Handwritten Character Recognition Based on the Specificity and the Singularity of the Arabic Language

Youssef Boulid[1], Abdelghani Souhar[2], Mohamed Youssfi Elkettani[1]

[1]*Department of Mathematics, Faculty of Sciences, University Ibn Tofail, Kenitra, Morocco*
[2]*Department of Computer Science, Faculty of Sciences, University Ibn Tofail, Kenitra, Morocco*

***Abstract* — A good Arabic handwritten recognition system must consider the characteristics of Arabic letters which can be explicit such as the presence of diacritics or implicit such as the baseline information (a virtual line on which cursive text are aligned and/join). In order to find an adequate method of features extraction, we have taken into consideration the nature of the Arabic characters. The paper investigate two methods based on two different visions: one describes the image in terms of the distribution of pixels, and the other describes it in terms of local patterns. Spatial Distribution of Pixels (SDP) is used according to the first vision; whereas Local Binary Patterns (LBP) are used for the second one. Tested on the Arabic portion of the Isolated Farsi Handwritten Character Database (IFHCDB) and using neural networks as a classifier, SDP achieve a recognition rate around 94% while LBP achieve a recognition rate of about 96%.**

***Keywords* — Handwritten Arabic Character Recognition, Feature Extraction, Texture Descriptor, Structural Feature.**

I. Introduction

Today even with the emergence of new technologies, people still use the paper as a physical medium of communication and information storage. The collection and archiving of papers and historical documents is one of the greatest goals of nations, as these archives are an inexhaustible mine of valuable information.

Many documents are stored in their original form (as papers). Scanning might be enough to preserve these documents from degradation, but it is not good enough to allow for quick access to information using text queries.

Intelligent Character Recognition (ICR) systems allow the conversion of handwritten documents into electronic version, while Optical Character Recognition (OCR) systems deal with printed documents (produced by typewriter or computer). ICR is more difficult to implement because of the wide range of handwritten styles as well as image degradation.

There are several applications which use ICR, such as document digitization, storing, retrieving and indexing, automatic mail sorting, processing of bank checks and processing of forms. The importance of these applications has leads to intensive research for several years.

1. The architecture of an ICR system consists generally of five stages (fig.1):

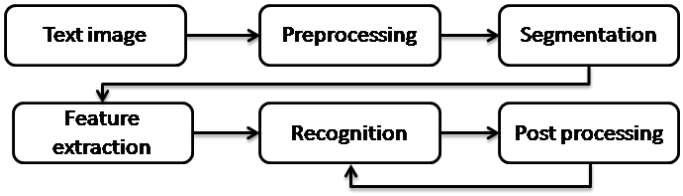


Fig. 1. Phases of character recognition process.

2. Preprocessing: contains techniques for image enhancement and normalization, such as: smoothing, noise reduction, slope normalization, contour detection, slant and skew correction …etc
3. Segmentation : the process of partitioning a document into homogeneous entities such as lines, words and characters;
4. Feature extraction: This deal with the extraction of some features from the image, which should permit discrimination between different classes (words or characters).
5. Learning/Recognition: which allows learning the classification rules based on the characteristics drawn from a training set.
6. Post-processing : It contains techniques for word verification (lexical, syntax and semantic)

Arabic language today is spoken by over 300 million people and it is the official language of many countries, and it contains a huge inheritance of documents to digitize. A robust system of recognition of Arabic handwriting documents can also serve other languages using the Arabic script such as Farsi, Urdu...

The Arabic script is written from right to left and is semi-cursive in both printed and handwritten versions. There are 28 letters, and the shape of these letters change depending on their position in the word, as they are preceded and/or followed by other letters or isolated, some letters can take four different forms: for example the letter 'Ain' (ع, عـ, ـعـ, ـع).

The diacritics play an essential role in reading, some letters have the same shape, and the only distinction is the number of points (diacritics) that can go up to three and their positions either on top or bottom of the letter. For example, three different letters 'ba','taa','thaa' (ب, ت, ث) have the same basic shape but the position and the number of points are different (fig.2).

All these facts make the Arabic script more challenging than other script like Latin.

Many collections of Arabic manuscripts are now in archives and libraries around the world, but unfortunately despite its importance, remained not exploited.

This paper focuses on the recognition of Arabic handwritten letters in their isolated form. We will investigate the efficiency of two feature extraction methods namely: Spatial Distribution of Pixels (SDP) and Local Binary Pattern (LBP) while taking into consideration the specificity of the Arabic script.

The rest of the paper is organized as follows: The next section, examines some related works on isolated Arabic character recognition. Section 3 describes the used dataset, the pre-processing and the classification stage. Section 4 provides a detailed overview and the results of the used feature extraction methods. Section 5 gives a comparative analysis of the proposed method with other works. Finally section 6 concludes the paper.

DOI: 10.9781/ijimai.2017.446

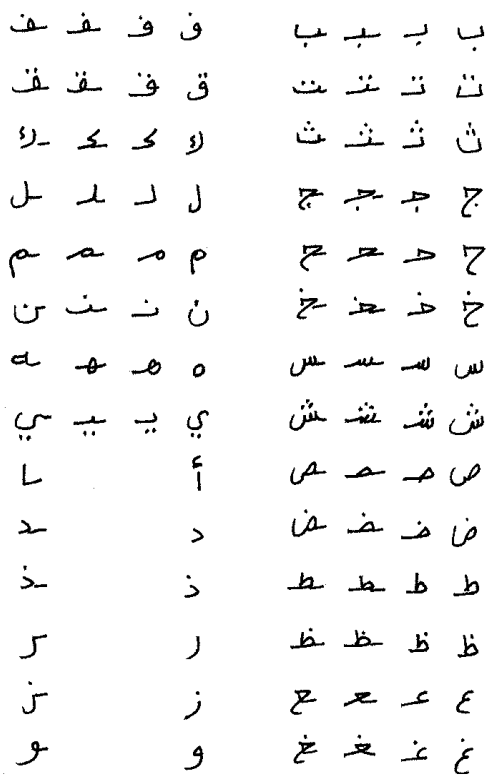


Fig. 2. Different form of the Arabic alphabets according to their positions.

