Sequential Primitive Classifier Combination for Amazigh Alphabets OCR

Mehdi Boutaounte
National School of Business and Management,
Dakhla, Morocco

Youssef Ouadid
Sultan Moulay Slimane University,
Beni Mellal, Morocco

ABSTRACT

The main difficulty in developing a successful Optical Character Recognition (OCR) system lies in the confusion between the characters. In the case of Amazigh writing (Tifinagh alphabets), some characters have similarities based on rotation or scale. Most of the researchers attempted to solve this problem by combining multiple descriptors and / or classifiers which increased the recognition rate, but at the expense of processing time that becomes more prohibitive. Thus, reducing the confusion of characters and their recognition times is the major challenge of OCR systems. In this paper, we present an off-line OCR system for Tifinagh characters. Using a proposed key point extraction algorithm, character skeleton is divided into several segments. The length and orientation of every segment are stored into a feature vector, then the relationship between these segments are represented by a graph in the form of an incidence matrix. Classification is done by searching for similarity between test images and their counterpart in the reference database by comparing their incidence matrix. When search provides multiple classes, feature victors are compared and input image is assigned to the closest class. Based on experiments done on 3300 images, an accuracy of 99% is achieved.

General Terms

Pattern Recognition, , Algorithms , OCR

Keywords

Feature points Extraction, Graph Theory, Incidence Matrix

1. INTRODUCTION

A large number of populations all over the world, especially in North Africa, speak Amazigh language. Since its normalization in 2001, efforts of research centers have manifested themselves in numerous and in-depth studies on the promotion of this alphabet and the widening of its scope. This has led to the appearance of Amazigh documents written in Amazigh. From then on, the automatic processing and recognition of these documents became a very active field of research.

Even though the OCR research is well advanced for Arabic, Latin and Chinese scripts, research on Amazigh scripts OCR s is still in the infancy stage. The goal is to elaborate a fast and accurate OCR to convert the text of such documents into a machine-readable representation easily reproducible by computers.

The main difficulty in developing an efficient OCR system is the confusion between characters. Indeed, the automatic recognition of characters consists in describing the content of the images automatically by features through an analysis of their visual content. This analysis is confronted by: the noise produced during acquisition and the problem of shape variability that can come from scale and rotation changes. This accentuates the problem of intra-class variation and inter-class resemblance. This visual variation creates complicated relationships between character classes and their visual content, which leads to confusion between the characters and makes the recognition problem very difficult to solve. Most of the researchers attempted to solve this problem by combining multiple descriptors and / or classifiers which increased the recognition rate, but at the expense of processing time that becomes more prohibitive.

Oulamara et al. (Oulamara and Duvernoy, 1988) used the Hough transform to extract straight segments with their attributes (length and orientation). By analyzing the characters in the parametric space, a reading matrix is constructed containing the feature vectors of the reference images in the learning phase. Using a local database, the authors achieved interesting results. However, according to Djematene et al. (Djematen et al., 1998), this technique is not appropriate because the segmentation by the Hough transform does not produce a correct segmentation.

Ait Ouguengay [1] have proposed a recognition system based on multilayer artificial neural networks (RNA) with a single hidden layer to classify characteristic vectors composed of geometric properties (horizontal and vertical projections, centers of gravity in x and y , perimeter, area, compactness and central moments of order 2). Interesting results are obtained by testing the system on a local database.

Amrouch [2]proposed an approach based on the extraction of directional information from the Hough transformation of each character in the form of a vector of observations. This information feeds a hidden Markov model (HMM). The results obtained are promising. However, the discrimination of these models is not very good because in the learning phase each character, according to the authors, is represented by a single reference image. To remedy this issue, Amrouch [3] have replaced the Hough transform with a new technique to express a set of structural features from the contour of the character based on points that have maximum deviation. In the learning and classification phase, they combined dynamic programming with continuous HMMs. This approach has the advantage of being independent of the number of recognition classes (in terms of memory and speed) since the model is built for all classes. The results, which are quite encouraging, have shown that continuous HMMs are more robust. However, the disadvantages of this approach are the detection of the points that have the maximum deviation for the features extraction phase that seem restrictive for some fonts of the Amazigh writing.

In an attempt to solve orientation and size change problems, El Ayachi [4] compared two robust descriptors. These are the invariant moments and the transform of Walsh. The authors presented two systems containing the same preprocessing and classification techniques. Using dynamic programming in the classification phase, the authors concluded that the invariant moments are greater than the Walsh transform in terms of execution time and discrimination. In order to improve the recognition rate, El Ayachi [5] have replaced the dynamic programming method with a single hidden layer neural network. The system gave a better recognition rate compared to dynamic programming and neural networks with 2 or 3 hidden layers. Recently, in (El Ayachi et al., 2014) the authors used their systems to display the braille code adapted to Tifinagh characters. The braille system is a system adopted to help blind and partially sighted people integrate into different areas of life.

