Word Extraction and Recognition in Arabic Handwritten Text
Nabil Aouadi and Afef Kacem Echi
Pages 17 – 23
DOI: http://dx.doi.org/10.21700/ijcis.2016.103

Word Extraction and Recognition in Arabic Handwritten Text

Nabil Aouadi and Afef Kacem Echi

LaTICE Laboratory, University of Tunis
Avenue Taha Hussein Montfleury, 1008 Tunis, TUNISIA
Nabil.aouadi@utic.rnu.tn, afef.kacem@esstt.rnu.tn

Abstract: *Segmenting arabic manuscripts into text-lines and words is an important step to make recognition systems more efficient and accurate. The major problem making this task crucial is the word extraction process: first, words are often a succession of sub-words where the space value between these sub-words do not respect any rules. Second, the presence of connections even between non adjacent sub-words in the same text-line, makes word's parts identification and the entire word extraction difficult. This work proposes an automatic system for arabic handwritten word extraction and recognition based on 1) localizing and segmenting touching characters, 2) extracting real sub-words and structural features from word images and 3) recognizing them by a Markovian classifier. The performance of the proposed system is tested using samples extracted from historical handwritten documents. The obtained results are encouraging. We achieved an average rate of recognition of 87%.*

Keywords: *Arabic handwriting recognition, Touching letters, Text-line Segmentation, Word Segmentation, Structural Feature Extraction, Word recognition, Hidden Markov Model*.

Received: June 10, 2016 | ***Revised:*** June 25, 2016 | ***Accepted:*** August 25, 2016

1. Introduction

Many systems or off-line recognition of Arabic script have been proposed in the objective to transliterate large number of Arabic manuscripts into machine readable. But, a lot of problems are raised by Arabic handwriting recognition systems due to text-line inclination, touching words, overlaps, ligatures, irregular spaces between words, etc.

This work is an attempt to overcome some of these problems. We mainly focused on the problem of touching letters which are connected components produced when adjacent letters touch or overlap each other (see Fig. 1). As Arabic alphabet is composed of 28 letters where 21 of them are ascenders and/or descenders, many vertically touching or overlapping letters can be found, especially in unconstrained Arabic manuscripts with small inter-lines spacing or when we practice calligraphy where terminal letters are with big descending. Segmenting these ambiguous connected components means their separation into individual letters. To achieve that, we propose an automatic system for Arabic handwritten word extraction and recognition based on 1) localizing and segmenting touching letters, 2) extracting real sub-words and structural features from word images and 3) recognizing them by a Markovian classifier.

This paper is organized as follows. In section 2, we present some related works. In section 3, we detail the different steps followed by the proposed system. Experimental results are reported in section 4 and some conclusions are drawn in section 5.

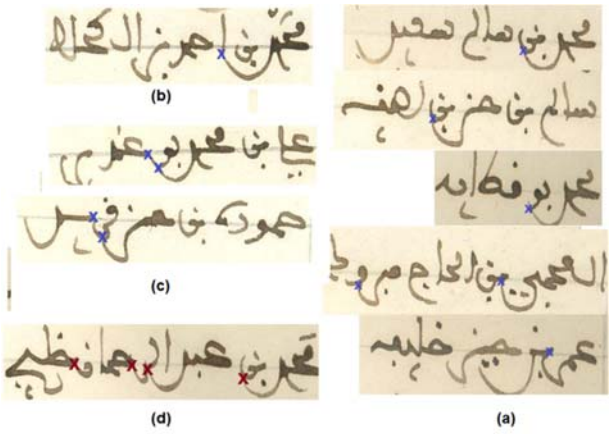


Figure 1. Samples of touching letters in the used Arabic Manuscripts

2. Related Works

The recognition of Arabic handwriting strongly depends on accurate segmentation. As many researchers have emphasized either segment-free

based methods or letter or stroke based approaches, words segmentation has not been well addressed, especially for Arabic writing.

In [22], authors have presented a method for segmenting Arabic handwritten documents into text-lines and words. Text-line segmentation is addressed by the horizontal projection profile. This technique promotes the estimation of text-line spacing. Word extraction is based on an adaptation of a known method, gap metrics. The method exploits the membership values of a clustering algorithm to identify segmentation thresholds as "within word" or "between words" gaps. The proposed method is tested on the benchmarking datasets of Arabic handwritten text recognition research, and very promising results were achieved, with an 84.8% correct extraction rate.

