

# Recognition of Isolated Printed Tifinagh Characters

M. OUJAOURA

Computer Science Department  
Faculty of Science and  
Technology, Sultan Moulay  
Slimane University  
Béni Mellal, Morocco.

B. MINAOUI, M. FAKIR

and R. EL AYACHI  
Computer Science Department  
Faculty of Science and  
Technology, Sultan Moulay  
Slimane University  
Béni Mellal, Morocco

O. BENCHAREF

Computer Science Department  
Higher School of Technology  
Cadi Ayyad University  
Essaouira, Morocco

## ABSTRACT

Most of the reported works in the field of character recognition systems achieve modest results by using a single method for calculating the parameters of the character image and a single approach in the classification phase of the system. So, in order to improve the recognition rate, this document proposes an automatic system to recognize isolated printed Tifinagh characters by using a fusion of some classifiers and a combination of some features extraction methods. The Legendre moments, Zernike moments, Hu moments, Walsh transform, GIST and texture are used as descriptors in the features extraction phase due to their invariance to translation, rotation and scaling changes. In the classification phase, the neural network, the Bayesian network, the multiclass SVM (Support Vector Machine) and the nearest neighbour classifiers are combined together. The experimental results of each single features extraction method with each single classification method are compared with our approach to show its robustness. A recognition rate of 100 % is achieved by using some combined descriptors and classifiers.

## General Terms

Pattern Recognition, Image processing, Machine learning.

## Keywords

Recognition system; Legendre moments; Zernike moment; Hu moments; Texture; GIST; Walsh transform; Neural Networks; Bayesian Networks; Multiclass SVM; nearest neighbour classifier; Tifinagh characters.

## 1. INTRODUCTION

Today, the work of computerization of the Amazigh research is concentrated in Morocco and Algeria. Morocco has three research centers: the Royal Institute of the Amazigh Culture (IRCAM) at Rabat, University Sultan Moulay Slimane at Beni Mellal and Ibn Zohr University at Agadir. In the recent years, numerous researchers have annually published valuable publications in the field of Optical Character Recognition (OCR) especially the recognition of Tifinagh characters. In their paper, Es Saady and all [1] are treated the printed and isolated Amazigh characters using an automatic character recognition based on the formalism of finite automata. They encountered integration difficulties for the 33 references characters. Bencharef and all [2] are used the metric descriptors based on the calculation of the Riemannian metric. They indicated that these descriptors are known for their reliability towards the change of scale, the existence of noise and geometric distortions. For the classification of Tifinagh characters, SVM and neural networks were used. El Ayachi and all [3] are developed an automatic system for recognition of Tifinagh characters using a multilayer neural networks in classification step and Walsh transform in the extraction phase. Amrouch and all [4] are used a Hidden Markov model

(HMM), and a Haugh transform for the recognition of amazigh character. A Markov model for each character complicates the integration of this system. Many other works was published in the same area but none of them was exceeded and achieved a recognition rate of the proposed approach [5, 6]. Also, the fact that some descriptors give a very good result with some classifiers in some situation leads us to conclude that their combination can be a good idea to develop a very efficient system for character recognition especially tifinagh characters. Therefore, in this paper, a new method based on the combination of multiple descriptors and classifiers is proposed in order to increase considerably the recognition rates of tifinagh characters.

The rest of the paper is organized as follows. The section 2 presents a tifinagh characters and the proposed approach while the Section 3 discusses the primordial task of any recognition system. It's the classification problems using some classifiers based on a combination of several algorithms like neural network, Bayesian network, SVM and nearest neighbor classifiers. The Section 4 is reserved for another important task which is the features extraction problems in addition to a brief formulation for Hu moments, Legendre moments, Zernike moments, Walsh transform, GIST and texture as features extraction methods. The Section 5 presents the experimental results for the recognition system. Finally, the conclusion and a future work are given in the section 6.

## 2. RECOGNITION SYSTEM OF TIFINAGH CHARACTERS

### 2.1 Tifinagh Characters

The Amazigh language is represented today by its spoken variants by people called Berbers or amazigh. These variants are present in Morocco, Egypt, Algeria, Tunisia, Niger and Mali. The used Berber script called Tifinagh alphabets is not absolutely united. She knows quite a number of variations through times and places. There are about thirty varieties.

In Morocco, the Berber-speaking population is divided between several dialects: Tarifite in the north, Tamazight in the High and Middle Atlas and tachelhite in the center of the country. Moroccan Berbers also use the Tifinagh characters as ornamental motifs in handicrafts (carpets, jewelery, pottery ...) and to decorate the interior of the traditional houses. The Royal Institute of Amazigh Culture (IRCAM) proposed the standardization of Tifinaghe Alphabet to the International Organization for Standardization since 2004. The Tifinagh alphabet adopted by the Royal Institute of Amazigh Culture (IRCAM) is composed from 33 characters as shown in Fig. 1.

