Pattern Recognition 47 (2014) 3477–3486

Contents lists available at ScienceDirect

Pattern Recognition

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

Arabic word descriptor for handwritten word indexing and lexicon reduction

Youssouf Chherawala *, Mohamed Cheriet [1]

Synchromedia Laboratory, École de Technologie Supérieure, 1100 Notre-Dame Ouest, Montreal, QC, Canada

ARTICLE INFO

Article history:
Received 30 May 2013
Received in revised form
10 March 2014
Accepted 27 April 2014
Available online 14 May 2014

Keywords:
Arabic word descriptor
Shape indexing
Holistic representation
Lexicon reduction
Arabic handwritten documents
IFN/ENIT
Ibn Sina database

ABSTRACT

Word recognition systems use a lexicon to guide the recognition process in order to improve the recognition rate. However, as the lexicon grows, the computation time increases. In this paper, we present the Arabic word descriptor (AWD) for Arabic word shape indexing and lexicon reduction in handwritten documents. It is formed in two stages. First, the structural descriptor (SD) is computed for each connected component (CC) of the word image. It describes the CC shape using the bag-of-words model, where each visual word represents a different local shape structure, extracted from the image with filters of different patterns and scales. Then, the AWD is formed by sorting and normalizing the SDs. This emphasizes the symbolic features of Arabic words, such as subwords and diacritics, without performing layout segmentation. In the context of lexicon reduction, the AWD is used to index a reference database. Given a query image, the reduced lexicon is obtained from the labels of the first entries in the indexed database. This framework has been tested on Arabic word databases. It has a low computational overhead, while providing a compact descriptor, with state-of-the-art results for lexicon reduction on the Ibn Sina and IFN/ENIT databases.

1. Introduction

Arabic word recognition is an active field of research [1–6]. Most word recognition systems (WRS) use a lexicon, which is made up of a set of accepted words, to limit their output to valid words. The recognition rate is improved by testing all the lexicon word hypotheses, although this is achieved at the expense of a loss of recognition speed. A processing time as long as 4 s for a single word [7,8], even in competitive Arabic WRS, is not acceptable in an industrial context. Lexicon reduction methods have been developed to alleviate this problem, which dynamically reduce the lexicon based on the input images. Unfortunately, the reduction process is prone to error, in which it may discard the true label of an input image. If this happens, not only does the accuracy decrease, but also the WRS will not recover the true label. The sources of error are the same as those for word classifiers, which are affected by the handwriting variability [9] of individuals, and even of a single individual, and the level of degradation of the documents, which is typically high in historical texts [10]. Lexicon reduction methods must manage the difficult trade-off between reducing the size of a lexicon and maintaining a high level of accuracy on the retained word hypotheses. In other words, these methods must improve the WRS processing speed without decreasing its recognition rate. In addition, a successful lexicon reduction system must be efficient to compute, in order to minimize its impact on the WRS processing speed, and it should capture discriminative lexicon word shape features to provide good performance.

Unlike Latin script, Arabic script is written from right to left, and the alphabet is composed of 28 letters instead of 26 (Fig. 1). The shape of the letters is dependent on their position in the word, and is usually different if they are at the beginning, middle, or end of a word. Six letters (‘ء’, ‘D’, ‘R’, ‘Z’, and ‘W’) can be connected only if they appear in a final position; if they appear in initial or medial position, a space is inserted after them and the word is broken into subwords. Several letters share the same base shape and are only distinguishable by diacritics in the form of one, two, or three dots appearing above or below the shape. The features of Arabic words are illustrated in Fig. 2.

The problem of lexicon reduction was initially investigated for Latin script. The simplest method is based on the length of the word, as it allows discrimination between short and long words. The most common feature extracted from a word image is a sequence of ascenders and descenders [11]. The sequence is matched against features extracted from synthetic images of

* Corresponding author.
E-mail addresses: ychherawala@synchromedia.ca (Y. Chherawala), mohamed.cheriet@etsmtl.ca (M. Cheriet).
[1] Tel.: +1 5143968972; fax: +1 5143968595.

http://dx.doi.org/10.1016/j.patcog.2014.04.025
0031-3203/

ص	ش	س	ز	ر	ذ	د	خ	ح	ج	ث	ت	ب	ا
Ṣ	Š	S	Z	R	Ḏ	D	Ḫ	Ḥ	Ǧ	Ṯ	T	B	ʾ
ي	و	ه	ن	م	ل	ك	ق	ف	غ	ع	ظ	ط	ض
Y	W	H	N	M	L	K	Q	F	Ġ	ʿ	Ẓ	Ṭ	Ḍ

Fig. 1. Arabic letters with their ISO 233 transliteration.

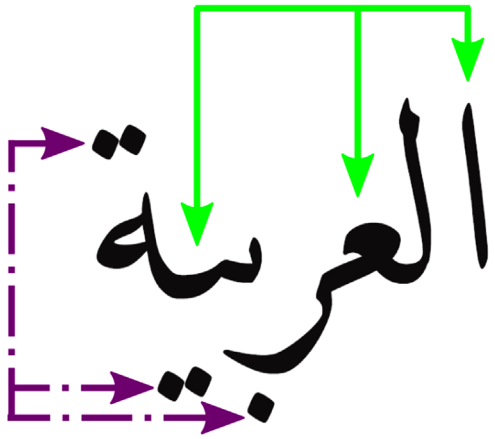

Fig. 2. An Arabic word with its subwords (solid lines) and diacritics (dashed lines).

words in the lexicon, using regular expressions [12] or the string edit distance [13]. Then, lexicon reduction is performed by discarding the unmatched lexicon entries. More advanced features are often used in combination with an analytic classifier. Zimmermann and Mao [14] form a regular expression from key characters, which represent an unambiguous recognition of a character-level classifier. Bertolami et al. [15] propose an HMM based on shape code models, where each shape code represents multiple letters. Then, a list of regular expressions is obtained from the top ranked shape code sequences of the HMM.

Research on lexicon reduction has been given new impetus in recent years with the increasing interest in Arabic script [16]. Novel methods are being built, based on the specificities of Arabic words, and they can be classified into two groups. One group of methods considers only the diacritic information and subword counts, ignoring the subword shape. Mozaffari et al. [17] proposed the first of these methods, in which the lexicon is pruned based on the estimated number of subwords, and then the diacritics are categorized according to their types (1, 2, or 3 dots) and their positions relative to the base shape (above or below); finally, a sequence of diacritics is formed and matched against synthetic models of the remaining lexicon words. The diacritic categorization step has since been improved by Wshah et al. [18], thanks to a better estimation of their positions and the use of a convolutional neural network to recognize their type. The other group of methods considers the subword shape, and is based on the skeleton image. Chherawala and Cheriet [19] propose a spectral method for indexing skeleton shapes, where the skeleton is modeled as a weighted graph using topological and geometrical features. Lexicon reduction is then performed by indexing a reference database of subword shapes and selecting the labels of the top ranked database entries. Asi et al. [20] propose a hierarchical organization of subword skeleton shapes, where the bottom layer represents the original shapes and the top layer their coarse representations. The shapes of a given layer are simplified and then clustered to form the next level. Given a query shape, the lexicon is reduced by traversing the hierarchy in a top-down fashion and by skipping the less promising clusters.