All these facts make the Arabic script more challenging than other script like Latin.

Many collections of Arabic manuscripts are now in archives and libraries around the world, but unfortunately despite its importance, remained not exploited.

This paper focuses on the recognition of Arabic handwritten letters in their isolated form. We will investigate the efficiency of two feature extraction methods namely: Spatial Distribution of Pixels (SDP) and Local Binary Pattern (LBP) while taking into consideration the specificity of the Arabic script.

The rest of the paper is organized as follows: The next section, examines some related works on isolated Arabic character recognition. Section 3 describes the used dataset, the pre-processing and the classification stage. Section 4 provides a detailed overview and the results of the used feature extraction methods. Section 5 gives a comparative analysis of the proposed method with other works. Finally section 6 concludes the paper.

II. Related Works

The challenge concerning the shape recognition problem such as handwritten character recognition remain in finding features that maximize the interclass variability while minimizing the intra-class variability [1]. Feature extraction methods can be categorized into two classes [2]:

- Structural features [3, 4], which extract geometrical and topological properties such as the number and position of dots, the presence of loops, the orientation of curves...etc.
- Statistical features [5], such as histograms of projection profile and transitions, moments, histograms of gray level distribution, Fourier descriptors and chain code...etc.

A Technique for recognizing hand printed Arabic characters using induction learning is presented in [6]. The method extracts structural features while tracing the path from the skeleton of the character where it finds an end or junction point. These features are primitives such as lines, curves and loops which are stored in a binary tree where the relationship between them is considered. The features such as the relation between primitives, the orientation of lines and curves, and the number of dots are used in an inductive learning program, in order to generate Horn clauses. As mentioned by the authors this can generalize over large degree of variation between writing styles. 30 samples from each character were selected for training and 10 samples were used for test. The average correct recognitions rate obtained using cross-validation was 86.65%.

The method in [7] uses preprocessing steps to remove noise, and then extract morphological and statistical features from the main body and secondary components. Using back propagation neural network on the CENPRMI [8] dataset, they report 88% of recognition rate.

Based on the extraction of normalized central and Zernike moments features from the main and secondary components, the work in [9], uses SVM as classifier with Non-dominated Sorting Genetic Algorithm for feature selection. The authors claim to reach 10% classification error on a dataset of isolated handwritten character of 48 persons.

The authors in [10] use neural network with the wavelet coefficients on a corpus of Arabic isolated letters from more than 500 writers. Using a network whose input vector contains 1024 input give 12% of error rate.

The work in [11] proposes a set of features to distinguish between similar Farsi letters. The first stage is based on the general shape structure to find the best match for a letter. In the second stage statistical feature such as distributive and concavity are extracted after partitioning a letter into smaller parts, this allows the distinction of structurally dissimilar letters. Vector quantization has been employed to test the features on 3000 letters and achieved 85.59% of accuracy.

The method proposed in [12], pre-processes the images in order to remove noisy points, and then uses moments, Fourier descriptor of the projection profile and centroid distance with Principal Component Analysis to reduce dimension of the feature vector. Using SVM on a database containing 1000 Arabic isolated characters, the authors achieved a 96.00% of recognition rate.

The paper in [13] presents a comparative study for window-based descriptors for the recognition of Arabic handwritten alphabet. Descriptors such as HOG, GIST, LBP, SIFT and SURF are first extracted from the entire image and evaluated using Support Vector Machine, Artificial Neural Network and Logistic Regression and then extracted from horizontal and vertical overlapped spatial portions of the image. The same paper introduces a dataset for Arabic handwritten isolated alphabet which contains about 8988 letters collected from 107 writers. The authors claims to reach 94.28% as the best recognition rate using SVM with RBF kernel, with SIFT features from the entire image.

In 2009 [14], a competition for handwritten Farsi/Arabic character and digit recognition, grouped four works. For character recognition the CENPARMI and IFHCDB [15] databases were used in training and testing steps. The best reported recognition rate is 91.85% that corresponds to a system based on hierarchy of multidimensional recurrent neural networks that works directly on raw input data with no feature extraction step.

The paper in [16] discusses the effectiveness of the use of Discrete Cosine Transform and Discrete Wavelet Transform to capture discriminative features of Arabic handwritten characters. On a dataset of 5600 characters, the coefficients of both techniques have been extracted in a zigzag fashion from the top-left corner of the image and used as input to the artificial neural network. Extracting 400 coefficients from the 128x128 resized image gives about 79.87% for DCT and 40.71% for DWT.

In [17] a two-stage SVM based scheme is proposed for recognition of Farsi isolated characters. After binarizing the image, the undersampled bitmaps and chain-code directional frequencies of the contour pixels are used as features. For the first stage the characters are grouped into 8 classes, and then the undersampled bitmaps feature is used to assign an input image to the class that belongs to. In the second stage the classifier are trained on those classes using this time chain-code feature in order to discriminate between the characters belonging to the same class. Using the IFHCDB database, the authors reach a recognition rate of 98% and 97%, respectively for 8-class and 32-class problems.

In [18], after applying some preprocessing technique and normalizing the image, zoning and crossing counts are combined to represent the feature set. Self-organized map is used to cluster the classes and for creation of binary decision tree. For each node the classifier among SVM, KNN and neural networks who gives the best recognition rate is considered as the main classifier of the node. Tested on the IFHCDB dataset, the authors claim the reach a recognition rates of 98.72, 97.3 and 94.82 respectively when considering 8, 20 and 33 clusters.