Es Saady [6] proposed a syntactical approach for the recognition of printed Tifinagh characters. This is done by representing the character images using Freeman coding. The classification is performed using the finite automata. An automaton is built from the specific automata of each image of the printed Tifinagh character. According to the authors, the tests carried out show the robustness of the system. However, the problem with this approach is that it does not deal with circular characters. To remedy this, the authors used the horizontal and vertical symmetry of the spelling of the Tifinagh alphabet. Indeed, the authors presented in [7] a system based on the position of the central lines of each character. The features are extracted based on the density of the pixels contained in a sliding window in the image of the character. According to the authors, this approach has proved its power of discrimination by testing it on a local database.

In order to complement the limits of the previous systems facing the problems of rotation and size change, Bencharef [8] proposed an approach based on a geometric description using geodetic descriptors. These descriptors served as input to the hybrid classification process that combines neural networks and decision trees.

The success of Bencharef's approach has motivated Oujaoura et al. to go in the same direction. In a first approach [9], the authors performed a comparison between the Walsh transform, GIST and texture using Bayesian networks as classifiers. The tests of these descriptors on a local database showed the superiority of the GIST method in terms of recognition rate and computation time. In a second approach [10], the authors have proposed a system that combines Zernike moments, Legendre moments, Hu moments, Walsh and GIST transforms in the feature extraction phase; As well as the neuron networks, SVM and NN in the classification phase. The results obtained are excellent in terms of recognition rate. However, the system is quite slow in terms of computation time.

Authors developed an OCR system based on exact graph matching described by incidence matrices [11]. Graphs are constructed using the key points, extracted from the skeletons of the different characters, by the neighborhood algorithm. Then, in order to improve the recognition rate of this system, in [12],[13] they adapted the thinning algorithm to their needs then they used Harris method to extract the key points. At the classification level, inexact graph matching is applied. It is based on a spectral approach that represent and distinguish the structural properties of graphs using the eigenvectors and eigenvalues of their adjacency matrices. The evaluation of the system thus carried out, using a database of printed characters [1], highlighted the good performances of this system in terms of recognition rate and execution time.

Properties of Amazigh scripts

Tifinagh alphabet is composed of 33 graphemes corresponding to the 33 phonemes of the standard Amazigh. The following table, presents these 33 characters as well as their Latin correspondents.

Table 1. The official directory of the alphabet Tifinagh-IRCAM with their correspondents in Latin characters

1	ya	ⴰ	12	yaḥ	ⵃ	23	yaṛ	ⵕ
2	yab	ⴱ	13	yaɛ	ⵄ	24	yaɣ	ⵖ
3	yag	ⴳ	14	yax	ⵅ	25	yas	ⵙ
4	yag^{w}	ⴳⵯ	15	yaq	ⵇ	26	yaṣ	ⵚ
5	yad	ⴷ	16	yi	ⵉ	27	yac	ⵛ
6	yaḍ	ⴹ	17	yaj	ⵊ	28	yat	ⵜ
7	yey	ⴻ	18	yal	ⵍ	29	yaṭ	ⵟ
8	yaf	ⴼ	19	yam	ⵎ	30	yaw	ⵡ
9	yak	ⴽ	20	yan	ⵏ	31	yay	ⵢ
10	yak^{w}	ⴽⵯ	21	yu	ⵓ	32	yaz	ⵣ
11	yah	ⵀ	22	yar	ⵔ		yaẓ	ⵥ

Amazigh script is written from left to right. Unlike Arabic and Latin language, it does not have the notion of uppercase and lowercase, nor that of pseudo-word. However, it includes the same punctuation marks as the Latin one. It is a non-cursive writing which facilitates the process of segmentation. This justifies the fact that most of the research done on this writing is focused on character recognition since existing works on the segmentation of Latin script documents are viable for that of the Amazigh.

As for the graphic models of characters, their shapes are composed of dots, small circles and straight segments. Indeed, we can divide these characters into three classes:

Linear letters: □, □, □, □, □, □□, □, □, □, □, □, □, □, □, □, □, □, □, □, □, □□;

Circular letters: □, □, □, □, □, □, □, □, □;

Other: □, □, □.

Ignoring the criteria of the orientation and the size in the graphical models of the characters, some characters are similar such as: (□ and □), (□ and □), (□ and □). It is important to note that most characters are composed of a single connected component except for (□□, □□, □, □, □) alphabets.

