IJDAR (2013) 16:285–294
DOI 10.1007/s10032-012-0189-5

ORIGINAL PAPER

Comprehensive synthetic Arabic database for on/off-line script recognition research

Raid M. Saabni · Jihad A. El-Sana

Received: 5 June 2011 / Revised: 19 January 2012 / Accepted: 10 May 2012 / Published online: 29 May 2012

Abstract Developing and maintaining large comprehensive databases for script recognition that include different shapes for each word in the lexicon is expensive and difficult. In this paper, we present an efficient system that automatically generates prototypes for each word in a lexicon using multiple appearances of each letter. Large sets of different shapes are created for each letter in each position. These sets are then used to generate valid shapes for each word-part. The number of valid permutations for each word is large and prohibits practical training and searching for various tasks, such as script recognition and word spotting. We apply dimensionality reduction and clustering techniques to maintain compact representation of these databases, without affecting their ability to represent the wide variety of handwriting styles. In addition, a database for off-line script recognition is generated from the on-line strokes using a standard dilation technique, while making special efforts to resemble pen's path. We also examined and used several layout techniques for producing words from the generated word-parts. Our experimental results show that the proposed system can automatically generate large databases, whose quality is at least as good as the manually generated ones.

Keywords Arabic · Script · Recognition · Database · Synthetic · PCA · Kmeans

R. M. Saabni (✉)
Triangle R&D center, Kafr Qari, Israel
e-mail: saabni@cs.bgu.ac.il

J. A. El-Sana
Department of Computer Science,
BG University in the Negev, Beersheba, Israel
e-mail: elsana@cs.bgu.ac.il

1 Introduction

The recognition of cursive handwriting is a challenging task because of the huge variations and individuality of personal handwriting. In scripts where cursive writing is optional, such as Latin scripts, it is possible to restrict the recognition to the non-cursive handwriting and provide a partial solution. However, such an option is not valid for inherently cursive scripts, such as the Arabic script.

The research in script recognition has distinguished between two main approaches—segmentation-based and segmentation-free. The segmentation-based approaches segment an input word into individual characters, which are then recognized and combined to identify the input word. The segmentation-free approaches recognize the whole word at once, without segmenting into characters. Recently, the holistic approach has been attracting more interest and become widely accepted in handwriting recognition research [5,9–11,18,30,31,42]. However, the holistic approach compares continuous words or sub-words and is required to maintain large databases—a recognition model for each word in the lexicon. In addition, the training and recognition demand the existence of the handwritten shapes for the entire lexicon and the generation of these shapes manually is expensive and time consuming.

Many Latin databases for handwritten script recognition have been developed, especially for English scripts [20,36,17,40]. In contrast, very few databases have been developed for the Arabic script and fewer have become publicly available [1,2,21,28,29]. Research groups have developed private databases [2,21,28], which rarely become publicly available. To the best of our knowledge, no database has been developed to include the entire handwritten words or word-parts in the Arabic language because of the tremendous effort needed for such a task. That explains the unavailability of

a standard comprehensive database (on-line or off-line) for Arabic handwriting script recognition.

In this paper, we present a new approach for efficient generation of an on demand synthetic comprehensive database for on-line and off-line Arabic script recognition research. The generated database includes multiple shapes for each word that represent multiple personal handwriting styles. We have developed a novel approach to generate synthetic shapes of any Arabic word using manually generated handwriting sets of prototypes representing the various ways of writing each letter. To keep the database compact, we reduce redundancy by applying clustering and dimensionality reduction techniques. The compact set still covers the huge variety of writing styles while keeping the database compact for practical applications. The shape of a word-part, which is generated from basic elements of characters of one-pixel width, does not reflect realistic scanned off-line writing. To overcome this limitation, we extend the stroke width based on the properties of each word-part.

The rest of this paper is organized as follows: in Sect. 2, we present a brief introduction to Arabic script characteristics; in Sect. 3, we explore the closely related work of Arabic script recognition and databases; the proposed system is discussed in detail in Sect. 4. Sections 5 and 6 present experimental results and discuss directions for future work.

2 Characteristics of Arabic script

The Arabic script is used as the alphabet for several languages, such as Farsi, Urdu, Malay, Swahili, Hausa, and Ottoman Turkish. It is written from right to left in a semi-cursive manner in handwriting as well as machine printing. The Arabic script is similar to western scripts in that it has a strict alphabet consisting of letters, numerals, and punctuation marks but is different in the way it combines letters into words and the way it treats vowels.

The Arabic script consists of 28 basic letters, 12 additional special letters such as (ة ٔ ى), and 8 diacritics. A letter in Arabic usually has several (2–6) different shapes—initial, medial, final, and isolated—according to its adjacent letters and its position within the word. As a result, the 28 basic letters in Arabic script have close to 120 different shapes when additional strokes (Dots, Hamza, Madda, and others) are included. Some letters interrupt the cursiveness of a word by prohibiting a connection to the following letters and splitting words into connected groups of letters called components. Each component includes one or more letters and, with its additional strokes, forms a part of word, which we call *word-part*.

An Arabic word-part, ω, has a *main part*, which is totally cursive, and a *complementary part*, which includes all the additional strokes of the letters within ω (the complementary part could be empty). Several letters share the same body part and differ by the complementary parts. For example, the word-parts *bayt* (بيت) and *tabeth* (تبث) share the same main body and differ in the complementary part.

