Noname manuscript No.
(will be inserted by the editor)

Effective Recognition of Handwritten Arabic Text

Zennaki Mahmoud · Sadouni Kaddour · Mamouni El Mamoun

Received: date / Accepted: date

Abstract This paper presents an effective method for segmenting handwritten Arabic words based on projection histograms, as well as a comprehensive study of overlap letters, which represent a major problem in the recognition of Arabic script. The characters resulting from segmentation are recognized by an SVM classifier that was trained on a corpus of handwritten Arabic characters that also includes the main overlap letters identified in Arabic script. The proposed method was tested using the IFN/ENIT database and encouraging results were obtained.

Keywords Arabic text · handwritten · segmentation · projection histogram · support vector machines

1 Introduction

The automatic recognition of Arabic handwriting has been the subject of intense research, mainly due to the complexity of handwritten Arabic script. Great progress has thus been made in many domains such as the automation of postal mail, the processing of checks, the processing of forms, the automatic indexing of old manuscripts, etc. Two main approaches are used in recognition systems: the global approach, which is based on the recognition of entire words, and the analytical approach, which consists in segmenting words into characters and recognizing the characters separately. The global approach is only effective if the vocabulary is small, limiting its use to very specific areas. Other hybrid approaches have been developed without significant contribution.

In an analytical approach, segmentation into characters is a crucial step. This step is simple in the case of printed Latin text, but very difficult in the case of cursive writing, especially in Arabic writing ([12], [17]). Another problem makes the recognition of Arabic script more difficult: overlap characters. In Arabic script, writers are free to form certain groups of two or three letters linked vertically. In general, they are very complex to segment and are usually recognized directly as they are.

An efficient method of segmenting handwritten Arabic words and a detailed study of overlap letters are proposed in this work. The characters resulting from segmentation are recognized by an SVM classifier previously trained on a corpus of handwritten Arabic characters including the main overlap letters identified in Arabic script. To our knowledge, no work has addressed overlap letters so thoroughly, although they are the main cause of deterioration in recognition rates. Furthermore, the proposed segmentation algorithm combines the advantages of the main segmentation algorithms described in the literature.

This paper is organized as follows: Section 2 presents some recent studies on the segmentation and recognition of handwritten Arabic script. Section 3 focuses on overlap letters, while section 4 describes the proposed approach. The results and their analysis are discussed in section 5. Section 6 draws conclusions.

2 Related work

Despite intense research, the recognition of Arabic script remains a relevant issue and many challenges are still

M. Zennaki, K. Sadouni, E. Mamouni
Departement Informatique, Faculté des Mathématiques et Informatique, Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf USTO-MB, BP 1505 El M'naoeur, Oran, Algeria
E-mail: {mahmoud.zennaki,kaddour.sadouni,elmamoun.mamouni}@univ-usto.dz

open, such as segmentation and overlap letters. Several studies have been carried out based on entire word recognition (global approach) without segmentation [6]. Other researchers have assumed that characters are already segmented to avoid the segmentation step ([4], [7]). Therefore, this step is a challenge for researchers and needs to be improved [5].

Dupre proposed a segmentation algorithm based on a skeleton [2]. The aim was to identify certain patterns to deduce the candidates from cut-points. The detection of these patterns takes into account the calculation of curvatures and angles, which are compared with adjusted thresholds to obtain the desired result. Dupre reported that this approach is wrong in approximately 10% of the cases.

S. Madhvanath et al. [8] proposed a segmentation method applied to contours. The authors determined the best candidate cut-off points between letters, based on local extrema of the contour, which are associated to a proximity criterion. Segmentation based on the contour requires many adjustments before finding the decision criteria [19]. This trial and error is the common point of many image processing related to handwriting recognition.

Another more common segmentation method is the segmentation based on histograms. This simple and effective method was proposed by B. Yanikoglu and P. Sandon [18]. It consists of calculating projection histograms in several directions close to the vertical. The lines chosen are those which intercept the least number of black pixels, with a constraint of regular spacing in the image. This method shows its limits when the letters are very close or overlapped.

