Amazigh Handwritten Character Recognition based on Horizontal and Vertical Centerline of Character

Youssef Es Saady, Ali Rachidi, Mostafa El Yassa, Driss Mammass

IRF-SIC Laboratory, University Ibn Zohr, Agadir, Morocco
essaady2110@yahoo.fr, rachidi.ali@menara.ma, melyass@gmail.com,
mammass@univ-ibnzohr.ac.ma

Abstract

We present in this paper a system of Amazigh handwriting recognition based on horizontal and vertical centerline of the character. After the image preprocessing, the text is segmented into lines and then into characters. The positions of the vertical and horizontal centerline of the character are used to obtain a set of independent and dependent features to those lines. These features are related to the densities of pixels and are extracted on binary images of characters using the sliding window technique. Finally, a multilayer perceptron is used for character classification. The system was tested on two bases of the Amazigh characters: on a printed database of Amazigh characters and on another one for handwritten characters created locally. The correct average recognition rate obtained using 10-cross-validation was 99.28 % for the 19437 Amazigh printed characters and 96.32% for the 20150 Amazigh handwritten characters.

Keywords: Handwritten Character Recognition; Amazigh; Tifinagh; MLP; Segmentation.

1. Introduction

The recognition of characters and handwriting has been a popular research area for many years because of its potentials and various applications, such as bank cheque processing, postal automation, documents analysis, reading postal codes and reading different forms, etc. Several scientific researchers have been carried out for handwritten recognition of English language, Chinese/Japanese/Hindi languages, and Arabic/ Farsi language [1], [2], [3], [4], [5], [6], [7]. Various approaches have been proposed by researchers for automatic recognition of these characters. However, less attention had been given to Amazigh language recognition. Recently, some efforts have been reported in literature for Amazigh characters: based on the Hough transformation [8], the statistical and geometrical approaches [9], [10], artificial neural network [11], [12], [13], Hidden Markov Models [14], [15], syntactical method based on the finite-state machines [16], [17] and dynamic programming [18].

In this paper, we propose a recognition system for handwritten Amazigh characters. The scanned document is pre-processed to ensure that the characters are in a suitable form. Then the text is segmented into lines and then into characters. Parameters such as vertical and horizontal centerlines are used to derive a subset of dependent centerline features from the segmented characters. For Latin and Arabic writing, several baselines have been used to extract features that depend on these baselines; For this reason, different approaches based on the positions of these lines have been proposed in the literature [19], [20] [21], [22]. For Amazigh writing, we use the positions of the vertical and horizontal centerline of the character to derive a set of dependent and independent features to those lines. Finally a

multilayer perceptron with one hidden layer are used for the training step of extracted features and the recognition of characters.

The paper is organized as follow: Section 2 presents an overview about the Amazigh language. In Section 3, we refer to recent related works on the recognition of character and handwriting Amazigh. Section 4 displays the system architecture of the proposed method and we will deeply discuss, respectively, pre-processing and feature extraction. Section 5 presents the database developed. In Section 6, we present the training and recognition steps. Results are given in section 7. Finally, conclusions and future work are discussed in section 8.

2. The Amazigh Language

The Amazigh (Berber) language is spoken in Morocco, Algeria, Tunisia, Libya, and Siwa (an Egyptian Oasis); it is also spoken by many other communities in parts of Niger and Mali. It is used by tens of millions of people in North Africa mainly for oral communication and has been integrated in mass media and in the educational system in collaboration with several ministries in Morocco [23]. In linguistic terms, the language is characterized by the proliferation of dialects due to historical, geographical and sociolinguistic factors (the orthographic details are discusses in [24]). In Morocco, the term Berber (Amazigh) encompasses the three main Moroccan varieties: Tarifite, Tamazighte and Tachelhite. More than 40% of the country's populations speak Berber. So, all the Moroccans are concerned with this language. The establishment of The "Royal Institute of the Amazigh Culture" (IRCAM) carried out a major action to standardize the Amazigh language. In the same tread, and since 2003, the Amazigh language has been integrated in the Moroccan Educational System. It is taught in the classes of the primary education of the various Moroccan schools, in prospect for a gradual generalization at the school levels and extension to new schools [25].

The Tifinagh is the writing system of the Amazigh language. An older version of Tifinagh was more widely used in North Africa. It is attested from the 3rd century BC to the 3rd century AD. The Tifinagh has undergone many changes and variations from its origin to the present day [17]. The old Tifinagh script is found engraved in stones and tombs in some historical sites in northern Algeria, in Morocco, in Tunisia, and in Tuareg areas in the Sahara. The Fig. 1 shows a picture of an old Tifinagh script found in site of rock carvings near from Intedeni Essouk in Mali [26].

Figure 1. Old Tifinagh Script, Site of Rock Carvings Near from Intédeni Essouk Mali [26]

The Amazigh alphabet which is called "Tifinagh-IRCAM", adopted by the Royal Institute of the Amazigh Culture, was officially recognized by the International Organization of Standardization (ISO) as the basic multilingual plan [27]. The Fig. 2 represents the repertoire of Tifinagh which is recognized and used in Morocco with their correspondents in Latin characters. The number of the alphabetical phonetic entities is 33, but Unicode codes only 31 letters plus a modifier letter to form the two phonetic units: ⵖⵯ (gᵂ) and ⴽⵯ (kᵂ).
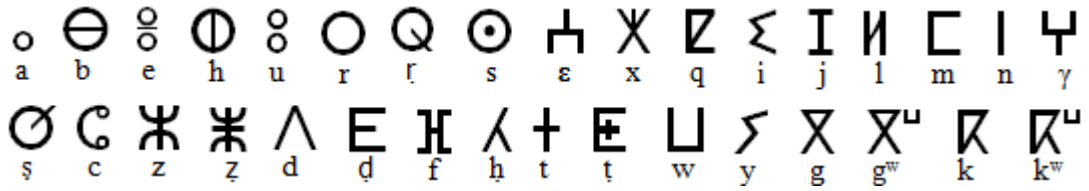
Figure 2. Tifinaghe Characters Adopted by the Royal Institute of the Amazigh Culture with their Correspondents in Latin Characters.