In order to construct a comprehensive synthetic database that includes the entire Arabic lexicon, we have explored large collections of Arabic texts and extracted 300,000 different words assembled from 48,000 different word-parts. After ignoring the additional strokes, the number of different word-parts went down to 28,500, organized in a *word-part lexicon*. Since the text collections were very large (over 5 million words) and came from different domains and periods, we believe these results closely represent a large fraction of the Arabic lexicon (see [32] for more details). We have used our novel technique to synthesize large sets of shapes for each word-part in the word-part lexicon.

3 Related work

In the research in Arabic script, compared to Latin and Chinese, recognition has recently attracted the interest of researchers. This delay exposed researchers to results and techniques used in other scripts, which were adapted and improved to meet the needs and challenges of Arabic scripts recognition. Segmenting cursive Arabic words into characters is a hard task due to the large variety of different writing styles and the absence of constraints and consistency. Attempts have been made to recognize isolated forms of Arabic letters and avoid word segmentation by forcing a non-cursive style of writing [6,16,24,27]. In parallel, segmentation-based methods were developed or adapted and improved [3,8,14,19,35].

Incorrect segmentation of a given word or word-part into letters results in poor recognition rates. The complexity and difficulties of segmenting Arabic handwritten words into letters shifted the research focus to the segmentation-free (holistic) approach. In the holistic approach [4,12,19,23], complete words are processed to be recognized without segmenting words into characters. Character-based recognition approaches are required to store the various forms of each character, while holistic approaches are required to maintain the large database that stores multiple shapes for each word in the lexicon, which are used for training and recognition. The holistic approach was initially used for recognition tasks that require a small vocabulary, such as check verification, mail sorting, and keyword searching. Recently, the development of efficient holistic-based recognition methods to handle large vocabularies has attracted more interest [22,32,33]. Such development requires large databases for training and evaluation as well as efficient processing in terms of time and space.

Wang et al. [41] presented a method to synthesize cursive handwriting words guided by a deformable model. The process concatenates ligatures, strokes, and isolated letters generated from learned models to determine word trajectory. Varga and Bunke [38] presented a method for generating synthetic handwritten text lines using images of text lines of cursive handwriting. Thinning/thickening and other geometrical transformations were used as perturbation models to generate the synthetic lines. They used the synthetic data to improve the learning process of HMM-based off-line cursive handwriting recognition. Varga et al. [39] present a method for synthesizing English cursive handwriting from text lines using templates of characters and the *delta log normal* model. To generate a text line, they concatenate a perturbed version of the characters in the text line based on the given templates. Overlapping strokes and delta log normal velocity profiles were used to draw the text line. Cheng and Lopresti [13] conducted experiments using a mixture of real English handwritten text lines and text lines altered from existing handwriting with various distortion degrees, based on a perturbation method from [37]. They aim to calibrate distortion parameter settings for Varga and Bunke's [37] perturbation model and compare the effects of parameter settings on different writing styles. Preliminary experimental results show that appropriate parameter settings for different handwriting styles make it possible to generate a large quantity of training and testing data for building better off-line handwriting recognition systems.

Many databases for Latin handwritten script recognition, such as UNIPEN [36], IAM [26] CEDAR [20], NIST [17], and IRONOFF [40], were developed. In contrast, very few databases were developed for Arabic script and fewer became publicly available. The IFN/ENIT off-line database for Arabic words is one of the first publicly available databases and became the first standard database for Arabic. It includes 946 Tunisian town/villages names and postal codes written by 411 people. A Persian version of the IFN/ENIT was recently released, including city names handwritten in Farsi and consists of 7,271 binary images of 1,080 Iranian province/city names, collected from 600 writers. Another known database for Arabic handwriting recognition is the CENPARMI Arabic checks, which was released in 2003 [2] and consisted of legal and courtesy amounts on bank checks and isolated handwritten digits. Several standard databases have been developed recently for research of Farsi/Arabic off-line handwritten recognition [15,28,34]. Mozaffari et al. [28] presented a new comprehensive database for isolated off-line handwritten Farsi/Arabic numbers and characters for use in optical character recognition research. It includes the gray scale images of 52,380 characters and 17,740 numerals, which were scanned from Iranian school entrance exam forms during the years 2004–2006 at 300 dpi. Solimanpour et al. [34] described an approach toward a standard handwritten Farsi database including isolated digits, letters, numerical strings, legal amounts used for checks, and dates. The ADAB database for on-line Arabic script recognition had been published by Haikal et al. [15] as part of a competition in on-line Arabic handwriting script recognition at the ICDAR2009. The database consists of 15,158 Arabic words from 937 Tunisian town/village names written by more than 130 different writers.

In conclusion, researchers have mostly developed their own small datasets or large databases that are not available to the public [2,7,12,21]. None of these databases were developed to include the entire Arabic lexicon.

4 Our approach

In this paper, we present a novel approach for generating synthetic comprehensive databases for training, testing, and evaluating Arabic script recognition systems. The system consists of three main procedures: *handwriting prototypes extraction, word-part generation, and shape clustering*. The handwriting prototypes extraction procedure is responsible for generating sets of different handwriting prototypes for each letter in each position by enabling users to manually enter complete words that cover the various shapes of the letters at different positions. The word-part generation procedure generates multiple shapes for each word-part ω in a given lexicon by concatenating the shapes of its constituting letters in the right position and order. The existence of multiple shapes for each letter and the need to consider many permutations generate a huge number of shapes for each word-part, which are too large for practical use. To overcome this difficulty, the third procedure clusters the shapes of each word-part into groups. We have found that selecting small representative shapes from each cluster dramatically reduces the size of the database without affecting its representativeness. Figure 1 shows the flow diagram and the different stages of our system. It starts with accepting the ASCII code of a given Arabic word-part (right end) and uses a predefined set of *handwriting prototypes* to synthesize word-part images and cluster them to generate a compact set of representatives.