Character number	Character	Character number	Character	Character number	Character
1	o	12	Λ	23	Q
2	Θ	13	h	24	Y
3	X	14	X	25	⊙
4	X <sup>u</sup>	15	z	26	⊙
5	Λ	16	Σ	27	Ⓒ
6	E	17	I	28	+
7	⊙	18	H	29	E
8	H	19	□	30	U
9	R	20	I	31	5
10	R <sup>u</sup>	21	⊙	32	⋈
11	⊙	22	○	33	⋈

Fig 1: Tifinagh characters adopted by the IRCAM.

## 2.2 Recognition system

In this work, the bloc diagram of the proposed system is illustrated by the Fig. 2. It contains three phases to recognize isolated printed Tifinagh characters: Pre-processing, Extraction and Classification phases.

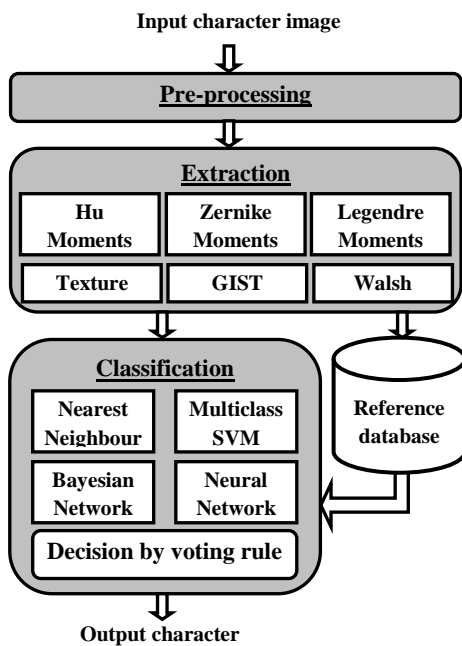


Fig 2: Bloc diagram of the proposed recognition system

In the pre-processing phase, normalization is applied to remove unwanted areas using the method of histogram; In this phase, we first calculate the horizontal and vertical histograms, then the histogram is scanned horizontally in both directions: respectively from the top to the bottom and from the bottom to the top until finding the first black pixels, thereafter, the vertical histogram is traversed in both directions: respectively from the left to the right and from the right to the left until finding the first black pixels. Finally,

after determining the positions of the first black pixels, we eliminate the unwanted areas.

To calculate the features of character image, Legendre moments, Hu moments, Zernike moments, Walsh transform, GIST and texture are used in the second phase which is the extraction phase. The classification phase adopts the neural network, Bayesian network, the SVM (Support Vector Machine) and the nearest neighbor classifier to recognize the correct character. Each one of the descriptors with each one of the classifiers votes for a given character. The character with maximum votes is selected to be the correct recognised character.

## 3. CLASSIFICATION

The robustness of the recognition system is based on the decision given by the classification phase. Many classifiers are used singly to classify the tifinagh characters. They can be combined together with descriptors in order to increase the recognition rates of amazigh characters. This document presents the combination of some classification approaches (Neuronal Network, Bayesian network, SVM, nearest neighbor classifiers) known in the recognition field [7].

### 3.1 Neural network

Neural networks (or artificial neural networks) learn by experience, generalize from previous experiences to new ones, and can make decisions [8, 9].

A multilayer neural network consists of an input layer including a set of input nodes, one or more hidden layers of nodes, and an output layer of nodes. Fig. 3 shows an example of a three layer network used in this paper, having input layer formed by M nodes, one hidden layer formed by L nodes, and output layer formed by N nodes [7].

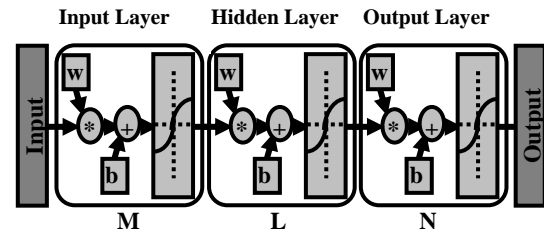


Fig 3: The three layer neural network

This neural network is trained to classify inputs according to target classes. The training input data are loaded from the reference database while the target data should consist of vectors of all zero values except for a one element, where its index is the class they are to represent. The transfer function used in this tree layer neural network is hyperbolic tangent sigmoid transfer function defined by:

$$tsig(x) = 2/(1 + \exp(-2x)) - 1 \quad (1)$$

According to authors in [10], the number of neurons in the hidden layer is approximately equal to:

$$L = E(1 + \sqrt{M(N + 2)}) \quad (2)$$

Where:

- E(x) denotes the integer part of x.

- M and N are respectively the number of neurons in the input and output layers.

### 3.2 Support vector machine (SVM)

Support vector machine (SVM) were originally designed for binary classification. As shown in Fig. 4, SVM is a classification method which is based on finding a hyper-plan that separates data sets into two classes. Several methods have been proposed to construct a multi-class classifier [12] by combining one-against-one binary classifiers or one-against-all binary classifiers. The data sets can be linearly separable or nonlinearly separable. The nonlinearly separable cases require the use of kernel function in order to obtain linearly separable data sets [11, 12].