In this paper, we propose to represent the shape of Arabic words using the Arabic word descriptor (AWD). It encodes the shape of the image connected components (CCs) while emphasizing the symbolic features of Arabic words, such as subwords and diacritics.

A structural descriptor (SD) is used to encode the shape of each CC, based on the bag-of-words (BOW) model [21], which has been successful for image retrieval and classification [22–25], as well as for shape matching [26]. A set of pattern filters representing different patterns at different scales is used to extract local features, called the pixel descriptor (PD), for each foreground pixel of the CC skeleton image. The PDs are assigned to their nearest visual word from a predefined codebook of the feature space. The SD is then formed as a histogram representing the number of occurrences of each visual word. The SD is well suited for lexicon reduction, because it allows efficient shape matching by vector comparison. Finally, the AWD is formed by sorting and normalizing the SDs of all the CCs. It incorporates information about the shape and the count of the subwords and diacritics into a single vector, without performing any word layout analysis. In the context of lexicon reduction, the AWD is used to index a reference database of word shapes. The labels of the top ranked database entries form the reduced lexicon. We show the AWD's high performance for lexicon reduction with low computational overhead.

This paper is an extension of the work published in [27]. In particular, the extension of the methodology includes a larger set of filters for image feature extraction. The experimental evaluation has also been significantly improved, by combining lexicon reduction with word recognition tasks.

The rest of this paper is organized as follows: Section 2 explains the pixel descriptor concept. Section 3 describes the structural descriptor formation. Section 4 explains the Arabic word descriptor formation. Section 5 gives an overview of the lexicon reduction system. Section 6 presents our experimental results.

2. Pixel descriptor

The pixel descriptor (PD) is a feature vector which describes the local shape structure. It is computed on the skeleton image, which highlights the shape structure. Note that only the skeleton pixels are considered, as they provide the most information on the shape of the word. The PD is formed from the output of various image filters, called the pattern filters. We first describe the pattern filters and PD formation, and then we provide a structural interpretation of the PD.

2.1. Pattern filters and pixel descriptor formation

Pattern filters are designed to detect specific structural patterns at a given scale around each skeleton pixel of the skeleton image. We assume that the skeleton image I is binary, having skeleton pixels with a value of 1 and background pixels with a value of 0. For computational efficiency, we have chosen rectangular filters, because they can be efficiently computed using the integral image [28]. We define a family of five patterns to describe the local structure of skeleton images. The patterns comprise a square and four lines of orientations: 0, 45, 90, and 135° (Fig. 3). The square filter is the most isotropic, given the rectangular filter constraint. All the filters are square windows of width w, which also represents their scales, with masked areas to form the pattern. The square pattern has no mask, the 0° and 90° lines are masked by two rectangles of size $w \times w/4$, while the 45° and 135° lines are masked by two squares of size $w/2 \times w/2$. The patterns are similar to the Haar-like features, the difference is that the value of masked area is ignored instead of being subtracted.

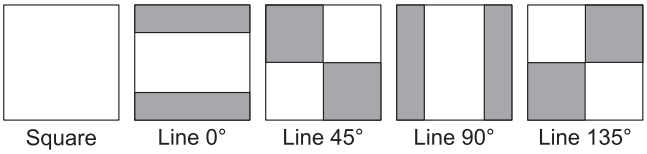

Fig. 3. Pattern filters. The gray areas are masked.

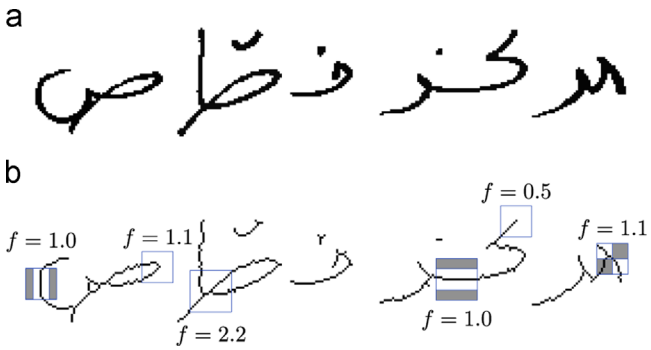

Fig. 4. Response of pattern filters. (a) Original word shape. (b) Response of various pattern filters on the skeleton image.

Each pattern filter defines a specific neighborhood around the skeleton pixel and it counts the number of skeleton pixels falling inside their patterns. The output of the filter is normalized by the filter scale. Considering the filter as an image, the value of the pixels of the pattern area is 1, and the value of the pixels of the masked area is 0. The output f of a filter F of scale w at a pixel of position (x,y) of the skeleton image I is given by

$$f = \frac{1}{w_0} \sum_{0 \le i,j < w} F(i,j) \cdot I\left(x + i - \left\lfloor \frac{w}{2} \right\rfloor, y + j - \left\lfloor \frac{w}{2} \right\rfloor\right) \quad (1)$$

where $F(i,j)$ and $I(i,j)$ are the values of F and I at the position (i,j) respectively, and $\lfloor \cdot \rfloor$ represents the floor function. The values outside the bounds of I are considered to be 0.

The PD is then formed from the concatenation of the output of n pattern filters $\mathrm{PD} = [f_1 \ldots f_i \ldots f_n]^T$, where f_i is the output of the filter F_i. All the filters F_i are unique, and are differentiated either by their patterns or by their scales. Each filter provides a different insight into the pixel neighborhood. The PD is therefore a signature of the local structure surrounding the pixel.

2.2. Structural interpretation

The outputs of the pattern filters composing the PD provide a geometrical and topological interpretation of the skeleton pixels (Fig. 4). When the response of a line filter is close to 1 for a given skeleton pixel, a local skeleton curve has the same orientation as the filter. The case of the square filter is more interesting. A response close to 1 indicates that the skeleton pixel belongs to a simple curve structure (curve with no self-intersection), while responses that are significantly smaller and bigger are indicators of pixels in the neighborhood of end points and branch points respectively. Therefore, the square filter is an indicator of the local skeleton topology. All these considerations only hold on the condition that the filter scale is small enough to not be perturbed by spatially close structures.

3. Structural descriptor

The structural descriptor (SD) is a feature vector describing word shapes. It is based on the BOW model, which represents the distribution of image features extracted at selected keypoints. The skeleton shape image is considered to highlight the shape topology and the geometry. All the skeleton pixels are considered as keypoints, as it has been shown that dense sampling provides better results for lexicon reduction [29]. Given a set of pattern filters, a set of PDs $\{\mathrm{PD}_1 \ldots \mathrm{PD}_p\}$ is extracted from the skeleton image, where p is the number of skeleton pixels in the image. In order to build the codebook, the entries of the PDs are normalized to zero mean and unit variance across the reference database. The PD is quantized using the k-means algorithm, which outputs k pixel prototypes, representing the skeleton image visual words. The SD of a given skeleton image is built by first assigning each of its PD to the visual word of its nearest prototype and then forming a histogram from the number of occurrences of each visual word in the image. The process of the formation of the SD is shown in Fig. 5 and summarized in Algorithm 1.