In our current implementation, we ignored additional strokes (dots) and concentrated on synthesizing the main component of a word-part. Nevertheless, additional strokes could be easily added in a post-processing phase.

Our system provides the ability to easily generate comprehensive databases for on-line and off-line script recognition. The generated database includes additional properties, such as local and global features at the character and word-part levels, which simplifies the study of various script recognition and word-part segmentation algorithms.

In the rest of this section, we discuss in detail the three procedures of the proposed system.

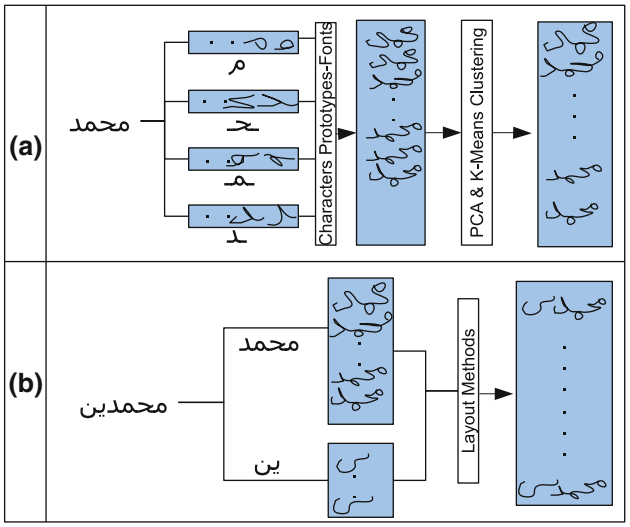


Fig. 1 In the *first row* (**a**), we can see the diagram of generating multiple appearances of a word-part given as a text using the different shapes of each letter within the word-part. In the *second row* (**b**), multiple appearances of the different word-parts within a word given as a text are used to generate different shapes of that word

4.1 Handwriting prototypes extraction

This procedure enables a human operator to enter multiple shapes for each letter of the Arabic script, while taking into account the variance in the shapes of Arabic letters according to their position in the word. For that purpose, it includes a graphic user interface (GUI) that simplifies entering various letter and word shapes by different means. Currently, our system supports three schemes to capture the different shapes of letters:

1. The operator enters the different shapes of a letter for each position in a word using a digital tablet.
2. The system posts a sequence of word-parts, and the operator enters, via tablet, these word-parts without delayed strokes (dots and small strokes) and then segments these word-parts manually to generate the various shapes for each letter.
3. Off-line word-parts (word-parts written on paper) are scanned into the system, which extracts the components and enables a human operator to manually segment each word-part to letters. The stroke thickness of each letter is computed and stored in the database (to be used for the generation of off-line shapes).

The set of word-parts feed into the system aims to cover various shapes of all the letters and ligatures in the Arabic script.

Our system is able to extract handwriting prototypes from existing databases and extend them to cover to larger lexicons. It scans word-parts in the database, extracts the various occurrences of each letter in the different positions, and generates sets of handwriting prototypes. Databases that do not include segmentation of the handwritten word-parts to individual characters are required to go through a manual segmentation phase for each word and word-part. To capture unusual ways of writing pairs of special ligatures such as (ﻻ) and (ﻹ) we have considered them as additional single letters to the alphabet. In a post-processing step, human operators eliminate unreal shapes of word-parts mostly generated using special cases of writing letters that are part of these special ligatures. For example, the shape of the letter (ﺎ) in the ligature (ﻹ) is a special way of writing this letter in it's medial form.

Typically, the shapes of each letter in a generated font are similar, and some are almost identical. To eliminate redundant samples and retain compact set of shapes, we apply hierarchical clustering, which produces sets that faithfully represent the variety of handwriting shapes of each letter.

The human operator is advised to use their personal handwriting styles and imitate different common writing styles for ligatures or letters in different positions. For example, the two letters (ﺮﺗ) have different writing shapes and users are guided to write them in vertical order to cover such a style (see Fig. 5). The system also guides the attention of operators to special ligatures, such as (ﻻ) and other common pairs of consequent letters, which may not concatenate naively (Fig. 2).

The system computes and sets global features, such as loops, for the generated prototypes per word-part in the database. The deterministic loop properties are extracted directly from the text of word-part. To consider these properties in a non-deterministic manner, we store the probability of computationally detecting a loop within a word-part shape. This probability is determined based on the size and the ratio between the short and the long diameters of the loop (see Figs. 3, 4). Nevertheless, researchers and developers can access the strokes representing each character and word-part and extract additional properties. These properties are used to generate the features for the generated words and integrate them into the database. Stroke-based features, such as end point, split point, or high curvature points, are extracted and added to the database (Fig. 5).

The generation of an off-line representation uses these features to imitate off-line writing by integrating off-line properties, such as end-point smoothing and thickening of split points. These feature points, the original stroke, and the concatenation points between letters within the word-part are available in the database. They can be utilized by researchers as a ground truth to test and evaluate thinning and character segmentation algorithms.

