Arabic Word Spotting Based on Key-Points Features

Hamza GHILAS*, Meriem GAGAOUA*, Abdelkamel TARI*, Mohamed CHERIET†
*Université de Bejaia, Algérie, {hamzaghilas, gagaouameriem}@yahoo.fr, tarikamel59@gmail.com
† Ecole de Technologie Supérieure, Montréal (CANADA), Mohamed.Cheriet@etsmtl.ca

Keywords: word spotting; historical Arabic documents; features extraction; word matching.

Abstract

Word spotting is an efficient alternative to OCR (Optical Character Recognition) for understanding historical manuscripts by matching features of words. However, due to the large variability in the Arabic script, the retrieving results are still not satisfactory. In this work a novel method based on key-points features is proposed. Key-points features can capture both topological and local characteristics. Feature vectors were extracted from key-points and a distance function was proposed for word matching. To reduce space matching, the connected components (CCs) were clustered into meta-classes in a soft manner using a Gaussian mixture model, then a query CC was spotted by matching it only with CCs which belong to its meta-class. The experiments were carried out on a benchmark consisting of Arabic historical manuscripts by IBEN SINA and shows promising results.

1 Introduction

The digitization of historical documents is an efficient way to preserve them. Nowadays a large amount of digitized documents is available; thus the need to make them searchable becomes necessary. Since manual transcription of these documents is time consuming and the automatic transcription by Optical Character Recognition (OCR) is far from to be practical, the word spotting technique becomes a promising alternative. Word spotting aims to locate in a target document, regions that are most similar to a query word without recognizing its characters.

Word spotting methods are classified in two main categories, learning based and template matching based methods. The first type is inspired by OCR; where models of keywords are trained using labeled data which are used to recognize queries in the target document [1-6]. These methods allow string querying but they suffer from the need of large labeled databases to train the system and the user is limited to choose the queries in a finite vocabulary. In other hand; template matching methods match directly the query image with word images in the target document without using labeled data. The query is arbitrary and can be selected in the target document or synthesized [7-10].

Word spotting has been widely implemented for Latin-based scripts, but studies investigating word spotting in Arabic handwritten documents are very scarce. Arabic script is complex by its cursive nature. Contrary to the Latin script, in Arabic script the form some letters change according to their position in the word. Generally, the Arabic words are not contiguous and consist of several connected components (CC) or sub-words. This complexity associated with degradations and noises in historical documents make the implementation of word spotting for Arabic handwritten documents more challenging.

Dynamic time warping (DTW) with profile features is the first word spotting method proposed by Rath and Manmatha[11]. This method is time consuming because it is a dynamic programming algorithm. In Latin script the small words are ignored in the matching process but this is not possible in Arabic script. Moghaddam and Cheriet [12] overcome this problem by using Self-Organizing Maps to initially cluster CCs to libraries depending on their shape complexity. Then a CC is spotted in its own library using Euclidean distance measure enhanced by rotation and DTW.

Wshah et al [6] proposed a script independent word spotting for handwritten documents, where the gradient and intensity features are extracted from a sliding window. The orientations of the gradient in eight directions are accumulated in a histogram and the intensity feature is the black-to-white pixel ratio. Gradient features are also used by Khayyat et al in [3]. A mean filter is applied to the binary images to obtain gray scale images which are used to compute the gradient. Initially the gradient is quantized in 32 directions in 9 x 9 blocks. After dimensionality reduction a feature vector of dimension 400 is obtained. These methods are learning based and they need a correctly labeled data which is a time consuming task.

Cheriet and Moghaddam [13] used the skeleton of CC image to extract topological and geometrical features. The shape of the script is analyzed by extracting descriptors using end-points, branch-points and dots. A distance function is proposed to rank the CCs in the target document according to their similarity with the CC query.

Sari and Kefali [14] used the structural features such as loops, ascenders and descenders. The document was segmented to CCs and a coding process was done to represent the document by an ASCII file and the spotting was done as text search. This method is efficient in documents with good quality like printed documents.

An efficient method for features extraction to capture the complex characteristics of the Arabic handwritten and tolerate noise and degradation is needed. In this paper and for the first time a feature extraction method based on key-points is proposed. First, a set of key-points is extracted using the skeleton of the CC image. Second, a feature vector is extracted at each key-point in the binarized image of the CC. A distance function is proposed to match two sequences of feature vectors extracted from CCs. In the remaining of this paper, we present our contribution and offer a detailed explanation of the proposed method. Also the experiment analysis on a benchmark of an historical Arabic document is provided.