In contrast to Latin and Arab, the Amazigh alphabet is never cursive which facilitates the operation of segmentation. The Amazigh script is written from left to right; it uses conventional punctuation marks accepted in Latin alphabet. Capital letters, nonetheless, do not occur neither at the beginning of sentences nor at the initial of proper names. So there is no concept of upper and lowercase characters in Amazigh language. Regarding the figures, it uses the Arabic Western numerals. The majority of graphic models of the characters are composed by segments. Moreover, all segments are vertical, horizontal, or diagonal. Fig. 3 shows some of these characteristics in a few Amazigh texts from a schoolbook.
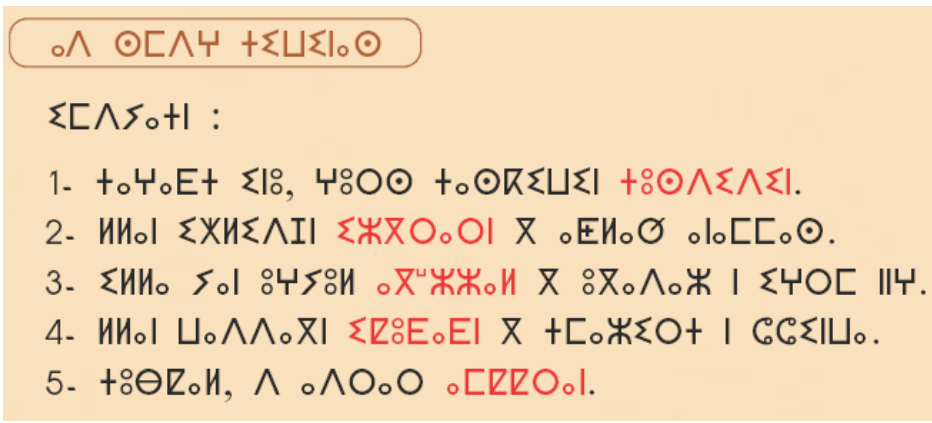
Figure 3. An example of a Amazigh Text from a Schoolbook

3. State of the Art

Comparing to English, Arabic or Chinese languages, researches on handwriting recognition of Amazigh language have not achieved perfection. As far as we know, few attempts have been carried out on the handwriting Amazigh recognition. In this section, we give a brief overview of recent works.

In [8], Oulamara and al. has described a statistical approach for Berber character recognition based on the Hough transform to extract straight line segments. The features are defined and measured in the parameter space obtained by the Hough transforming of the character shape. A single master frame that generates all alphabetic characters has been found; it allows the implementation of a new coding scheme in the Hough space. Then, a reading matrix is defined in order to match input characters with a reference alphabet. An original encoding is inferred and then used as the basis for construction of the matrix representing a coded reading of this alphabet. The results seem interesting on a set of printed Amazigh characters on a local database.

Djematene and al. consider that the approach in [8] is not a suitable technique for Amazigh handwritten characters since it produces incorrect segmentation [10]. To overcome this difficulty when the characters consist of curved strokes, Djematene and al. propose in ([9], [10]) a complete geometrical feature-points method for multiple font styles and handwritten script. The character is described by a set of feature points: the points of intersection, end points, corner points, dominant points, and significant dots, incorporated in a letter. By examining the position of these points in the input image bounding-rectangle; the skeleton of the normalized image is coded into a 27-letter string. The coding of character is often unique, so the character can be identified unambiguously by using a distance metric. The results obtained are encouraging especially for character consisting of straight strokes.

In [11], Ait Ouguengay and al. proposed an artificial neural network (ANN) approach for Amazigh characters recognition. The latter was trained on a database that contains the Amazigh spelling patterns of different fonts and sizes. The simulation of the neural network was carried out by the Free Software JavaNNS (Java Neural Network Simulator). A vector of primitives such as horizontal and vertical projections, gravity center, perimeter, area, compactness and central moments of order 2 are extracted on each character and feeds a multilayer perceptron with one hidden layer. The approach was tested on a database of printed Amazigh patterns. This approach has produced good results on all the training patterns. However, test results are still far from satisfactory because of the test database which is very low compared to the weight of ANN to be determined.

In [15], Amrouch and al. have given a simple and global approach for the Amazigh handwritten characters recognition based on Hidden Markov Models. The input is a vector of features extracted directly from an image of Amazigh characters using the Hough transform. This vector is translated into sequence of observations that is used for the learning phase. This task involves several of processing steps, typically, pre-processing and feature extraction. Finally the class of the input characters is determined by the Forward classifier. The authors obtained promising results on a locally database of handwritten Amazigh characters. However, the discrimination of these models is not very good because each Hidden Markov Model uses the learning of a single character. The error rate was recorded mainly due to bad writing and the training data.

In [18], El Ayachi and al. have presented a method for recognizing script's Tifinagh using dynamic programming, which contains three main parts: pre-processing, features extraction and recognition. In the preprocessing process, the scanned image document is midline skew corrected then segmented into lines using vertical histogram, lines into words, and words into character using the horizontal histogram. In the features extraction process, invariant moments and Walsh coefficients are computed. Finally, the dynamic programming is adopted in the recognition part. The tests are applied on several images. The experimental results showed that the recognition method using invariant moments give better results comparing to the method based on Walsh transform in terms of recognition rate and computing time.

In previous works, we had proposed two methods that contributed to increase the performance of Amazigh characters and handwriting recognition: a Syntactic Approach using Finite Automata [17] and an Automatic Recognition of Amazigh script based on Centerline of writing [13]. The choice of the syntactic approach [17] was justified by the fact that most Amazigh characters have segmented forms. After the pre-processing, appropriate algorithms allow to build the representative chain of the character from the Freeman coding. The chain will be used in the input of a finite automaton, which recognizes all the segmented Amazigh characters. This global automaton is built from specific automata of recognition for each Amazigh character. This automaton has been stored in a text file. The experimental results show the robustness of the approach. We have tested the proposed system on 660 characters of a local database of isolated printed Amazigh characters. 617 characters were well recognized that represents a rate of 93, 48% of good recognition. The errors come from some form of unrecognized characters skeleton which includes non-orthogonal segments. The limitation of this approach is that it is not applicable for non-segmented Amazigh characters.

In [14], we have developed a system of Amazigh handwriting recognition based on horizontal centerline of writing. After the pre-processing step, the text is segmented into lines and then into characters using the analysis techniques histogram of horizontal and vertical projections. The positions of the baselines of character (a central line, upper and lower line of writing) are used to derive a subset of baseline independent and dependent features. A subset of the features is related to the position of the horizontal centerline to take into account the association of the majority of Amazigh characters by this line. These features are related to the densities of pixels are extracted on binary images of characters using the sliding window technique. Some of these features have been used for Arabic and Farsi Handwriting Recognition in [19] and [28]. We used a multilayer neural network in the learning stages and recognition. The system showed good performance on a database of 19437 printed Amazigh characters developed in [11]. Significant increase in accuracy levels has been found in comparison of our method with the others for Amazigh character recognition. Furthermore, the experimental results have showed a significant improvement in recognition rate when integrating the features dependent on the horizontal centre line. The causes of errors are mainly due to the resemblance between certain Amazigh characters. To overcome these limitations, we will add, in this paper, other features based on the vertical centerline of the letter that will improve the results.