In our approach, characters and ligatures have different writing styles and thus encounter different numbers of loops, which call for careful treatment to avoid eliminating legiti-

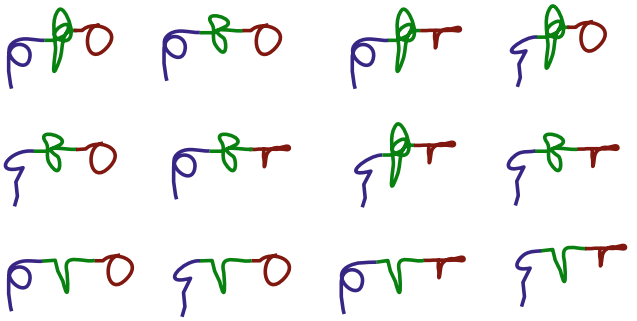


Fig. 2 Shapes generated for the word (محمد), where the letters (د) and (م) include different numbers of loops. The *shapes* in each *row* are ordered according to number of loops (in a decreasing order)

Fig. 3 Examples of loop existences probabilities of the letter (م): the three columns show high, medium, and low probabilities (*left* to *right* order). The size of the suspected loop and the ratio between the diameters are used to calculate these probabilities

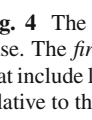
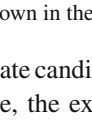
Shape	Human Operator Properties		
	Position	Type	Aspect Ratio
	Up / Down	Triple	0.49
	Up / Down	Double	0.56
	None	None	None
	Down	Single	0.985
	Down	Single	0.05
	Down	Single	0.62

Fig. 4 The data integrated with each loop within a shape in the database. The *first column* includes shapes for different Arabic characters that include loops. The *second column* presents the position of the loops relative to the base line. The types of the loops and the aspect ratio are shown in the *third column* and f*ourth column*, respectively

mate candidates in a loop-based candidate filtering; for example, the existence of loop in the letter (ط) is not consistent even in different printed fonts—the medial form may include zero, one, two, or three loops (see Fig. 2 for details)—and letters (ح), (خ) and (ج) can be written with or without a loop. Such inconsistency in the number of loops complicates the preprocessing and post-processing candidate pruning phase.

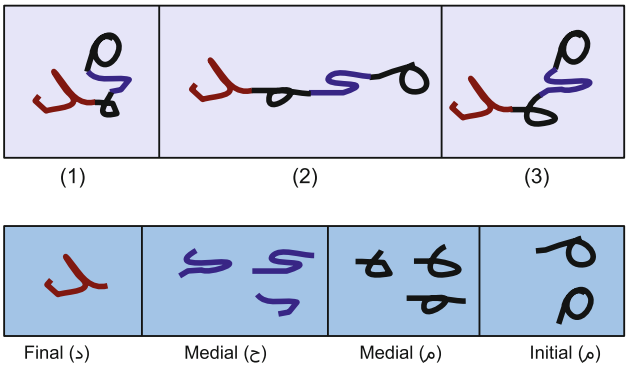


Fig. 5 Three samples of synthetically generating the word (محمد)

In our approach, loops are extracted and counted separately for each different letter shape, and filtering is applied across the different word shapes (see Fig. 4). We have adopted this policy to avoid deterministic pruning, which can eliminate words with degenerated loops or different writing styles of the same letters, as mentioned previously. Different shapes of a letter may contribute different local and global features to a word-part. These data are highly important, as it is used to filter out candidates in a non-deterministic manner when considering global features such as loops.

4.2 Synthesizing word-parts and word shapes

The synthesizing process uses the designed handwriting prototypes to generate a polyline representation of each word-part that forms the on-line database. These polylines are used to generate off-line databases by increasing the width of the lines and adding noise to simulate scanned off-line word-parts.

The generated fonts are used to construct a writer-independent open vocabulary database of word-part shapes. For each word-part ω in the lexicon Σ, the system determines the shape of the letters in ω, while taking into consideration the position of each letter and ignoring the additional strokes. The shapes representing ω are generated by concatenating the various shapes for each letter in the right order—generating the valid permutations of ω. This simple concatenation is performed by stitching the endpoint of each letter shape to the start point of the following one and smoothing the stitching region. Our current system uses an Arabic language word-parts lexicon that includes a large fraction of all word-parts in the Arabic language—around 48,000 word-parts. It is also capable of generating a database for any given lexicon, such as Farsi or Urdo, which uses the Arabic alphabet. For some languages, it may require adding shapes of letters that do not exist in the Arabic language.

Let l_i^p be a vector $(v_0, \ldots, v_{n_i})$ with length n_i, representing the ith shape of the letter l in the position p. To generate one shape for the word-part $\omega = (l_1, l_2, \ldots, l_m)$ with length

m, we concatenate the vectors $l_{i_1}^{\text{ini}}, l_{i_2}^{\text{med}}, \ldots, l_{i_m}^{\text{fin}}$; each represents one appearance of a letter in a specific form, where ini, med, and fin stand for the initial, middle, and final positions of a letter within a word-part, respectively. The concatenation is performed by joining the endpoint of the vector l_i with the start point of l_{i+1}, while taking into account the appropriate positions of the two letters. Points in the vector $l_{i+1}^{p_{i+1}}$ are adjusted to be aligned to the end point of the previous letter's vector. To achieve seamless stitching, we apply a simple smoothing to the stitching region of each two adjacent shapes of letters.

Obviously, no concatenation is applied for one-letter word-parts. Two-letter word-parts are constructed by concatenating the initial and final vectors of the corresponding letters. As expected, such a scheme for word-part shape generation produces huge sets of shapes for each word-part in the lexicon. For example, a word-part that contains five letters with eight different shapes for each letter have $32{,}768 = 8^5$ different shapes. Many of these shapes are similar as they display only minor differences, which calls for techniques to reduce redundancy.