In [23], authors proposed a statistical analysis to determine an optimal threshold for word segmentation. By using knowledge of potential positions of the baseline, more accurate results are obtained in comparison with those without knowledge support. A component-based method is introduced to segment words from handwritten text. As noted by the author, this method is useful and more flexible than segment-free based methods as it can make good use of the component parts of image in further recognition. It is also simpler and more robust than letter-based methods because the letter has much difficulty in effectively segmenting arbitrary handwritten characters.

In Arabic writing, it is difficult to separate words from each other, especially when people write with calligraphy. We think that distance information is very useful for segmenting words, but improvements still desirable.

3. Proposed System

The proposed system segments the used manuscripts into text-lines and recognizes words composing them. Below is a description of the different steps followed by our system.

3.1 Touching Letter Extraction

Text-lines are a sequence of connected components belonging to the same alignment. Touching words are connected components running simultaneity into two adjacent words or text-lines. Before touching letter extraction and segmentation, we applied Sauvola local image thresholding [6] followed by a morphological operation to restitute thinned letters.

Note that TLs can be horizontally/left-right, when they occur between successive letters or words of the same text-line or vertically/up-down, when they belong to consecutive text-lines.

To extract vertically/up-down TLs, we adapted the Ouwayed and Belaïd's method [7] where each connected component *ccx*, belonging to two vertically adjacent text-lines, involve TLs of connected words or sub-words (see Fig. 2). To extract these TLs:

1. Baselines are detected from the input image,
2. A labeling process is accomplished and *ccx* that belong simultaneously to both adjacent text-lines are localized,
3. The text-line skeletons are extracted sing Zhang and Suen thinning algorithm [8],
4. The junction point (pixel having at least three neighbors in skeleton images), near the minima axis (valley in the horizontal projection profile between the two connected text-lines which coincides with the middle line between their baselines), is localized,
5. Using the junction point as center, the TL's connection zone is delimited where the TL's width is the $\frac{W_{ccx}}{4}$ and the TL's height is the $\frac{H_{ccx}}{4}$.

For horizontally/left-right TLs, we propose a method based on curve convexity analysis. The proposed method includes three main steps: 1) Baseline extraction, 2) Text-line's skeleton computing and 3) Junction point localization.

We find that most of touching words occur at terminal letters under text-line's baselines in the used manuscripts. Text-line baselines are extracted as the maxima in the manuscript's horizontal projection profile. To localize the TLs in words of the same text-line, the text-line lower part's skeleton is extracted. The image's skeleton is formed by several branches of thickness equal to one pixel around junction or intersection points. We note that junction point, resulting from an overlap between two disjoint letters, is usually associated to a convex curve turned to right. To discard junction points which do not correspond to TLs, only big branches were retained (ignoring diacritical and noise components). We then analyze the convexity of these branches. When a retained branch B (a set of pixels) has not an imperfect convex, we will consider it as part of TL only if there is a sufficient number (an empirical threshold fixed to 310) of pixel's couple *(x,y)* that satisfies convexity equation. Let X be a convex real vector space and let $B: X \rightarrow \Re$ be a function, B is convex if $(x,y) \in X \times X$ and $\lambda \in]0;1[$; $B(\lambda * x + (1-\lambda) * y) \leq (\lambda * B(x) + (1-\lambda) * B(y))$.

We used 1500 text-lines to evaluate the TL detection. We found that in 94% of cases, the junction points which correspond to real horizontally TLs are well localized. We also achieved almost the same accuracy for vertically TLs. The main errors are due to the text-line's skeleton computing step.

Figure 2. Detection and extraction of vertically TLs: (a) Original image and the identified starting pixel for baseline detection, (b) Binarisation followed by a morphological operation and baseline detection, (c) ccx identification, (d) ccx labeling, (e) Skeleton image, (f) Middle line identification, (g) Junction point localization, (h) TL extraction (in this case it is a connection between letters ج and أ).

3.2 Touching Letter Segmentation

To segment found TLs, we approximate them to models stored in a codebook with their known segmentation (part *A* and part *B* composing respectively the first and the second letter). Thus, there are two steps before TLs can be segmented: 1) a recognition step to find the most similar model for an input TL and 2) an approximation step to estimate the transformation aligning the found model to the TL. Once the TL is aligned to its most similar model, we use and adjust the midpoints of the model's parts to segment the TL.