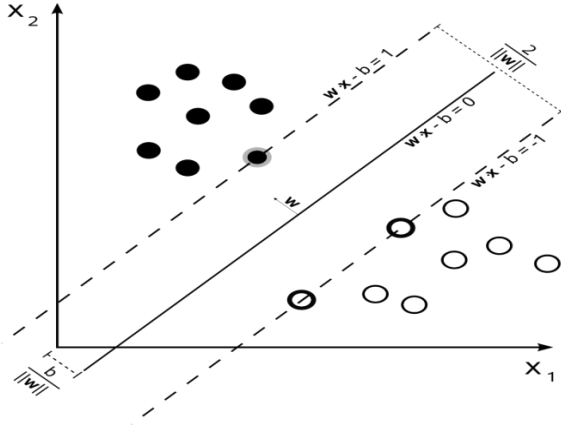


Fig 4: Binary SVM with samples from two classes

In this paper, the one-against-one and the one-against-all multiclass classifier are used. Those classifiers are based on the Gaussian kernel function defined by:

$$K(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right) \quad (3)$$

Where  $\sigma > 0$  equal to 1.

Many other kernel functions can be used for each binary classifier.

#### 3.2.1 One-against-one multiclass SVM

From N class in data sets, the one-against-one multiclass SVM method constructs  $N(N-1)/2$  binary classifier where each one is trained on data from two classes. The Structure the one-against-one multiclass SVM classifier can be represented by the Fig. 5.

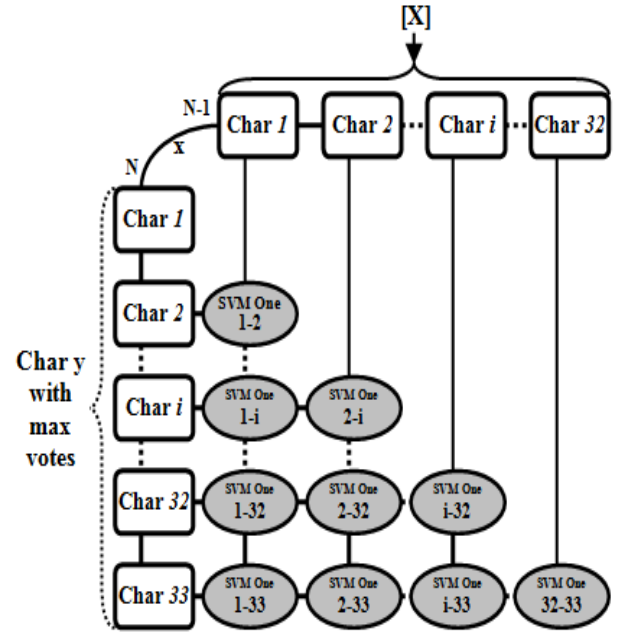


Fig 5: Structure of the one-against-one multiclass SVM classifier

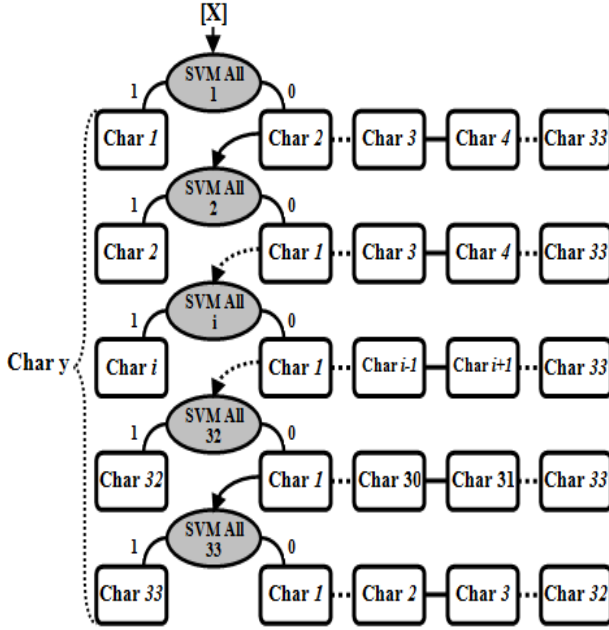
To design and extend SVM binary classifier into a one-against-one multiclass SVM, two groups of data examples are constructed from two classes. The obtained SVM binary classifier is trained to decide if the class is from the first class or it belongs to the second group of classes. This process is repeated for another couple of classes until finishing all the possible couples of the classes from data sets. So, by following this way, multiclass SVM is transformed to a multiple  $N(N-1)/2$  SVM binary classifier. Each SVM binary classifier is trained using a matrix of training data, where each row corresponds to the features extracted as an observation from a class. When classifying an object with an input features vector, each binary classifier from the multiclass SVM one-against-one model decides and votes for only one class. The class with the majority votes is the correct class which the object belongs to.

#### 3.2.2 One-against-all multiclass SVM

The one-against-all multiclass SVM classifier contains N binary classifier, where N is the number of class in data sets. The ith binary SVM is trained with all of the data examples in the ith class with positive labels, and all other data examples with negative labels.