4. Our System Architecture

Generally, a system of off-line handwritten character recognition includes three stages: pre-processing, feature extraction, and classification. The process of handwriting recognition involves extraction of some defined characteristics called features permitting to classify an unknown character into one of the known classes. The pre-processing is primarily used to reduce variations of handwritten characters, to correct the skew of the text lines and to segment the text into isolated characters. The feature extraction step is essential for efficient data representation and extracting meaningful features for later processing. The classifier assigns the characters to one of classes. The architecture of our system is shown in Fig. 4.

4.1 Pre-Processing

Such as was presented in figure 4, the procedure of pre-processing which refines the scanned input image includes several steps: Binarization, for transforming gray-scale images in to black and white images, noises removal, and skew correction performed to align the

input paper document with the coordinate system of the scanner and segmentation into isolated characters.
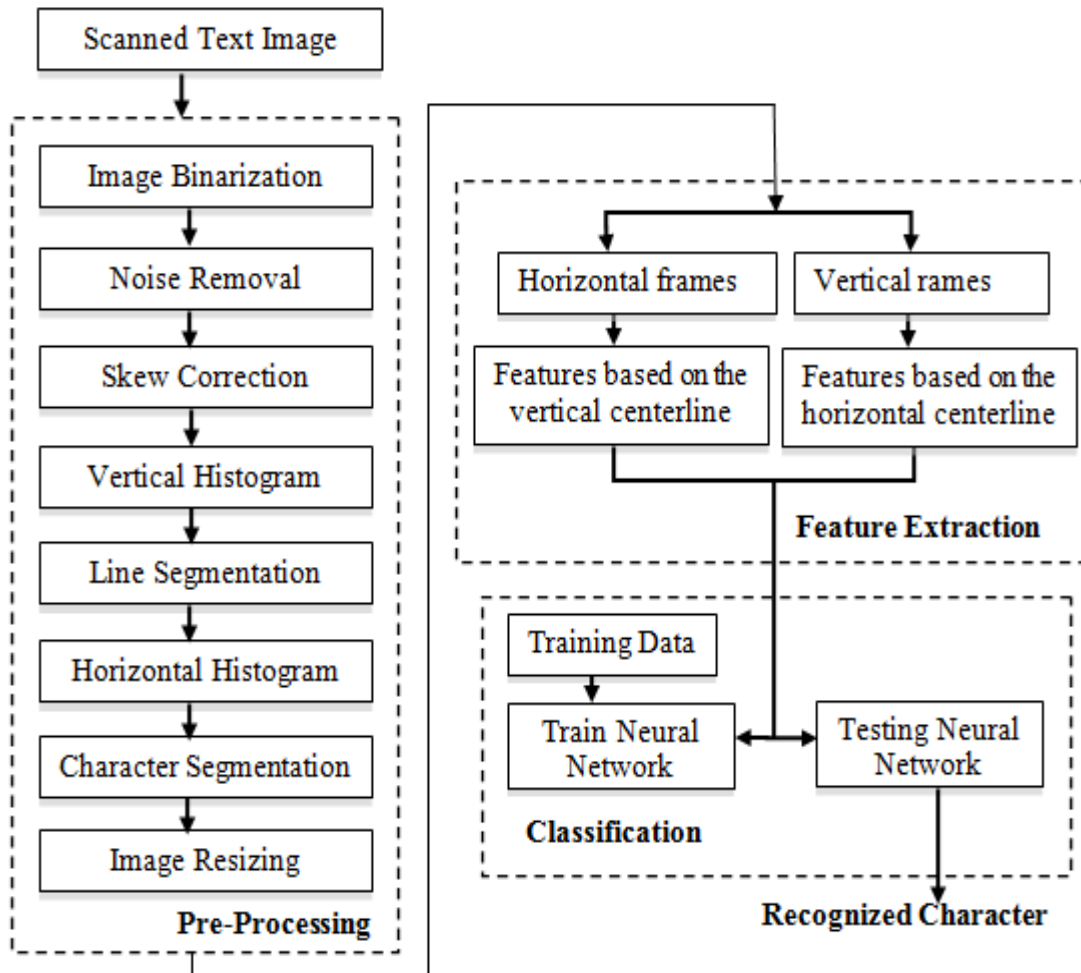
Figure 4. Architecture of our System

4.1.1. Binarization and Noise Removal: We used the Otsu method for binarization [29]. This method of thresholding is performed as a preprocessing step to remove the background noise from the picture prior to extraction of characters and recognition of text. This method performs a statistical analysis of histograms to define a function to be maximized to estimate the threshold. Fig. 5 (a) shows a sample input handwritten character image and Fig. 5 (b) shows the binarized image after the thresholding step using Otsu's method.
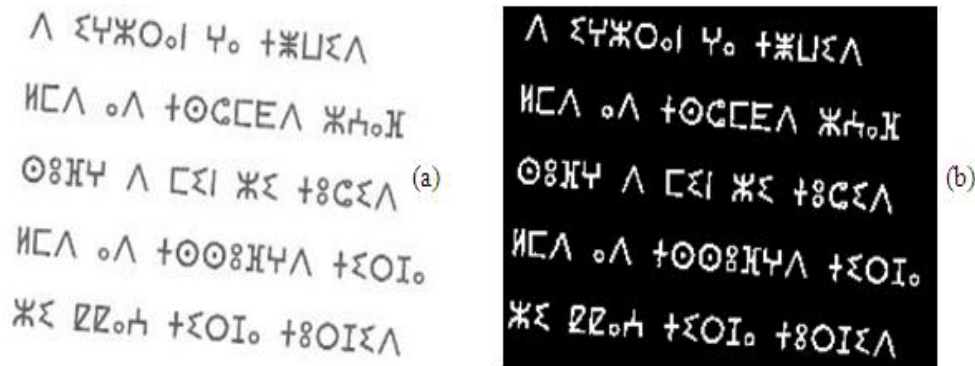
Figure 5. (a) Example of an input image of a text Amazigh, (b) Thresholded image with Otsu's method.

4.1.2. Skew detection and correction: Skew correction methods are used to align the paper document with the coordinate system of the scanner. Main approaches for skew detection include line correlation [30], projection profiles [31], Hough transform [32], etc. For this purpose two steps are applied. First, the skew angle is estimated. Second, the input image is rotated by the estimated skew angle. In this paper, we use the Hough transform to estimate a skew angle θs and to rotate the image by θs in the opposite direction.