A method was proposed [19] for the recognition of Persian isolated handwritten characters, after preprocessing the images, the derivatives of the projection profile histograms in four directions is used as feature with a Hamming neural network. This classifier is implemented using CUDA on GPU in order to speed-up the classification time. On a subset of the Hadaf database, the author claim to reach a 94.5% of recognition rate while accelerating the algorithm 5 times.

III. Dataset, preprocessing and classification stage

A. Dataset

The database used in this work, is the same used in the ICDAR 2009 competition [14] under the name of Isolated Farsi Handwritten Character Database (IFHCDB) [15], it contains 52380 characters and 17740 numbers. The images are scanned in grayscale at a resolution of 300 dpi, and each character has a size of 95x77. The distribution of characters in this database is not uniform, which means that the number of samples is not the same for all characters.

In this work only of the Arabic portion (28 Arabic characters) is considered, which represents approximately 97% of all character set (Arabic and Farsi), and which is divided into 35989 images for learning and 15041 for the testing.

B. Preprocessing

In the feature extraction phase the used LBP descriptor (see section IV) which allows to describe textures relies heavily on the distribution of the grayscale level in the image and since the characters in the dataset are extracted from specific areas; these areas sometimes contain noise and degradation of gray level. Extracting the LBP histogram from the entire image of the character poses a risk of capturing useless information and can thus have a negative impact on the recognition results.

To remedy to this problem; first the image is binarized using a global threshold (i.e. Otsu), then convolved with a “low pass” filter, which results in smoothing the contour of the character (fig.3).

This pretreatment allows LBP to capture the grayscale level of pixels that lie within the contour which will better discriminate between different characters.

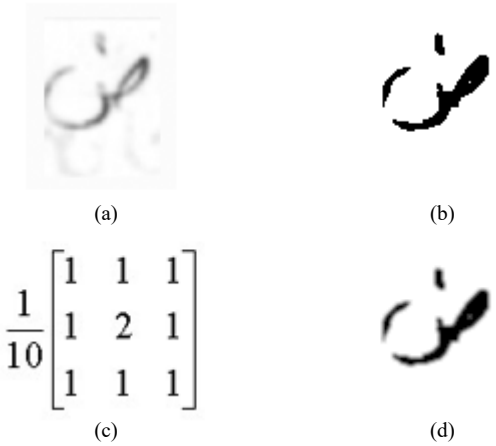


Fig. 3. Preprocessing steps: (a) The original image, (b) the binary image using Otsu, (c) the smoothed image using the filter in (d).

C. Classification

The interest here is to investigate the feature extraction methods, in order to choose the one that works best. So initially we choose Artificial Neural Networks (ANN) as classifier. Afterwards, if the used classifier is replaced with another one that outperforms neural networks, it will certainly improve the results.

A feed-forward neural network with Scaled Conjugate Gradient as training algorithm is used here. After experimental tests, a configuration of 100 hidden layers gives the optimal recognition rates.

IV. Feature extraction

To recognize the Arabic characters, a descriptor must be able to recognize two different characters that are written similarly (many characters have the same body shape and can only be differentiated using the information about the diacritical points), but at the same time has the capability to recognize the same character that is written differently as shown in the Fig. below.

Fig. 4. Some Arabic characters writing in a similar way. (a) samples of handwritten character, (b) printed form of the characters in (a).

Furthermore, the baseline information which is a special characteristic of the Arabic script has to be taken into consideration to further improve the recognition of similarly written characters.

A. The concept of baseline in the Arabic script

The baseline is a virtual line on which cursive text are aligned and join. According to the shape of Arabic characters, the baseline can be either at the top, at the bottom or across the body of the character (the horizontal lines in Fig. 5).

Words of Arabic language are made up of characters written from right to left and linked with each other. These linking parts allow continuity and smoothness in the writing which in turn allows easy and fast reading (the red portions in Fig. 5).

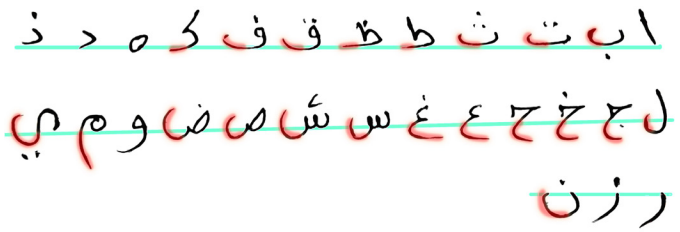
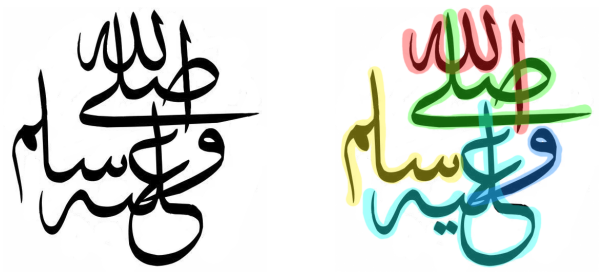


Fig. 5. The position of the baseline (bottom, across and top) according to different characters.

In fact most of isolated Arabic characters have two parts: one contains the identity of the character (the useful part) and the other contains the linking part with the next character. This gives Arabic script more flexibility in writing, such as the case of calligraphy as shown in Fig. 6, we steel read the words easily and this thanks to the presence of the baseline information which lies in useful part of the characters.



(a) (b)

Fig. 6. Example of Arabic calligraphy: (a) the original sentence, (b) detection of different word in the sentence according to the baseline information.

When dealing with the recognition of Isolated Arabic characters one must take into consideration that the linking parts in the characters can lead to confusion between those having similar ones. As a solution, in the feature extraction phase, the character must not be considered as one part; rather it must be segmented into different parts allowing us to distinct the useful one from the linking one. So we seek a unique way to divide all the characters in a way that approximates their baseline.