The paper is structured as follows: section 1 gives a brief introduction about the offline character recognition domain. The architecture of the proposed system is introduced in section 2. The experiments and results obtained are presented in section 3. Finally, section 4 concludes the paper.

Literature review that has been done author used in the chapter "Introduction" to explain the difference of the manuscript with other papers, that it is innovative, it are used in the chapter "Research Method" to describe the step of research and used in the chapter "Results and Discussion" to support the analysis of the results [4]. If the manuscript was written really have high originality, which proposed a new method or algorithm, the additional chapter after the "Introduction" chapter and before the "Research Method" chapter can be added to explain briefly the theory and/or the proposed method/algorithm [5].

2. RESEARCH METHOD

The proposed system consists of three main phases:

preprocessing, feature extraction and classification. The scanned image is first preprocessed to produce a clean version by removing as much unwanted information as possible. In the feature extraction, preprocessed image is divided into several segments using a proposed key point extraction algorithm. Then structural properties are represented by a feature vector containing segments properties and a graph representation using an incidence matrix. The incidence matrix of test images are compared with their counterpart in reference images. In case if the exact comparison output produces multiple classes, feature vector is then classified using normal distance criteria. The architecture of the proposed system is shown in figure 1.

2.1 Pre-processing

In order to reduce as much distortion as possible, the Otsu's global thresholding (Xu et al., 2011) [14] is applied to the scanned image. Since the operations applied in the proposed system are structural and pixel type ones to the image are pixel type ones, the gap between image borders and the smallest rectangle containing the character is removed first.

Then, the character skeleton is extracted using Zhang-Suen thinning algorithm (Zhang and Suen, 1984) [15]. To ensure that no relevant information is removed after thinning, Zhang-Suen algorithm is applied to the 8-connected components of the image. This will allow us to keep all components composing the character such as the filled hole in yas character (□) as shown in figure 2. The character skeleton is then smoothed by removing pixels that have differences between the numbers of foreground pixels and the number of transitions from a background pixel with a foreground pixel. This operation is applied in the 8-neighborhood of every foreground pixel in the image. Figure 3 show results of skeleton smoothing further details can be found in [13].

2.2 Feature extraction

Feature extraction is the key of the success rate of the underlying OCR system. Here, character skeleton is divided into several segments using key points as delimiters. Based on these segment and points sets, two feature sets are extracted. The first set is incidence matrix. It describes the relationship between extracted segments and key points. The second set is a vector containing the length and orientation of every segment.

Key points extraction

Key points (so-called interest/corner points), are point-like features where fast change happens in direction in the structure of the character skeleton.

There are two types of key points:

Primary key points that consists of end and intersection points. The end points are foreground pixels where the number of transitions (from background neighbor to foreground neighbor) in the 8-neighborhood is equal to 1. Intersection points are foreground pixels where the number of transitions is more than 2.

Secondary key points are points where the change of direction happens in the curved parts of character skeleton.

The proposed algorithm for the extraction of key points is well described in [16]. It is proceeded as follows:

- Based on the skeleton of the character, extract primary key points;
- Divide the skeleton into segments using primary key points as delimiters;
- Check the level of curvature of every segment using the threshold k where (distance refers to Euclidian distance) :

$$\mathrm{k} = \frac{\text{segment length}}{\text{distance between segment ends}} \qquad (1)$$

$$\mathrm{k} = \frac{\text{segment length}}{\text{distance between segment ends}}$$

- If the threshold is bigger than an emperically chosen value, then secondary key points is required;
- If a secondary key point is required, then choose the pixel that have the biggest orthogonal distance from the straight line connecting segment ends as a key point;
- Update key points list and repeat steps 1 to 5 till no secondary key point is required;

2.3 Incidence matrix

Incidence matrix is a way to define the graphical representation of the character. A graph is a formal mathematical representation of a set of objects and their relationships. Each object is called a vertex. Relationships between objects are called edges. More formally, we define a graph G as an ordered pair of $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ where V is a set of vertices, E is a set of edges and each edge is a pair of vertices. Incidence matrix is fast and compact representation of a graph. It is a $\mathrm{m} \times \mathrm{n}$ matrix M where m is the number of nodes and n is the number of edges. The entry in row i and column j is non-zero, if and only if, the vertex i is incident at the edge j, which means:

$$\begin{cases} \mathrm{M(i,j)} = 1 \text{ si node i is incident to the edge j} \\ \mathrm{M(i,j)} = 0 \text{ else} \end{cases} \qquad 2)$$

In this work, the nodes represent key points and the edges represent segments linking these points in pairs.