We consider that the SDs are embedded in the vector space $\mathbb{R}^k$. Using the Euclidean metric, the complexity for computing the distance between two SDs is $O(k)$. Given two structural descriptors SD_1 and SD_2, representing two different shapes, where $\mathrm{SD}(i)$ represents the ith entry of the descriptor, the distance between the two SDs is

$$\sqrt{\sum_{i=1}^{k} (\mathrm{SD}_1(i) - \mathrm{SD}_2(i))^2} \quad (2)$$

The distance is commensurate with the number of unmatched visual words. This approach is therefore similar to the pairing of similar substructures between two skeleton images. It is adapted to the description of Arabic word shapes, as the vector quantization provides some tolerance to handwriting variability.

Algorithm 1. Structural descriptor computation.

Input: shape image; pattern filter set E; database pixel descriptor statistics (mean and standard deviation); k-means prototypes $\{P_i\}$
Output: Structural descriptor
Compute the shape skeleton
Compute the pixel descriptor for each skeleton pixel using E
Normalize the pixel descriptors with the database statistics and assign them the visual word of their nearest P_i
Form the SD as the histogram of the visual words in the skeleton image

4. Arabic word descriptor

The SD is holistic which means that it considers the image as a whole. This approach fails to incorporate symbolic information related to the various units forming the Arabic words, i.e. the subwords and the diacritics. In this section, we adapt the SD to the Arabic word descriptor (AWD), which further integrates information on the subword counts and diacritics. We assume that the subwords and the diacritics correspond to CCs of the image. First, the SD of each CC is computed. Then, like the idea introduced in [19] that was never developed, the SDs are sorted in the descending order with respect to the number of pixels in their respective CC skeleton. This ordering is expected to rank the largest subwords first and the diacritics last. The sorted descriptors are then concatenated into the Arabic word descriptor $\mathrm{AWD} = [\mathrm{SD}_1 \ldots \mathrm{SD}_c]^T$, where c is the number of CCs in the image and $\{\mathrm{SD}_i\}$ are the sorted CC descriptors ($1 \le i \le c$) – see Fig. 6 for an illustration. This ordering has three main advantages:

- It avoids the difficult task of explicit classification of the CCs into diacritics or subwords, as confusion arises with single letter subwords [18].

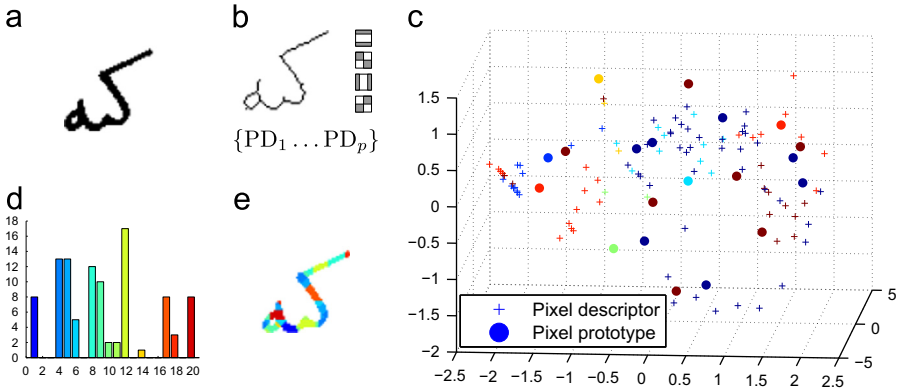

Fig. 5. Formation of the structural descriptor. (a) Shape image. (b) Set of extracted pixel descriptors, given the skeleton image and a set of pattern filters. (c) Assignment of each pixel descriptor to the visual word of its nearest pixel prototype. (d) Structural descriptor: histogram of the occurrence of visual words. (e) Illustration of the structure encoded by each visual word on the original shape, the shape pixels are shown with the color of their pixel prototypes (for clarity, the original shape image is shown, instead of the skeleton image). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

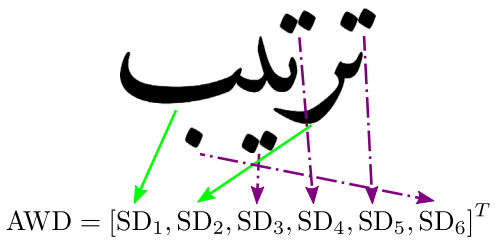

Fig. 6. Construction of the Arabic word descriptor (AWD)—see text for details.

- It avoids spatial ordering of the CCs, which is also a difficult problem because Arabic subwords can overlap each other horizontally, and the vertical ordering for the diacritics is based on the estimation of the baseline, which is a problem in itself [30].
- It is more tolerant to changes in topology, such as touching, broken or missing CCs like diacritics, as in most cases these modifications have a relatively small impact on the number of pixels in the original CC.

As the SDs are sorted, the first entries of the AWD will be more prominent than the last entries. In order to give equal importance to the subwords and the diacritics, all the AWD entries are normalized to have zero mean and unit standard deviation. The AWD size is set to contain m SDs. If the number of CCs in an image is smaller than m, the AWD is padded with zeros (absence of CCs). Otherwise, it is truncated.

5. Lexicon reduction system

5.1. System overview

The lexicon reduction system is based on the shape indexing. A reference database is composed of word shape images with their corresponding labels L_i. The set of labels contained in the database forms the application lexicon, and so each lexicon word must be represented by at least one image. The more the images there are per lexicon word, the better the modeling of handwriting variability. This database is processed by computing the AWD for each of its images, given a set of pattern filters for local feature extraction. During the lexicon reduction phase, the system takes a word image segmented from the original document as input. First, the AWD of the query word is computed, and it is compared to the AWDs in the reference database in the AWD vector space. Then, the reference database entries are sorted in the ascending order, according to their distance from the query word AWD. The reduced lexicon is finally obtained by considering the labels of the first max_{rank} entries of the sorted database, where max_{rank} is a parameter provided to the system. Then, the reduced lexicon is fed to the word recognition system. This lexicon-reduction system is illustrated with images segmented at the subword level in Fig. 7.

5.2. Performance measure

When a query word is submitted to a lexicon-reduction system, two criteria are important to assess its performance. The first is accuracy, with value 1 if the reduced lexicon contains the true label of the query word, otherwise 0, in which case the WRS is bound to fail. The second is the lexicon-size reduction, which is expressed as $1-R/L$, where L is the size of the original lexicon and R is the size of the reduced lexicon. If we consider accuracy and reduction as random variables over a test dataset, their expected values are noted as the accuracy of reduction α and the degree of reduction ρ respectively. A system with an accuracy of reduction and a degree of reduction that are both close to 1 achieves good performance. However, it is difficult to optimize α and ρ at the same time, as a high degree of reduction increases the chances that the true label will be discarded. The reduction efficacy $\eta = \alpha \cdot \rho$ is also used as a unified measure. In this case, a lexicon-reduction system is evaluated using α, ρ, and η [13].

