Impact of Features and Classifiers Combinations on the Performances of Arabic Recognition Systems

Afef Kacem Echi
Université of Tunis
LaTICE, Tunis, Tunisia
Email: afef.ensit@rnu.tn

Abdel Belaïd
Université of Lorraine
LORIA, Nancy, France
Email: abdel.belaid@loria.fr

***Abstract*—Arabic recognition is a very challenging task that begins to draw the attention of the OCR community. This work presents our latest contributions to this task, exploring the impact of several features and classifiers combinations on the performances of some developed systems. Different types of writings were considered (machine-printed, multi-fonts, handwritten, unconstrained, multi-writers, bi-dimensional, large vocabulary, ancient manuscripts). For each type of writing, we have considered both the most appropriate features and classifiers: contextual primitives to compensate the Arabic morphology variation, statistical features to recognize mathematical symbols and spectral features, mainly run lengths histogram-based features and histogram of oriented gradient-based descriptors to discriminate between machine-printed/handwritten and Arabic/Latin words. We have also used the shape context descriptor, for touching characters segmentation, which has been useful to train the models in the template-based recognition system. We have taken advantage of the Hough generalized transform to spot separator words in ancien arabic manuscripts. Otherwise Bayesian networks are used to apprehend the writing uncertainty and transparent neural networks to exploit the morphological aspect of Arabic language and integrate linguistic knowledge in the recognition process. The proposed systems are designed based on the characteristics, the similarities and the differences of Arabic writings.**

I. Introduction

The automatic script recognition for Latin and others languages has been extensively investigated. In contrast, written Arabic recognition only has been researched in recent years and research has been growing very slowly. This slow growth is mainly due to the lack of standardized database and to the inherent difficulties in written Arabic recognition. Nowadays, commercial OCRs integrate this language into their possibilities and we find ICR for Arabic as efficient as for Latin. The challenge remains, however, for large vocabulary, mathematical formulas, unconstraint documents and ancient Arabs, where the calligraphic aspect is more intense and complex to deal with, offering challenges for decades.

Typically, the recognition systems can be divided into three main steps: 1) preprocessing step which aims to improve the text image by enhancing contrast and removing noise, 2) feature extraction and selection step which describe the input image so that features of interest are highlighted, and 3) classification and recognition step which assign a label to a word based on the information provided by its description. This paper focuses on feature extraction and classification steps and their impact on the performances of Arabic recognition systems. Note that the performance of a classifier can rely as much on the quality of the features as on the classifier itself. A good set of features should represent characteristics that are particular for one class and be as invariant as possible to changes within this class. A feature set made to feed a classifier can be a mixture of different types of features. This paper attempts to understand the difference in performance of various features and classifiers when applied to different recognition problems. Different types of writings were considered (machine-printed, multi-fonts, handwritten, unconstrained, multi-writers, bi-dimensional, large vocabulary, ancient manuscripts) and many features and classifiers are tested and used in different contexts. This should give researchers the chance to gain more insight into the natures of the problems and, thus, be better able to do classifier-problem matching.

This paper is organized as follows. In section II, we discuss different types of features used in the proposed recognition systems. Different models, trained and tested, are provided with illustrations in the remaining sections: 1) Transparent Neural Network-based system to recognize a large vocabulary of Arabic words, 2) Probabilistic Graphical Model-based system to handle with non-linear distortions in the Arabic handwriting, 3) Template-based system to extract and segment touching components in Arabic manuscripts and 4) a rule-based system to recognize Arabic mathematical formulas. In each section, we report and analyze the performances of the proposed systems. We finally give some conclusions.

II. Feature Extraction

Different feature extraction methods can be divided into two categories structural and statistical.

A. Structural Features

There is a close interrelation among various parts of the word. Such interrelation representing the structural properties must be taken into account to achieve the best recognition performance. Additionally to the fact that structural features serve to better describe the topological and geometrical characteristics of words, we believe that words can be represented by this type of features with high tolerance to distortions and style variations. However, their extraction is not easy as it

requires a lot of precaution considering the cursive aspect of the Arabic script.