Another technique, using sliding windows, was presented by Tay et al. [15] and Mamouni et al. [9], which consists of cutting the image into vertical bands. This cutting can be regular or not, possibly with partial overlapping of the successive bands. This technique has the advantage of being simple, robust to noise, and independent to connectivity. The drawback of this method is that the generated sequence contains a lot of noise (overlapping of two successive letters).

Comparing these segmentation algorithms and recognition systems is a difficult task. The appearance of large-scale public databases such as the IFN/ENIT database [13] (32,492 images for a vocabulary of 937 names of Tunisian cities, and 411 writers) and the organization of competitions as part of the ICDAR conference, has enabled comparisons between systems and has allowed for rapid progress in recent years.

Today, the performance of the best Arabic script recognition systems seems to be close to what one would expect for Latin script for an equivalent vocabulary size, provided that overlap letters are not used frequently ([1], [3]). This is because overlap letters significantly degrade recognition rates, and few studies have addressed this problem in detail. For this end, a comprehensive review of overlap letters is presented in this paper.

Table 1 Arabic alphabet

Khaa خ	Haa ح	Jeem ج	Tha ث	Ta ت	Baa ب	Alif ا
Saad ص	Shin ش	Seen س	Zain ز	Raa ر	Thaal ذ	Dal د
Qaaf ق	Faa ف	Ghayn غ	Ayn ع	Dhad ظ	Taa ط	Dad ض
Yaa ي	Waw و	Ha ه	Noun ن	Meem م	Lam ل	Kaf ك

3 Arabic overlap letters

The Arabic alphabet includes 28 letters (Table 1) whose shape depends on their position in the word. Some letters can have four different forms: beginning, middle, end and isolated (Fig. 1), but, for most letters, the beginning/middle and end/isolated forms are identical to the near ligature. The presence of a ligature with the previous letter or with the next letter does not change the shape of the letter significantly (not any more than in Latin cursive handwriting).

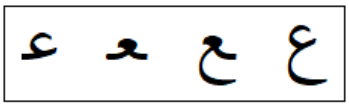

Fig. 1 Different form of an Arabic letter

Ligatures are always located at the level of the writing line, but some vertical ligatures can be found. In Arabic alphabet, 15 of the 28 letters have one or more dots. These diacritical marks are located either above or below, but never both. A diacritical mark is therefore a secondary component of a letter, which completes it or changes its meaning. The scripters can form certain groups of two or three letters linked vertically. These are the overlap letters (Fig. 2). Generally, they are very complex to segment, it would be more efficient to recognize them as they are. Based on the observation that the main weakness of current Arabic recognition systems is the overlap letters, an exhaustive search of these special letters was carried out and then the most relevant ones were selected for inclusion in a reduced alphabet. Therefore, the IFN/ENIT database (32,492 words) as well as the Holy Qur'an (over 77,000 words) in different versions of scripts (Naskh, Othoman, etc.) and also various handwritten words containing overlap letters were analyzed. The result obtained is impressive since more