6. Experiments

6.1. Databases

We evaluate our approach on two Arabic word databases. The first is the Ibn Sina database [31], which is based on a commentary on an important philosophical work by the Persian scholar Ibn Sina (Fig. 8). It contains 60 pages from a manuscript copied by a single writer, and is labeled at the subword level. This represents approximately 25,000 subword images and 1200 different classes using archigrapheme label encoding [19], which ignores diacritic information. The first 50 pages are used for the evaluation of our lexicon reduction system. The second is the IFN/ENIT database [32], which contains the names of Tunisian cities and villages (Fig. 9). Approximately 400 writers participated in its creation. It is labeled at the word level, and contains 26,459 word images representing 946 classes. It is composed of five sets (A, B, C, D, and E), the first four being used for the evaluation of our lexicon reduction system. As the image resolution is high in this database,

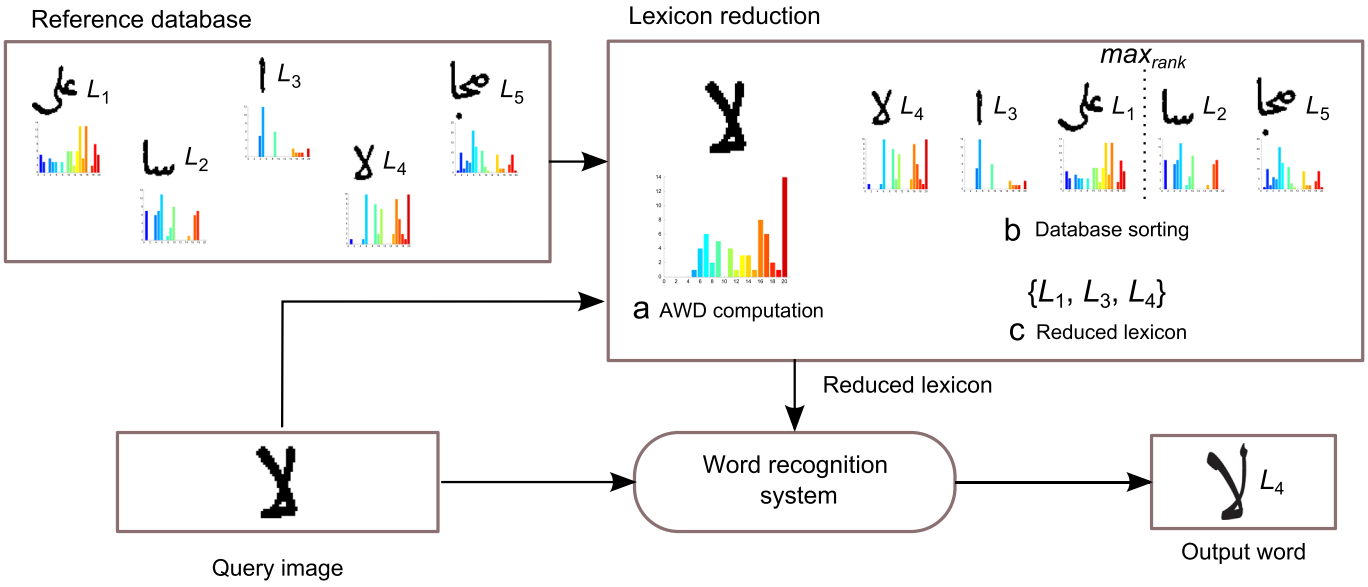

Fig. 7. Lexicon reduction system overview.

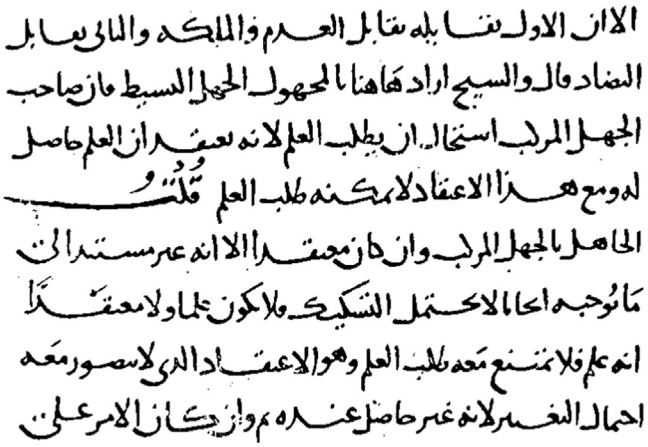

Fig. 8. Text sample from a page of the Ibn Sina database.

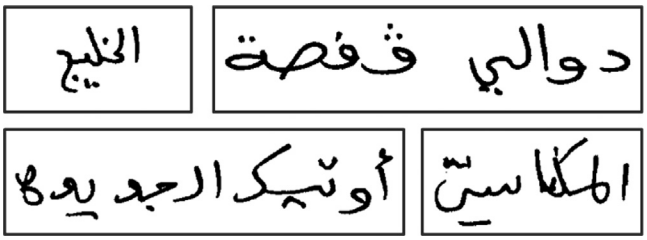

Fig. 9. Sample words from the IFN/ENIT database.

it has been decreased by 2 to improve the processing speed. Also it brings both databases approximately to the same scale.

6.2. Experimental protocol

The skeleton image is obtained with the thinning algorithm of MATLAB. Then, a set of 40 pattern filters is used for the feature extraction, containing the five patterns (square and lines at 0, 45, 90, and 135°) each at eight different scales (5, 9, 15, 21, 25, 31, 41, 51). The largest scale is bigger than the average size of the database subwords. No feature selection algorithm has been used because the total number of pattern filters is small. Therefore, the only free parameters of the system are m, the maximum number of CCs in the AWD, and k, the number of pixel prototypes. The choice of m is guided by the level of segmentation of the database. For the Ibn Sina database, labeled at the subword level with archigraphemes encoding, the AWD is built from only one CC ($m=1$). This setting allows us to focus on the subword body and implicitly ignore the diacritics. For the IFN/ENIT AWD, m is set to 20, in order to take into account all the subwords and most of the diacritics, even for large words. For the choice of k, different values have been tested on a subset of the database (results not shown). For Ibn Sina, the values {10, 20, 30, 40, 50, 60, 70, 80} were considered, and we obtained the best results for 60 but with only a slight improvement of performance over 50. We therefore favored the simplest model among these two and chose $k=50$. For IFN/ENIT, the values {1, 5, 10, 15} were considered. These values are smaller than that for Ibn Sina, in order to limit the total size of the AWD ($k \times m$). The best results were obtained for $k=5$. For the construction of the SD, the seeds of the k-means clustering are initialized using the k-means++ algorithm [33].

Our framework is evaluated by cross validation. The whole database is split into 10 folds for the evaluation (outer folds), where the folds are each considered successively as the test database and the remaining folds form the reference database of the system. The results on the outer folds are averaged, in order to provide a measure of performance on the whole database.