TL's Model Recognition and Transformation: In this step, the objective is to look for the most similar model for the TL to be segmented. Different TL's models are stored in a codebook. For each model, we associate its two parts to it as references for TL segmentation. These models are organized into levels with a representative element for each group using a clustering algorithm [9]. To find the most similar model for an input TL, we compare these latter to all representative elements in the codebook, using a similarity metric computed from the shape context descriptor as proposed by Belongie [10]. This descriptor has the advantage to cover both steps of recognition and transformation since similarity is computed by solving the correspondence between shapes (TL and model) and estimating the aligning transform with the Thin Plate Spline (TPS) function [11].

TL Segmentation: So far, we have the most similar model for a given TL and the estimated TPS transformation parameters that align the model to the TL. It then becomes possible to exploit the correct segmentation of the model and to use the two model's parts midpoints to assign TL's pixels to the closest model's part midpoint. As the assignment can be ambiguous for pixels in the shared zone which can be allocated to the wrong part, we proposed to adjust midpoints before being used. For more details, see [20]. In Fig. 3, we display some TL segmentation results.

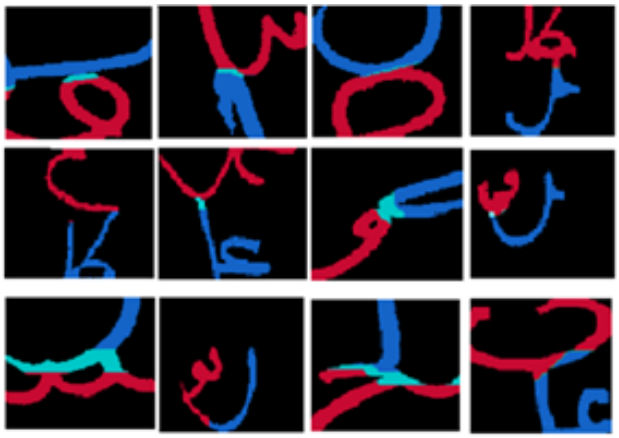


Figure 3. Samples of TL segmentation in Arabic manuscripts

3.3 Word Extraction

Once TLs are segmented, each text-line is composed of separated sub-words as shown in Fig. 4 and word extraction can be performed.

Figure 4. Sub-words extraction after TL segmentation

To extract words, we combined sub-words based on the analysis of geometric relationship of the adjacent sub-words, as accomplished by [21]. We first combined the letter "ا" (recognized based on its width) with the nearest sub-word and then computed the white space distances between them. We classified the previously white spaces as either inter-word or inter-part of word gaps. A threshold equal to 4 has been fixed for the maximum combined sub-words. Results are encouraging but needed to be enhanced (See Fig. 5).

Figure 5. Word extraction result: in red cases of over-segmentation

3.4 Word Recognition

Since it is difficult to segment words into letters, we opted for a global approach. The idea is to extract structural features from word image and submit them to a Markovian classifier, as explained in the following subsections. We have implicitly divided the word image horizontally, into an upper and a lower band according to baseline and vertically, into three strips (begin, middle and end). These three portions representing the beginning, middle and the end of the word are not equal in size. The middle portion is twice the portion of the beginning and the end. This inequality is due to that in Arabic writing, the information is concentrated in the middle.

Feature Extraction: We extracted structural features which are intuitive aspects of writing, such as loops, stems, legs, diacritic signs because we believe that words can be represented by this type of features with tolerance to style variations and distortions. In fact, structural features are based on topological and geometrical properties and the handwritten Arabic character has no fixed pattern, but has fixed geometrical features. That is the shapes of handwritten Arabic characters differ between writers, but the geometrical features are always the same. Structural features may also encode some knowledge about word structure or may provide knowledge as to what sort of components make up that word.

Note that Arabic manuscripts reveal the complexity of the feature extraction, especially for the features choice (discontinuity of the writing, multiple connections of sub-word, complex ligatures, etc.). Due to variability, complexity and imprecision in Arabic handwriting, extracting structural features that represent words is a hard task. We proposed a method to extract some structural features, mainly loops, stems, legs, diacritic, considering their position in the word: at the beginning, in the middle or at the end, in the upper, central or lower bands. It is a free-segmentation method where the objective is to detect the presence of letters and to get a global vision of words while avoiding word segmentation problems.