2.4 Feature vector

As it is said earlier, the feature vector represents the orientation and length of every segment. Since the graphical representation deals with most of orientation problems, length is given more weight than the orientation as follow:

$$\text{segment feature} = 2 \times \text{Length} + \text{Orientation}$$

The size of the feature vector depends on the number of segments of the character.

2.5 Classification

In this phase, the resulting graphical representation of input images are compared with the reference images. In order to benefit from the representational potency of graphs and the reliability of statistical classification techniques in a timely manner, we propose a flexible hybrid classification based on matrix matching and Bayesian naïve classification. This hybridization is performed sequentially as indicated in Figure 1. Alternatively, if the search results provide images with multiple classes, features vectors are compared using Euclidian distance and input image is affected to the closest class. The reason behind using such a simple way to classify is the fact that the number of observations is very small to build and compared to other classifier such as Naïve Bayes are almost the same. The algorithm is proceeded as follows:

Input 1: A cell array (2475x3) that contains required information (Incidence matrix, feature vector and class label (1 to 33)) for trainning dataset;

Input 2: A cell array (825x3) that contains required information (Incidence matrix, feature vector and Actual class label) for test dataset;

Output: 2D array (825x2) that represents actual and predicted classes.

For each incidence matrix of the test cell array is compared to all incidence matrices of trainning cell array;

Two incidence matrices are similar if they have the same size and the same values;

If a similar incidence matrix is found, its row coordinate is stored in an array; So, its related information such as class label and feature vector is accessible when needed;

Once the comparison is done. The number of class labels of similar incidence matrices are verified;

If all labels predicted the same class; then the actual class and predicted class are affected to the output array;

Else if the prediction gave several class labels; then their feature vectors are compared with test feature vector using Euclidian distance; the class label of the closest vector is designed as the predicted class;

Else if both options (h and i) are not available then the predicted class is zeros label defining a rejection case;

These steps are repeated till all elements of test dataset are processed.

3. RESULTS AND DISCUSSION

3.1 Experimentations Setup

Several experimentations have been conducted under a compatible Dell Precision 5520 with the following configuration:

- Quad core processor intel i7 6820hq 2.7 GHz;

- 16 Gb RAM DDR4;

- Implementation under MATLAB programing langage.

As said earlier, the threshold that allow us to divide the character skeleton is calculated empirically. Accuracy, All possible threshold within the interval [0.1,0.3].

3.2. Database

In order to evaluate the performance of the proposed system, A database provided by (Ait Ouguengay and Taalabi, 2009b)[1] is used. It's composed of 3300 images; every alphabet is represented by 100 images with variable scale. Training dataset is composed of 2475 images and test dataset is composed of 825 images. A random selection is performed in the process of splitting database into training and test sets

3.2. Classification Performance

After performing Feature Engineering, Extraction and getting output in the forms of a class, the next step is to see how effective is the classifier test dataset. In Machine Learning, there are several performance metrics to evaluate algorithms. The Confusion matrix (also know as the Contingency Table) is of simplest metrics used for determining the effectiveness of the classifier which record actual and predicted classes. Table 2 illustrates the Confusion Matrix of a binary classification and some performance measurements.

The Contingency table in itself is not a performance metric as such, but almost all of the performance metrics are based on it and the numbers inside it. An example of metrics are Precision, Recall, Accuracy and F-measure which can be used for sorting algorithms primarily used by search engines.

Where:

$$\text{Precision} = \text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TP} + \text{FP} + \text{FN} + \text{FP}} \quad (4)$$

$$\text{Recall} = \text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (5)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (6)$$

$$\text{F} - \text{measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

Table 2. Confusion Matrix of a binary classification and some performance measurements

	Positives Actu a Negatives l	
Positives	True Positive (TP)	True Negative (TN)
Negatives	False Negative (FN)	False Positive (FP)

For the threshold of feature extraction process, we performed a series of runs by changing the threshold value starting from 0.1 till 0.3. The starting value is minimums value that produce a reasonable number of key points. As shown in figures 5-9 the value 0.21 produced the best results in term of Accuracy, Precision, Recall, Specificity and F-measure. Figure 4 illustrates contingency table based of the best value.

Based on this value learning process of 3300 images is elapsed in 61 sec and an accuracy of 99.97% while 8.7% of images are reject in the search phase. Rejection can be minimized using an alternative classifier on rejected images. Tables presents the overall performance of the proposed system.