The experiments are performed on a computer with a 2.3 GHz AMD Phenom(tm) 9600B Quad-Core processor, 4 GB of RAM and Windows 7 Enterprise as OS. The code is single threaded, and has been implemented as MATLAB scripts, except the feature extraction with pattern filters which has been implemented in C++. The processing times are given for this configuration.

6.3. Lexicon reduction performance

The results of the lexicon reduction performance on both databases are shown in Fig. 10. The degree of reduction ρ is plotted for different accuracies of reduction α. The degree of reduction remains high, even for $\alpha > 70\%$, and then it drops quickly with a large standard deviation as α approaches 100%. Detailed results are shown in Table 1 for specific reduction accuracies. The system performs better on the Ibn Sina database

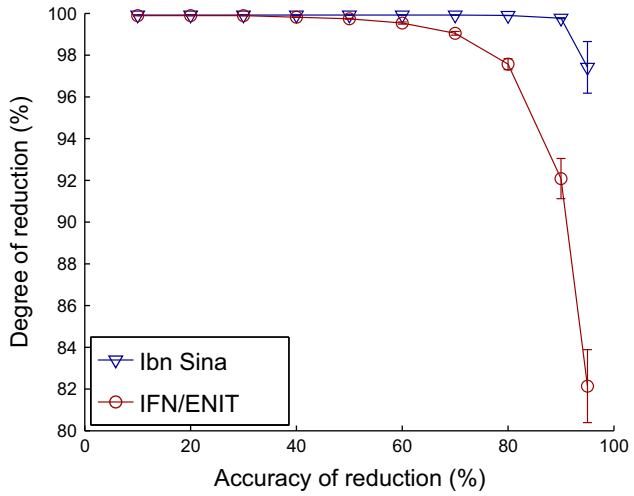

Fig. 10. Lexicon reduction performance.

Table 1
Lexicon-reduction performance on the Ibn Sina and IFN/ENIT databases.

α (%)	Ibn Sina		IFN/ENIT	
	ρ (%)	η (%)	ρ (%)	η (%)
90.0	99.8 ± 0.0	89.8	92.1 ± 1.0	82.9
95.0	97.4 ± 1.2	92.6	82.1 ± 1.8	78.0

than on the IFN/ENIT one. In particular, for α up to 70%, $max_{rank} = 1$ on the Ibn Sina database which means that the SD would achieve a recognition rate of 70% by considering the label of its nearest neighbor. Some results of the reference database indexing are shown in Fig. 11.

Representative pixels of visual words are shown in Fig. 12 (figure best viewed by zooming on a computer screen). A different color is assigned to each visual word. Note that the pixels are clustered according to their topology and geometry, with a different color for branch points and end points, as well as for different orientations.

The proposed approach produces compact descriptors. The AWD is a 50D vector for the Ibn Sina database, and a 100D vector for the IFN/ENIT database. Also, the computational overhead is relatively small. The average processing time for each word, from the raw image to the formation of the AWD, is 7.0 ms on the Ibn Sina database and 14.0 ms on IFN/ENIT. The average time of lexicon reduction for each query word against the full database is 5.0 ms on Ibn Sina and 6.7 ms on IFN/ENIT.

6.4. Analysis of the ADW formation steps

The AWD formation relies on two main steps, sorting the SDs and normalizing its entries. In this section, we analysis the relevance of these two steps for lexicon reduction. First, we compare the proposed sorting approach, based on the number of pixels of each CC's skeleton, against three simple approaches based on the position of each CC. These approaches sorts the CCs from right to left based on three different criteria: the right end, the left end, and the centroid horizontal position of each CC. All the formed descriptors are compared with and without normalization. The IFN/ENIT database is used for this experiment. The database is separated into 10 folds, nine folds form the reference database and the last fold the test database. The results are shown in Table 2 for target accuracies of reduction of 90% and 95%. We first notice that for all the sorting approaches, the normalization increases the degree of reduction.

Fig. 11. Database indexing based on the AWD. For each row, the first element is the query word image, while the remaining images are the first elements of the sorted reference database. The elements sharing the same label as the query are surrounded by a solid line box. (a) Ibn Sina database. (b) IFN/ENIT database.

Fig. 12. Visual words on Ibn Sina and IFN/ENIT databases. The original word image (black) and its partition into visual words, where each color corresponds to one visual word. For clarity, the partition is shown on the original image instead of the skeleton, using the visual word color of the nearest skeleton pixel. (a) Ibn Sina database. (b) IFN/ENIT database. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Among the position based sorting approaches, the best results are obtained for the right end criterium. This is certainly linked to the fact that Arabic is written from right to left. For $\alpha = 90\%$ and with

Table 2
Comparison of different AWD steps for lexicon reduction.

Sorting approach	Norm.	ρ (%) ($\alpha = 0.90$)	ρ (%) ($\alpha = 0.95$)
Right end		91.8	73.7
	✓	93.6	77.3
Left end		83.6	58.2
	✓	88.6	62.7
Centroid		85.2	63.4
	✓	90.8	67.5
Num. pixel		87.3	72.3
	✓	92.3	83.8

Table 3
Lexicon reduction influence on a holistic word recognition system on the Ibn Sina test set.

max_{rank} (%)	α (%)	ρ (%)	Classifier recognition rate (%)	Avg. proc. time (ms)
100	100	–	86.2	6376
15	93.5	85.1	86.2	893
10	93.4	87.9	86.1	608
5	93.0	92.4	85.9	320
1	91.6	98.0	85.6	85
0.1	87.9	99.7	83.3	35

descriptor normalization, the right end approach is slightly better (1.4%) than the proposed approach based on the number of pixels. Nevertheless, for $\alpha = 95\%$, it is clearly outperformed (6.5%) by the proposed approach. It shows that sorting based on the number of pixel is more robust than position based criteria for high accuracy of reduction.

6.5. Combination with a holistic word recognition system

The chosen holistic WRS performs recognition at the subword level. Each subword is described by the square-root velocity (SRV) representation [34]. The subword contour is projected on a Riemmanian manifold, where it is represented as a sequence of velocity points normalized by their square-root velocity. This representation is invariant to scaling and rotation, but the rotation invariance is removed for this application. The SRV is also tolerant to elastic deformations, which often occur during the handwriting process. The dynamic programming algorithm from [35] is used for optimal SRV sequence alignment. The subwords are classified using a nearest neighbor classifier (1-NN) with the SRV metric. This system has been implemented in C++, and is evaluated on the Ibn Sina database, with the first 50 pages used as the reference database and the remaining 10 pages as the test set. The pixel prototypes are computed on the reference database, and then the SDs are formed for the whole database. During recognition, lexicon reduction is implicitly performed by ignoring all the reference database entries with a rank larger than max_{rank} in the indexed database.

The recognition rate, along with the actual degree of accuracy and degree of reduction on the test set, as well as the average processing time per subword, is shown in Table 3 for different values of max_{rank}, expressed here as a percentage of the size of the reference database. The value of max_{rank} goes from 0.1% of the reference database up to 100%, which corresponds to the case where the WRS is run without lexicon reduction. We can see that, as max_{rank} decreases, the accuracy of reduction and the classifier recognition rate both decrease, while the degree of reduction increases. A high accuracy of reduction is achieved with max_{rank} as small as 1% of the reference database, for a system 75 time faster and a drop in the recognition rate of just 0.6% compared to the classifier with the full lexicon.