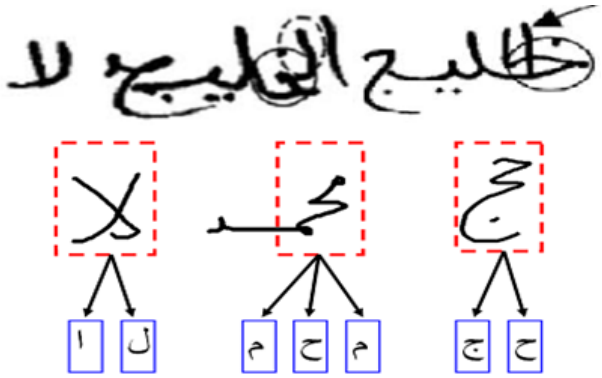

Fig. 2 Examples of overlap letters

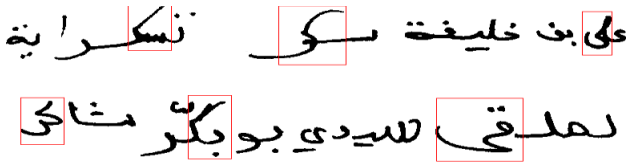

Fig. 3 Examples of overlap letters caused by 'kaf' and 'ya' letters

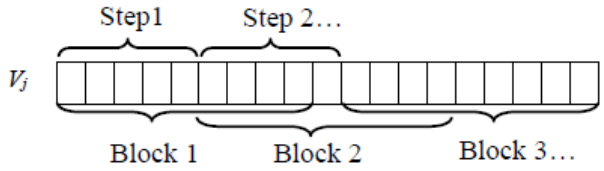

Fig. 4 An example of vertical projection vector with B_s=9 and S_p=5

Fig. 5 Example of extraction of segmentation points

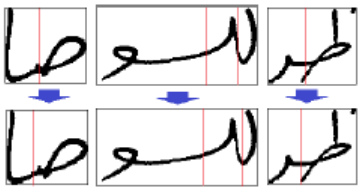

Fig. 6 Improvement of segmentation points

than 100 vertical ligatures were identified (Table 2). Diacritical marks were ignored in this analysis.

It can be noticed that the letters 'kaf' and 'ya' are at the origin of several overlap letters. Figure 3 shows some examples extracted from the IFN/ENIT corpus. Based on these statistics, the most relevant overlap letters were selected. The IFN/ENIT database provides a statistical analysis of the number of overlap letters and their distribution (Table 3). This analysis was enriched with statistics on overlap letters caused by the letters 'kaf' and 'ya' (Table 4). These relevant overlap letters were integrated into the alphabet presented in the following section.

4 Segmentation and recognition

4.1 Segmentation

This phase aims to extend the segmentation algorithm proposed by Mamouni et al. [9] with several successful ideas used in the segmentation of both Arabic and Latin scripts. The algorithm consists of several steps:

- Detection of connected components;
- Detection and deletion of diacritical marks;
- Segmentation of each connected component;
- Reintegration of diacritical marks.

The algorithm proposed by Mamouni et al. [9] is based on vertical and horizontal projection and the sliding windows approach. The vertical projection is used to detect local minima which represent the potential position of segmentation points, while horizontal projection is used to determine the baseline position. The vertical projection is defined by the vector V_j as follows:

$$V_j = \sum_j P(i,j) \quad (1)$$

where $P(i,j)$ is a pixel of the binary image of the script and is either 0 or 1, i and j refer to indexes of the row and column; V_j represents the number of black pixels in each column of the image.

The algorithm can extract segmentation points using overlap blocks by setting three parameters: Block size (B_s), Step size (S_p) and Threshold (T). The value of the parameter S_p must be less than or equal to B_s. If the two parameters are equal there is no overlapping between the block and the next one, as shown in Figure 4. The algorithm creates a segmentation point (Figure 5) if there is a significant decrease between the sums of the current values of the block compared to the previous block (above the threshold). Then a local search around each segmentation point is performed to find the best position of these points. The final segmentation point is the nearest point that has the smallest value in the vertical projection vector around the initial point, as shown in Figure 6. Finally, some generated segmentation points are deleted if they are far from the baseline (Figure 7a) or if the number of transitions (i.e total number of times the pixel state changes) is greater than 2 (Figure 7b). Even if the recognition rate obtained by this algorithm on IFN/ENIT database is 89.5%, this