For handwritten Arabic word recognition [2] and large vocabulary recognition [3], we extracted some structural features, mainly ascenders, descenders, loops and dots and their positions relative to the baseline. As handwritten words are not usually written on a single baseline, we extracted a sequence of sub-baselines and formed the entire word baseline by juxtaposition of its PAW baselines. Some heuristics are employed for such local extractions. For example, if the PAW consists of ascending letter like the Alif, its baseline corresponds to the bottom of its bounding box. Thus, the word baseline is estimated to mainly cope with the problems of inclination and overlap. More details are given in [2]. In Fig. 1, we report some comparisons between some existing methods and the one we proposed. To provide for the greatest number of

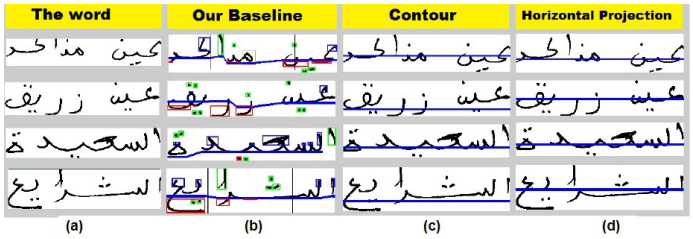


Fig. 1. (a) Words, (b) Extracted baseline by our method, (c) Baseline extracted by contour method, (d) Baseline extracted by horizontal projection.

opportunities to find such features, the effective word central band is estimated based on the extracted baseline (see Fig. 2).

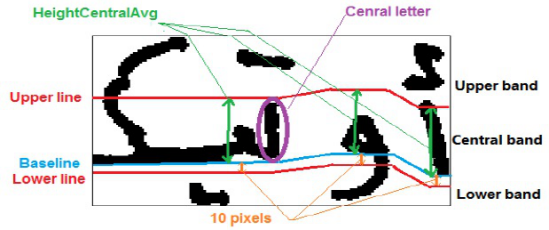


Fig. 2. Upper, central and lower band extraction based on baseline.

B. Statistical Features

Statistical features are derived from the statistical distributions of pixels. In [6], we used Hu and Zernike moments, run-length histograms, bi-level co-occurrence and white pixel portion to recognize mathematical symbols. In general, statistical features are easy to extract. Such features may not have physical meaning. But, they are usually effective and computable. Nevertheless, they may be misleading due of noise.

For machine-printed/handwritten and Arabic/Latin word discrimination, we computed black run lengths (BRL)[4], histogram of oriented gradients (HOG), pyramid HOG [5] (PHOG) and cooccurrence matrices of oriented gradients (CoMOG) [5]. The idea is to exploit the writing orientation as a discriminative descriptor between Arabic and Latin scripts. In fact, letters in Arabic words, as written from right to left, are generally tilted to the left, following the writing direction (see Fig. 3(a)). In contrast, letters in Latin script, as written from left to right, tend to be inclined to the right (see Fig. 3(b)). Thus, Arabic letter strokes are generally diagonally down whereas those written in Latin are diagonally up. Furthermore, machine-printed Arabic words are characterized by the use of horizontal ligatures, more or less long depended on the used font (see Fig. 3(c)). Oppositely, machine-printed Latin words are composed by successive letters without any ligature between them (see Fig. 3(d)). Consequently, horizontal strokes would be more frequent in Arabic words than in Latin words. Both scripts use vertical strokes for ascenders.

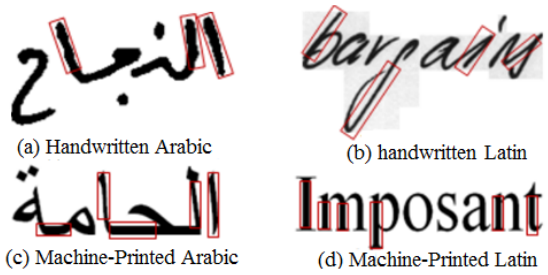


Fig. 3. Machine-printed/Handwritten and Arabic/Latin word identification based on the writing orientation.

In [4], a BRL vector P is defined as follows. Each element $P(i)$ represents the number of runs with black pixels and length of run equal to i in a given direction. The BRL vector's size is M which corresponds to the maximum run length in words. An orientation is defined using a displacement vector $d(x, y)$, where x and y are the displacements for the x-axis and y-axis respectively. The typical orientations are horizontal, right diagonal, vertical and left diagonal, then calculating the run-length encoding for each direction will produce four BRL vectors. The four obtained BRL vectors are then concatenated into a single vector characterizing the word and implicitly its script.Various texture features are then derived from BRL vectors which measure the distribution of short and long runs, the similarities of gray level values and of the length of runs through out the word's image and the homogeneity and the distribution of runs of the word's image in a specific direction. These features aim to capture the coarseness of a texture in specified directions [9]. We tested with different classifiers and we find that *1*-NN achieves the best accuracy: 99%.