Inspired by these observations and in order to cope with the problem of similarity of Isolated Arabic characters, the problem can be seen from two different perspectives:

- The shape of character is a set of pixels that are spatially distributed. The way these pixels are arranged informs us about the structural features of such character and gives us idea about the location of the useful parts. Here, techniques that capture the distribution of pixels from different regions can be used.
- The way how the character is written influences on it shape i.e. the texture in the beginning of the character (the useful part) is not the same as in its ending. So statistical techniques can be used to better capture this information.

In the following, we introduce each of the chosen feature extraction method namely Spatial Distribution of Pixels and Local Binary Patterns corresponding to the above perspectives.

B. Spatial Distribution of Pixels (SDP)

1) Definition

The image of the character can be seen as a set of pixels that are linked together and have a certain position and dispersion in the space. Capturing this information will allow to better distinguish between the Arabic letters.

To measure this dispersion of pixels, the rectangle enclosing the character and its diacritical points is divided using a grid, which will allow for a better comparison between different characters even if they could have different sizes.

As explained below, partitioning of the image into an odd number will be adequate to measure the symmetry and the arrangement of the Arabic characters:

- The division of the character into four equal and symmetrical parts allows the distinction between symmetrical characters such as (ب, ت, ن…) from asymmetrical ones (ص, ش, و, م…).
- The location of diacritics in columns C2, C3 and C4 (fig.7), allows to differentiate characters that have similar shapes but differ in the number and position of diacritics such as (س, ش, ر, ز, ت, ث).
- The location of the baseline in rows R2, R3 and R4 (fig.7), provides additional information to differentiate between characters such as (ب, ن, ا, ر…)
- The division of the character in four regions: up, down, left and right, allows it to be precisely located in the bounding box i.e. (و, م, ل, ن…).

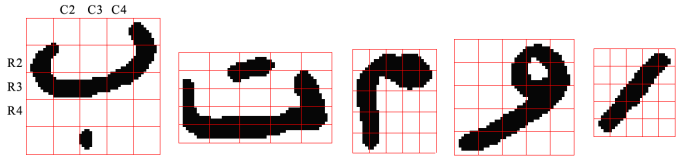


Fig. 7. Partitioning the rectangle enclosing the character in 25 regions of the same size.

The reason we choose to divide the character into a 5x5 grid is justified by the fact that 5 is the smallest number that permits the above points while reducing the computation time.

In what follows, certain blocks in the grid that will allow us to capture the information mentioned above are explained.

2) SDP application

To extract feature vector of spatial distribution of pixels, four configurations are considered:

a) The first configuration

The number of black pixels in the four blocks of the grid is divided by the area of these blocks (each block contains four regions as shown in fig.8.a). In the same way, the percentage of black pixels in the middle column and the middle row of the grid is computed as shown in fig.8.b and fig.8.c.

Finally these percentages are used to differentiate between characters.

Using the feed-forward neural network, we have got a 81.22% of recognition rate.

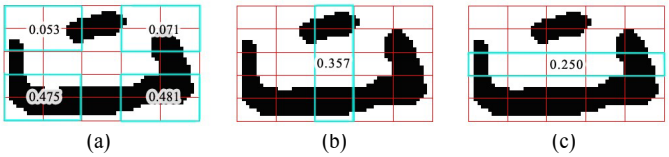


Fig. 8. The calculation of percentages according to the first configuration.

b) The second configuration

In order to integrate the information about the location of the character's body, the following percentages are calculated:

- The percentage of black pixels in the two rows at top (fig.9.a).
- The percentage of black pixels in the two rows at down (fig.9.a).
- The percentage of black pixels in the two columns at left (fig.9.b).
- The percentage of black pixels in the two columns at right (fig.9.b).

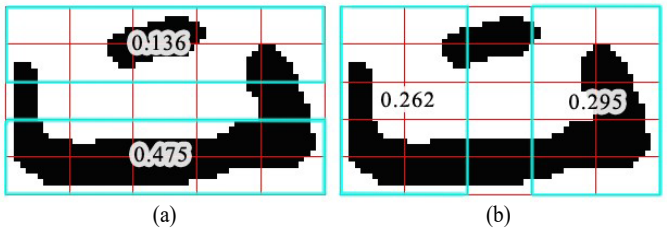


Fig. 9. The calculation of percentages according to the second configuration.

In addition to the previous configuration, the feature vector in this one contains the four percentages about the location of the character.

Using the feed-forward neural network with this new configuration, we have achieved a rate of 87.65%.

c) The third configuration

After splitting the rectangle enclosing the character and its diacritics in 25 regions of the same dimensions, the percentage of black pixels in each region is calculated by dividing their number by the total number of black pixels in the rectangle. In this case the overall shape of the image is the one of interest (like zoom out).

These percentages are inserted line by line in a vector which will be considered as descriptor of the character (fig.10).

Using the feed-forward neural network, we found 92.99% of recognition rate.

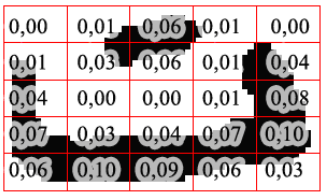


Fig. 10. The calculation of percentages according to the third configuration.

d) The fourth configuration

In this configuration, the percentage of black pixels in each region is calculated by dividing their number by the area of the region where they are located. In this case, the interest is the information about the presence and the fill of local pixels in each region (like zoom in).

These percentages are inserted line by line in a vector which is used to describe the character (fig.11).

Extracting the vector according to this configuration has achieved a rate of 93.84%.

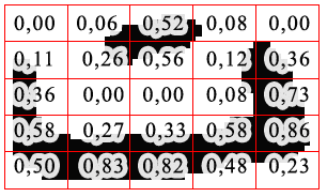


Fig. 11. The calculation of percentages according to the fourth configuration.