Table 2 Exhaustive list of Arabic overlap letters

ﻻ	ل.ا	ﻟ	ل.ح	ﺤ	ـ.ح	ﻠ	ل.م	ﻼ	ل.ا	ﺤ	م.ح
ﻟ	ل.م.ح	ﺨ	ـ.ح	ﻔ	ـ.ح	ﻷ	ر.ا	ﻤ	ح.م	ﻤ	ـ.م
ﻢ	ـ.م	ﺴ	س.ح	ﻫ	ه.ـ	ﻣ	ـ.م	ﻢ	ـ.م	ﻣ	م.م
ﻜ	ك.م	ﻛ	ك.م	ﻛ	ك.ر	ﻜ	ك.ع	ﻛ	ك.ه	ﻛ	ك.ى
ﻛﻮ	ك.و	ﻛ	ـ.ك	ﻛ	ح.ك	ﺴ	س.ك	ﻛ	ص.ك	ﻋ	ع.ك
ﻋ	ع.ك	ﻓ	ف.ك	ﻛ	م.ك	ﻬ	ه.ك	ﻛ	ك.ـ	ﻛ	ك.ح
ﻛ	د.ك	ﺴ	س.ك	ﻫ	ص.ك	ﻫ	ط.ك	ﻛ	ف.ك	ﻫ	ه.ك
ﻛ	ك.ى	ﻄ	ط.ك	ﻟ	ل.ك	ﻄ	ط.ى	ﻋ	ع.ى	ﻠ	ـ ل ى
ﻰ	ع.ى	ﺣ	ح.ى	ﻫ	ه.ى	ﻰ	ـ.ى	ﻟ	ل.ى	ﻣ	م.ى
ﻰ	ـ.ى	ﺳ	س.ى	ﻜ	ل.ك.م	ﻠ	ل.ح.ك	ﻠ	ل.ـ.ى	ﻠ	ـ.ـ.ى
ﻤ	ـ.ح.م	ﺦ	ل.ح	ﺢ	م.ح	ﺢ	ـ.ح	ﻤ	م.ح.م	ﺤ	ح.ح
ﺢ	ح.ح	ﻟ	ا.ـ	ﻟ	ل.ر	ﺤ	ص.ح	ﺢ	ص.ح	ﻟ	ا.ل.ح
ﻟ	ا.ل.م	ﺮ	ح.ر	ﻋ	ع.ر	ﺮ	ع.ر	ﺲ	س.ح	ﻮ	ـ.و
ﺴ	س.م	ﺴ	س.م	ﻤ	ل.ح.م	ﻤ	ل.ح.م	ﻔ	ف.ـ.ح	ﻒ	ف.ـ.ح
ﻠ	ل.ح.ح	ﻠ	ل.ح.ح	ﻜ	ـ.م.ح	ﻢ	ـ.م.ح	ﻟ	ل.ل.م	ﻟ	ل.ل.م
ﻄ	ط.ح	ﻂ	ط.ح	ﻫ	ع.م	ﻣ	ع.م	ﻛ	ك.ك	ﻰ	ص.ى
ﻰ	ـ.م.ى	ﻠ	ـ.م.ى	ﻢ	ـ.ح.م	ﻋ	ع.ح	ﺠ	ـ.ح.ح	ﻣ	م.م

Table 3 IFN/ENIT statistical analysis of overlap letters

Overlap letter	Quantity
ﻻ	1266
ﻟ	1197
ﻷ	1066
ﻠ	450
ﻜ	100
ﻠ	64

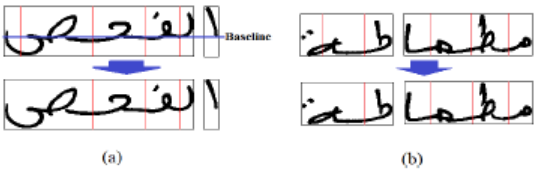

Fig. 7 Removing segmentation points

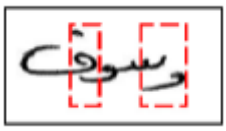

Fig. 8 Detection of connected components can avoid overlap letters

work introduces some extensions to improve its performance including the detection of connected components and the removal of diacritical marks. The connected component detection was used to separate the pieces of Arabic words (PAW) which can avoid overlap letters (Figure 8) and then skew the segmentation procedure. To this end, a classical algorithm introduced by Srihari et al. [14] was applied. Detecting diacritical marks has two advantages. On one hand, too many diacritical marks can disturb the histograms used to detect the baseline and segmentation points. On the other hand, it reduces the alphabet size and improves the time required for learning. Thus, an extension of the algorithm described by Menasri [11] was used.

First the connected components (CC) are filtered according to their surface to detect most of the diacrit-

Table 4 Statistical analysis of 'Kaf' and 'Ya' overlap letters on IFN/ENIT

Overlap letter	Quantity	Overlap letter	Quantity
بک	192	لک	104
لی	92	کر	73
فک	37	حی	32
یک	30	لی	28
مک	27	سی	25
کو	19	سک	16
فی	14	کم	12
مک	9	می	4
کس	3	که / کی	2/2

ical marks. Two tests are carried out, the first is based on the surface; connected components that are too large cannot be diacritical marks. Then, a special treatment is carried out for certain connected components which, despite a small surface, do not correspond to diacritical marks such as the case of the letter 'Alif', or the letter 'Daal'. For the first case, the CC are identified by their vertical extent; for the second, they are identified by their relative position with respect to other connected components. If a connected component C1 of the reduced area is located above another connected component C2, and that C2 overlaps C1 vertically at more than 75%, then C1 is a diacritical mark. Certain diacritical marks, which exceed the threshold of the surface like the chadda (ω), can also be ruled out because they are located either above or below the baseline.