2 Contributions

In this paper a template matching based word spotting system for Arabic historical documents is proposed. The CCs are spotted instead of Arabic words to overcome the word segmentation issue. The main contributions in this paper are:

- A candidate selection algorithm based on soft clustering was proposed. The candidate selection reduced the space matching at spotting stage.
- Key-points were extracted from the skeleton of a CC and an algorithm was used to remove the noisy and unnecessary key-points.
- The key-points were then described by a set of features extracted from the binarized image of a CC.
- A distance function was proposed to rank the CCs in the target document according to their similarity with the query CC.

3 Candidate selection

Unlike Latin script where small words are insignificant, in Arabic script all parts of the text are important, then the pruning techniques are not relevant. To reduce the number of images to match with the query, CCs were clustered into meta-classes according to their shape complexity. Moghaddam and Cheriet [12] used as described above Self-Organizing Maps to create libraries of CCs. The hard clustering of the CCs is not necessary, then in this work, the CCs are clustered to meta-classes in a soft manner where meta-classes can overlap and a CC can belong to more than one meta-class. This may reduce errors due to clustering. A Gaussian mixture model is used to cluster CCs in a soft manner to meta-classes.

For each CC, a vector of six features was extracted. Namely, the aspect ratio, the ink density (the ratio between black and white pixels), the number of minima in the vertical projection, the number of holes and the number of ascenders and descenders.

Given a set C = $\{CC_1 \ldots CC_n\}$ of CCs extracted from the document. To cluster the CCs in k classes, a Gaussian mixture model with k component is fitted using the expectation maximization algorithm [15]. The density function is given by(1).

$$F(cc_i)=\sum_{j=1}^{K} w_j N\left(\mu_j,\sigma_j\right) \qquad (1)$$

The component $w_j N\left(\mu_j,\sigma_j\right)$ in (1) is the probability that the connected component CC_i belongs to the class j, then a vector of probability with k elements is associated to each CC. Finally, the overall of CCs probabilities is given by a matrix P(n x k); Where P_{ij} is the probability that the CC_i belongs to the class j. Once the probability matrix is estimated Algorithm 1 is used to classify CCs in k classes.

Assuming a CC can be assigned at most to m classes. Then the CC is assigned to the dominant classes having their probability greater than 1/(m+1). If all classes probability are less than 1/(m+1); the m classes with best probabilities are considered. By classification, each CC_i was labeled with one, two or more classes allowing the reduction of clustering errors and ensures that a CC query will be matched with the most of its occurrences during the spotting process.

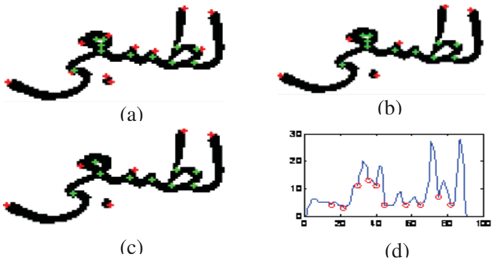
```
Algorithm 1
Inputs: p(nxk): probability matrix.
        M: number of cluster overlaps.
Outputs: classes of CCs.
For all Cci do
        Classes(CCi) = {$C_j$, $P_{ij} \geq 1/(m+1)$};
                If classes(CCi) = Ø then
                Classes(CCi) = {the m best classes};
                End if.
End for.
```

4 Key-points extraction

To capture characteristics of handwritten, it is not necessary to analyze the overall of word images known as time consuming [10]. In this work both end-points (EP) and branch-points (BP) are denoted by Key-points (KP). They were used to capture the salient information in the word image. First, all Key-points were extracted from the skeleton of the word by the *bwmorph* Matlab function. The KPs extracted by this function were noisy and must be cleaned (Figure 1). The cleaning was performed using two important parameters which are the average ink width (AIW) and a distance d in the skeleton. AIW estimated the average width of the stroke in the dataset. The distance d between two key-points is the number of pixels separating them in the skeleton image. A hypothesis assumes that two key-points must be separated by a distance greater than AIW.

4.1 Average ink width estimation.

For each CC in the dataset, the vertical projection profile was computed and the local minima were calculated in the projection profile. The mean of these local minima was considered as the average ink width for the word (Figure 1.d).



(a) (b) (c) (d)

Figure 1: (a): raw key-points, (b): noizy key-points deleted, (c): clean key-points, (d): minima in the projection profile.