4.1.3. Line Segmentation: Once the image of the text cleaned, the text is segmented into lines. This is used to divide text of document into individual lines for further preprocessing. For this, we used analysis techniques of horizontal projection histogram of the pixels in order to distinguish areas of high density (lines) of low-density areas (the spaces between the lines) (see Fig. 6). These techniques were often used to extract lines in printed texts, which do not show much variability in the spatial arrangement of related entities such as writing Amazigh.
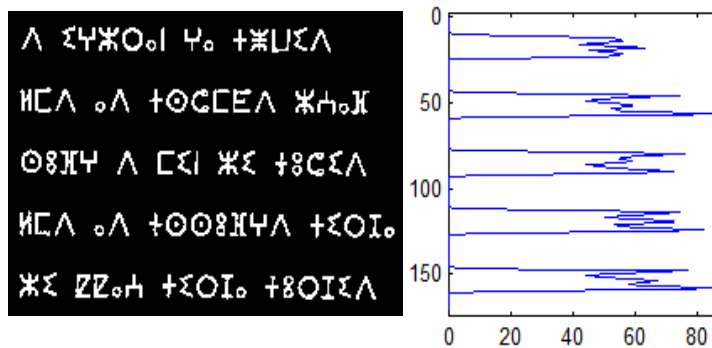
Figure 6. Image Text and Horizontal Projection Histograms Corresponding.

4.1.4. Character Segmentation: The Amazigh writing is not cursive which facilitates the segmenting of a text line of characters. We used the vertical projection histogram to segment each text line of characters. Fig. 7 shows a text line, the vertical histogram and the result of segmentation into characters.

(a)

(b)

(c)
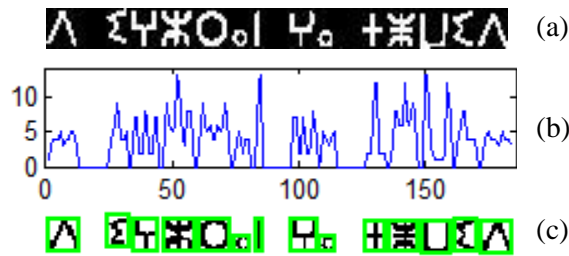
Figure 7. (a) A text Line, (b) the Vertical Histogram, (c) the Result of Segmentation.

4.2 Features Extraction

After pre-processing, a feature extraction method is applied to extract the most relevant characteristic of the character to recognize. The performance of a character recognition system largely depends on the quality and the relevance of the extracted features. In this paper, the extracted features consist of two groups. The features of the first group based on the horizontal centerline of the letter and, the features of the second group based on the vertical centerline of the letter.

4.2.1. Features based on the horizontal centerline: This step separates the image of the character into two zones: an upper zone that corresponds to the area above the baseline, which is the horizontal central line, and a lower zone which corresponds to the area below the baseline. Fig. 8 shows the line position of writing on a few Amazigh characters.

Figure 8. Line position of writing on a few Amazigh characters.

To create the first group of the feature vector, the letter is then scanned from left to right and from top to bottom with a sliding window. The image is divided into vertical frames (Fig. 9). The height and width of the frame are constant and are considered as parameters. The window height varies according to each image.
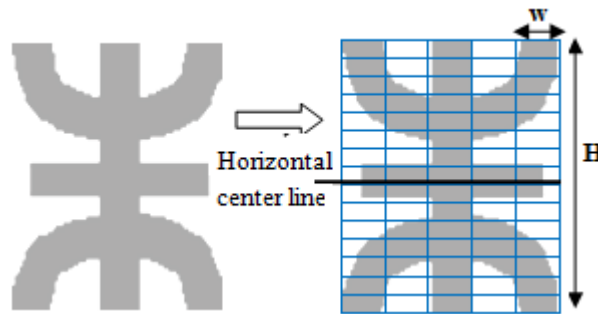
Figure 9. Dividing letters into vertical frames

Each frame is divided into cells where the height cell is fixed (in our experiments to 4 pixels) [19]. This yields to variable number cells in each frame according to the word image height. In each window, we generate a set of 9 features. These are representative of the densities of pixels.

Let H be the height of the frame in an image, h be the fixed height of each cell, and w the width of each frame. The number of cells in each frame n_c is: $n_c = H/h$.

Suppose n(i) is the number of black pixels in cell i, r(j) is the number of black pixels in the jth row of a frame, and b(i) is the density level of cell i. If the n(i) of cell i is equal to 0 then we assign 0 to it, otherwise, we assign 1. This procedure is presented in (1).

$$b(i) = 0 \ if \ n(i) = 0 \ else \ b(i) = 1 \tag{1}$$

Let LH the position of the horizontal centerline, we consider it the baseline of the letter. For each frame 9 features are extracted. The first feature, f1, is the density of foreground pixels (black).

$$f_1 = \frac{1}{H \times w} \sum_{i=1}^{n_c} n(i) \tag{2} \qquad f_2 = \sum_{i=2}^{n_c} |b(i) - b(i-1)| \tag{3}$$

The second feature, f2, is the number of transitions between two consecutive cells of different density levels of a frame.

The third feature, f3, is the derivative feature which represents the difference between the gravitational center of black pixels of frame t and its previous frame.

$$f_3 = g(t) - g(t-1) \tag{4} \qquad \text{Where g is the position of gravitational center.}$$

The vertical position of the gravity center in each frame is considered as the forth feature. This feature is normalized by the height of each frame.

$$g = \frac{\sum_{j=1}^{H} j.r(j)}{\sum_{j=1}^{H} r(j)} \tag{5} \qquad f_4 = \frac{g - LH}{H} \tag{6}$$

The fifth and sixth features that represent the density of foreground pixels over and under the horizontal center line for each frame.

$$f_5 = \frac{1}{H \times w} \sum_{j=LH+1}^{H} r(j) \tag{7} \qquad f_6 = \frac{1}{H \times w} \sum_{j=1}^{LH-1} r(j) \tag{8}$$

The seventh and eighth features are similar to the third one. But only those cells that are above the horizontal center line are considered in, and the cells that are below the horizontal center line are considered in, f8.

$$f_7 = \sum_{i=2}^{k} |b(i) - b(i-1)| \tag{9} \qquad f_8 = \sum_{i=k}^{n_c} |b(i) - b(i-1)| \tag{10}$$

The ninth feature, f9, is the density of foreground pixels in the horizontal centerline. This feature was added to distinguish between some resembling Amazigh characters.