In some cases, diacritics may not form connected components separate from the text body. An example is given in Figure 9. This kind of diacritical mark was not treated in the study carried out by Menasri [11]. This work aims to detect them by performing a skeletonization operation which in many cases can detect this kind of diacritical mark. The experiments showed a high success rate. Figure 9 illustrates this procedure:

a) Initial image;
b) Image obtained after application of the diacritical marks removal algorithm;
c) Skeletonization and highlighting of the diacritical mark;
d) Removal of the diacritical mark.

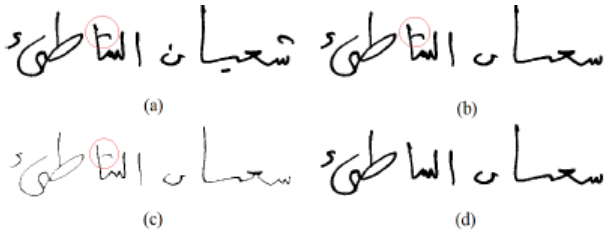

Fig. 9 Presence of a diacritical mark stuck to the text body (chadda)

After generating segmentation points, the diacritical marks are reintegrated in the image used in the recognition phase. Experiments showed that the detection of connected components and removal of diacritics improved the recognition rate obtained by Mamouni et al. [9] even if their position in some cases does not match the corresponding letter. Finally, the particular case of the letter 'Seen' is verified with a test to reconstruct this letter if the shapes (ι) are close to each other and to avoid the recognition of the letter 'Seen' as a succession of shapes (ι).

4.2 Recognition

A learning algorithm must be used to recognize the characters obtained by the segmentation to complete a recognition system. In this study, the well-known Support Vector Machines (SVM) approach was used. It was trained on a corpus of handwritten Arabic letters based on a reduced alphabet including the main overlap letters. The idea of the reduced alphabet was proposed by Menasri [11] to improve learning time by ignoring diacritical marks.

However, the removal of diacritics can lead to a drawback, which was not mentioned in the work of Menasri. Indeed, some characters can easily be confused with others characters, which degrades the recognition rate. As an example, the results of the experiments carried out show that the letters 'Faa' and 'Qaaf' without diacritics are in many cases confused with the letter 'Waw'. For other letters with diacritical marks, the confusion with other characters appears very rarely. To this end, it is recommended to keep only the letters (Faa) and (Qaaf) in the vocabulary with diacritical marks. This allows to extend it by including the most used overlap letters (Table 5) without degrading the learning time too much.

The SVM approach is used for training and prediction which is a kernel method of statistical learning introduced by Vapnik [16] and has been successfully used for the solution of a large class of supervised machine learning tasks such as categorization, prediction, nov-

Table 5 Complete list of the 60 symbols of the proposed alphabet

ا	ب	ه	ه	ح	ح	د	ر	س	س
ص	ص	ط	ع	ع	ع	ع	ف	ق	ل
ك	م	م	ن	ه	ه	و	ى	ى	لا
لـ	حـ	حـ	ا	لا	كـ	كـ	كـ	كـ	كه
ى	كو	ك	ك	ك	ك	ك	ك	ك	ك
ها	ى	ق	ق	ق	ق	ى	ى	لـ	و

Table 6 Selected SVM parameters

Multi-Class training	One-against-One
Kernel function	Gaussian
Regularization parameter C	117405
Kernel function parameter σ	1.83 E-05

elty detection and ranking. The key idea of SVM is that given a training set $Z = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, the classifier is found by solving the following linearly constrained convex quadratic programming problem:

$$Maximize \sum_i \alpha_i - 1/2 \sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$
$$Subject\ to\ 0 \leq \alpha_i \leq C\ and \sum_i \alpha_i y_i = 0 \quad (2)$$

where C is a regularization parameter which influences classifier generalization when classes are strongly intertwined.