3) Summary

Table I shows the TOP-5 measures of recognition rates of the four configurations for extracting SDP feature using the feed-forward neural network described above.

For instance the TOP2 and TOP4 are respectively the percentages of samples that the true class is among the two first and the four positions in the list of candidates.

TABLE I

TOP-5 MESURES FOR DIFFERENT CONFIGURATIONS OF SDP EXTRACTION

Configuration of SDP extraction	TOP1	TOP2	TOP3	TOP4	TOP5
First configuration	81.22	91.52	95.76	97.50	98.37
Second configuration	87.65	95.39	97.83	98.82	99.25
Third configuration	92.99	97.61	98.80	99.21	99.43
Fourth configuration	**93.84**	**97.76**	**98.84**	**99.27**	**99.44**

The first configuration includes the concept of symmetry, location of the baseline and diacritical points. In addition to the first configuration, the second one includes information about the location of the character in the grid.

In the third configuration, the percentage of black pixels in all the cells of the grid, allows to encompass all the concepts of the first two configurations, which results by an increase in the recognition rate.

Finally, in the fourth configuration, the calculation of the percentages of pixels with respect to the regions they are located in, which also helps to make the descriptor more insensitive to the size of the character, and therefore improves the recognition rate evermore.

C. Local Binary Pattern

1) Definition

Developed by Ojala et al [20], LBP for Local Binary Pattern is a descriptor designed firstly to analyze textures in terms of local spatial patterns and gray level contrast.

In its basic form, a 3x3 window is used to produce labels for each pixel by thresholding to neighboring pixels with the center pixel in the window and considers the result as a binary number. The histogram of the $2^8 = 256$ different labels is used to describe a texture.

In [21], LBP has been improved to be used with a circular neighboring with a radius and a number of neighborhoods.

The decimal value of a pixel is calculated as:

$$LBP_{P,R} = \sum_{p=0}^{P-1} f(g_p - g_c)2^p \qquad f(x) = \begin{cases} 1, & f \quad x \geq 0; \\ 0, otherwise \end{cases}$$

With, P stands for the number of neighbors, R is the radius of the circle, and g_p is the gray level of the pixel in the position p.

Fig. 12 shows an example of LBP calculation. The neighboring pixels that are greater or equal to the value of the central pixel are replaced by 1 and the lower pixels are replaced by 0. The LBP label corresponds to the binary value taken in a circular order (11101101) which is then converted to decimal (237).

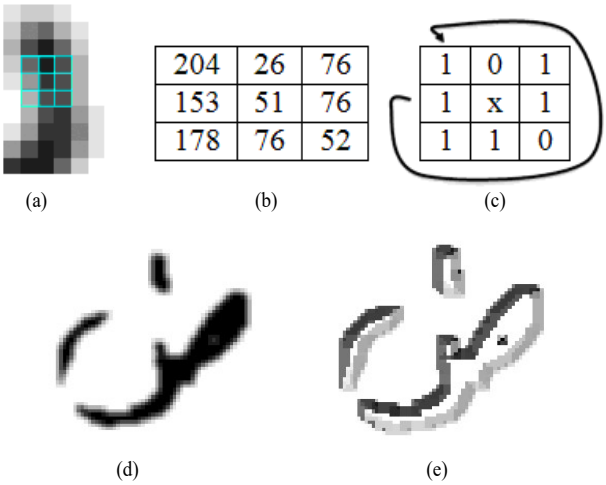


Fig. 12. Example of LBP computation: (a) a 3x3 window centered on a pixel, (b) the gray levels in the window, (c) thresholding the neighborhood compared to pixel x, (d) an image after preprocessing, (e) result of LBP application with a configuration $LBP_{8,1}$

Another extension of LBP is called Uniform Pattern [21], it allow reducing the size of the vector while keeping the same performance. The idea comes from the fact that there are patterns that occur frequently compared to others and must be grouped in a single pattern.

A pattern is called uniform if it contains at most two bit transitions from 0 to 1 and vice versa. For example 00000000 (0 transitions), 01110000 (two transitions) and 11001111 (two transitions) are Uniform, but the patterns 11001001 (4 transitions) and 01010010 (6 transitions) are not. For a neighborhood of 8 pixels, there are 256 patterns, 58 of them are uniform.

After the calculation of the label of each of pixel $f_{lbl}(x,y)$, the LBP histogram can be defined by:

$$H_i = \sum_{x,y} I\{f_{lbl}(x,y) = i\}, i = 0,...,n-1$$

N is the number of labels, *I{A}* equal 1 if A is true, otherwise it equal 0.

In addition to its success in several areas such as: face recognition [22], writer identification [23], and handwritten and digits recognition [24], the motivation behind the use of LBP as a descriptor is justified by the fact that it has proved its discriminating power for textures through the combination of information of spatial local patterns and intensity which we believe can be useful in the shape recognition problem.

To detect the variation in the strokes of the character, we can look at the neighborhood of the pixels in the contour. We found out that it is more appropriate to use LBP method which offers technical possibilities to implement this vision.

In this work, the uniform version of LBP with a configuration of $LBP_{8,1}$ is used.

2) LBP application

The natural idea is to extract the LBP histogram from the entire image (fig.13). Using the feed-forward neural network, we have got an 85.26% of recognition rate.

To improve this recognition rate, the useful part of the image is considered, i.e. the part containing only the character, by cropping it.