Table 3. Classifier Performance metrics based on Threshold value 0.21

Metrics	Accuracy (%)	Precision (%)	F-measure (%)	Recall (%)	F-measure2 (%)	Test Time (s)
Results	99.9678	99.5194	99.98335	99.510101	99.50990	0.88824

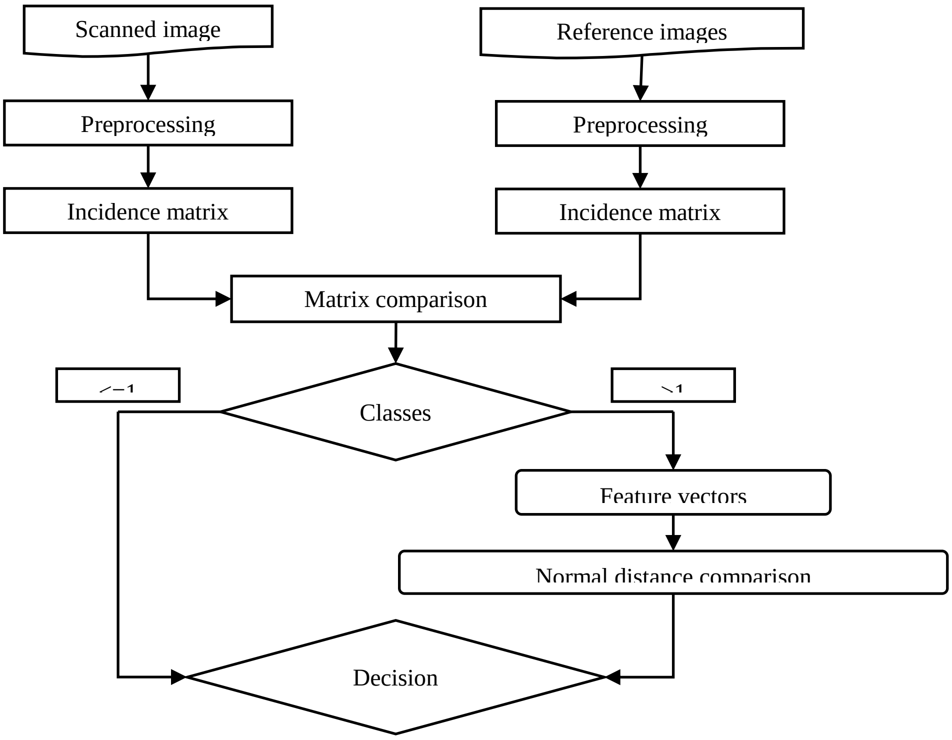
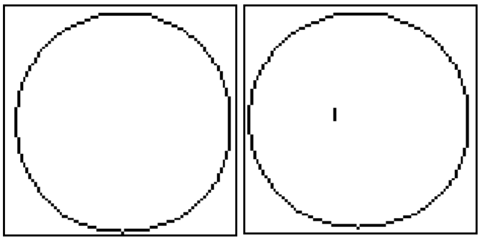
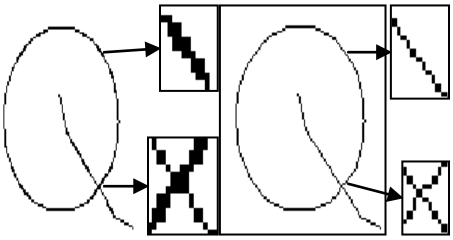


Fig 1: Proposed system architecture



(a) (b)

Fig 1: Thinning enhancement: (a) Zhang-Suen applied on the image, (b) Zhang-Suen applied to the 8-connected component of the image



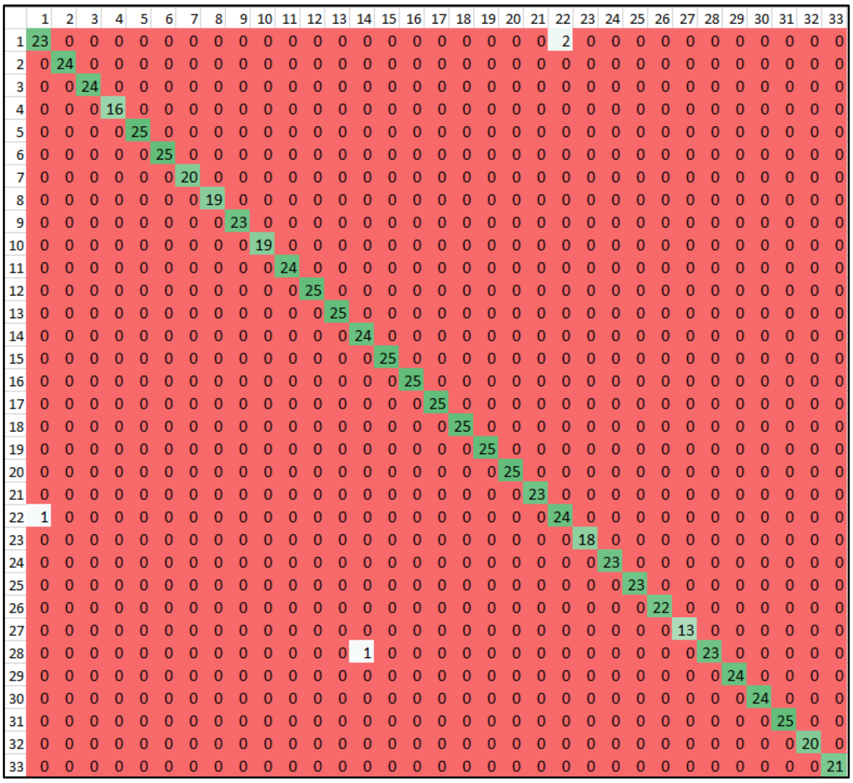
(a) (b)