The QP objective function involves the Kernel function $k(x_i, x_j)$. Many successful Kernel functions have been used like Linear ($k(x_i, x_j) = x_i.x_j$), Polynomial ($k(x_i, x_j) = (ax_i.x_j + b)^d\ a > 0$), Sigmoid ($k(x_i, x_j) = tanh(ax_i.x_j + b)$) and Gaussian or RBF ($k(x_i, x_j) = exp(-||x_i - x_j||^2/\sigma^2)$). A study carried out by Mamouni et al. [10] show the effectiveness of support vector machines (SVM) in learning handwritten Arabic characters. In this study, the optimal configuration of SVM led to the best prediction rate. The concerned SVM parameters are the multi-class learning approach (one-against-all or one-against-one), the Kernel function, the regularization parameter C and the parameter(s) corresponding to the Kernel function. Table 6 summarizes the values of the SVM parameters.

5 Computational experiments

In this work, an open-source SVM engine developed by Thorsten Joachims in 2008 was used[1]. It contains documentation, examples, and bibliographic references. For the learning phase, three databases of handwritten Arabic letters were used: the characters obtained by the proposed segmentation algorithm on IFN/ENIT database, a corpus of Arabic letters used by Mamouni et al. [9] and a database containing exclusively overlap letters. These three databases contain:

- Corpus IFN/ENIT: 212211 letters (141471 for training and 70740 for test);
- Corpus used by Mamouni et al. [9]: 4840 letters (3840 for training and 1000 for test);
- Overlap letters database: 2000 letters (1300 for training and 700 for test).

Therefore, the proposed training dataset includes about 2/3 of the examples available in these three databases, i.e. (141471 + 3840 + 1300) = 146 611 letters, while the test dataset includes about 1/3 of these examples, i.e. (70470 + 1000 + 700) = 72 170 letters. The results obtained are comparable with the state of the art and recent work on the Arabic handwriting recognition (See Figure 10). For overlap letters, which were been recognized directly, the results are not very good, but they remain acceptable thanks to the reduced alphabet, which contains only the most relevant overlap letters. However, some of them were not recognized because they were not retained in the alphabet. A compromise must therefore be found between the number of overlap letters to be included in the alphabet (which affects the time required for learning) and the prediction quality. Another problem that influences the results is that some overlap letters are easily confused with Arabic letters or other overlap letters, as shown in Figure 11.

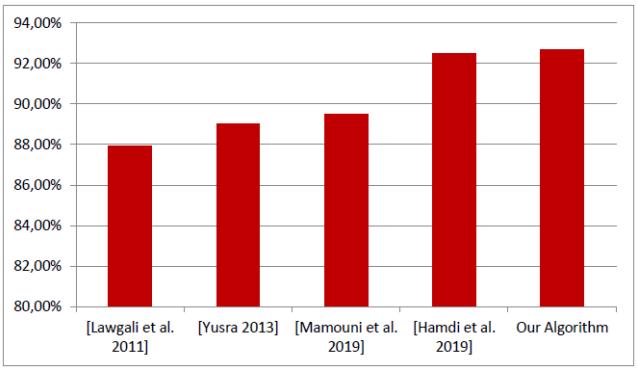

Fig. 10 Comparison with previous works

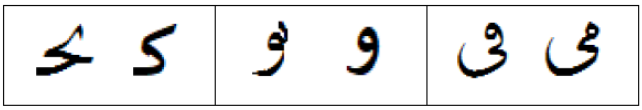

Fig. 11 Some confused Arabic letters

[1] Available at: http://www.cs.cornell.edu/people/tj/svm_light/svm_struct.html

6 Conclusion

This study focused on the recognition of Arabic handwriting. The IFN/ENIT database (names of Tunisian cities) was used for the development and evaluation of a recognition system of Arabic writing. Unlike other systems, which only consider Arabic alphabetic letters, the proposed alphabet was enriched with the most relevant overlap letters.

A comprehensive study of these special letters and their frequency of use in the Arabic script was also carried out, which facilitated the enrichment of the proposed alphabet. In addition, an extension of the segmentation mechanism used in the literature was proposed. It used diacritical signs to reduce the size of the alphabet used in learning.