4.2.2. Features based on the vertical centerline: For exploiting the information coming from the detection of the vertical centerline of the character, we generated a new group of the features. To create the second group of the feature vector, the letter image is then scanned from top to bottom and from left to right with a sliding window. The image is divided into horizontal frames (Fig. 10). The number of the frame is constant and is considered as one of the system parameters (5 frames in our experiments). Each frame is divided into cells where the cell height is fixed (in our experiments to 4 pixels). In each window we generate a set of 9 features. These 9 features (f10 to f18) are similar to (f1 to f9), respectively. We use the same formula used for vertical windows.
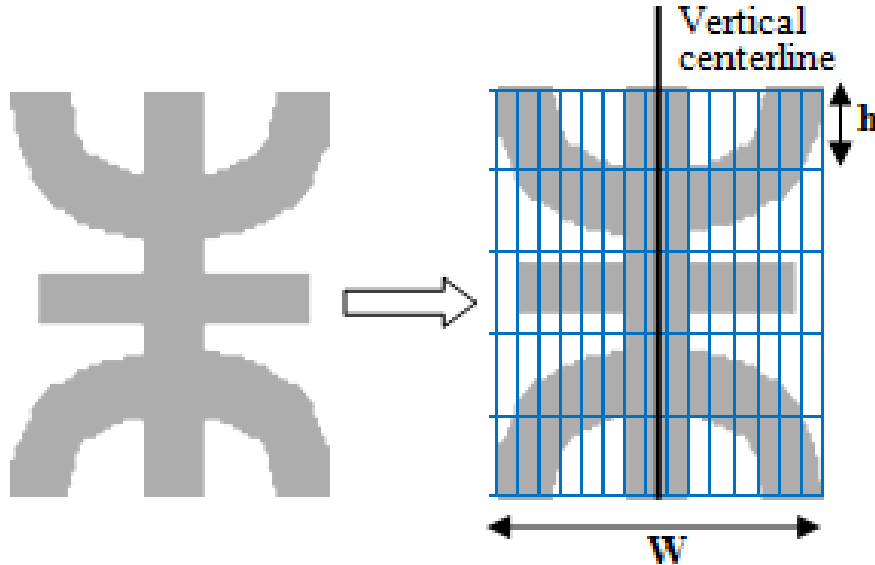
Figure 10. Dividing Letters into Horizontal Frames

5. Database

To evaluate the performance of the proposed method, the experiments were performed on both bases of the Amazigh patterns: a printed database of Amazigh characters and another one for handwritten characters created locally. The first is a database of printed Amazigh patterns of different fonts and sizes [11]. It contains a total of 12 fonts and sizes of 10 points to 28 points for each model.

Our database of isolated Amazigh handwritten characters was collected from 50 people. These people were selected to represent various age, gender, and educational background groups. The samples were collected by asking the participants to write on a form 13 examples for each Amazigh character. We collected 350 documents forms. After the digitizing these documents, we have developed an automatic system to processes and segment them into isolated characters. In total, we obtained 650 variations for each character led to an overall of 21450 isolated Amazigh handwritten characters (650×33). Some Amazigh handwriting samples are given in Table 1.

Table1. Some Amazigh Handwriting Samples.

Amazigh characters	Writer 1	Writer 2	Writer 3	Amazigh characters	Writer 1	Writer 2	Writer 3
ⵕ				ⵄ			
ⵝ				ⵀ			
ⵣ				ⵞ			
ⵟ				ⵔ			
ⵥ				ⵚ			
ⵉ				ⵗ			
ⵉ				ⵇ			
ⵣ				ⵇ			
ⵛ				ⵖ			
ⵍ				ⵛ			
ⵓ				ⵎ			
ⵜ				ⵎ			
ⵟ				ⵝ			
ⵓ				ⵗ			
ⵙ				ⵉ			
				ⵝ			

6. Training and Recognition

Before feature extraction, each letter is normalized to 60×50 pixels. Then, two different sets of features (45 features based on the vertical centerline and 45 features based on the horizontal centerline giving a total of 90 features) are used in the training and testing modules.

There are several available architectures and learning methods which may be selected depending on a given problem. In this paper, we choose a multi-layer perceptron architecture using the back propagation with momentum learning scheme. The multilayer perceptron architecture consists of an input layer of information processing nodes, a hidden layer with hidden nodes and an output node which consists of output nodes that usually equal the number of data classes. The Fig. 11 presents the Artificial Neural Networks (ANN) Architecture. The number of neuron outputs is 31. The numbers of hidden nodes vary across different applications and this number must be experimentally determined [7], [33], [34], [35], [36]. In our case, we adopted a number of hidden layer neurons equivalent to the

number of attributes and classes divided by two. The network learned on the entire training set using the back propagation method, and was then tested (with the validation set) for its performance with a limit of 1000 epochs. The network is trained with 0.3 learning rate and 0.2 momentums constant. This multilayer perceptron implementation is available in the Weka collection of machine learning algorithms [37].
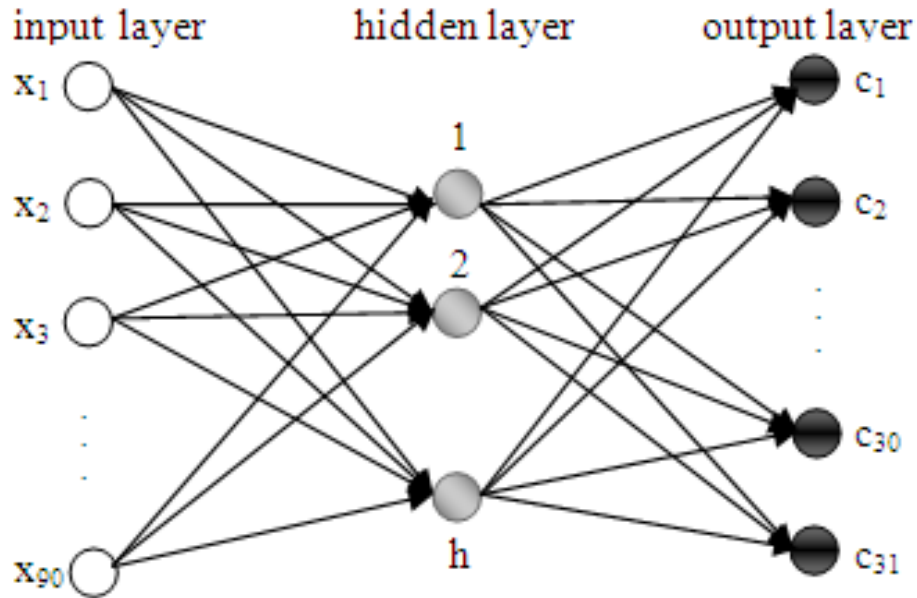
Figure 11. ANN Architecture (three layers: input, hidden and output)

7. Results and Analysis

We have tested the proposed system on two bases of the Amazigh characters: a printed database of Amazigh characters [11] and another one for handwritten characters presented previously. The tests were performed according to the integration of features dependent and independent on the vertical and horizontal centerline. Also, we have used a 10-fold cross validation scheme for recognition result evaluation. For 10 fold cross validation, 90% of the data was used for training and this performance was tested on the remaining 10%. We divided database in 10 major groups and measured recognition accuracy for each group separately. The recognition rates of all the 10 test subsets of the dataset are averaged to get the recognition result. Table 2 shows the experiments results using ten-fold cross-validation on the base of printed Amazigh characters and on the base of handwritten characters.