During the generation process, global and local features are extracted from the font classes, for example, the number of loops in a word-part, ω, is determined based on the assigned loop count for each letter in ω. Note that the same word-part may have various shapes that have different properties depending on the complexity of its constituting letter shapes, for example, different number of loops. Such diversity in word-part representation enables delicate treatment of various features in a non-deterministic manner, which is essential for holistic-based recognition approaches.

We create the off-line images of the word-parts from the generated on-line word-part shapes—the ordered strokes—by applying a standard dilation process. In general handwriting, the curves near end points are usually smooth and thin due to a pen lift, and areas around a split point and curved strokes are often thicker than the average width of the stroke. Our off-line handwriting generation algorithm determines these properties based on a small number of user-determined parameters. In the final step, we use two methods to simulate the process of printing and scanning. In the first method, we replace the expected process of printing and scanning by the methods presented in [25], and to simulate the scanning process, we use different degradation factors. The second method uses a convolution with a Gaussian kernel (see Fig. 6) to add noise to the generated images.

The generation of words from word-parts is performed based on a predefined *layout scheme*, which determines the position of the shapes of word-parts with respect to each other. To represent the different writing styles, these schemes apply different layout methods, such as tilting word-parts horizontally, and semi-vertically with homogeneous, heterogeneous, or zero distances. In the Arabic script, there are only six disconnective letters (و, ر, ز, د, ا, ذ) which means that any non-final Arabic word-part has to end with one of these six letters. Observing the different writing styles and their different behaviors, we have noticed that the three letters (و, ر, ز) at the end of a word-part may allow or encourage the consequent word-part to overlap or touch the current word-part within a word. The other three of the six letters prohibit overlapping or touching, unless the consequent word-part starts with the letters (ك, س and ش). We have utilized this observation to generate various shapes of a given word by tilting the different word-parts within a word using the different layout schemes (see Fig. 6).

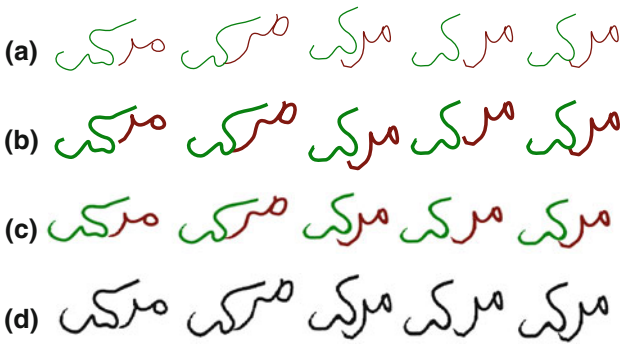


Fig. 6 Different samples of the shapes generated for writing the word (مركب): **a** one-pixel width shapes, **b** after applying dilation, **c** after applying feature points smoothing, and **d** after applying scanning imitation

We generate words from the given lexicon using the generated word-part's shapes and based on the following three different layout schemes that determine different handwriting styles.

- A reasonable gap to concatenate the word-parts within a word on the same base line. This gap has been determined by calculating the average gap between different word-parts within the same word in a text collection that includes one hundred full pages of different handwriting styles.
- Enabling selected word-parts, based on their constituting letters, to be aligned vertically or in any other direction, while allowing their bounding boxes to overlap. This is done using the results of layout schemes discussed earlier.
- Based on the first and last letters of the different word-parts, we enable some selected shapes to touch each other.

Even though these methods do not represent all the different styles, still they include most of them. It is important to note that these styles are easily extended to cover more layout schemes.

4.3 Dimensionality reduction and clustering

The generated representation for each word-part in the lexicon is too large for practical use. Fortunately, we have realized that a large fraction of these representations have very few or no differences. Such a high percentage of redundancy of the generated shapes for each word-part, which often include tens of thousand of items, could be reduced dramatically by clustering and dimensionality reduction techniques. This step aims to generate compact sets, defined as the smallest sets that represent the wide variety of shapes for each word-part. We have adopted three techniques to build compact sets: Hierarchical clustering, principal component analysis (PCA), and K-means clustering.

Let $S(\omega) = \{s_i(\omega)\}_{i=1}^{n}$ be a set of n vectors $s_i = (v_1, v_2, \ldots, v_{n_i})$, which represent shapes of the word-part ω. To enable efficient and accurate processing, we simplify the stroke s_i in a semi-uniform manner. Let us denote the simplified vector s_i by s_i^{δ}, where δ is the error tolerance used to control the simplification process. We define the feature α_j at the point p_j, on a given vector (point sequence), as the angle between the segment $\overline{p_j, p_{j+1}}$ and the following segment. For each point vector s_i^{δ}, we generate a feature vector f_i^{δ} using the features α_j for $0 \leq j < n_i$. We also use a parameter k to determine the desired cardinality.

We first apply PCA on the covariance matrix of the n vectors f_i and use the m eigenvectors derived from the largest m eigenvalues for dimensionality reduction. The original samples transformed by the m eigenvectors are clustered using the K-Means clustering technique. The results are then transformed back to the original vectors and used as the k centroids to extract the representative vectors within each cluster. The result of the third step is a set of k vectors representing the k shapes in our desired compact set. The constants k and m are fixed for each word-part as a percentage of the different shapes for each letter and the length of the word-part. In these clustering methods, we adopted the Euclidean distance to measure differences between shapes, which requires applying length normalization on the feature vectors.