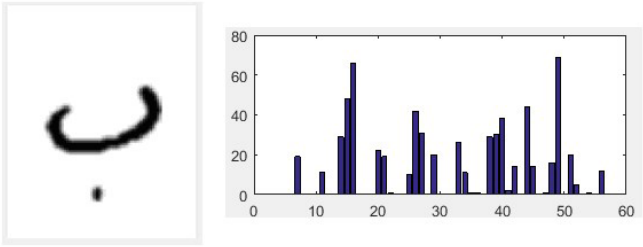


Fig. 13. (a) The input image, (b) the LBP histogram extracted from the entier image.

a) The first configuration

In this configuration, the character and its diacritical point are included within the smallest rectangle (fig.14).

Extracting LBP histogram in this configuration has achieved a rate of 88.47%, an improvement of about 3.21% compared to the extraction of LBP from the entire image.

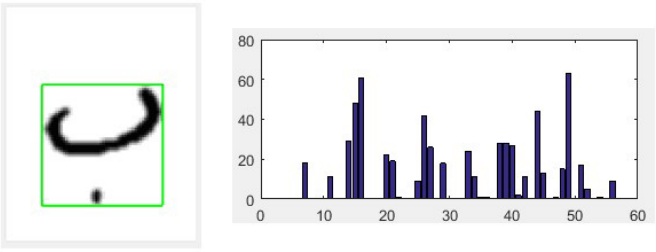


Fig. 14. (a) In green the rectangle enclosing the character, (b) the LBP histogram extracted from the rectangle.

b) The second configuration

In their isolated forms several Arabic characters have similar endings or terminations (ج, ع, س, ص). If we only extract the LBP histogram from the whole image, it does not effectively allow differentiating between different characters that are written almost in the same way.

To remedy to this problem, we try to approximate the baseline of the character by dividing the rectangle enclosing the character into four regions from the center of gravity of the body of the character, in order to break it into the four quadrants while highlighting areas which do not look the same. As shown in fig.15, the portions a2 and a4 respectively look similar to the portions b2 and b4, on the other hand portions a1, a3, b1 and b3 are different.

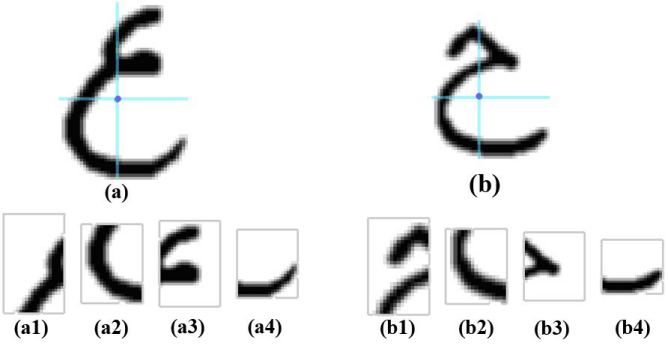


Fig. 15. Different regions after spliting the character into four regions from its centroid.

After splitting the bounding box into four regions (quadrants) from the centroid of the character, LBP histograms are extracted from each region and concatenated to form one feature vector of 236 elements (fig.16).

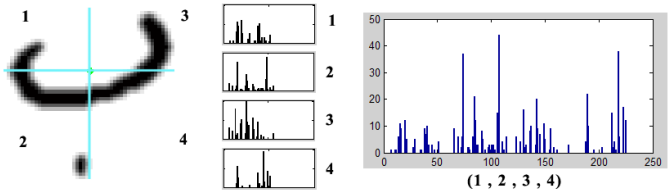


Fig. 16. Concatenation of the histograms extracted from the four regions from the centroid of the character.

This configuration has significantly improved the recognition rate, until we achieve a rate of 96.10%.

c) The third configuration

In the previous configuration we were interested on the body shape of the character, but in the Arabic language, there are more characters that are distinguishable only by the number and position of diacritics, for example (ح, خ, ب, ت, ن). In addition the histogram of LBP of the entire image does not capture the information of the existence of diacritical points.

To address this problem and to better approximate the baseline, the rectangle enclosing the character is divided into four regions from the point situated halfway between the centroid of the body and the centroid of diacritics. This will highlight the region that contains diacritical point(s). As shown in fig.17, the a2 and b3 portions contain information of diacritical points.

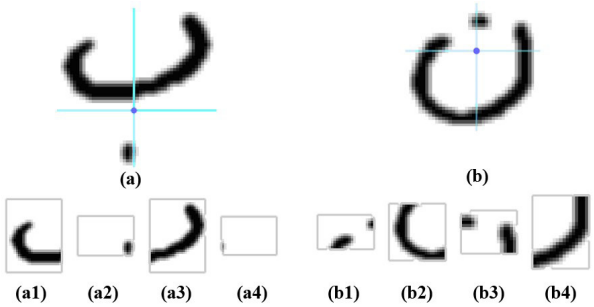


Fig. 17. Different regions after spliting the character into four regions from the centroid of both the character and its diacritics.

After splitting the bounding box in four regions from the center point of the centroid of the character and the centroid of diacritics, LBP histograms are extracted from each region and concatenated to form a vector of 236 elements (fig.18).

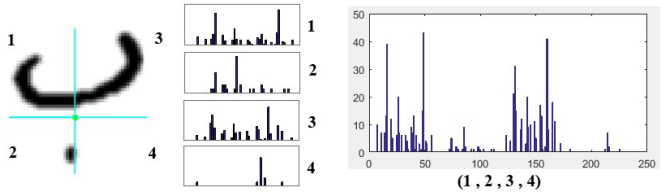


Fig. 18. Concatenation of the histograms extracted from the four regions from the centroid of of the character and its diacritics.

This configuration enables to further improve the recognition rate and achieve a rate of 96.31%.

3) Summary

Table II illustrates the top-5 recognition rate measures of different ways of extracting LBP features.