For the base of printed Amazigh characters, the recognition rate is 98.49% when integrating the features based on the position of the horizontal centerline and increases to 99.28% when adding features based on the position of the vertical centerline. This demonstrates that the features based on the position of central lines offer a significant improvement in the recognition performance.

Table 2. Recognition Results Using Ten-fold Cross-validation

Features integrated	Printed Amazigh characters		Handwritten Amazigh characters	
	Database size	*Recognition rate %*	*Database size*	*Recognition rate %*
Features independent on the centerline	19437 characters	88.6814 %	20150 characters	84.4963 %
Features dependent and independent on the horizontal centerline	19437 characters	98.4977 %	20150 characters	92.2333 %
Features dependent and independent on the horizontal and vertical centerline	19437 characters	99.2849 %	20150 characters	96.3275 %

Table 3. Confusion Matrix of the System on the Base of Printed Amazigh Characters

ya	yab	yach	yad	yadd	yae	yaf	yag	yagh	yah	yahh	yaj	yak	yal	yam	yan	yaq	yar	yarr	yas	yass	yat	yatt	yaw	yax	yay	yaz	yazz	yey	yi	yu	
602	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	24	0	0	0	0	0	0	0	0	0	0	0	0	0	ya
0	616	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	3	0	5	0	0	0	0	0	0	0	0	1	0	0	yab
0	0	627	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	yach
0	0	0	627	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	yad
0	0	0	0	625	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	yadd
0	0	0	0	0	626	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	yae
0	0	0	0	0	0	624	0	0	0	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	yaf
0	0	0	0	0	0	0	624	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	yag
0	0	0	0	0	0	0	0	626	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	yagh
0	0	0	0	0	0	0	0	0	626	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	yah
0	0	0	0	0	0	0	0	0	0	627	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	yahh
0	0	0	0	0	0	0	0	0	0	0	627	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	yaj
0	0	0	0	0	0	0	0	0	0	0	0	627	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	yak
0	0	0	0	0	0	0	0	0	0	0	0	0	627	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	yal
0	0	0	0	0	0	0	0	0	0	0	0	0	0	627	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	yam
0	0	1	0	0	0	0	0	0	0	0	0	0	0	37	587	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	yan
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	626	0	0	0	0	0	0	0	0	0	0	0	0	0	0	yaq
17	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	603	0	0	1	0	1	0	0	0	0	0	3	0	0	yar
0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	623	0	0	0	0	0	0	0	0	0	2	0	0	yarr
0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	625	0	0	0	0	0	0	0	0	0	0	0	yas
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	625	0	0	0	0	0	0	0	0	0	0	yass
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	627	0	0	0	0	0	0	0	0	0	yat
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	627	0	0	0	0	0	0	0	0	yatt
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	627	0	0	0	0	0	0	0	yaw
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	627	0	0	0	0	0	0	yax
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	627	0	0	0	0	0	yay
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	627	0	0	0	0	yaz
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	614	0	0	0	yazz
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	621	0	5	yey
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	627	0	yi
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	627	yu

Table 3 shows the confusion matrix of the system on the base of printed Amazigh characters. Only four letters from 31 had few errors. The first error is due to the resemblance between some letters in different fonts. For example, 37 images of letter yan (|) are

recognized as letter yaj (I). Moreover, in the database used, the form of the letter yan (|) on the 'tassafut' font resembles entirely to the letter yaj (I), such as shown in the Fig. 12. The second error consists of twenty four replacements of the letter ya (∘) by the letter yar (O) out of 627 instances. This problem is due to the database. Furthermore, in this database used, the images of patterns are stored in a way which does not allow the possibility of renormalizing their sizes to a fixed average size. Indeed, this can be particularly troublesome because of the resemblance of the characters ya (∘) and yar (O), which differ only in size: the character ya (∘) is a small circle, while the character yar (O) is a big circle. In some cases, there will be a real confusion between the images of these two characters. This problem has an influence on the results. It is something that has been resolved in our database of handwritten Amazigh characters. The third error consists of thirteen replacements of the letter yazz (Ⅹ) by the letter yaz (Ⅹ). This problem caused by the similarity between these two letters. The only difference between these two letters is the dash in the centre of the letter.