To this end, we extracted the word's baseline and deduced its upper, central and lower bands (see Fig. 6(a)). Afterwards, stems, legs and loops are respectively extracted from the upper, the lower and the central bands. Diacritic signs are extracted from both upper and lower bands since they generally occur over and/or below the word's baseline. Word description is then performed from right to left as a sequence of structural features gathered from each band. For further details, see [18].

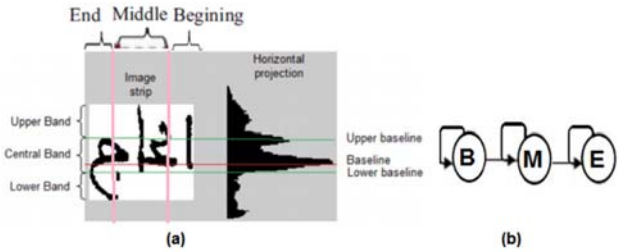


Figure 6. (a) Possible positions of the extracted features, (b) HMM structure

Table 1 presents a codification of the extracted features. To evaluate feature extraction results, let C be a feature's code, Ch_{Ex} be the feature's codes chain that should be extracted and Ch_{Ret} be the returned feature's codes chain by the proposed system, then we can compute the following metrics:

- Recovery means how many codes of the exact feature chain, exist in the returned chain,
- Configuration means how many codes of the exact feature chain exist in order in the returned chain,
- Similarity expresses the rate of resemblance between the exact chain of features and the returned chain. It is computed as follows where λ, set at a selected value (here 0.5) is a variable that favors an evaluation criteria (recovery or configuration) and varies from 0 to 1:

$$Similarity = (1-\lambda) * \text{Recovery} + \lambda * Configuration$$

We also used a second evaluation method based on Levenshtein distance which is a string metric for measuring the amount of difference between two sequences. This distance is defined as the minimum number of edits needed to transform one sequence into the other, with the allowable edit operations being insertion (case of feature extracted in superfluous), deletion (case of not extracted feature), or substitution (case of not correctly extracted feature) of a single feature. Table 2 displays some obtained results evaluated using the Recovery, Configuration and Similarity criteria, the Levenshtein distance, the Recall (the number of correctly extracted features, divided by the total number of features that must be extracted) and Precision values (the number of correctly extracted features, divided by the total number of extracted features).

Most of feature extraction errors can be attributed to the writing style and the poor quality of some data samples like:

- A group of two diacritic points can be written in the form of one or two related components. A group of three points may result in one, two or three related components depending on the writing style,
- A group of two or three letters is linked vertically in the beginning of the word,
- The same letter can be written in two different ways at the end of the word. Consequently the same word can be written in different ways.

Word Modeling: In this step, each word is modeled by a unique HMM. As Arabic is written from right to left, HMM's topology is sequential from right to left as shown in figure Fig. 6(b). There are only three states where each state corresponds to a word's region (the beginning, the middle and the end). The training step is performed by the Baum-Welch algorithm.

Note that when dealing with higher length words, we will see more structural features. Loop in the HMM structure tells about staying in the same state, which means that the proposed HMM takes care small and large words. Therefore, the HMM, could eliminate paths that are not promising early by computing probability from the first observations. To recognize a given word, its image is tested on all HMMS and it is assigned to the HMM class which gives the highest probability.

Table 1. Structural Features Codification

Feature	Code	Feature	Code	Feature	Code
Loop at the Beginning of the word	a	Loop in the Middle of the word	b	Loop in the End of the word	c
One diacritic Point Up at the Beginning	d	Two or three Points Up at the Beginning	e	One diacritic Point Up in the Middle	f
Two or three Points Up in the Middle	g	One diacritic Point Up at the End	h	Two or three Points Up at the End	i
One diacritic Point Down at the Beginning	j	Two or three Points Down at the Beginning	k	One diacritic Point Down in the Middle	l
Two or three Points Down in the Middle	o	One diacritic Point Down at the End	p	Two or three Points Down at the End	q
Stem Alif at the Beginning	r	Stem Alif in the Middle	s	Stem Alif at the End	t
Stem Kef in the Beginning	u	Stem Kef in the Middle	v	Stem Kef the End	w
Leg Noun at the Beginning	x	Leg Noun in the Middle	y	Leg Noun at the End	z
Leg Raa at the Beginning	A	Leg Raa in the Middle	B	Leg Raa at the End	C
Leg Haa at the Beginning	D	Leg Haa in the Middle	E	Leg Raa at the End	F