To synthesize an off-line word, we applied the same technique of clustering using the contour of the shape. The resulting compact sets were very similar to those we obtained using the one-pixel width stroke. Therefore, we decided to apply the dilation on the clustering results, yield from stroke, for efficient processing. Holistic approaches, which rely on contour or sliding windows techniques can use the original—non-compact—and apply their own techniques for clustering. Holistic approaches using words as one component can adapt their own technique to reduce the size of the lexicon for each word, if needed. We believe that our layout methods represent various writing styles; nevertheless, additional reduction techniques can be used to obtain different compact sets.

5 Experimental results

To test and evaluate our approach, we performed several experiments on two Arabic databases: Adab [1] and IFN/ENIT [29], which are often used to train and evaluate Arabic handwriting recognition systems. We adapt the Adab [1] and the IFN/ENIT [29] databases to be used with our on-line and off-line Arabic handwriting recognition systems, which expect individual word-parts as the basic units for recognition. The database contents were slightly modified to include each word-part as one connected component, that is, incorrectly split components for single word-parts rejoined to form a single one and touching components split manually to individual word-parts. The resulting databases include word-parts, most of them with multiple appearance and demarcations that indicate the split of word-parts to letters. We also extended the properties of the IFN/ENIT word-parts to include the beginning and end drawing points.

We have evaluated the synthetically generated on-line and off-line databases against the manually generated Adab and IFN/ENIT databases, respectively. To compare the performance of two different recognition systems, we train and compare their recognition performance using the same database. To compare two databases, we use them to train two instances of the same recognition system and compare their recognition performance.

We used our on-line Arabic Script recognition system [33] for on-line handwriting recognition and adapted it to off-line handwriting recognition by using the bounding contour of a word-part as the input stroke. We compared the recognition precision and recall rates of each system using a synthetically generated database versus the manually generated benchmarks (IFN/ENIT and Adab). Next, we overview the handwriting prototypes and datasets used to evaluate the quality of a synthesized database.

The following three sets were used to evaluate on-line handwriting recognition:

1. *Manually modified Adab database (MM-Adab)*: A modified version of the *Adab* database that includes 2,200 word-parts with 16,356 different shapes. The modification aims to ensure the correctness of the main component, that is, each word-part is represented as a single connected component.
2. *Synthetic generation from Adab (SG-Adab)*: The letters in the MM-Adab are used to generate 22,000 different shapes from the 2,200 word-parts.
3. *Synthetic generation from user (SG-ON-User)*: 48 writers trained the system to generate three sets of prototypes based on the three proposed style schemes (see Sect. 4.1). In each set, we generated 31,230 different shapes for the word-parts in the MM-Adab set.

Table 1 Precision and recall results in columns 1 and 2 are for the on-line recognition system

Dataset	Off-line recognition		On-line recognition	
	Precision (%)	Recall (%)	Precision (%)	Recall (%)
Manually modified	80.21	81.32	78.21	79.21
Synthetic generation	81.16	80.57	78.64	77.96
User synthetic generation	81.09	80.10	80.43	78.12

The three sets (MM-Adab, SG-Adab, and SG-On-User) are in rows 1, 2, and 3, respectively. Results for the off-line system are in the third and fourth columns with the different sets in the respective rows

The following three sets were used to evaluate off-line handwriting recognition:

1. *Manually modified IFN/ENIT database (MM-IFN)*: A modified version of the IFN/ENIT database that includes 2,070 word-parts with 18,245 different shapes. The modifications were similar to those applied to MM-Adab set.
2. *Synthetic generation from IFN/ENIT (SG-IFN)*: This set includes prototypes from the IFN/INIT images with demarcation marks (manually generated) that indicate the letters of each word-part. The extracted list of handwriting prototypes were used to generated 21,300 different shapes of the 2,070 word-parts in the MM-IFN set.
3. *Synthetic generation from user (SG-OFF-User)*: 48 writers have trained the system to generate three sets of prototypes (see Sect. 4.1) using the proposed style schemes. In each set, we generated 27,220 different shapes for the word-parts in the MM-IFN set.

We split each set to two sub-sets, one for training and the other for testing. The training set includes 60 % of the samples, and the testing includes the rest (40 %) of the samples. Our experimental study aims to evaluate the quality of the synthesized databases, which is estimated based on the ability of the generated samples to cover the variety of handwriting as well as is done by manually generated datasets.

Table 1 presents recognition performances of the manual set compared to the synthetic sets. The three rows for the on-line recognition system represent the Manually Modified (MM-Adab), Synthetic Generation based on the Adab database (MM-Adab), and Synthetic Generation based on users on-line handwriting (SG-On-User). The same rows represent the (MM-IFN,SG-IFN and SG-Off-User) for the off-line system, using the IFN/ENIT database and images of the users handwriting. The similarity in the recognition results shows the ability of the synthesized samples to capture the various writing styles of different users. The synthesized samples can also provide better performance (as can be seen in the third column), which results from extending the variety of handwriting by combining various handwritings into one shape (word-part). The richness of the synthesized samples depends on the number of writers and the variety of their handwriting.

Table 2 The recognition accuracy rates and the time reduction when using the loops' numbers as a global feature to pick the right candidates class

Loops case	1 of 5 (%)	Recognition rate (%)	Time reduction (%)
No loop used	85.31	81.27	100
Loops in lexicon	86.51	83.19	20.12
Loops integrated with shapes	91.76	89.68	21.17

Results were conducted using the selected set of 400 word-parts

In order to evaluate the efficiency of deterministic and non-deterministic utilization of the loop as a global features, we experimented with the following three options:

- The recognition system ignores the loop feature.
- The loop feature was used as a filtering step directly on the text lexicon of the Arabic words.
- We use the loop feature across the shapes of the word-parts and apply filtering on the set of shapes instead of text words. For example, filled or degenerated loops, which may not be detected as loops, were not counted as loops in the filtering process (see Fig. 3).