TABLE II

TOP-5 MESURES FOR DIFFERENT CONFIGURATIONS OF LBP EXTRACTION

Configuration of LBP extraction	TOP1	TOP2	TOP3	TOP4	TOP5
The entire image	85.26	93.26	96.28	97.72	98.43
First configuration	88.47	94.98	97.18	98.26	98.84
Second configuration	96.10	98.88	99.45	99.68	99.80
Third configuration	**96.31**	**98.91**	**99.44**	**99.65**	**99.78**

It is clear that it is important to use the bounding box enclosing the character instead of the whole image. The concatenation of LBP histograms extracted according to the third configuration gives the best results.

These results are obtained because the third configuration process respected the nature and the characteristics of the Arabic language:

- Using the centroid information to get close to the baseline of the character, which is in fact within the character itself.
- Dividing the bounding box into four regions, allows differentiating different characters having resembling shapes (ع, ح).
- Integrating the centroid information of diacritics, allows to differentiate between similar letters shapes but differs in the diacritics (ن, ب, ث, ت).

V. COMPARATIVE ANALYSIS

From the results it is clear that LBP descriptor outperforms SDP, in what follows a comparative analysis of the proposed method (the third configuration of LBP) with the existing systems working on IFHCDB dataset is performed.

The dataset used here contains the Farsi letters (The known 28 Arabic letters plus the letters پ, ژ, چ, گ, which makes it 32 letters). So in order to test the robustness of the proposed method it will be interesting to experiment it on the Farsi set too.

In some works the 32 Persian characters are grouped in 8 or 20 classes containing character with similar shapes. Therefore in order to be able to compare the proposed method with the existing ones we consider the same classes in those studies to construct the 8-class, 32-class and 33-class problems.

Table III shows both versions adopted to build the 8-class:

TABLE III

THE TWO USED VERSION OF 8-CLASS PROBLEM

	8-class version 1 (8-v1)	8-class version 2 (8-v2)
Class 1	ظ – ط – آ - ا	ه – د – ر – و – ا
Class 2	ث - ت – پ – ب – ن – ل – ق – ف	م – آ
Class 3	غ – ع – خ – ح – چ – ج	پ – ب
Class 4	و – ژ – ز – ر – ذ – د	ذ – ز – ژ
Class 5	ي – ض – ص – ش – س	ش – س – ض – ص
Class 6	ك – گ	چ – خ – ح – ج – غ – ع
Class 7	ه	ظ – ط – ك – گ – ل
Class 8	م	ث – ت – ق – ف – ن – ي

It is to be noted that the authors in [17], have implemented the work in [27] for the problem of 8-class in order to compare the results.

Table IV shows the results of our method compared to others when considering 8, 32 and 33 class problems.

TABLE IV

COMPARISON OF THE PROPOSED SYSTEM WITH OTHER METHODS ON IFHCDB DATASET

Algorithms	Train size	Test size	Number of classes	Recognition rate (%)
Dehghan and Faez [26]	36682	15338	8 – v1	81.47
Alaei et al [17]	36682	15338	8 – v1	98.10
Rajabi et al [18]	36000	13320	8 – v2	98.72
Alaei et al [17]	36682	15338	32	96.68
Rajabi et al [18]	36000	13320	33	94.82
Proposed system	36437	15233	8 – v1	95.63
Proposed system	36437	15233	8 – v2	95.56
Proposed system	36437	15233	32	95.87
Proposed system	36437	15233	33	96.04

As can be seen from the results, the proposed method is competitive even if we can see that assigning some character in the same group can lead to misclassification. It is to be noted that our focus was only on the Arabic portion of the IFHCDB dataset.

Table V shows some characters in IFHCDB dataset that have been correctly recognized despite their similarity.

TABLE V

SAMPLES OF SIMILAR ARABIC HANDWRITTEN CHARACTERS WHICH WERE CORRECTLY RECOGNIZED

Character image					
Character class	ب	ت	ث	ن	ن
Character image					
Character class	ن	ف	ق	ك	ا
Character image					
Character class	د	ذ	ر	ر	م
Character image					
Character class	م	ي	ي	ل	ه
Character image					
Character class	س	ش	ح	خ	ع

VI. CONCLUSION

This paper deals with the recognition of Arabic handwritten characters. The goal is to find the best way of features extraction from two perspectives (textural and structural) while taking into consideration the specificity and the nature of the Arabic script.

SDP and LBP feature extraction methods are configured and used to implement these two perspectives. Tested on IFHCDB dataset and using ANN as classifier, we have found that the percentages of pixels extracted after dividing the image into a 5x5 grid, allows to reach a recognition rate around 94%, while extracting LBP histograms after dividing the image into four regions from the centroid of the character's body and its diacritics allows to achieve a recognition rate around 96%.

Since the recognition rate in TOP2 is about 99%, keeping two results in the recognition of a character belonging to a word is beneficial in the stage of word recognition, since we will have more than one possibility.

In future work, we will investigate the collaboration between what we call the word agent (the entity responsible to recognize a word) and the character agent.

ACKNOWLEDGMENT

The authors are thankful to S. Mozaffari for providing the dataset for the experiment. The authors are also thankful to the Maxware Technology stuff for their moral support and professional help, without forgetting faculty of science, University Ibn Tofail for providing the infrastructural facilities that helped to complete this work.

REFERENCES

[1] P.A. Devijver, J. Kittler, Pattern Recognition: A Statistical Approach, 1982.

[2] V. Govindan, A. Shivaprasad, Character recognition—a review, Pattern Recognit. (1990).

[3] L. Heutte, T. Paquet, J. V Moreau, Y. Lecourtier, C. Olivier, A structural/statistical feature based vector for handwritten character recognition, Pattern Recognit. Lett. 19 (1998) 629–641.

[4] M. Parvez, S.A. Mahmoud, Arabic handwriting recognition using structural and syntactic pattern attributes, Pattern Recognit. 46 (2013) 141–154.

[5] J. Cai, Integration of structural and statistical information for unconstrained handwritten numeral recognition, IEEE Trans. Pattern Anal. Mach. Intell. 21 (1999) 263–270. doi:10.1109/34.754622.