The speed improvement is commensurate with max_{rank}, as only the entries ranked below it are considered during the nearest neighbor search. The computation of a single SRV distance is 0.26 ms, and the matching against large databases takes several seconds. Therefore, database indexing is needed for fast Arabic handwriting recognition using shape analysis methods, such as the SRV or the shape context [36]. In the general case of holistic WRS, where there are as many word models as there are entries in the lexicon, the speed improvement is commensurate with the degree of reduction.

Table 4
Lexicon reduction influence on an analytic word recognition system on the IFN/ENIT set E.

max_{rank} (%)	α (%)	ρ (%)	Classifier recognition rate (%)	Avg. proc. time (s)
100	100	–	88.1	4.7
15	95.5	45.5	84.9	3.9
10	92.5	55.8	82.6	3.8
5	85.9	70.3	77.7	3.5
1	64.7	89.7	60.5	2.8
0.1	32.6	98.2	31.6	1.4

6.6. Combination with an analytic word recognition system

The analytic word recognition system is based on the well known HMM. We implemented the system proposed in [37] which we first describe. A set of 16 concavity features is extracted from the word image using the sliding window approach. The frame width is of six pixels, and there is an overlap of three pixels between consecutive frames. The delta and acceleration features are also computed, leading to a total of 48 features for each frame. An HMM model with six emitting states and a mixture of 64 Gaussians per state are trained for each symbol of the alphabet. The word-level HMM is built by concatenating the HMMs of the symbols forming the word. During the recognition, all the word-level HMMs of the lexicon are tested and the word hypothesis having the highest likelihood is chosen as the recognized word. We used the HTK [38] implementation of HMM to build this system. It has been trained on the sets A, B, C, and D of the IFN/ENIT database, and tested on the set E.

Here as well, the pixel prototypes are computed from the training sets and then the AWDs are formed for the whole IFN/ENIT database. The lexicon is dynamically reduced using our approach for different values of max_{rank}. The results are shown in Table 4. We see that the accuracy of reduction drops progressively with respect to max_{rank}. Therefore, it is harder to achieve a high performance for both the accuracy of reduction and the degree of reduction. A good compromise between the classifier recognition rate and the average processing time per image is achieved by considering $max_{rank} = 15\%$, for a drop of the recognition rate of 3.2% for a speed improvement of approximately 20%, compared to the WRS with a full lexicon.

6.7. Combination with a dense descriptor

The AWD preserves only to a small extent the image spatial coherence because it is based on the BOW model and the SD sorting ignores spatial relations. Therefore, we combine in this section the AWD with a descriptor preserving the spatial

coherence to alleviate this shortcoming. Descriptors based on the dense sampling are eligible for this purpose [39], and we chose the HOG descriptor [40]. It decomposes the image into small regions called the cells, and a histogram of gradient orientation is computed for each cell, with normalization over larger areas called the blocks. The AWD and the HOG descriptor are combined using a weighted concatenation: $[\beta \cdot \mathrm{HOG}^T, (1-\beta) \cdot \mathrm{AWD}^T]$, where $\beta = [0, 1]$ is the weight parameter.

In order to compute the HOG descriptor, all the database images are first centered inside a fixed size image to align them. Also, the HOG descriptor has one free parameter, which is the cell size. It controls the scale at which the descriptor is extracted, and it has a direct influence on the length of the descriptor; a small cell size leads to a large descriptor which has important memory requirements for large databases. Different values have been tested on a subset of the databases (results not shown). For Ibn Sina, the values $\{5, 10, 15, 20, 25, 30, 35\}$ were considered, and we obtained the best result for the value 30. For IFN/ENIT, the values $\{20, 25, 30, 35, 40, 45, 50\}$ were considered, and we obtained the best result for the value 20. The lexicon reduction performance of the AWD and HOG combination for $\alpha = 0.95$ and for different values of β is shown in Fig. 13. First, we observe that the performance of the HOG descriptor alone ($\beta = 1$) is similar to that of the AWD on Ibn Sina, with $\rho = 0.96$. Nevertheless, on IFN/ENIT, the HOG performance is better than that of the AWD, with $\rho = 0.89$. This result highlights the importance of spatial coherence. Nevertheless, the AWD is complementary to the HOG descriptor because it implicitly includes symbolic information. The best degree of reduction is obtained for $\beta = 0.9$ on Ibn Sina ($\rho = 0.98$) and for $\beta = 0.75$ on IFN/ENIT ($\rho = 0.93$), which is a significant improvement. Global descriptors have the advantage to preserve the spatial coherence, but they increase the memory requirement, the HOG descriptor length is of 465 and 2790 for the Ibn Sina and IFN/ENIT databases respectively. We used the VLFeat implementation of the HOG descriptor [41].

6.8. Comparison with other methods

The proposed method has been compared with other available approaches (Table 5). The ideal diacritic matching method extracts a sequence of diacritics directly from the subword label and reduces the lexicon by removing unmatched sequences. Therefore, it represents an upper bound for all the methods based only on diacritic matching on the Ibn Sina database, as there is no error in the sequence extraction process. The sparse descriptor and the Arabic word descriptor come from the earlier version of this work [27], where only a single square pattern filter is used. The other methods were briefly detailed in Section 1. Our method combined with the HOG descriptor shows the best reduction efficacy on both databases. Furthermore, only the descriptor-based methods (AWD, HOG) are competitive at both the subword and the word level. Note that, because a training set and a testing set were not clearly defined in the previous experimental protocols, we used cross validation to estimate our system parameters. Our protocol is therefore slightly different from the one used in previous methods, but we believe that the results are comparable.

Table 5
Comparison with other lexicon-reduction methods.

Database	Method	α (%)	ρ (%)	η (%)
Ibn Sina	Ideal diacritics matching	100	75.0	75.0
	W-TSV [19]	90.0	92.9	83.6
	Sparse descriptor [27]	90.0	95.2	85.7
	HOG [40]	95.0	96.2	91.4
	Proposed method	**95.0**	**97.4**	**92.6**
	Proposed method + HOG	**95.0**	**98.4**	**93.5**
IFN/ENIT	Subword and diac. [17]	74	92.5	68.5
	Improved diacritics [18]	94.6	85.6	81.0
	W-TSV [19]	90.0	83.6	75.2
	Arabic word desc. [27]	90.0	90.1	81.1
	HOG [40]	90.0	97.6	87.8
	Proposed method	**90.0**	**92.1**	**82.9**
	Proposed method + HOG	**95.0**	**93.7**	**89.0**

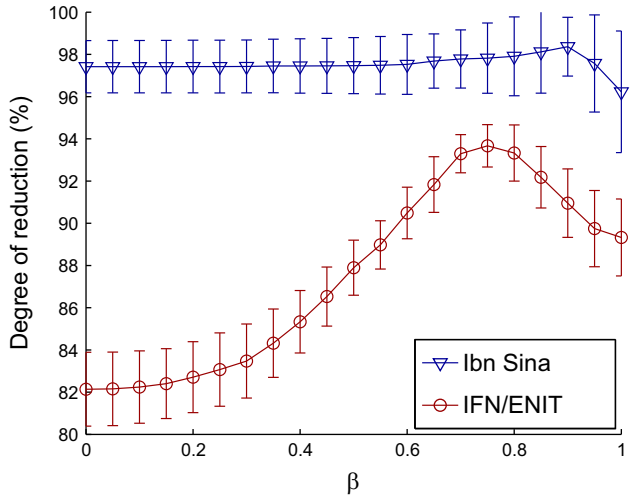

Fig. 13. Lexicon reduction performance of the AWD and the HOG descriptor combination for different values of β. Results shown for $\alpha = 0.95$.