Fig 1 : Skeleton smoothing: (a) Before smoothing, (b) After smoothing

4. CONCLUSION

An effort is made to provide a fast and accurate Amazigh OCR system without relying on the parallel combination of multiple descriptors/classifiers. The idea is to use the simplest ways to achieve our goal. It consists on structural information in the form of incidence matrix, then a search of similar images is performed based on simple comparison between the graphical representation of input images and their counterpart in training dataset. Then alternatively, Euclidian distance is used to refine the search.

According to the analyzes, the proposed system gives good results in term confusion. Yet, due to the rejection rate the proposed system is far from perfection. In future work, refinement will be applied to this system in order to minimize rejection without affecting performance.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
1	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
2	0	24	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	0
3	0	0	24	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
4	0	0	0	16	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
5	0	0	0	0	25	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
6	0	0	0	0	0	25	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
7	0	0	0	0	0	0	20	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
8	0	0	0	0	0	0	0	19	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
9	0	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0
22	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0
26	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	22	0	0	0	0	0	0	0
27	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	0	0	0	0	0	0
28	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	0	23	0	0	0	0	0
29	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	24	0	0	0	0
30	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	24	0	0	0
31	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	25	0	0
32	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	0	20	0
33	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	0	0	21

Fig 4 : Contingency Table based on thresholds best value (Alphabets numbered from 1 to33)