To construct a one-against-all multiclass SVM model from binary classifier, the classes are divided into two groups: the first group is formed by one class, and the second group is all the other classes. The obtained SVM binary classifier is trained to decide if the class is from the first group or it belongs to the second groups of classes. This process is repeated for the second group that contains more than two classes until having only one class for each group. The process must stop there. So, by following this way, multiclass SVM is transformed to a multiple SVM binary classifier. Each SVM binary classifier is trained using a matrix of training data, where each row corresponds to features extracted as an observation from a class. After training phase, the multiclass SVM model is able to decide the correct class for an input features vector. To classify an object, its input features vector is presented iteratively to the ith against all binary classifier from the first to the N<sup>th</sup> classifier while the result is negative.

When the  $i$ th binary classifier gives a positive result, the process is stopped. This means that the object belongs to the  $i$ th class. The Structure of the one-against-all multiclass SVM classifier is given by the Fig. 6.



**Fig 6: Structure of the one-against-all multiclass SVM classifier**

### 3.3 Nearest neighbour

The nearest neighbour classifier is used to compare the feature vector of the input image and the feature vectors stored in the database. It is obtained by finding the distance between the prototype image and the database. The class is found by measuring the distance between a feature vector of input image and feature vectors of images in reference database. The Euclidean distance measurement is used in this paper, but other distance measurement can be also used [13].

Let  $X_1, X_2, \dots, X_k$  be the  $k$  class features vectors in the database and  $X_q$  the feature vector of the query image. The feature vector with the minimum distance is found to be the closest matching vector. It is given by:

$$d(X_q, X_j) = \min_{j \in \{1, 2, \dots, k\}} \left\{ \sqrt{\sum_i (x_q(i) - x_j(i))^2} \right\} \quad (4)$$

The nearest neighbour classifier doesn't need any training phase.

### 3.4 Bayesian networks

Bayesian networks are based on a probabilistic approach governed by Bayes' rule. The Bayesian approach is then based on the conditional probability that estimates the probability of occurrence of an event assuming that another event is verified. A Bayesian network is a graphical probabilistic model representing the random variable as a directed acyclic graph. It is defined by [14]:

- $G = (X, E)$ , Where  $X$  is the set of nodes and  $E$  is the set of edges,  $G$  is a Directed Acyclic Graph (DAG) whose vertices are associated with a set of random variables  $X = \{X_1, X_2, \dots, X_n\}$ ;

- $\theta = \{P(X_i | Pa(X_i))\}$  is a conditional probabilities of each node  $X_i$  relative to the state of his parents  $Pa(X_i)$  in  $G$ .

The graphical part of the Bayesian network indicates the dependencies between variables and gives a visual representation tool of knowledge more easily understandable by users. Bayesian networks combine qualitative part that are graphs and a quantitative part representing the conditional probabilities associated with each node of the graph with respect to parents [14].

Pearl and all [15] have also shown that Bayesian networks allow to compactly representing the joint probability distribution over all the variables:

$$P(X) = P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | Pa(X_i)) \quad (5)$$

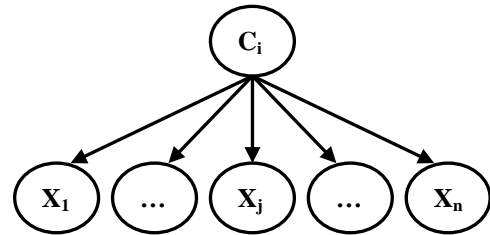
Where  $Pa(X_i)$  is the set of parents of node  $X_i$  in the graph  $G$  of the Bayesian network.

This joint probability could be actually simplified by the Bayes rule as follows [16]:

$$\begin{aligned} P(X) &= P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | Pa(X_i)) \\ &= P(X_n | X_{n-1}, \dots, X_1) \times P(X_{n-1} | X_{n-2}, \dots, X_1) \times \dots \times P(X_1) \\ &= P(X_1) \times \prod_{i=2}^n P(X_i | X_{i-1}, \dots, X_1) \end{aligned} \quad (6)$$

The construction of a Bayesian network consists in finding a structure or a graph and estimates its parameters by machine learning. In the case of the classification, the Bayesian network can have a class node  $C_i$  and many attribute nodes

$X_j$ . The naive Bayes classifier is used in this paper due to its robustness and simplicity. The Fig. 7 illustrates its graphical structure.



**Fig 7: Naive Bayes classifier structure.**

To estimate the Bayesian network parameters and probabilities, Gaussian distributions are generally used. The conditional distribution of a node relative to its parent is a Gaussian distribution whose mean is a linear combination of the parent's value and whose variance is independent of the parent's value [17]:

$$P(X_i|Pa(X_i)) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left\{-\frac{1}{2\sigma_i^2} \left(x_i - \left(\mu_i + \sum_{j=1}^{n_i} \frac{\sigma_{ij}}{\sigma_j^2} (x_j - \mu_j)\right)\right)^2\right\} \quad (7)$$

Where,

- $Pa(X_i)$  Are the parents of  $X_i$ ;
- $\mu_i, \mu_j, \sigma_i$  and  $\sigma_j$  are the means and variances of the attributes  $X_i$  and  $X_j$  respectively without considering their parents;
- $n_i$  is the number of parents;
- $\sigma_{ij}$  is the regression matrix of weights.