Also, we discuss the computational cost of the proposed algorithm with the other methods. First, we discuss the lexicon reduction strategies. We distinguish between the methods based on subword counts and diacritics [17,18] and those based on shape descriptors [19,27] such as the proposed method. The first kind of methods has a complexity of $O(L)$ for lexicon pruning based on the subword count, and $O(L' \cdot M \cdot M')$ for lexicon reduction using the string edit distance, where L' is the size of the pruned lexicon from the first step, M the average lexicon string length and M' the query image string length. The total complexity is $O(L + L' \cdot M \cdot M')$. The descriptor-based approaches have a complexity of $O(N \cdot D)$ for the nearest-neighbors search in the reference database, based on the Euclidean distance, where D is the size of the descriptor database and N the size of the descriptor, and $O(max_{rank})$ to retrieve the distinct labels of the retrieved nearest neighbors. The total complexity is $O(N \cdot D + max_{rank})$. Theoretically, methods based on the nearest-neighbors search with the Euclidean metric have a lower computational cost than those based on a string edit distance. The drawback of the latter has been alleviated by first pruning the lexicon entries with a less costly approach, such as pruning based on the subword count.

We now compare the feature extraction cost for the methods based on shape descriptors. The W-TSV has a complexity of $O(F)$ for the thinning algorithm, where F is the number of foreground pixels of the image, $O(p)$ for the skeleton keypoints detection, where p is the number of pixels of the skeleton, $O(t|V|(|E| + |V|\log|V|))$ to build the DAG with the Dijkstra algorithm, where $|E|$ is the number of edges of the DAG and $|V|$ its number of vertices, and $O(|E|)$ for the descriptor extraction from the DAG. The total complexity for the W-TSV computation is $O(F + p + |V|(|E| + |V|\log|V|) + |E|)$. The AWD has a complexity of $O(F)$ for the thinning algorithm, $O(n \cdot p)$ for the n pattern filters application and finally $O(k \cdot n \cdot p)$ for the descriptor computation from the pixels descriptors. The total complexity is $O(F + n \cdot p \cdot (1 + k))$. The AWD is a less

abstract descriptor than the W-TSV. Therefore, its complexity is more dependent on the number of skeleton pixels than the W-TSV. Nevertheless, it provides a greater flexibility of representation through the different number of filters and pixel prototypes. The complexity of the HOG descriptor is $O(c)$, where c is the number of cells of the image.

7. Conclusion

In this work, we proposed an Arabic word descriptor for word indexing and lexicon reduction. It encodes the shape of each connected component of the image through a structural descriptor (SD) based on the bag-of-words model. The sorting and the normalization of the SDs emphasize the symbolic features of Arabic words, such as the subwords and the diacritics. Experiments on Arabic word databases demonstrate the suitability of the AWD for lexicon reduction, thanks to its computation efficiency and high accuracy of reduction. In future work, the AWD will be combined with complementary shape representations, in order to improve its performance for very high accuracy of reduction, and spatial constraints will be directly added as features. In order to reduce the impact of the errors introduced by the lexicon reduction system, a rejection mechanism will be added at the output of the word recognition system. The broader scope of this work is to reduce the processing time of individual word recognition systems, so that multiple word recognition systems can be run efficiently, in order to improve the recognition accuracy by combining their outputs.

Conflict of interest

None declared.

Acknowledgments

The authors thank the NSERC (CRSNG RGPIN 138344) and SSHRC (CRSH 412-2010-1007 (via McGill)) of Canada for their financial support and the reviewers for their constructive and helpful suggestions.