4.2 End-points pruning

Lets $\mathrm{EP}_{\mathrm{raw}} = \{\mathrm{EP}_{\mathrm{i}}(y_i, x_i),\ \mathrm{i} = 1 \dots \mathrm{N}_{\mathrm{ep}}\}$ and $\mathrm{BP}_{\mathrm{raw}} = \{\mathrm{BP}_{\mathrm{i}}(y_i, x_i), \mathrm{i} = 1 \dots \mathrm{N}_{\mathrm{bp}}\}$ to be the raw sets of the extracted KPs by the *bwmorph* matlab function (Figure 1.a). EPs that are far from other EPs or BPs with a distance less than AIW were considered as noise and thus deleted. The clean EP set is defined by Equation (2).

$$\mathrm{EP}_{\mathrm{clean}} = \left\{\mathrm{EP}_{\mathrm{i}}(\mathrm{y}_{\mathrm{i}}, \mathrm{x}_{\mathrm{i}}) \in \mathrm{EP}_{\mathrm{raw}},\ \mathrm{d}\left(\mathrm{EP}_{\mathrm{i}}, \mathrm{EP}_{\mathrm{j}}\right) > AWI, \forall\ \mathrm{EP}_{\mathrm{j}} \in \mathrm{EP}_{\mathrm{raw}} \cup \mathrm{BP}_{\mathrm{raw}}\right\} \quad (2)$$

4.3 Branch-point pruning

To clean the raw BPs, the cleaned EPs were considered as raw BPs; then $\mathrm{BP}_{\mathrm{raw}} = \mathrm{BP}_{\mathrm{raw}} \cup \mathrm{EP}_{\mathrm{clean}}$. A weight w was associated for each BP in $\mathrm{BP}_{\mathrm{raw}}$ which was considered as the accumulated distances to its neighbors key-points. If the distance between two branch-points was less than AIW*;* the branch-point with minimum weight was deleted. The clean set of BP is defined by Equation (3).

$$\mathrm{BP}_{\mathrm{clean}} = \left\{\mathrm{BP}_{\mathrm{i}}(\mathrm{y}_{\mathrm{i}}, \mathrm{x}_{\mathrm{i}}) \in \mathrm{BP}_{\mathrm{raw}},\ \mathrm{d}\left(\mathrm{BP}_{\mathrm{i}}, \mathrm{BP}_{\mathrm{j}}\right) > AWI, \forall\ \mathrm{BP}_{\mathrm{j}} \in \mathrm{BP}_{\mathrm{raw}} \cup \mathrm{EP}_{\mathrm{clean}}\right\} \quad (3)$$

The cleaning process for the key-points was done by Algorithm 2, where Neighbors(kp) returns the key-points connected to kp by skeleton pixels.

Algorithm 2

```
Inputs: EPraw,BPraw: The raw sets of KPs.
        W: weights of BPs.
Outputs: EPclean,BPclean: The clean sets of KPs.

EPclean = Ø;
BPclean = Ø;
For all $ep_i$ in EPraw do
  If d($ep_i$, Neighbors($ep_i$)) > AIW then
     EPclean = EPclean U {$ep_i$};
  End if.
End for.

BPraw= BPraw U EPclean;
For all $bp_i$ in BPraw do
 Nbrs = Neighbor($bp_i$);
  For all $Nbr_j$ in Nbrs do
   If d($bp_i$,$Nbr_i$) < AIW then
     BPclean=BPclean U {$\mathrm{argmax}(W(bp_i), W(Nbr_j))$}
   End if.
  End for.
End for.
```

5 Image matching

A feature vector was extracted from each key-point then a distance function is used to compute the dissimilarity between the query and all CCs in the same meta-class to the query.

5.1 Feature extraction.

After KPs extraction, let's consider P = {p_1... p_n} to be the clean set of KPs extracted from a CC. For each KP p_i a feature vector v_i was extracted then the CC was represented by a set of features F={v_1... v_n}. Lets $C(y_c,x_c)$ to be the center of the CC image and $P_i(y_i,x_i)$ a KP (Figure 2), the features extracted from p_i are:

- The distance from P_i to the centre of the CC: it was computed as the amplitude of the vector $\overrightarrow{\mathrm{CP_i}}$
  $$\left\|\overrightarrow{\mathrm{CP_i}}\right\| = \sqrt{(\mathrm{y_i} - \mathrm{c_y})^2 + (\mathrm{x_i} - \mathrm{c_x})^2} \quad (4)$$
- The angle of the vector $\overrightarrow{\mathrm{CP_i}}$ to the horizontal axe. $\theta = \tan^{-1}\left((\mathrm{y_i} - \mathrm{c_y})/(\mathrm{x_i} - \mathrm{c_x})\right)$ (5)
- The multi-scale intensity at the KP p_i. It was computed by the Ink/background ratio in a circle centered at p_i with a radius r. Three scales are considered r=AIW, r=2xAIW and r=3xAIW.

$$\mathrm{I}_{s=1..3} = \frac{\sum \mathrm{I(y,x)}}{\sum 1\text{-}\mathrm{I(y,x)}}\ ,\sqrt{\left(\mathrm{y}\text{-}\mathrm{y_c}\right)^2 + (\mathrm{x}\text{-}\mathrm{x_c})^2} \leq \mathrm{s}{*}\mathrm{r} \quad (6)$$

5.2 Distance in features space.

Given two CCs images CC^a and CC^b described by key-points features of length L_a and L_b respectively. $\mathrm{KP^a} = \{\mathrm{kp}_1^{\mathrm{a}} \dots \mathrm{kp}_{\mathrm{La}}^{\mathrm{a}}\}$ and $\mathrm{KP^b} = \left\{\mathrm{kp}_1^{\mathrm{b}} \dots \mathrm{kp}_{\mathrm{Lb}}^{\mathrm{b}}\right\}$ where i =1…La and j = 1…Lb. The dissimilarity of CC^a and CC^b is computed based on the Euclidian distance between two key-point features $\mathrm{d_{euclidian}}(\mathrm{kp_i^a}, \mathrm{kp_j^b})$. For each key-point $\mathrm{kp_i^a}$ in KP^a we find in Kp^b the best matching key-point $\mathrm{kp_j^b}$ with the lowest distance. Then $\mathrm{match}(\mathrm{kp_i^a}) = \mathrm{kp_j^b}$ if :

$$\mathrm{d_{euc}}\left(\mathrm{kp_i^a}, \mathrm{kp_j^b}\right) < \mathrm{d_{euc}}\left(\mathrm{kp_i^a}, \mathrm{kp_k^b}\right), \forall\ \mathrm{k} \neq \mathrm{j}, \mathrm{k} = 1 \dots \mathrm{L_b}.$$

The final dissimilarity between two CCs images CC^a and CC^b is then computed by summing the distances between CC^a key-points and their best matches. The dissimilarity is defined by Equation (7).

$$\mathrm{D}\left(\mathrm{KP^a}, \mathrm{KP^b}\right) = \sum_{\mathrm{i}=1}^{\mathrm{La}} \mathrm{d_{euc}}\left(\mathrm{kp_i^a}, \mathrm{kp_j^b}\right) \quad (7)$$

Where $\mathrm{kp_j^b} = \mathrm{match}(\mathrm{kp_i^a})$

In order to enhance the similarity between images having more matching key-points, we normalize the dissimilarity value between two CCs images CC^a and CC^b as defined in Equation (8). Where $M_{a,b}$ is the number of matching points between CC^a and CC^b. An example of matching points is shown in Figure 3.

$$\mathrm{Dist}\left(\mathrm{KP^a}, \mathrm{KP^b}\right) = \mathrm{D}\left(\mathrm{KP^a}, \mathrm{KP^b}\right)\left(\frac{\left(\mathrm{La}-\mathrm{M_{a,b}}\right)^2 + \left(\mathrm{La}-\mathrm{M_{a,b}}\right)^2}{\left(\mathrm{La}^2 + \mathrm{M_{a,b}^2}\right)\left(\mathrm{Lb}^2 + \mathrm{M_{a,b}^2}\right)}\right) \quad (8)$$

Finally a CC is spotted by computing distances to other CCs which are in the same meta-class. A ranked list is created by sorting CCs in the ascending order according to their distances.

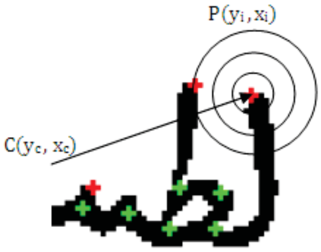


Figure 2: Key-points features extraction.