After the parameter and structure learning of a Bayesian network, The Bayesian inference is used to calculate the probability of any variable in a probabilistic model from the observation of one or more other variables. So, the chosen class  $C_i$  is the one that maximizes these probabilities [18, 19]:

$$P(C_i|X) = \begin{cases} P(C_i) \prod_{j=1}^n P(X_j|Pa(X_j), C_i) & \text{if } X_j \text{ has parents} \\ P(C_i) \prod_{j=1}^n P(X_j|C_i) & \text{else} \end{cases} \quad (8)$$

For the naive Bayes classifier, the absence of parents and the variables independence assumption are used to write the posterior probability of each class as given in the following equation [20]:

$$P(C_i|X) = P(C_i) \prod_{j=1}^n P(X_j|C_i) \quad (9)$$

Therefore, the decision rule d of an attribute X is given by:

$$\begin{aligned} d(X) &= \arg \max_{C_i} P(C_i|X) \\ &= \arg \max_{C_i} P(X|C_i) P(C_i) \\ &= \arg \max_{C_i} P(C_i) \prod_{j=1}^n P(X_j|C_i) \end{aligned} \quad (10)$$

The class with maximum probability leads to the suitable character for the input image.

## 4. EXTRACTION

Extraction is the second phase to apply in the recognition system. It is an important step, because the computed parameters will be used in the classification stage.

In this paper, the used approaches are: Hu moments [21], Legendre moments [22, 23, 24], Zernike moments [25, 26], walsh transform, Texture and GIST descriptors.

### 4.1 Hu moments

For a discrete image of M x N pixels with intensity function f(x, y), Hu [21] defined the following seven moments that are invariant to the change of scale, translation and rotation:

$$\phi_1 = \mu_{20} + \mu_{02} \quad (11)$$

$$\phi_2 = (\mu_{20} + \mu_{02})^2 + 4\mu_{11}^2 \quad (12)$$

$$\phi_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2 \quad (13)$$

$$\phi_4 = (\mu_{30} - \mu_{12})^2 + (\mu_{21} - \mu_{03})^2 \quad (14)$$

$$\begin{aligned} \phi_5 &= (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12}) \times \\ &\quad [(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] + \\ &\quad (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03}) \times \\ &\quad [3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \end{aligned} \quad (15)$$

$$\begin{aligned} \phi_6 &= (\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \\ &\quad + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03}) \end{aligned} \quad (16)$$

$$\begin{aligned} \phi_7 &= (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12}) \times \\ &\quad [(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] - \\ &\quad (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03}) \times \\ &\quad [3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \end{aligned} \quad (17)$$

Where

$$\mu_{pq} = \frac{\alpha_{pq}}{\alpha_{00}^{\frac{p+q}{2}+1}} \quad \text{are the Normalized central moments}$$

$$\alpha_{pq} = \int_{a_1}^{a_2} \int_{b_1}^{b_2} \left(x - \frac{M_{10}}{M_{00}}\right)^p \left(y - \frac{M_{01}}{M_{00}}\right)^q f(x, y) dx dy$$

are the central moments, and

$$M_{pq} = \int_{a_1}^{a_2} \int_{b_1}^{b_2} x^p y^q f(x, y) dx dy \quad \text{are the two-dimensional geometric moment of order (p + q) of an image f(x, y).}$$

### 4.2 Legendre moments

The Legendre moments were first introduced by Teague [22]. They were used in several pattern recognition applications [23]. The orthogonal property of Legendre polynomials implies no redundancy or overlap of information between the moments with different orders. This property enables the

contribution of each moment to be unique and independent of the information in an image [24].

The Legendre moments for a discrete image of  $M \times N$  pixels with intensity function  $f(x, y)$  is the following:

$$L_{pq} = \lambda_{pq} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} P_p(x_i) P_q(y_j) f(x, y) \quad (18)$$

Where

$$\lambda_{pq} = \frac{(2p+1)(2q+1)}{M \times N}, \quad x_i \text{ and } y_j \text{ denote the normalized pixel coordinates in the range of } [-1, +1], \text{ which are given by:}$$

$$\begin{cases} x_i = \frac{2x - (M-1)}{M-1} \\ y_j = \frac{2y - (N-1)}{N-1} \end{cases} \quad (19)$$

$P_p(x)$  is the  $p^{\text{th}}$ -order Legendre polynomial defined by:

$$P_p(x) = \sum_{k=0}^p \left\{ \frac{(-1)^{\frac{p-k}{2}} (p+k)! x^k}{2^p k! \left(\frac{p-k}{2}\right)! \left(\frac{p+k}{2}\right)!} \right\}_{p-k=\text{even}} \quad (20)$$

In order to increase the computation speed for calculating Legendre polynomials, the recurrent formula of the Legendre polynomials is used, it is defined by:

$$\begin{cases} P_p(x) = \frac{(2p-1)x}{p} P_{p-1}(x) - \frac{(p-1)}{p} P_{p-2}(x) \\ P_1(x) = x, \quad P_0(x) = 1 \end{cases} \quad (21)$$

### 4.3 Zernike moments

Zernike moments are the mapping of an image onto a set of complex Zernike polynomials. As these Zernike polynomials are orthogonal to each other, Zernike moments can represent the properties of an image with no redundancy or overlap of information between the moments [25]. Due to these characteristics, Zernike moments have been utilized as feature sets in many applications [26].