4. Experimentation

All proposed methods, described here, have been tested on a subset of the Tunisian National Archive collection and the IFN-ENIT [19]. We are especially interested by Arabic manuscripts of the 19th century which include Tunisian ancient personal names. To evaluate the word recognition method, we used a database containing 234 different words. Table 3 shows the basic allocation for some names

Table 3. Features Extraction Results

Feature set	Recall	Precision	F-Measure
Personnal names	0.89	0.89	0.89
IFN-ENIT names	0.78	0.82	0.80

5. Conclusion and Future Work

For manuscript segmentation into text-lines and words, we firstly proposed to extract touching letters between successive text-lines or words of the same text-line. Then, we segmented the TCs them with reference to a set of models, stored in a codebook with their prior-known segmentation, using shape context descriptor, an interpolation function: the thin plate spline transformation and the midpoints of the most similar model's parts. For word recognition, we extracted some structural features from word's image and trained a classic right-left Hidden Markov Model. Experiments are carried on a set of ancient Arabic manuscripts and the IFN-ENIT standard database. The obtained results are encouraging: an average recognition rate is around 87%. This opens opportunities in the field of text-line segmentation, feature extraction and selection and classifiers for Arabic manuscript recognition.

References

[1] A. Amin, *Off-line Arabic character recognition: a survey*, Proceedings of the Fourth International Conference on Document Analysis and Recognition, Volume 2, pp. 596-599, 1997.

[2] M. A. Mahjoub, N. Ghanmi N., K. Jayech and N. Amara Essoukri, *Proposition d'un modèle de réseau bayésien dynamique appliqué à la reconnaissance de mots arabes manuscrits*, Journées Francophones sur les réseaux bayésiens, pp. 11-13, 2012.

[3] J. H. lKhateeb, J. Ren, J. Jiang and H. Al-Muhtaseb, *Offline handwritten Arabic cursive text recognition using Hidden Markov Models and re-ranking*, Pattern Recognition Letters, volume 32, Number 8, pp. 1081- 1088, Elsevier, 2011.

[4] S. Masmoudi and H. Amiri, *Reconnaissance de mots arabes manuscrits par modélisation markovienne*, 6ème Colloque International Francophone sur l'écrit et le Document, pp. 473-482, 2000.

[5] K. Jayech, N. Trimech, M. A. Mahjoub and N. Ben Amara, *Dynamic hierarchical Bayesian network for Arabic handwritten word recognition*, Fourth International Conference on Information and Communication Technology and Accessibility (ICTA), pp. 1-6, 2013.

[6] J. Sauvola, T. Seppänen, S. Haapakoski and M. Pietikäinen, *Adaptive document binarization*, Proceedings of the Fourth International Conference on Document Analysis and Recognition, pages 147-152, 1997.

[7] N. Ouwayed and A. Belaïd, *Separation of overlapping and touching lines within handwritten Arabic documents*, Computer Analysis of Images and Patterns, pp. 237-244, 2009.

[8] T. Y. Zhang and C. Y. Suen, *A fast parallel algorithm for thinning digital patterns*, Communications of the ACM, Volume 27, number 3, pp. 236-239, 1984.

[9] E. Schaeffer Satu, *Graph clustering*, Computer Science Review, Volume 1, Number 1, pp. 27-64, Elsevier 2007.

[10] S. Belongie, J. Malik and J. Puzicha, *Shape Matching and object Recognition Using Shape Context*, IEEE transactions on pattern analysis and machine intelligence, pp. 509-522, 2002.

[11] F.L. Bookstein, *Principal Warps: Thin-Plane Spline and the Decomposition of Deformations*, IEEE transactions on pattern analysis and machine intelligence, 1999.

[12] N. Aouadi, S. Amiri and A. Kacem Echi, *Segmentation of Connected Components in Arabic Handwritten Documents*, Procedia Technology, Vol. 10, pp. 738-746, Elsevier 2013.

[13] N. Aouadi, A. Kacem and A. Belaïd, *Segmentation of Touching Component in Arabic Manuscripts*, 14th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 452-457, IEEE, 2014.

[14] N. Aouadi, A. Kacem Echi and A. Belaïd, *A recognition based approach for segmenting touching components in Arabic manuscripts*, 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 21-25, IEEE, 2015.

[15] N. Azizi, N. Farah, M. T. Khadir and M. Sellami, *Arabic handwritten word recognition using classifiers*, Recent Advances in Intelligent Information Systems, pp. 978-83, 2009.