Table 4. Confusion Matrix on the Base of Handwritten Amazigh Characters

ya	yab	yach	yad	yadd	yae	yaf	yag	yagh	yah	yahh	yaj	yak	yal	yam	yan	yaq	yar	yarr	yas	yass	yat	yatt	yaw	yax	yay	yaz	yazz	yey	yi	yu		
649	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	ya
0	626	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	3	7	3	0	6	0	1	1	0	0	1	0	1	yab
0	2	625	0	0	0	0	1	0	0	0	1	1	0	2	0	0	4	7	1	2	0	1	0	0	0	1	0	0	2	yach
0	0	0	643	0	1	0	0	0	0	2	0	0	2	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	yad
0	0	1	0	599	0	0	0	0	0	0	1	3	0	9	0	0	0	0	0	0	29	0	2	0	0	0	0	6	0	yadd
1	0	0	4	0	638	0	0	0	0	3	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	yae
0	0	0	0	0	0	635	1	1	0	0	1	0	1	0	0	3	0	0	3	0	0	0	1	0	0	1	2	1	0	yaf
0	0	0	1	0	0	1	641	0	0	0	0	3	0	0	0	1	0	2	0	0	0	0	0	0	0	0	0	0	1	yag
0	0	0	0	1	0	0	0	639	0	0	0	0	3	0	0	0	0	0	4	1	0	1	0	0	1	0	0	0	0	yagh
0	0	0	0	0	0	0	1	0	627	0	2	1	3	1	0	0	1	6	3	1	1	0	0	3	0	0	0	0	0	yah
0	0	0	2	0	2	0	0	0	0	644	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	yahh
0	0	0	0	0	0	0	4	0	1	0	628	0	0	6	3	1	0	0	1	0	2	0	0	0	0	3	0	0	1	yaj
0	0	0	1	1	0	0	6	0	3	6	0	623	0	0	0	0	0	1	2	0	1	0	6	0	0	0	0	0	0	yak
0	0	0	0	0	0	0	0	1	2	1	0	0	640	0	0	1	0	0	1	3	0	0	0	0	1	0	0	0	0	yal
0	0	4	0	3	0	0	0	0	0	0	1	0	0	640	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	yam
0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	644	0	0	0	0	2	1	1	0	0	0	0	0	0	0	yan
0	0	0	0	0	0	1	2	2	0	5	0	0	1	0	0	622	0	0	0	2	0	0	0	0	13	2	0	0	0	yaq
0	2	5	0	2	0	1	0	0	1	0	0	0	0	2	1	0	620	2	2	9	0	0	2	0	0	0	0	0	1	yar
0	1	4	0	0	2	2	0	1	4	0	0	3	0	0	0	0	5	617	6	1	1	0	0	2	0	1	0	0	0	yarr
0	1	2	0	2	0	0	1	2	0	0	1	0	0	0	0	6	7	611	6	0	4	0	4	0	0	2	0	0	1	yas
0	2	1	0	0	0	2	0	0	3	2	1	0	1	0	0	2	8	0	9	614	0	1	1	3	0	0	0	0	0	yass
0	0	0	0	3	2	1	0	0	0	1	1	0	0	0	0	0	0	0	0	634	1	0	0	0	0	3	4	0	0	yat
1	9	0	0	23	0	1	0	0	0	0	2	3	0	1	2	0	0	1	1	1	0	589	1	7	0	0	1	0	4	3	yatt
0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	648	0	0	0	0	0	0	0	yaw
0	0	0	1	1	0	0	0	0	3	0	2	4	0	1	1	0	1	0	2	1	0	0	633	0	0	0	0	0	0	yax
0	0	0	1	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	2	0	0	0	644	0	0	0	1	0	yay
0	0	0	0	0	3	1	1	0	0	1	4	0	3	0	0	18	0	0	3	3	2	0	0	0	577	33	1	0	0	yaz
0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	6	0	0	0	2	2	0	0	1	1	40	593	1	0	1	yazz
0	0	0	0	2	0	0	0	0	0	0	1	0	0	0	3	0	0	1	0	0	2	0	0	0	3	1	2	621	4	10	yey
0	0	1	0	3	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	2	1	0	5	635	1	0	yi
0	0	1	0	0	0	0	3	0	1	0	2	0	0	0	1	1	0	0	0	0	2	0	1	1	0	2	9	0	626	yu

Figure 12. Some Examples of Character yan (I) in the Base, which is 'tassafut' Font.

In Table 4, we present the confusion matrix of the system on the base of handwritten Amazigh characters. The study of this confusion matrix showed that most errors are mainly due to the resemblance between some Amazigh characters. For examples, confusions between the two letters yaz (ⵣ) and yazz (ⵥ), between the two letters yatt (ⴻ) and yadd (ⴹ), between the two letters yaz (ⵣ) and yaq (ⵇ) and between the two letters yey (ⵉ) and yu (ⵓ).

8. Conclusion and Perspectives

In this paper, we have presented a system for automatic recognition of Amazigh handwriting based on the position of the vertical and horizontal centerline of each character. Several features have been studied and compared. The importance of using the position of the centerline in the image of the character has been proved. The extracted features are based on the density of pixels derived from a sliding window. The developed system was tested on two databases, a database of printed Amazigh patterns and another base of handwriting Amazigh character developed locally. These results show a significant improvement in recognition rate when integrating the features dependent on the vertical and horizontal centerline. In future work, we will add other features that improve the results for some characters, such as information on the possible inclination of writing using the windows inclined. In addition, we will experiment our approach on a database of Amazigh handwritten documents.

References

[1] R. M. Bozinovic and S. N. Srihari, "Off-line cursive script word recognition", IEEE Trans. on Pattern Anal. Mach. Intell., vol. 11, no. 1, 1989, pp. 68-83, Jan.

[2] I. Bazzi, R. Schwatz and J. Makhoul: "An Omni font Open-Vocabulary OCR System for English and Arabic. IEEE Transactions on pattern Analysis and Machine Intelligence". vol. 21, no 6. 1999, pp. 495-504..

[3] Z. Zhang, L. Jin, K. Ding, X. Gao: Character-SIFT: "A Novel Feature for Offline Handwritten Chinese Character Recognition", 10th International Conference on Document Analysis and Recognition, 2009, pp. 763-767.

[4] Pooja Agrawal, M. Hanmandlu, Brejesh Lall, "Coarse Classification of Handwritten Hindi Characters", International Journal of Advanced Science and Technology, Vol. 10, September 2009, pp. 43-54.

[5] A. Adnan: "Off-line Arabic character recognition - the state of the art", Pattern Recognition, vol. 31, no. 5, 1998, pp.517-530.

[6] R. Ebrahimpour, R. D. Vahid, B. M. Nezhad, "Decision Templates with Gradient based Features for Farsi Handwritten Word Recognition", International Journal of Hybrid Information Technology, vol. 4, No. 1, January, 2011, pp.1-12.

[7] K.W. Wong, C.S. Leung & S.J. Chang, "Handwritten digit recognition using multi-layer feedforward neural networks with periodic and monotonic activation functions", ICPR, vol. 3, 2002, pp. 106–109.

[8] A. Oulamara, J Duvernoy, "An application of the Hough transform to automatic recognition of Berber characters", Signal Processing, vol. 14, 1988, pp.79-90.

[9] A. Djematen, B. Taconet, A. Zahour, "A Geometrical Method for Printing and Handwritten Berber Character Recognition", ICDAR'97, 1997, p. 564.

[10]   A. Djematen, B. Taconet, A. Zahour: "Une méthode statistique pour la reconnaissance de caractères berbères manuscrits'', CIFED'98, 1998,p 170-178.

[11] Y. Ait ouguengay, M. Taalabi, ''Elaboration d'un réseau de neurones artificiels pour la reconnaissance optique de la graphie amazighe: Phase d'apprentissage'', Systèmes intelligents-Théories et applications, 2009.

[12] R. El Yachi and M.Fakir, "Recognition of Tifinaghe Characters using Neural Network", International Conference on Multimedia Computing and Systems, Actes DE ICMCS'09, Ouarzazate, Maroc, 2009.

[13] Y. Es Saady, A. Rachidi, M. El Yassa, D. Mammass, "Reconnaissance Automatique de l'Ecriture Amazighe à base de Ligne Centrale de l'Écriture", 4ème Atelier international sur l'amazighe et les TIC, 2011, IRCAM, Maroc.

[14] M. Amrouch, Y. Es Saady, A. Rachidi, M. Elyassa, D. Mammass, "Printed Amazigh Character Recognition by a Hybrid Approach Based on Hidden Markov Models and the Hough Transform", International Conference on Multimedia Computing and Systems, Actes de ICMCS'09, Ouarzazate, Maroc, 2009.