The discrete form of the Zernike moments of an image size  $M \times N$  represented by  $f(x, y)$  is expressed, in the unit disk  $x^2 + y^2 = 1$ , as follows:

$$Z_{pq} = \frac{p+1}{\lambda} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} R_{p,q}(r_{xy}) e^{-jq\theta_{xy}} f(x, y) \quad (22)$$

Where

$$\begin{cases} R_{p,q}(r) = \sum_{s=0}^{(p-|q|)/2} \frac{(-1)^s (p-s)! r^{p-2s}}{s! \left(\frac{p+|q|}{2} - s\right)! \left(\frac{p-|q|}{2} - s\right)!} \\ p-|q| = (\text{even}), \quad |q| \leq p, \quad p \geq 0 \end{cases} \quad (23)$$

$\lambda$  is the number of pixels located in the unit circle, and the transformed phase  $\theta_{xy}$  and the distance  $r_{xy}$  at the pixel of coordinates  $(x, y)$  are [27]:

$$\begin{cases} \theta_{xy} = \tan^{-1} \left( \frac{(2y - (N-1))/(N-1)}{(2x - (M-1))/(M-1)} \right) \\ r_{xy} = \sqrt{\left( \frac{2x - (M-1)}{M-1} \right)^2 + \left( \frac{2y - (N-1)}{N-1} \right)^2} \end{cases} \quad (24)$$

Most of the time taken for the computation of Zernike moments is due to the computation of radial polynomials. Therefore, researchers have proposed faster methods that reduce the factorial terms by utilizing the recurrence's relations on the radial polynomials [27, 28]. In this paper, Zernike moments are obtained using the direct method and other method based on the recurrence's relations, defined by:

$$\begin{cases} R_{p,q}(r) = \sum_{s=0}^{(p-|q|)/2} \frac{(-1)^s (p-s)! r^{p-2s}}{s! \left(\frac{p+|q|}{2} - s\right)! \left(\frac{p-|q|}{2} - s\right)!} \\ \text{if } q=0, p-|q| = (\text{even}), \quad |q| \leq p, \quad p \geq 0 \\ R_{p,q}(r) = r^p \\ \text{if } p=q \\ R_{p,q}(r) = \frac{2rp R_{p-1,q-1}(r) - (p-q) R_{p-2,q}(r)}{p+q} \\ \text{if } p \neq q, q \neq 0 \end{cases} \quad (25)$$

The usage of direct method to compute radial polynomials in the case of  $q=0$  will considerably increase the computation time especially when  $p$  is very large.

### 4.4 Texture Descriptors

The texture descriptor is extracted using the co-occurrence matrix introduced by Haralick in 1973 [29]. So for a grayscale image  $I$  of size  $N \times N$ , for  $(k, l) \in [1, \dots, N]^2$  and  $(a, b) \in [1, \dots, G]^2$ , the co-occurrence matrix  $M_{k,l}[I]$  of the character image  $I$  is defined by:



$$M_{k,l}(a,b) = \frac{1}{(N-k)(N-l)} \times \sum_{i=1}^{N-k} \sum_{j=1}^{N-l} \delta(I(i,j)-a, I(i+k,j+l)-b) \quad (26)$$

Where  $\delta$  is the unit pulse defined by:

$$\delta(x, y) = \begin{cases} 1 & \text{if } x = y = 0 \\ 0 & \text{else} \end{cases} \quad (27)$$

As they measure local interactions between pixels, they are sensitive to significant differences in spatial resolution between the images. To reduce this sensitivity, it is necessary to normalize these matrices by the total number of the considered co-occurrences matrix:

$$M_{k,l}([I], a, b) = \frac{M_{k,l}([I], a, b)}{\sum_{i=0}^{T-1} \sum_{j=0}^{T-1} M_{k,l}([I], i, j)} \quad (28)$$

Where T is the number of quantization.

To reduce the large amount of information of these matrices, the 14 Haralick indices [29] are used.