To evaluate the contribution of the integrated properties to the accuracy and time response, we selected a set of 400 word-parts from the datasets based on the Adab database. These word-parts were selected to include at least one loop. Table 2 reports the recognition rates and reduction of time obtained using each of the three options. For a query shape s to be recognized, the process starts by determining the number of loops in s. The first option did not use number of loops in the filtering process. The second option filtered out the shapes (word-parts) that have different numbers of loops in their textual representation. The third option filtered out the shapes by comparing the number of loops integrated into the database, which were determined directly from the shapes without considering their textual representation.

Table 2 shows that using the loop feature to filter candidate words from the lexicon reduces the average recognition time by 80 %. The recognition rates are less encouraging when the loop feature is used in a deterministic manner. The second and third rows in Table 2 show that using global features for pruning candidates improves response time. The recognition rates are improved when the loop feature is used in a

non-deterministic manner across different shapes as seen in the third line.

Research in segmentation into characters can use the recorded split points and the global features for training, testing, and evaluation. Thinning algorithms can use the synthetically generated off-line words with the original strokes as skeletons and feature points, such as end, split, and curvature points for testing and evaluations.

6 Conclusion and future work

We have presented an efficient approach for generating large datasets of synthetic shapes of a given lexicon of words in order to generate a comprehensive database that includes different shapes for each word in the Arabic script. Our approach requires a lexicon that determines the set of words and word-parts, and a set of *handwriting prototypes*. These fonts could be generated manually by human writers or extracted automatically from a given small dataset of word shapes. The results we have presented show the credibility of the procedure and report a small improvement on the recognition rates that result from the inherited ability of this approach to generate many shapes representing the wide variety of writing styles. Our approach aims to produce a comprehensive handwriting database for the Arabic language and can be used to construct different databases for additional languages that use the Arabic alphabet. We also present a simple and efficient technique for extending a given small database to a comprehensive one. Extending this approach for western scripts such as handwriting Latin and English can be done following different rules for connecting letters to each other.

The databases we generated are compact and comprehensive—they include all the words and word-parts in a given Arabic lexicon. For each character, word-part, and word shape, the database includes integrated data for local and global features that could be used by researchers for developing, training, and evaluating new techniques. We believe this database can contribute to various research approaches in Arabic script recognition, word spotting, and character/word skeletonization.

References

1. ADAB: Arabic DAta Base, for on-line recognition of the cursive Arabic handwritten word
2. Al Ohali, Y., Cheriet, M., Suen, C.: Databases for recognition of handwritten arabic cheques. Pattern Recogn. **36**(1), 111–121 (2003)
3. Al-Yousefi, H., Udpa, S.: Recognition of arabic characters. IEEE Trans. Pattern Anal. Mach. Intell. **14**(8), 853–857 (1992)
4. Alma'adeed, S.: Recognition of off-line handwritten arabic words using neural network. In: GMAI '06: Proceedings of the conference on Geometric Modeling and Imaging, pp. 141–144. IEEE Computer Society, Washington, DC, USA (2006)
5. Alsallakh, B., Safadi, H.: Arapen: an arabic online handwriting recognition system. In: Information and Communication Technologies, 2006 (ICTTA '06). 2nd, vol. 1, pp. 1844–1849 (April 2006)
6. Alshebeili, S.A., Nabawi, A.A.F., Mahmoud, S.A.: Arabic character-recognition using 1-d slices of the character spectrum. Signal Process. **56**(1), 59–75 (1997)
7. Amin, A.: Off-line arabic character recognition: the state of the art. Pattern Recogn. **31**(5), 517–530 (1998)
8. Amin, A., Mari, J.: Machine recognition and correction of printed arabic text. IEEE Trans. Syst. Man Cybern. **19**(5), 1300–1306 (1989)
9. Ataer, E., Duygulu, P.: Matching ottoman words: an image retrieval approach to historical document indexing. In: CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval, pp. 341–347. ACM, New York, NY, USA (2007)
10. Ball, G., Srihari, S., Srinivasan, H.: Segmentation-free and segmentation-dependent approaches to arabic word spotting. In: Proceedings of International Workshop on Frontiers in Handwriting Recognition (IWFHR-10), pp. 53–58. La Baule, France (October 2006)
11. Biadsy, F., El-Sana, J., Habash, N.: Online Arabic handwriting recognition using hidden Markov models. In: Proceedings of the 10th International Workshop on Frontiers of Handwriting and Recognition, pp. 3278–3286 (2006)
12. Biadsy, F., Saabni, R., El-Sana, J.: Segmentation-free online arabic handwriting recognition. Int. J. Pattern Recogn. (page to appear) (2011)
13. Cheng, W., Lopresti, D.: Parameter calibration for synthesizing realistic-looking variability in offline handwriting. In: Document Recognition and Retrieval XVIII IS&T/SPIE International Symposium on Electronic Imaging, p. 157. IEEE Computer Society, San Francisco, CA (2011)
14. El-Emami, S., Usher, M.: On-line recognition of handwritten arabic characters. IEEE Trans. Pattern Anal. Mach. Intell. **12**(7), 704–710 (1990)
15. El Abed, H., Kherallah, M., Margner, V., Alimi, A.M.: Arabic online handwriting recognition competition. In: 10th International Conference on Document Analysis and Recognition (ICDAR), pp. 1383–1387 (2009)
16. El-sheikh, T., Guindi, R.: Automatic recognition of isolated arabic characters. Signal Process. **14**(2), 177–184 (1988)
17. Garris, M.: Design and collection of a handwriting sample image database
18. Gatos, B., Konidaris, T., Ntzios, K., Pratikakis, I., Perantonis, S.J.: A segmentation-free approach for keyword search in historical typewritten documents. In: Proceedings of Eighth International Conference on Document Analysis and Recognition, 2005, pp. 54–58, vol. 1. 29 August–1 September (2005)
19. Gillies, A., Erl, E., Trenkle, J., Schlosser, S.: Arabic text recognition system. In: Proceedings of the Symposium on Document Image Understanding Technology (1999)
20. http://www.cedar.buffalo.edu/Databases/
21. Kharma, N., Ahmed, M., Ward, R.: A new comprehensive database of hand-written arabic words, numbers and signatures used for ocr testing. In: IEEE Canadian Conference on Electrical and Computer Engineering, pp. 766–768 (1999)
22. Koerich, A.L., Sabourin, R., Suen, C.Y.: Large vocabulary off-line handwriting recognition: a survey. Pattern Anal. Appl. **6**(2), 97–121 (2003)
23. Maddouri, S., Amiri, H.: Combination of local and global vision modelling for arabic handwritten words recognition. In: Proceedings of Eighth International Workshop on Frontiers in Handwriting Recognition, 2002, pp. 128–135 (2002)
24. Mahmoud, S.A.: Arabic character recognition using fourier descriptors and character contour encoding. Pattern Recogn. **27**(6), 815–824 (1994)