References

[1] L.M. Lorigo, V. Govindaraju, Offline Arabic handwriting recognition: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 712–724.
[2] R. Al-Hajj Mohamad, L. Likforman-Sulem, C. Mokbel, Combining slanted-frame classifiers for improved HMM-based Arabic handwriting recognition, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2009) 1165–1177.
[3] A. Giménez, A. Juan, Embedded Bernoulli mixture HMMs for handwritten word recognition, in: Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR '09), 2009, pp. 896–900.
[4] V. Märgner, H. El Abed, ICDAR 2011—Arabic handwriting recognition competition, in: Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR '11), 2011, pp. 1444–1448.
[5] F. Slimane, S. Kanoun, H. El Abed, A. M. Alimi, R. Ingold, J. Hennebert, ICDAR 2011—Arabic recognition competition: multi-font multi-size digitally represented text, in: Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR '11), 2011, pp. 1449–1453.
[6] P. Dreuw, D. Rybach, G. Heigold, H. Ney, RWTH OCR: A Large Vocabulary Optical Character Recognition System for Arabic Scripts, Springer, London, UK, 2012, pp. 215–254. ISBN 978-1-4471-4071-9.
[7] V. Märgner, H. El Abed, ICDAR 2009 Arabic handwriting recognition competition, in: Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR '09), 2009, pp. 1383–1387.
[8] V. Märgner, H. El Abed, ICFHR 2010—Arabic handwriting recognition competition, in: Proceedings of the 12th International Conference on Frontiers in Handwriting Recognition (ICFHR '10), 2010, pp. 709–714.
[9] J. Park, An adaptive approach to offline handwritten word recognition, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 920–931.
[10] R. Hedjam, R. Farrahi Moghaddam, M. Cheriet, A spatially adaptive statistical method for the binarization of historical manuscripts and degraded document images, Pattern Recognit. 44 (2011) 2184–2196.
[11] S. Carbonnel, E. Anquetil, Lexicon organization and string edit distance learning for lexical post-processing in handwriting recognition, in: Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition (IWFHR '04), IWFHR '04, IEEE Computer Society, Washington, DC, USA, 2004, pp. 462–467.
[12] S. Palla, H. Lei, V. Govindaraju, Signature and lexicon pruning techniques, in: Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR '04), IWFHR '04, IEEE Computer Society, Washington, DC, USA, 2004, pp. 474–478.
[13] S. Madhvanath, V. Krpasundar, V. Govindaraju, Syntactic methodology of pruning large lexicons in cursive script recognition, Pattern Recognit. 34 (2001) 37–46.
[14] M. Zimmermann, J. Mao, Lexicon reduction using key characters in cursive handwritten words, Pattern Recognit. Lett. 20 (1999) 1297–1304.
[15] R. Bertolami, C. Gutmann, H. Bunke, A.L. Spitz, Shape code based lexicon reduction for offline handwritten word recognition, in: Proceedings of the Eighth IAPR International Workshop on Document Analysis Systems (DAS '08), 2008, pp. 158–163.
[16] S. Mozaffari, K. Faez, V. Märgner, H. El Abed, Strategies for large handwritten Farsi/Arabic lexicon reduction, in: Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR '07), vol. 1, 2007, pp. 98–102.
[17] S. Mozaffari, K. Faez, V. Märgner, H.E. Abed, Two-stage lexicon reduction for offline Arabic handwritten word recognition, Int. J. Pattern Recognit. Artif. Intell. 22 (2008) 1323–1341.
[18] S. Wshah, V. Govindaraju, Y. Cheng, H. Li, A novel lexicon reduction method for Arabic handwriting recognition, in: Proceedings of the 20th International Conference on Pattern Recognition (ICPR '10), 2010, pp. 2865–2868.
[19] Y. Chherawala, M. Cheriet, W-TSV: Weighted topological signature vector for lexicon reduction in handwritten Arabic documents, Pattern Recognit. 45 (2012) 3277–3287.
[20] A. Asi, J. El-Sana, V. Märgner, Hierarchical scheme for Arabic text recognition, in: Proceedings of the 11th International Conference on Information Sciences, Signal Processing and their Applications: Special Sessions (ISSPA2012: Special Sessions), 2012, pp. 1299–1304.
[21] J. Yang, Y.-G. Jiang, A. G. Hauptmann, C.-W. Ngo, Evaluating bag-of-visual-words representations in scene classification, in: Proceedings of the 9th international Workshop on Multimedia Information Retrieval, MIR '07, ACM, New York, NY, USA, 2007, pp. 197–206.
[22] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06), vol. 2, 2006, pp. 2169–2178.
[23] P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica-Perez, T. Tuytelaars, A thousand words in a scene, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2007) 1575–1589.
[24] L. Wu, S.C. Hoi, Enhancing bag-of-words models with semantics-preserving metric learning, IEEE Multimed. 18 (2011) 24–37.
[25] L. Zhou, Z. Zhou, D. Hu, Scene classification using a multi-resolution bag-of-features model, Pattern Recognit. 46 (2013) 424–433.
[26] G. Mori, S. Belongie, J. Malik, Efficient shape matching using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 1832–1837.
[27] Y. Chherawala, R. Wisnovsky, M. Cheriet, Sparse descriptor for lexicon reduction in handwritten Arabic documents, in: Proceedings of the 21th International Conference on Pattern Recognition (ICPR '12), 2012, pp. 3729–3732.
[28] P. Viola, M.J. Jones, Robust real-time face detection, Int. J. Comput. Vis. 57 (2004) 137–154.
[29] Y. Chherawala, R. Wisnovsky, M. Cheriet, TSV-LR: topological signature vector-based lexicon reduction for fast recognition of pre-modern Arabic subwords, in: Proceedings of the 1st Workshop on Historical Document Imaging and Processing (HIP '11), 2011, pp. 6–13.
[30] M. Pechwitz, V. Märgner, Baseline estimation for Arabic handwritten words, in: Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition (IWFHR '02), 2002, pp. 479–484.
[31] R. Farrahi Moghaddam, M. Cheriet, M. M. Adankon, K. Filonenko, R. Wisnovsky, Ibn Sina: A database for research on processing and understanding of Arabic manuscripts images, in: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, DAS '10, ACM, New York, NY, USA, 2010, pp. 11–18.
[32] M. Pechwitz, S. Snoussi Maddouri, V. Märgner, N. Ellouze, H. Amiri, IFN/ENIT-database of handwritten Arabic words, in: Proceedings of the 7th Colloque International Francophone sur l'Ecrit et le Document (CIFED '02), 2002, pp. 129–136.
[33] D. Arthur, S. Vassilvitskii, k-means++: the advantages of careful seeding, in: Proceedings of the 18th annual ACM-SIAM symposium on Discrete algorithms (SODA '07), Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2007, pp. 1027–1035.
[34] A. Srivastava, E. Klassen, S.H. Joshi, I.H. Jermyn, Shape analysis of elastic curves in Euclidean spaces, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2011) 1415–1428.
[35] Y. Chherawala, M. Cheriet, Shape recognition on a Riemannian manifold, in: Proceedings of the 11th International Conference on Information Sciences,

Signal Processing and their Applications: Special Sessions (ISSPA2012: Special Sessions), 2012, pp. 1205–1210.
[36] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 509–522.
[37] S.A. Azeem, H. Ahmed, Off-line Arabic handwriting recognition system based on concavity features and HMM classifier, in: Proceedings of the 21th International Conference on Pattern Recognition (ICPR '12), Tsukuba Science City, Japan, 2012, pp. 705–708.
[38] S.J. Young, G. Evermann, M.J.F. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, P. C. Woodland, The HTK Book, version 3.4, Cambridge University Engineering Department, Cambridge, UK, 2006.
[39] A. Vedaldi, V. Gulshan, M. Varma, A. Zisserman, Multiple kernels for object detection, in: Proceedings of the 12th IEEE International Conference on Computer Vision (ICCV '09), 2009, pp. 606–613.
[40] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05), vol. 1, 2005, pp. 886–893.
[41] A. Vedaldi, B. Fulkerson, VLFeat: An Open and Portable Library of Computer Vision Algorithms, url: ⟨http://www.vlfeat.org/⟩, 2008.

Youssouf Chherawala received his M.Sc. degree in Electrical Engineering from the École de Technologie Supérieure (University of Québec), in 2007, and his Ph.D from the same university in 2013. Currently, he is a postdoctoral fellow at the Synchromedia Laboratory for Multimedia Communication in Telepresence. His research interests include pattern recognition, shape analysis and handwriting recognition.

Mohamed Cheriet was born in Algiers (Algeria), in 1960. He received his B.E. from USTHB University (Algiers), in 1984 and his M.Sc. and Ph.D. degrees in Computer Science 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 at the École de Technologie Supérieure (University of Quebec), Montreal, and was appointed as a full professor there, in 1998. He co-founded the Laboratory for Imagery, Vision and Artificial Intelligence (LIVIA) at the University of Quebec, and was its director from 2000 to 2006. He also founded the SYNCHROMEDIA Consortium (Multimedia Communication in Telepresence) there, and has been its director since 1998. His interests include document image analysis, OCR, mathematical models for image processing, pattern classification models and learning algorithms, as well as perception in computer vision. Dr. Cheriet has published more than 250 technical papers in the field, and has served as a chair or co-chair of the following international conferences: VI'1998, VI'2000, IWFHR'2002, and ICFHR'2008. He currently serves the editorial board and is an associate editor of several international journals: IJPRAI, IJDAR, and Pattern Recognition. He co-authored a book entitled, "Character Recognition Systems: A guide for Students and Practitioners," John Wiley and Sons, Spring 2007. Dr. Cheriet is a senior member of the IEEE and the chapter chair of IEEE Montreal Computational Intelligent Systems (CIS).