In [5], we made use of HOG-based features. Being based on shape descriptors, HOG has interesting properties for script characterization. It is a histogram which counts the gradient orientation at pixels in a given image. The number of features depends on the number of cells and orientation.

While HOG counts occurrences of gradient orientation in localized portions of an image, Pyramid HOG captures perceptually salient features taking into account the spatial property the local shape while representing an image by HOG. The spatial information is represented by tiling the

image into regions at multiple resolutions based on spatial pyramid matching (see Fig. 4). Table I displays HOG results

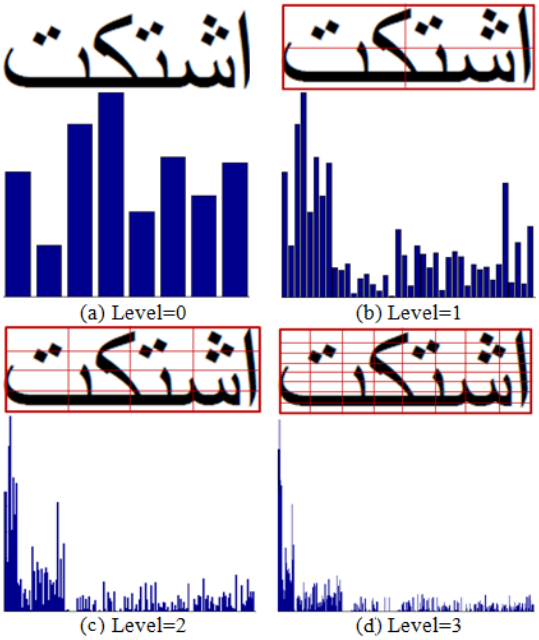


Fig. 4. A schematic illustration of PHOG at each resolution level.

according to the cell number using different classifiers. Table II shows the effects of orientation bins number on the accuracy using PHOG. We considered 24000 word images from public databases: IAM database for Latin handwritten, IFN-ENIT and AHTID/MW for Arabic handwritten and APTI for Arabic and Latin printed words.

TABLE I
SCRIPT IDENTIFICATION ACCURACY USING HOG.

Cell Number	8
Features Number (selected)	64 (38)
SVM Accuracy	80.56%
k-NN Accuracy	82.87%
AODEsr Accuracy	**93.47%**

TABLE II
SCRIPT IDENTIFICATION ACCURACY USING PHOG.

Orientation Bin Number	8
Features Number (selected)	680 (268)
Accuracy(%)	**97.49**

In [5], we took advantage of Co-MOG to express the distribution of gradient information over an image. These feature captures more spatial information than PHOG by counting the frequency of co-occurrences of oriented gradients between pairs of pixels. A total of 324 features are extracted from each word image. Genetic algorithm is applied for reducing the dimensionality of the features vectors to 248. We used SVM, *k*-NN and AODEsr classfiers and we respectively achieved 96.43%, 99.85% and 99.71% as identification rates. Overall, features extracted from BRL seem quite valuable for word level script identification. Features from HOG descriptors are characterized with high robustness against intra-class variability and fast enough for real time classification. In addition, they are scale invariant which means that words can be classified without any need for size normalization. They do not require word segmentation. Although Co-MOG features only include angular orientation of gradient and co-occurrence of angular orientation of gradient, they achieve the best performance in all script identification experiments.

The main drawback with the statistical feature extraction is that it is difficult to take contextual dependence into consideration. In [8], we investigated the use of the shape context descriptor for touching characters (TC) segmentation. The idea is to find the most similar model, among those stored in a code book with their prior known segmented parts, for the TC to be segmented using a similarity metric computed from the shape context descriptor. Finding correspondences between a model and a TC consists on searching for each point p_i of TC's contour, the best matching point q_j on the model's contour by comparing their edge point's shape context histogram, as illustrated in Fig. 5.