25. Margner, V., Pechwitz, M.: Synthetic data for arabic ocr system development. In: Sixth International Conference on Document Analysis and Recognition (ICDAR'01), pp. 1159–1163 (2001)
26. Marti, U., Bunke, H.: The iam-database: an english sentence database for off-line handwriting recognition. Int. J. Document Anal. Recogn. **5**, 39–46 (2002)
27. Mezghani, N., Mitiche, A., Cheriet, M.: Bayes classification of online arabic characters by gibbs modeling of class conditional densities. IEEE Trans. Pattern Anal. Mach. Intell. **30**(7), 1121–1131 (2008)
28. Mozzaffari, S., Faez, K., Faradji, F., Ziaratban, M., Golzan, M.: A comprehnsive isolated farsi/aarabic character database for handwritten ocr research. In: Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition, France, pp. 385–389 (October 2006)
29. Pechwitz, M., Maddouri, S.S., Margner, V., Ellouze, N., Amiri, H.: Ifn/enit—database of handwritten arabic words. In: Proceedings of CIFED 2002, pp. 129–136 (2002)
30. Plamondon, R., Guerfali, W.: Why handwriting segmentation can be misleading? In: Proceedings of International Conference on Pattern Recognition, pp. 369–400. Vienna, Austria (1996)
31. Plamondon, R., Srihari, S.N.: On-line and off-line handwriting recognition: a comprehensive survey. IEEE Trans. Pattern Anal. Mach. Intell. **22**, 63–84 (2000)
32. Saabni, R., El-Sana, J.: Justifying holistic approach for arabic script recognition. Technical report, Ben Gurion University of the negev, Israel (2008)
33. Saabni, R., El-sana, J.: Hierarchical on-line arabic handwriting recognition. In: 10th International Conference on Document Analysis and Recognition (ICDAR), pp. 867–871. Barcelona, Spain (2009)
34. Solimanpour, F., Sadri, J., Suen, C.Y.: Standard databases for recognition of handwritten digits, numerical strings, legal amounts, letters and dates in farsi language. In: Proceedings of the 10th IntlWorkshop on Frontiers in Handwriting Recognition (IWFHR), pp. 3–7, France (October 2006)
35. Souici, S.T., Sellami, L.M.: Off-line handwritten arabic character segmentation algorithm: Acsa. In: Eighth International Workshop on Frontiers in Handwriting Recognition, pp. 452–457 (2002)
36. The Unipen Website: http://hwr.nici.kun.nl/unipen/unipen-history.html
37. Varga, T., Bunke, H.: Comparing natural and synthetic training data for on-line cursive handwriting recognition. In: 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR'04), pp. 221–225 (2004)
38. Varga, T., Bunke, H.: Generation of synthetic training data for an hmm-based handwriting recognition system. In: ICDAR '03: Proceedings of the Seventh International Conference on Document Analysis and Recognition, pp. 618–622, IEEE Computer Society, Washington, DC, USA (2003)
39. Varga, T., Kilchhofer, D., Bunke, H.: Template-based synthetic handwriting generation for the training of recognition systems. In: Proceedings of the 12th Conference of the International Graphonomics Society, pp. 206–211 (2005)
40. Viard-Gaudin, C., Lallican, P.M., Binter, P., Knerr, S.: The ireste on/off (ironoff) dual handwriting database. In: Proceedings of the Fifth International Conference on Document Analysis and Recognition, ICDAR '99, pp. 455–458. IEEE Computer Society, Washington, DC, USA (1999)
41. Wang, J., Wu, C., Xu, Y.-Q., Shum, H.-Y., Ji, L.: Learning-based cursive handwriting synthesis. In: IWFHR '02: Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02), pp. 157–162. IEEE Computer Society, Washington, DC, USA (2002)
42. Zagoris, K., Papamarkos, N., Chamzas, C.: Web document image retrieval system based on word spotting. In: IEEE International Conference on Image Processing, 2006, pp. 477–480, 8–11 October 2006