[15] M. Amrouch, A. Rachidi, M. Elyassa, D. Mammass, "Handwritten Amazigh Character Recognition Based On Hidden Markov Models, ICGST-GVIP Journal, Vol.10, Issue 5, pp.11-18, 2010.

[16] Y. Es saady, A. Rachidi, M. Elyassa, D. Mammass, ''Une méthode syntaxique pour la reconnaissance de caractères amazighes imprimés'', CARI'08, MAROC, 2008, octobre 27-31.

[17] Y. Es Saady, A. Rachidi, M. El Yassa, D. Mammass , " Printed Amazigh Character Recognition by a Syntactic Approach using Finite Automata", ICGST-GVIP Journal, Vol.10, Issue 2, 2010, pp.1-8.

[18] R. El Yachi, K. Moro, M. Fakir, B. Bouikhalene, "On the Recognition of Tifinaghe Scripts", Journal of Theoretical and Applied Information Technology, Vol.20, No.2, 2010, pp.61-66.

[19] R. El-Hajj, L. Likforman-Sulem, C. Mokbel, "Arabic handwriting recognition using baseline dependent features and Hidden Markov Modeling", ICDAR 05, Seoul, Corée du Sud, 2005.

[20] M. Razzak, M. Sher, S. Hussain, "Locally baseline detection for online Arabic script based languages character recognition", International Journal of the Physical Sciences Vol. 5(7), 2010, pp. 955-959.

[21] R. Aida-zade and Z. Hasanov, "Word base line detection in handwritten text recognition systems", International Journal of Electrical and Computer Engineering 4:5 2009, pp. 310-314.

[22] A. AL-Shatnawi, O. Khairuddin, "Methods of Arabic Language Baseline Detection – The State of Art", IJCSNS International Journal, Vol.8 No.10, 2008, pp. 137-143.

[23] M. Outahajala, L. Zenkouar, P. Rosso, A. Martí, "Tagging Amazighe with AncoraPipe", Proc. Workshop on LR & HLT for Semitic Languages, 7th International Conference on Language Resources and Evaluation, LREC-2010, Malta, May 17-23, 2010, pp. 52-56.

[24] M. Ameur, A. Bouhjar, F. Boukhris, A. Boukouss, A. Boumalk, M. Elmedlaoui, E. Iazzi, "Graphie et orthographe de l'Amazighe", Publications de Institut Royal de la Culture Amazighe, Rabat Maroc, 2006.

[25] A. Boumalk, E. El Moujahid, H. Souifi, F. Boukhris, "La Nouvelle Grammaire de l'Amazighe", Publications de Institut Royal de la Culture Amazighe, Centre de l'Aménagement Linguistique, Rabat Maroc, 2008.

[26] http://fr.wikipedia.org/wiki/Tifinagh

[27] L. Zenkouar, "L'écriture Amazighe Tifinaghe et Unicode", in Etudes et documents berbères. Paris (France). n° 22, 2004, pp. 175-192.

[28] J. Shanbehzadeh, H. Pezashki, A. Sarrafzadeh, "Features Extraction from Farsi Handwritten Letters", Proceedings of Image and Vision Computing, Hamilton, New Zealand, December 2007, pp. 35–40.

[29] N. Otsu, "A threshold selection method from gray-level histograms", IEEE Trans. Sys,Man., Cyber, vol. 9, 1979, pp. 62–66.

[30]   H.Yan, "Skew correction of document images using interline cross-correlation", CVGIP: Graphical Models Image Process 55, 1993, 538-543.

[31] T. Pavlidis and J. Zhou, "Page segmentation and classification", Comput. Vision Graphics Image Process. 54, 1992, 484-496.

[32] D. S. Le, G. R.Thoma and H. Wechsler, "Automatic page orientation and skew angle detection for binary document images", Pattern Recognition 27, 1994, 1325-1344.

[33] S. Singh, and A. Amin, "Neural Network Recognition of Hand Printed Characters", NeuralComputing and Applications, vol. 8, no. 1, 1999, pp. 67-76.

[34] L. Fu, "Neural Networks in Computer Intelligence", McGraw-Hill.Singapore, 1994, 331-348.

[35] G.P. Zhang. "Neural Networks for Classification: A Survey". IEEE Trans on Systems, Man and Cybernetics - Part C, vol. 30, no. 4, 2000, pp. 641–662.

[36] Jayanta Kumar Basu, Debnath Bhattacharyya, Tai-hoon Kim, "Use of Artificial Neural Network in Pattern Recognition", International JInternational JInternational Journal of Software Engineering and Its Application, Vol. 4, No.2, April 2010, pp.23-34.

[37] I. H. Witten and E. Frank, "Data Mining: Practical Machine Learning Tools and Techniques", San Francisco: Morgan Kaufmann Publishers, second edition, 2005.

Authors

Youssef ES SAADY received his Master's degree in 2007 from Faculty of Sciences University Ibn Zohr Agadir Morocco. He is currently working on his Ph. D in the IRF-SIC Laboratory at Ibn Zohr University. His current research interests include pattern recognition, image analysis, document processing and automatic processing of natural languages.

Ali RACHIDI is currently research professor at the National School of Commerce and Management, University Ibn Zohr, Agadir Morocco. He received the doctorate at Mohamed V University, Rabat in 1999 and the clearance to conduct research (HDR) at the University Ibn Zohr, Agadir in 2007. His research topics are: artificial intelligence, automatic processing of natural language and pattern recognition.

Mostafa EL YASSA is a Professor of informatics at the Faculty of Sciences, University Ibn Zohr, Agadir, Morocco. He received a Doctorat in Informatics and Statistics in 1984 from Strasbourg University (France) and a Doctorat d'Etat es-Sciences degree in Pattern recognition by pretopologique approach from Faculty of sciences, University Ibn Zohr, in 2007. His research areas include: Pattern recognition, pretopologique approach, verification of Handwritten Signatures, image analysis and document processing.

Driss MAMMASS is professor of Higher Education at the Faculty of Sciences, University Ibn Zohr, Agadir Morocco. He received a Doctorat in Mathematics in 1988 from Paul Sabatier University (Toulouse - France) and a doctorat d'Etat-es-Sciences degrees in Mathematics and Image Processing from Faculty of Sciences, University Ibn Zohr Agadir Morocco, in 1999. He supervises several Ph.D theses in the various research themes of mathematics and computer science such as remote sensing and GIS, digital image processing and pattern recognition, the geographic databases, knowledge management, semantic web, etc. He is currently Director of High School of Technology Agadir and the head of IRF-SIC Laboratory (Image Reconnaissances des Formes, Systèmes Intelligents et Communicants) and an unit of formation and research in doctorat on mathematics and informatics.