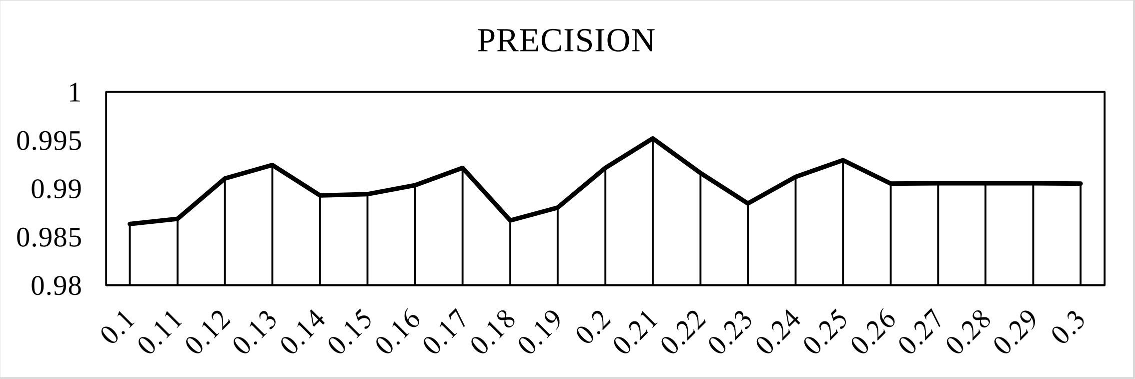


Fig 5 : Precision measurement per threshold

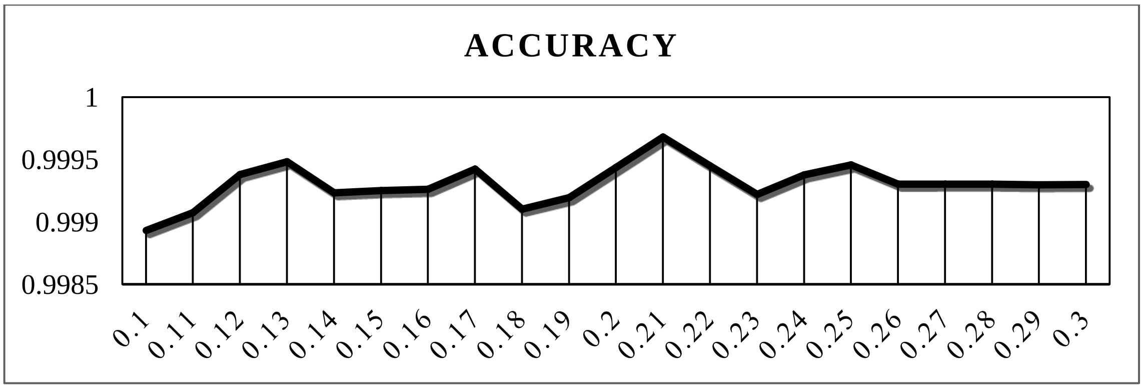


Fig 6 : Accuracy measurement per threshold

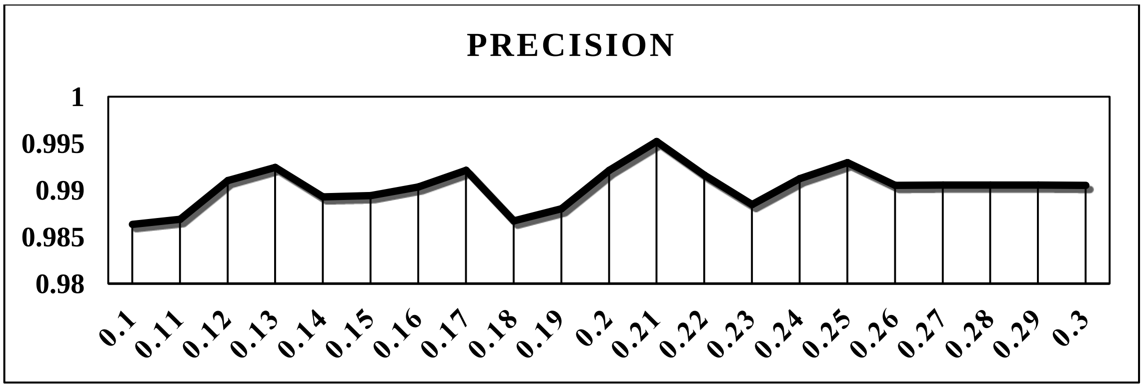


Fig 7 :Precision measurement per threshold

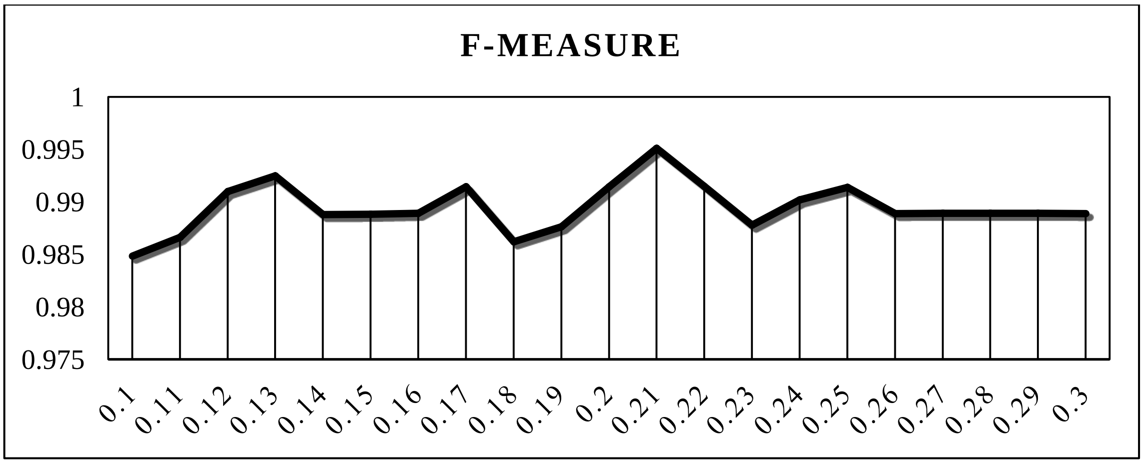


Fig 8 : F-measure measurement per threshold

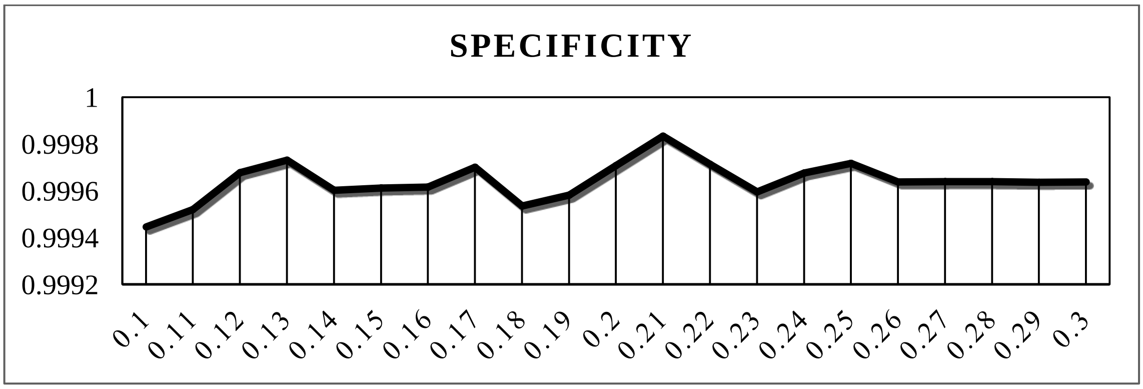


Fig 9 : Spicificity measurement per threshold

5. REFERENCES

[1] Ait Ouguengay, Y., Taalabi, M., 2009a. Elaboration d'un réseau de neurones artificiels pour la reconnaissance optique de la graphie amazighe: Phase d'apprentissage. Systèmes Intell.-Théories Appl.

[2] Amrouch, M., Rachidi, A., El Yassa, M., Mammass, D., 2009. Printed amazigh character recognition by a hybrid approach based on Hidden Markov Models and the Hough transform, in: Multimedia Computing and Systems, 2009. ICMCS'09. International Conference On. IEEE, pp. 356–360.

[3] Amrouch, M., Es-Saady, Y., Rachidi, A., El-Yassa, M., Mammass, D., 2012. A novel feature set for recognition of printed amazigh text using maximum deviation and hmm. Int J Comput Appl 44.

[4] Ayachi, R.E., Fakir, M., Bouikhalene, B., 2011. Recognition of TIFINAGHE Characters Using A multilayer Neural Network. Int. J. Image Process. IJIP 5, 109.

[5] El Ayachi, R., Oujaoura, M., Fakir, M., Minaoui, B., 2014. Code Braille et la reconnaissance d'un document écrit en Tifinagh, in: Proceedings of the International Conference on Information and Communication Technologies for the Amazigh.

[6] Es Saady, Y., Rachidi, A., El Yassa, M., Mammas, D., 2010. Printed Amazigh Character Recognition by a Syntactic Approach using Finite Automata. Int. J. Graph. Vis. Image Process. 10, 1–8.

[7] Es Saady, Y., Rachidi, A., Elyassa, M., Mammass, D., 2008. Une méthode syntaxique pour la reconnaissance de caractères Amazighes imprimés. CARI'08.