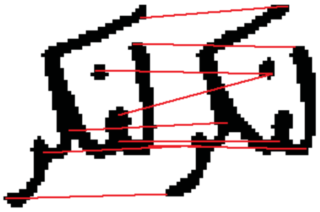


Figure 3: Matching points between two connected components

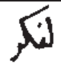
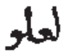
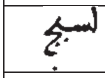
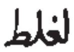
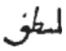
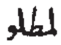
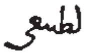
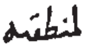
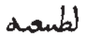
Queries	Number of instances	Proposed method				DTW with profile features			
		Top 10		AP (%)	Time response in seconds	Top 10		AP (%)	Time response in seconds
		Precision (%)	Recall (%)			Precision (%)	Recall (%)		
	22	70	31.8182	51.6452	5.9204	80	36.3636	44.8626	12.3687
	13	60	46.1538	52.1361	4.5012	90	69.2308	79.5586	5.0611
	12	90	75.0000	**95.9366**	5.6067	40	33.3333	45.4300	9.6093
	8	60	75.0000	83.6310	7.1940	50	62.5000	66.2957	23.0787
	27	100	37.0370	92.6246	6.2512	90	33.3333	70.8364	22.1770
	5	30	60.0000	68.3333	6.1887	30	60.0000	66.8421	14.5825
	11	90	81.8182	91.7063	4.7317	90	81.8182	95.3604	7.3350
	6	50	83.3333	89.5833	4.9860	40	66.6667	48.6652	7.3791
	5	40	80.0000	73.0952	7.7338	10	20.0000	22.8383	40.4709
			Mean	**77.63**	5.9492		**Mean**	0.6008	15.7847

Table 1: experimental results.

6 Experimental results

The preprocessing step was not the aim of this study, then it was needed to use a cleaned and binarized images. The IBEN SINA database was chosen [16]. This database includes 50 folios of historical Arabic manuscripts. In IBEN SINA, the images are binarized and segmented to connected components. A set of 10 pages were used to evaluation and include 5684 CCs. A truth label is associated for each CC. A set Q of 9 queries of the most occurring CCs in the document were manually selected. Queries were 4, 5, 6 and 7 character length (Table 1).

Candidate selection was performed with k = 8 chosen according to the number of characters of CCs ranging from 1 to 8 in the dataset. The number of classes overlaps was determined empirically and was optimal for m = 3. The candidate selection was evaluated by measuring the mean

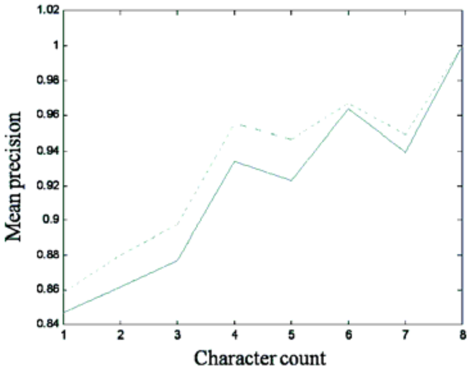


Figure 4: Mean precision by character count. Soft clustering (m=3) outperform hard clustering(m=1).

precision at each level for CCs character length which range from 1 to 8. For each CC the precision was measured by Equation (9) and the average precision for each category of CCs according to their character length was considered. In Figure 4, the average precision by character count of CCs is given. This allow to show that soft clustering by m=3 outperform hard clustering by m=1.

$$\text{Precision}(\text{CC}_\text{i})=\frac{\text{Max}_{\text{k}=1}^{\text{m}}(|\text{class}(\text{CC}_\text{j})=\text{k}|)}{|\text{CC}_\text{j}|} \qquad (9)$$

Where

CC_j is an occurrence of CC_i.

For word spotting evaluation, the experiments were carried out by comparing the proposed method with the well known DTW and profile features method. The recall and precision were considered in the TOP 10 retrieved results. The Average precision (AP) was measured for each query as the mean of the precisions obtained after retrieving each relevant CC. The Mean Average Precision (MAP) was measured as the mean value of AP of each CC.

$$\text{precision} = \frac{\text{TP}}{\text{TP+FP}} \qquad \text{recall} = \frac{\text{TP}}{\text{TP+FN}} \qquad (10)$$

Where TP is true positive, FN false negative and FP false positive. If the set of relevant CCs for a given query $q_j \in Q$ is $\{\text{RCC}_1, \dots \text{RCC}_{\text{mj}}\}$ and R_{jk} is the set of ranked retrieval results from the top result until the relevant RCC_k , then

$$\text{AP}(\text{q}_\text{j}) = \frac{1}{\text{mj}}\sum_{\text{k}=1}^{\text{mj}} \text{precision}(\text{R}_{\text{jk}}) \qquad (11)$$