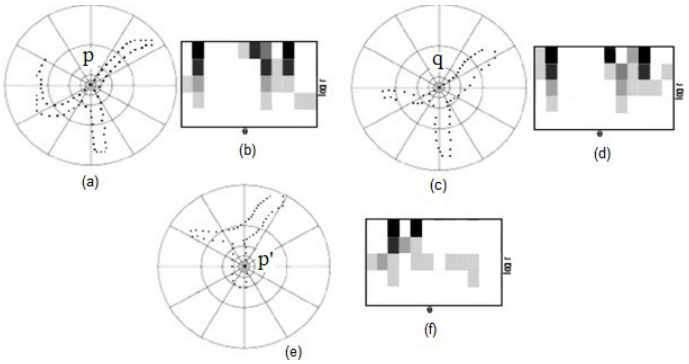


Fig. 5. Matching with SC: (a) SC of p in TC_1,(b) log-polar histogram for p (c) SC of q in TC_2, (d) log-polar histogram for q which is similar to that in (b), but different of p' in (f). The best matching is between p in TC_1 and q in TC_2. Black bins correspond to a higher number of pixels in that bin, gray bins contains fewer pixels than the black cells. Log-polar histogram similarity is according to the χ^2 distance.

In [7], we used the Generalized Hough Transform to spot separator words in ancien Arabic manuscripts. This technique provides features invariants to global deformation and seems to be tolerant of gaps in feature descriptions and relatively unaffected by image noise.

III. RECOGNITION SYSTEMS

The proposed recognition systems are based on template-matching, probabilistic graphical models and Transparent neural networks.

A. Template-based Recognition System

For templates without strong features, or when the bulk of the template image constitutes the matching image, a template-based approach may be effective. In [8], we proposed a template-based recognition system to address the problem of

TC extraction and separation in Arabic manuscripts. The proposed system first extracts the TCs in the document whether between successive text-lines or words of the same text-line. For vertically/up-down TCs, we improved an existing method, that of Ouwayed and Belaïd [11]. For horizontally/left-right TCs, we proposed a novel extraction method based on the morphology analysis of the terminal letters of Arabic words. To evaluate the vertically inter-connected TC extraction, we used 500 TCs and we achieved an extraction rate of 94%. For horizontally inter-connected TCs, we used 600 TCs and achieved an extraction rate of 94.2%. We then recognized the TCs relying on templates, using shape context descriptor and an interpolation function the Thin Plate Spline transformation (see Fig. 6). Note that the shape context descriptor has the advantage to summarize a global shape in a rich and local descriptor. It greatly simplifies recovery of correspondences between points of two given shapes and it is tolerant to all common shape deformations.

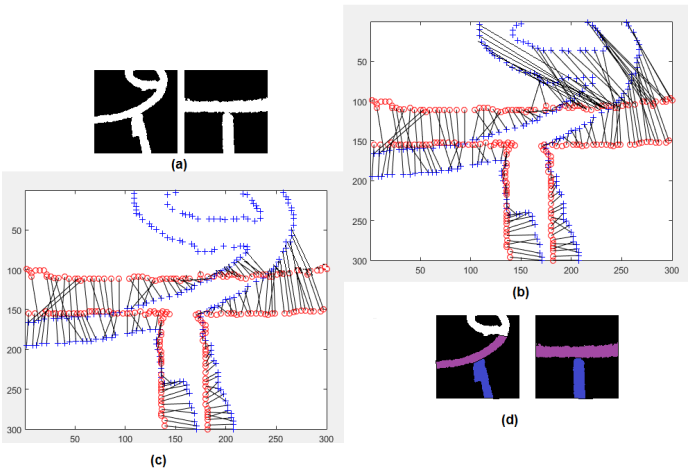


Fig. 6. (a) Matching between a TC and its most similar model, (b) The point-to-point matching, (c) Matching based on region, (d) Illustration of region-to-region matching.

We finally segmented the TCs based on the distance from the central points of the recognized template's parts. Tests are performed using a large dataset of TCs and three metrics: Manhattan, Euclidean and Canberra distances and we respectively achieved 94%, 92.6% and 92% as segmentation rates. For the TC segmentation evaluation, we used 3000 TCs (670 horizontally inter-connected TCs and 2330 vertically inter-connected TCs). We achieved a segmentation rate of 94%. Obtained results strongly support the efficiency of the proposed TC extraction and segmentation methods and outperform results of some related works, taking into account the different types, variability and complexity of the TCs.