[8] Bencharef, O., Fakir, M., Minaoui, B., Bouikhalene, B., 2011. Tifinagh Character Recognition Using Geodesic Distances, Decision Trees & Neural Networks. IJACSA Int. J. Adv. Comput. Sci. Appl. Spec. Issue Artif. Intell. 1–5.

[9] Oujaoura, M., El Ayachi, R., Minaoui, B., Fakir, M., Bouikhalene, B., Bencharef, O., 2013. Invariant descriptors and classifiers combination for recognition of isolated printed Tifinagh characters, in: Third International Symposium on Automatic Amazigh Processing (SITACAM'13). Beni-Mellal, Morocco.

[10] Oujaoura, Mustapha, Minaoui, B., Fakir, M., 2013. Walsh, Texture and GIST Descriptors with Bayesian Networks for Recognition of Tifinagh Characters. Int. J. Comput. Appl. 81.

[11] Ouadid, Youssef, Fakir, M., Minaoui, B., 2016. Tifinagh Printed Character Recognition through Structural Feature Extraction. Int. J. Comput. Vis. Image Process. IJCVIP 6, 42–53.

[12] Ouadid, Y., Minaoui, B., Fakir, M., 2016. Spectral Graph Matching for Printed Tifinagh Character, in: Computer Graphics, Imaging and Visualization (CGiV), 2016 13th International Conference On. IEEE, pp. 105–111.

[13] Ouadid, Y., Minaoui, B., Fakir, M., Abouelala, O., 2014. New Approach ofTifinagh Character Recognition using Graph Matching.

[14] Xu, X., Xu, S., Jin, L., Song, E., 2011. Characteristic analysis of Otsu threshold and its applications. Pattern Recognit. Lett. 32, 956–961.

[15] Zhang, T.Y., Suen, C.Y., 1984. A fast parallel algorithm for thinning digital patterns. Commun. ACM 27, 236–239.

[16] Djematen, A., Taconet, B., Zahour, A., 1998. Une méthode statistique pour la reconnaissance de caractères berbères manuscrits. CIFED '98 170–178.

[17] Oulamara, A., Duvernoy, J., 1988. An application of the Hough transform to automatic recognition of Berber characters. Signal Process. 14, 79–90.