And

$$\text{MAP}(\text{Q}) = \frac{1}{|\text{Q}|}\sum_{\text{j}=1}^{|\text{Q}|} \text{AP}(\text{q}_\text{j}) \qquad (12)$$

Table 1 shows the results obtained at the spotting stage with a corrected candidate selection. The results show that the proposed method outperforms DTW in both precision and time response for the majority of queries. An MAP of **77.63** was obtained for the proposed method with a mean time response of **5.94** seconds. On the other hand, an MAP of **0.60** with a mean time response of **15.78** seconds was the results of the well known DTW method. The word spotting system was also evaluated when the errors of the clustering stage are included and an MAP of **49.10%** was obtained for our method without any correction of clustering errors, against an MAP of **36.36%** was obtained for the DTW method.

Conclusion

In this work, a method for word spotting in historical Arabic documents was proposed. The proposed method outperforms the well known DTW method in both precision and time response. Word spotting was performed by matching CCs images using a set of features extracted from key-points instead of analyzing the overall of the ink pixels. The proposed method showed that representing Arabic script by key-points is a promising method.

The proposed method requires segmenting the document to CCs. Thus, it may be more interesting to segment the document to lines and perform the spotting in a set of line descriptors instead of CCs.

References

[1] Fischer, A., Keller, A., Frinken, V., and Bunke, H., Lexicon-free handwritten word spotting using character HMMs. Pattern Recognition Letters, 2012. 33(7): p. 934-942.

[2] Frinken, V., et al., Keyword spotting for self-training of BLSTM NN based handwriting recognition systems. Pattern Recognition, 2014. 47(3): p. 1073-1082.

[3] Khayyat, M., L. Lam, and C.Y. Suen, Learning-based word spotting system for Arabic handwritten documents. Pattern Recognition, 2014. 47(3): p. 1021-1030.

[4] Rodriguez-Serrano, J.A. and F. Perronnin, Synthesizing queries for handwritten word image retrieval. Pattern Recognition, 2012. 45(9): p. 3270-3276.

[5] Rodríguez-Serrano, J.A. and F. Perronnin, Handwritten word-spotting using hidden Markov models and universal vocabularies. Pattern Recognition, 2009. 42(9): p. 2106-2116.

[6] Wshah, S., G. Kumar, and V. Govindaraju, Statistical script independent word spotting in offline handwritten documents. Pattern Recognition, 2014. 47(3): p. 1039-1050.

[7] Almazán, J., Gordo, A., Fornés, A. and Valveny, E., Segmentation-free word spotting with exemplar SVMs. Pattern Recognition, 2014. 47(12): p. 3967-3978.

[8] Can, E.F. and P. Duygulu, A line-based representation for matching words in historical manuscripts. Pattern Recognition Letters, 2011. 32(8): p. 1126-1138.

[9] Rusiñol, M., et al., Efficient segmentation-free keyword spotting in historical document collections. Pattern Recognition, 2015. 48(2): p. 545-555.

[10] Zhang, X. and C.L. Tan, Handwritten word image matching based on Heat Kernel Signature. Pattern Recognition, 2015. 48(11): p. 3346-3356.

[11] Rath, T.M. and R. Manmatha. Word image matching using dynamic time warping. in 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings. 2003.

[12] 12. Moghaddam, R.F. and M. Cheriet. Application of Multi-Level Classifiers and Clustering for Automatic Word Spotting in Historical Document Images. in 2009 10th International Conference on Document Analysis and Recognition. 2009.

[13] 13. Cheriet, M. and R.F. Moghaddam, A Robust Word Spotting System for Historical Arabic Manuscripts, in Guide to OCR for Arabic Scripts, V. Märgner and H. El Abed, Editors. 2012, Springer London: London. p. 453-484.

[14] Sari, T. and A. Kefali, A search engine for Arabic documents. Actes du dixième Colloque International Francophone sur l'Écrit et le Document, 2008: p. 97-102.

[15] McLachlan, G. and D. Peel, ML Fitting of Mixture Models, in Finite Mixture Models. 2000, John Wiley & Sons, Inc. p. 40-80.

[16] Moghaddam, R. F., Cheriet, M., Adankon, M. M., Filonenko, K., and Wisnovsky, R., 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. 2010, ACM: Boston, Massachusetts, USA. p. 11-18.