B. Transparent Neural Network-based Recognition System

Most of the actual research in writing recognition focuses on specific applications where the vocabulary is relatively small. Many applications can be opened up when handling with large vocabulary. In [3], we proposed a collaborative combination of Neuron-Linguistic classifiers for large Arabic word vocabulary recognition. The proposed approach is based on three networks, which exploit the morphological aspect of the Arabic word and collaborate for a better word recognition. We focused on decomposable words which are derived from healthy tri-consonantal roots and easy to proof the decomposition. To perform word recognition, the system extracts a set of global structural features. Then, it learns and recognizes roots, schemes and conjugation elements that compose the word. To help the recognition, some local perceptual information is used in case of ambiguities. This interaction between global recognition and local checking makes easier the recognition of complex scripts as Arabic (see Fig. 7). Several experiments have been performed using a vocabulary of 5757 words, organized in a corpus of more than 17271 samples. We achieved a recognition rate of 95.33%.

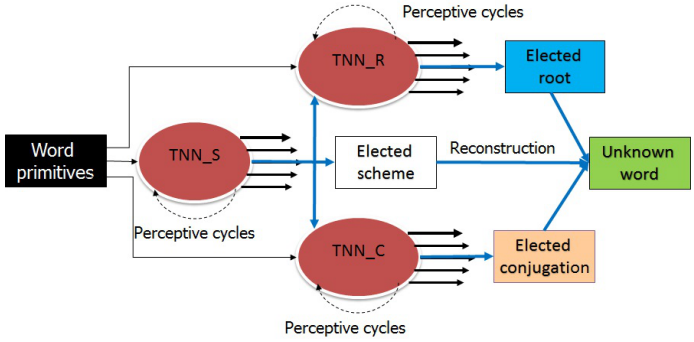


Fig. 7. TNN Propagation.

C. Probabilistic Graphical Model-based Recognition System

We explored a better understanding of the relative performance of probabilistic graphical models, known by their power in several recognition problems that require reasoning under uncertainty, for recognizing handwritten Arabic words. The objective was to exploit the Arabic writing process and the ability of these models to handle with nonlinear distortions of the Arabic script and uncertainly of human writing. In [1] we firstly proposed a Naïve Bayes (NB) which represents the word with a set of random variables and where the probability distribution is factorized on the variables by the product of elementary probabilities of the variable knowing the causes. For that, we extracted 102 structural features and composed 83 classes from the IFN-ENIT database.

With NB, we can look up all the probabilities with a single scan of the database and store them in a small table. Thus, NB is fast to train and to classify. It is not sensitive to irrelevant features. But, one of NB problems is its assumption of features independence. As solutions to alleviate the conditional independence assumption are to extend the structure of NB by considering the relationships between features as proposed in TAN. With NB and TAN, we respectively achieved 78.57% and 81.64% using 83 classes and 7888 samples from the IFN/ENIT database. However, if static BN handles well the

multi-dimensionality, it does not handle the temporal aspect. So, we used the DBN which can be seen as a repetition of static BNs called time slices. Transitions between time slices and observations are inside each slice. For that, we proposed to model words by coupling two HMM architectures into a single DBN, each one is for one direction: horizontal and vertical. We added direct links between hidden state nodes to represent dependencies between state variables of both vertical and horizontal HMMs and used two streams for word observations. In a first proposed DBN1 (see Fig. 8(a)), a state of a HMM is connected to the adjacent state in the same time slice t of the other HMM. To enhance the influence of the vertical stream, the edges are directed from the vertical stream to the horizontal one. Experimentally, it has been proved that V-HMM is more reliable than H-HMM, when tested separately.

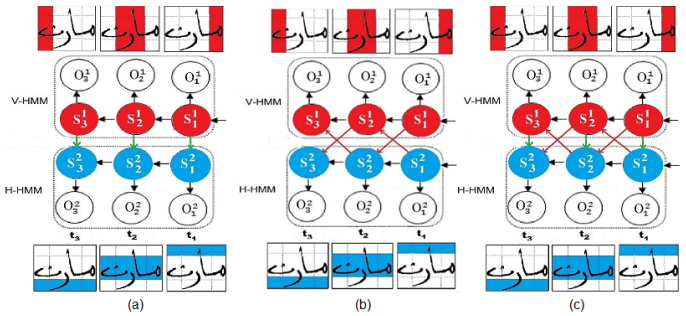


Fig. 8. DBN's architectures, from [2]: (a) DBN1, (b) DBN2 and (c) DBN3.