[16] R. El-Hajj and L. Likforman-Sulem and C. Mokbel, *Arabic handwriting recognition using baseline dependant features and hidden Markov modeling*, Proceedings of Eighth International Conference on Document Analysis and Recognition, pp. 893-897, IEEE, 2005.

[17] A. Kacem, N. Aouïti and A. Belaïd, *Structural features extraction for handwritten Arabic personal names recognition*, International Conference on Frontiers in Handwriting Recognition (ICFHR), 2012.

[18] A. Kacem, N. Aouïti, A. Khémiri and N. Aouadi, *Système à base de MMC pour la reconnaissance de noms propres manuscrits Arabes*, Proceedings of colloque International sur le Document Electronique, Tunisia, 2012.

[19] M. Pechwitz, S. Maddouri Snoussi and V. Märgner, N. Ellouze, H. Amiri and others, *IFN/ENIT-database of handwritten Arabic words*, Proceedings of CIFED, Volume 2, pp. 127-136, Citeseer, 2002.

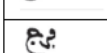
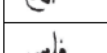
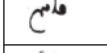
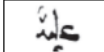
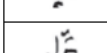
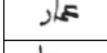
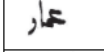
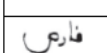
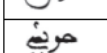
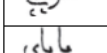
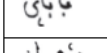
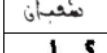
[20] N. Aouadi and A. Kacem, *A proposal for touching component segmentation in Arabic manuscripts*, Pattern Analysis and Applications, pp. 1-23, Springer, 2016.

[21] G. Louloudis, B. Gatos, I. Pratikakis, C. Halatsis, *Text-line and word segmentation of handwritten document*, Pattern Recognition, Volume 42, No 12, pp. 3169-3183, 2009.

[22] A. Al-Dmour and F. Fraij, *Segmenting Arabic Handwritten Documents into Text-lines and Words*, International Journal of Advancements in Computing Technology, pp. 109-119, 2014.

[23] J. H. AlKhateeb, J. Jiang, J. Ren and S. Ipson, *Interactive Knowledge Discovery for Baseline Estimation and Word Segmentation in Handwritten Arabic Text*, Recent Advances in Technologies, Maurizio A Strangio (Ed.), 2009.

Table 2. Samples for Structural Features Extraction

Word	Features to be extracted	Extracted features	Recovery	Configuration	Similarity	Decision	Precision	Recall	Distance
	qppbAnnlj	qppbAnnlj	1	1	1	oui	9/9 =1	9/9=1	0
	cDvlj	cDvlj	1	1	1	Oui	9/9 =1	9/9=1	0
	qdD	qdD	1	1	1	Oui	3/3 =1	3/3=1	0
	qiil	qiim	0.75	1	0.81	Non	0.75	0.75	1
	qzCee	qzee	1	1	1	Oui	1	4/5=0.8	0
	qA	qA	1	1	1	Oui	2/2	2/2	0
	Stszygmd	zygmd	1	1	1	Oui	5/5	5/7=0.73	2
	pbwzd	pbwzd	1	1	1	Oui	5/5	1	1
	qaCzoim	aCzoim	1	1	1	Oui	6/6	6/7=0.86	1
	qpxmlj	qpxmlj	1	1	1	Oui	6/6	6/6	0
	rxhle	rxhle	1	1	1	Oui	5/5	1	0
	qDe	qDe	1	1	1	Oui	3/3	3/3	0
	Azz	Azz	1	1	1	Oui	3/3	3/3	0

Nabil Aouadi received the engineer diploma degree in 1993 in Computer Sciences from the Faculty of Science of Tunis in 1993 and MS in 2008 from College of Science and Technology of Tunis. Currently he is a PhD student and a member of LaTICE: Laboratory for Technologies of Information and Communication at the High School of Sciences and Techniques of Tunis.

Dr. Afef Kacem Echi received M.Sc. and Ph.D. degrees in Computer Sciences from the National School of Computer Sciences of Tunis in 1997 and 2001 respectively. Since 2000, she has been an assistant in the computer science department at the Faculty of sciences of Monastir, and was appointed Assistant Professor there in 2002. Dr. Kacem is a responsible member of the research area: Analysis and Recognition of Handwriting and document in LaTICE: Laboratory for Technologies of Information and Communication at the National High School of Engineers of Tunis. She has authored over 50 articles in various national and international journals and conference proceedings.