciarelli and J. Luettin, "Off-Line Cursive Script Recognition Based on Continuous Density HMM," *Proc. Int'l Workshop Frontiers in Handwriting Recognition,* pp. 493-498, 2000.

[43] A. Yardeni, *AventureLettres.* Association Alphabets, Mar. 2004.

**Rami Al-Hajj Mohamad** received the BS degree in applied mathematics and the DEA degree in mathematical modeling and intensive computation from the Lebanese University, Beirut, in 1999 and 2001, respectively, and the PhD degree in signal and image processing from the Ecole Nationale Supérieure des Télécommunications (ENST)-Paris in 2007. He is currently an associate professor in the Department of Computer Sciences and Information Technology at the Lebanese International University, Beirut. His research interests include the automatic recognition of handwriting, pattern recognition, neurocomputing, and artificial intelligence. He and his coauthors won first place at the ICDAR '05 Competition on Arabic handwritten word recognition in Seoul.

**Laurence Likforman-Sulem** received the degree in engineering from the Ecole Nationale Supérieure des Télécommunications (ENST)-Bretagne in 1984 and the PhD degree from ENST-Paris in 1989. She is an associate professor in the Department of Signal and Image Processing, TELECOM ParisTech, where she serves as a senior instructor in pattern recognition and document analysis. Her research interests include document analysis dedicated to handwritten and historical documents, document image understanding, and character recognition. She is a founding member of the francophone Groupe de Recherche en Communication Ecrite (GRCE), an association for the development of research activities in the field of document analysis and writing communication. She chaired the program committee of the last Conference Internationale Francophone sur l'Ecrit et le Document (CIFED) held in Fribourg, Switzerland, in 2006 and the program committee of the Document Recognition and Retrieval (DRR) Conference in 2009 in San Jose. She is a member of the IEEE.

**Chafic Mokbel** received the degree in electronic engineering from the Lebanese University in 1988, the DEA (MSc) degree in electronic systems from the Institut National Polytechnique de Grenoble (INPG), France in 1989, and the PhD degree in signal processing from the Ecole Nationale Supérieure des Télécommunications-Paris in 1992. His PhD research focused on speech recognition in adverse environments. He studied and proposed several techniques covering speech enhancement, HMM adaptation by spectral transformation, as well as Lombard effects. He then joined the Centre National des Etudes des Télécommunications (CNET), where he participated in and led several projects concerning the robustness of speech recognition systems and the application of these systems in real-life telecommunication services. In 1999, he joined IDIAP, Switzerland, where he was the speech group leader. In 2000, he joined Jinny Software, Dublin, Ireland, where he worked on the development of telecommunications services, mainly for the mobile environment and developed a voicexml gateway. Since 2001, he has been with the University of Balamand, Lebanon. As an associate professor, his research activities cover different domains of speech, language, handwriting, and video processing. From 1996 to 1998, he was an associate editor of *Speech Communication*. He is senior member of the IEEE and the IEEE Signal Processing Society.