In a second proposed DBN2 (see Fig. 8(b)), we considered diagonal links, instead of vertical ones. Thus, a state of a HMM is connected to the adjacent state in the next time slice of the other HMM. We finally proposed a third DBN3 where both vertical and diagonal dependencies are involved (see Fig. 8(c)), in the attempt to consider all possible correlations between parts of word. We respectively achieved 90.02%, 84.10% and 86.07% recognition rates for DBN1, DBN2 and DBN3 on the same database. An explanation for this may be as follows. The dependencies between current states of the V-HMM and H-HMM in DBN1, seem to be relevant as they reflect the Arabic writing process: top-to-bottom and right-to-left and lead to the best rate: 90.02%. In DBN2, the cross dependencies, instead of vertical ones, lowered the rate to 84.10% due to their redundancy, since they did not express any additional feature in Arabic writing. When considering both vertical and cross-dependencies in DBN3, the rate was improved a little bit: 86.07%, probably thanks to relevance of vertical dependencies, but it remains lower than DBN1 recognition rate.

IV. Rule-based Recognition System

In [6], we addressed the problem of Arabic mathematical formula recognition. For the formula structure analysis, we proceeded by top-down and bottom-up parsing scheme based on operator dominance. A set of replacement rules is defined by a coordinate grammar based on symbol recognition and symbol arrangement analysis results. In the proposed system, the recognition and parsing modules interact more closely. Thus, we can use the context information collected during structural analysis to help us guess about the symbols, overcoming our incorrect assumption of perfect symbol recognition. The syntax-directed recognition system, described here, has been successfully demonstrated in many types of formulas and achieved satisfactory results. 91% of formulas are correctly recognized.

V. Conclusion

We presented research results on off-line Arabic recognition using structural and statistical approaches for the specific tasks of large vocabulary recognition, handwriting recognition, script identification, touching characters segmentation and mathematical formulas recognition. The differences between statistical and structural approaches can be summarized as follows. In the statistical approach, the word is described in a quantitative manner, whereas the structural approach generates a description composed of primitives. The statistical approach discriminates via numeric differences among features from different classes, while grammars are used by the structural approach to define a language which includes the acceptable configurations of primitives for each class. Structural features can be used with either a statistical or structural classifier. Statistical features cannot be used with a structural classifier because they miss relational information, however statistical information can be associated with structural primitives and used to resolve ambiguities, as when parsing with attributed or stochastic grammars. Having divergent theoretical foundations, the two approaches focus on different data characteristics and employ distinctive techniques for the word representation and classification.

References

[1] A. Khemiri, A. Kacem, A. Belaïd and M. Elloumi, *A System for off-line Arabic Handwritten Word Recognition based on Bayesian Approach*, In Proceeding of ICFHR, Schenzen, China, 2016.

[2] A. Khémiri, A. Kacem, A. Belaïd and M. Elloumi, *Arabic Handwritten Words Off-line Recognition based on HMMs and DBNs*, In Proceeding of ICDAR, 2015.

[3] A. Kacem Echi, I. Ben Cheikh and A. Belaïd, *Collaborative combination of neuron-linguistic classifiers for large Arabic word vocabulary recognition*, IJPRAI, Vol. 28, No. 1, pp. 1-39, 2014.

[4] A. Kacem and A. Saïdani, *A Texture-based Approach for Word Script and Nature Identification*, PAA, pp. 267-272, 2016.

[5] A. Saïdani, A.Kacem and A. Belaïd, *Co-occurrence Matrix of Oriented Gradients for Word Script and Nature Identification*, In Proceedings of ICDAR, pp. 16-20, 2015.

[6] K. Khazri Ayeb, A. Kacem Echi and A. Belaïd, *A Syntax Directed System for the Recognition of Printed Arabic Mathematical Formulas*, Proceedings of ICDAR, pp. 186-190. 2015

[7] N. Aouadi and A. Kacem, *Word Spotting for Arabic Handwritten Historical Document Retrieval using Generalized Hough Transform*, In proceeding of PATTERN, 2011.

[8] N. Aouadi and A. Kacem, *A proposal for touching component segmentation in Arabic manuscripts*, PAA, pp. 1-23, 2016.

[9] M.M. Galloway, *Texture analysis using gray level run lengths, Computer*, Graphics Image Process, vol. 4, no. 2, pp. 172-179, 1975.

[10] A. Sharaf Eldine, *Arabic manuscripts information system*, In Proceeding of the 6th International Conference and Exhibition on Multi-Lingual Computing, pp. 411-418, Cambridge, UK, 1998.

[11] N. Ouwayed, *Segmentation en lignes de documents an- ciens: application aux documents arabes*, PHD thesis, Nancy University, 2010.