The proposed system provided results comparable to those of the state of the art, in addition to effective treatment of overlap letters. However, a compromise had to be found between the size of the alphabet and thus the number of overlap letters to be taken into account and the quality of the results. This issue will be addressed in future work.

References

1. Alaasam R., Kurar B., El-Sana J., "Layout Analysis on Challenging Historical Arabic Manuscripts using Siamese Network", International Conference on Document Analysis and Recognition ICDAR, 738-742 (2019).
2. Dupre X., "Contributions à la reconnaissance de l'écriture cursive à l'aide de modèles de Markov cachés", PhD thesis, Univ. Rene Descartes - Paris V (2003).
3. Hamdi Y., Boubaker H., Dhieb T., Elbaati A., Alimi A.M, "Hybrid DBLSTM-SVM based Beta-elliptic-CNN Models for Online Arabic Characters Recognition", International Conference on Document Analysis and Recognition ICDAR, 545-550 (2019).
4. Khorsheed M. S., "Off-line Arabic character recognition – a review", Pattern Analysis & Applications, vol. 5, no. 1, pp. 31-45 (2002).
5. Lawgali A., "A survey on Arabic character recognition", International Journal of Signal Processing, Image Processing and Pattern Recognition, vol. 8, no. 2, pp. 401-426 (2015).
6. Lawgali A., Bouridane A., Angelova M., Ghassemlooy Z., "Automatic segmentation for Arabic characters in handwriting documents", Proceedings of the IEEE International Conference on Image Processing, pp. 3529-3532 (2011).
7. Lorigo L. M., Govindaraju V., "Offline Arabic handwriting recognition: a survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 5, pp. 712-724 (2006).
8. Madhvanath S., Krpasundar V., Govindaraju V., "Syntactic methodology of pruning large lexicons in cursive script recognition", Pattern Recognition, 34(1):37–46 (2001).
9. Mamouni E., Zennaki M., Sadouni K., "Efficient Analysis of Vertical Projection Histogram to Segment Arabic Handwritten Characters", Computers, Materials & Continua CMC, vol.60, no.1, pp.55-66 (2019).
10. Mamouni E., Zennaki M., Sadouni K., "SVM Model Selection Using PSO for Learning Handwritten Arabic Characters", Computers, Materials & Continua CMC, vol.61, no.3, pp.995-1008 (2019).
11. Menasri F., "Contributions à la reconnaissance de l'écriture arabe manuscrite", Phd Thesis, Paris Descartes University (2008).
12. Naz S., Umar A. I., Ahmed S. B., Shirazi S. H., Razzak M. I., Siddiqi I., "Segmentation techniques for recognition of Arabic-like scripts: a comprehensive survey", Education and Information Technologies, vol. 21, no 5, pp. 1225-1241 (2016).
13. Pechwitz M., Maddouri S., Maergner V., Ellouze N., Amiri H., "Ifn/enit-database of handwritten Arabic words", In CIFED (2002).
14. Srihari P. B. S., Srinivasan H., Bhole C., "Spotting words in handwritten Arabic documents", In Procs. of SPIE, San Jose, CA, USA (2006).
15. Tay Y., Lallican P., Khalid M., Viard-Gaudin C., Knerr S., "An offline cursive handwritten word recognition system", Proceedings of IEEE Region 10 International Conference on Electrical and Electronic Technology, TENCON 2001 (Cat. No.01CH37239), Singapore, pp. 519-524 vol.2, doi: 10.1109/ TENCON. 2001.949649 (2001).
16. Vapnik V., "Statistical Learning Theory", John Wiley & Sons Inc (1998).
17. Yasser M. A., "A survey on Arabic character segmentation", International Journal on Document Analysis and Recognition, vol. 16, no. 2, pp. 105-126 (2013).
18. Yanikoglu B., Sandon P., "Segmentation of off-line cursive handwriting using linear programming", Pattern Recognition, Volume 31, Issue 12, pp. 1825-1833 (1998).
19. Yusra O., "Segmentation algorithm for Arabic handwritten text based on contour analysis", Proceedings of the IEEE International Conference on Computing, Electrical and Electronic Engineering, pp. 447-452 (2013).