[6] A. Amin, Recognition of hand-printed characters based on structural description and inductive logic programming, in: Proc. Int. Conf. Doc. Anal. Recognition, ICDAR, 2001: pp. 333–337.

[7] A. Sahlol, C. Suen, A Novel Method for the Recognition of Isolated Handwritten Arabic Characters, arXiv Prepr. arXiv1402.6650. (2014).

[8] H. Alamri, J. Sadri, Ching Y Suen, N. Nobile, C.Y. Suen, A Novel Comprehensive Database for Arabic Off-Line Handwriting Recognition, Elev. Int. Conf. Front. Handwrit. Recognit. (2008) 664–669, Montreal.

[9] G. Abandah, N. Anssari, Novel moment features extraction for recognizing handwritten Arabic letters, J. Comput. Sci. 5 (2009) 226–232. doi:10.3844/jcssp.2009.226.232.

[10] A. Asiri, M. Khorsheed, Automatic Processing of Handwritten Arabic Forms, in: Proc. World Acad. Sci. Eng. Technol., 2005: pp. 313–317.

[11] J. Shanbehzadeh, H. Pezashki, A. Sarrafzadeh, Features Extraction from Farsi Hand Written Letters, Image Vis. Comput. (2007) 35–40.

[12] R.Hamdi, F.Bouchareb, M.Bedda, Handwritten Arabic character recognition based on SVM Classifier, in: 3rd Int. Conf. Inf. Commun. Technol. From Theory to Appl., 2008: pp. 1–4.

[13] M. Torki, M.E. Hussein, A. Elsallamy, M. Fayyaz, S. Yaser, Window-Based Descriptors for Arabic Handwritten Alphabet Recognition: A Comparative Study on a Novel Dataset, arXiv Prepr. arXiv1411.3519. (2014).

[14] S. Mozaffari, H. Soltanizadeh, ICDAR 2009 handwritten Farsi/Arabic character recognition competition, in: Proc. Int. Conf. Doc. Anal. Recognition, ICDAR, 2009: pp. 1413–1417. doi:10.1109/ICDAR.2009.283.

[15] S. Mozaffari, K. Faez, F. Faradji, M. Ziaratban, S.M. A Golzan, A Comprehensive Isolated Farsi/Arabic Character Database for Handwritten OCR Research, Tenth Int. Work. Front. Handwrit. Recognit. (2006) 385–389.

[16] A. Lawgali, A. Bouridane, Handwritten Arabic Character Recognition: Which feature extraction method, Int. J. Adv. Sci. Technol. 34 (2011) 1–8.

[17] A. Alaei, P. Nagabhushan, U. Pal, A new two-stage scheme for the recognition of persian handwritten characters, in: Proc. - 12th Int. Conf. Front. Handwrit. Recognition, ICFHR 2010, 2010: pp. 130–135.

[18] M. Rajabi, N. Nematbakhsh, S. Amirhassan Monadjemi, A New Decision Tree for Recognition of Persian Handwritten Characters, Int. J. Comput. Appl. 44 (2012) 52–58. doi:10.5120/6271-8433.

[19] M. Askari, M. Asadi, A.A. Bidgoli, Isolated Persian/Arabic handwriting characters: Derivative projection profile features, implemented on GPUs, J. AI. (2016).

[20] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture

measures with classification based on featured distributions, Pattern Recognit. 29 (1996) 51–59. doi:10.1016/0031-3203(95)00067-4.

[21] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24.7 (2002): 971-987. doi:10.1109/TPAMI.2002.1017623

[22] T. Ahonen, A. Hadid, M. Pietikainen, Face Description with Local Binary Patterns: Application to Face Recognition, TPAMI. 28 (2006) 2037–2041.

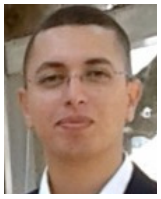
[23] Y. Hannad, I. Siddiqi, M.E.Y. El Kettani, Writer identification using texture descriptors of handwritten fragments, Expert Syst. Appl. 47 (2016) 14–22. doi:10.1016/j.eswa.2015.11.002.

[24] M. Biglari, F. Mirzaei, J. Neycharan, Persian/Arabic Handwritten Digit Recognition Using Local Binary Pattern, Int. J. Digit. (2014).

[25] S. Tulyakov, S. Jaeger, V. Govindaraju, D. Doermann, Review of classifier combination methods, Stud. Comput. Intell. 90 (2008) 361–386. doi:10.1007/978-3-540-76280-5_14.

[26] Poh, N., & Bengio, S. Database, protocols and tools for evaluating score-level fusion algorithms in biometric authentication. Pattern Recognit, (2006) 39(2), 223-233.

[27] M. Dehghan, Farsi handwritten character recognition with moment invariants, in: Digit. Signal Process. Proc., 1997: pp. 507–510.



Youssef Boulid received his M.S. degree in Decision Support Systems and Project Management in 2012 from University Ibn Tofail, Faculty of science, Kenitra- Morocco. Currently he is preparing a PhD at the same faculty. His research interests include image processing, handwritten document analysis, Arabic handwritten recognition and Artificial intelligence.

Abdelghani Souhar received M.S. degree in applied Mathematics in 1992, PhD degree in computer science in 1997 from University Mohammed 5 in Rabat - Morocco. Now he is a Professor at university Ibn Tofail in Kenitra - Morocco. His research interests include CAE, CAD and Artificial Intelligence.

Mohamed Elyoussfi Elkettani received M.S. degree in applied mathematics in 1980 and PhD degree in Statistics in 1984 from Orsay Faculty of Science, University of Paris XI. Now he is a Professor at university Ibn Tofail in Kenitra - Morocco. His research interests include Multivariate statistics and Image recognition algorithms.