#### 4.5 Walsh Transformation

The Walsh transformation W (u, v) can be calculated using the following formula:

$$W(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) g(x, y, u, v) \quad (29)$$

Where  $f(x, y)$  is intensity of the pixel with the coordinates (x, y) in the original binary image,  $u, v = 0, 1 \dots N-1$ . The size of the image f is N x N, and function g is the kernel function of the transformation and has the following form:

$$g(x, y, u, v) = \frac{1}{N} \prod_{i=0}^{n-1} (-1)^{b_i(x)b_{n-i-1}(u) + b_i(y)b_{n-i-1}(v)} \quad (30)$$

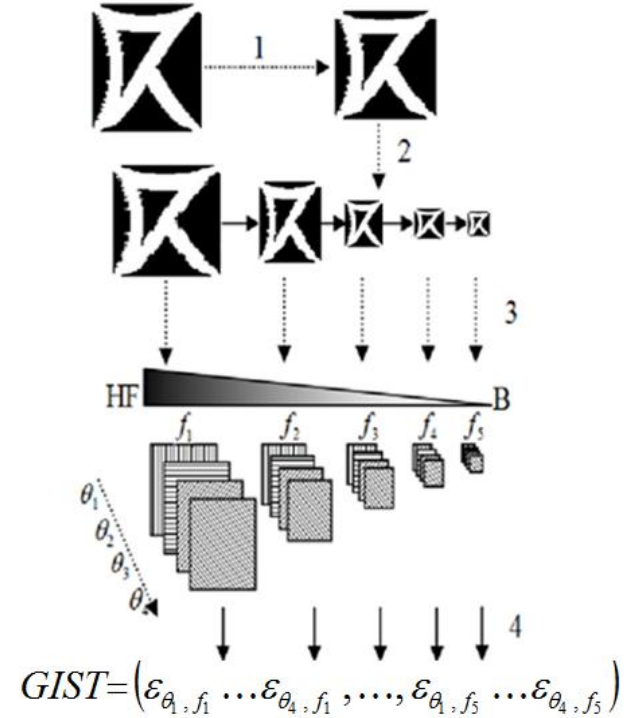
Where  $b_i(x)$  is the ith bit in the binary expansion of x, so it is equal either 0 or 1, and  $N=2^n$ .

The Walsh transform is unique in the sense that if we consider two different binary images, the corresponding feature vectors are also different. Since the Walsh transformation is invariant under size changes, to perform and calculate the Walsh transformation, the original image is firstly resized to the size of  $2^n \times 2^n$  for  $n=8$ .

#### 4.6 GIST descriptors

In computer vision, GIST descriptors are a representation of a low-dimensional image that contains enough information to identify the scene in an image. Global GIST descriptors allow a very small size representation of an image. These descriptors were introduced by Oliva and Torralba in 2001 [30, 31]. They can represent the dominant spatial structure of

the scene from a set of perceptual dimensions. The authors have tried to capture the gist descriptor of the image by analyzing the spatial frequency and orientation. The global descriptor is constructed by combining the amplitudes obtained at the output of the K Gabor filters [32] at different scales E and orientations O. To reduce the feature vector size, each filtered output image is scaled and divided into N \* N blocks (N between 2 and 16), which gives a vector of dimension N \* N \* K \* E \* O. This dimension can be further reduced by a principal component analysis (PCA), which also gives the weights applied to different filters [33].



**Fig 8: Block diagram for computation and extraction of GIST Descriptor.**

As specified above, the Fig. 8 presents a block diagram summarizing the different steps for the computation and extraction of GIST descriptors. After the pre-processing step of the input character image, the next step consists on changing the character image into different scales and orientations. Finally, the features vectors are calculated for each scale, orientation and frequency. Those feature vectors are combined to form a global feature descriptor which can be reduced by a principal component analysis (PCA).

#### 5. RESULTS

To test the accuracy of the combined classifiers and descriptors, the recognition rate of each classifiers and descriptors are calculated separately. Only 3 descriptors with 4 classifiers are firstly combined and 6 descriptors with 5 classifiers are secondly combined in order to see if the recognition rates can be increased by combination of a lot of descriptors and classifiers. Finally, from the single result of each descriptor with each classifier, only those descriptors and classifiers with recognition rate higher than 80% are selected and combined in order to decrease considerably the execution time while preserving and improving the recognition rate of the Tifinagh character recognition system.

In experiments, voting rule classifier combination schemes are used. Then, for each descriptor, each one of the combined

classifiers votes for the appropriate character. The character with maximum votes is selected and considered to be the suited character. The total number of the votes is given by the product of the number of the used descriptors and the number of the used classifiers. The efficacy of the recognition system is based on the decision given in the classification phase. The efficient description of the character image is also very important for the accuracy of the recognition system.

The computation of recognition and error rates with execution time is presented for each descriptors and each classification approaches in the following table 1:

**Table 1. Recognition and error rates with Execution times of single classifier and descriptor**

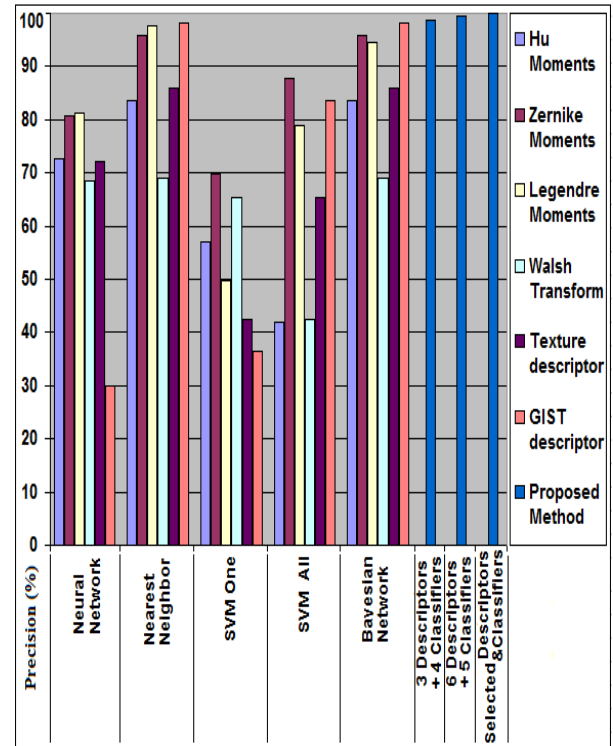
	Descriptor	Classification Approaches				
		Neural Network	Nearest Neighbor	SVM One	SVM All	Bayesian Network
Recognition rate (%)	Hu Moments	72.73	83.64	56.97	41.82	83.64
	Zernike Moments	80.61	95.76	69.70	87.88	95.76
	Legendre Moments	81.21	97.58	49.70	78.79	94.55
	Walsh Transform	68.48	69.09	65.45	42.42	69.09
	Texture descriptor	72.12	86.06	42.42	65.45	86.06
	GIST descriptor	30.03	98.18	36.36	83.64	98.18
	Proposed Method					
Error rate (%)	Hu Moments	17.86	16.36	43.03	58.18	16.36
	Zernike Moments	19.39	4.24	30.30	12.12	4.24
	Legendre Moments	18.79	2.42	50.30	21.21	5.45
	Walsh Transform	31.52	30.91	34.55	57.58	30.91
	Texture descriptor	27.88	13.94	57.58	34.55	13.94
	GIST descriptor	69.97	1.82	63.64	16.36	01.82
	Proposed Method					
Execution time (s)	Hu Moments	8.34	1.42	94.01	3.36	71.28
	Zernike Moments	859.92	943.90	1152.36	754.65	645.70
	Legendre Moments	853.16	805.33	892.41	806.00	1112.17
	Walsh Transform	15.34	2.36	137.23	8.39	476.88
	Texture descriptor	14.60	3.43	113.42	9.10	528.65
	GIST descriptor	29.68	25.27	153.33	26.56	298.62
	Proposed Method					

The recognition rate and error rate with execution times of the proposed method are presented in the following table 2:

**Table 2. Recognition rate and error rate with execution times of the proposed methods**

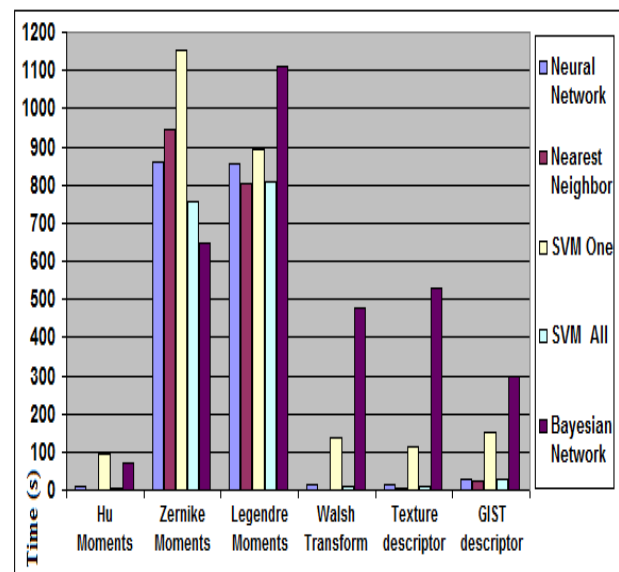
Proposed Method	Recognition rate (%)	Error rate (%)	Execution time (s)
Method based on 3 descriptors combined with 4 classifiers	98.79	1.21	1900.17
Method based on 6 descriptors combined with 5 classifiers	99.39	0.61	3503.44
Method based on descriptors and classifiers with a recognition rate higher than 80%	100.00	0.00	2513.59

From the results in Table 1 and Table 2, the recognition rate of the proposed system is improved in the recognition field of the isolated printed Tifinagh characters, despite of the processing time which is significantly increased.



**Fig 9: Comparison of the recognition rates between each single descriptor with a single classifier and a proposed method.**

The Fig. 9 presents a comparison between each single descriptor with a single classifier and a proposed method based on different combination of the descriptors and classifiers. The proposed method based on the combination of the descriptors and classifiers with recognition rate higher than 80% give a good result.



**Fig 10: Comparison of the execution times between each single descriptor with a single classifier.**



The Fig. 10 presents a comparison of the execution times between each descriptor and each classifier. It shows that Zernike and Legendre moments are very slow regarding the execution time.

In order to show the robustness of the proposed approach, further details are given by calculating the confusion matrix of some method as presented in Fig. 11 and Fig. 12.

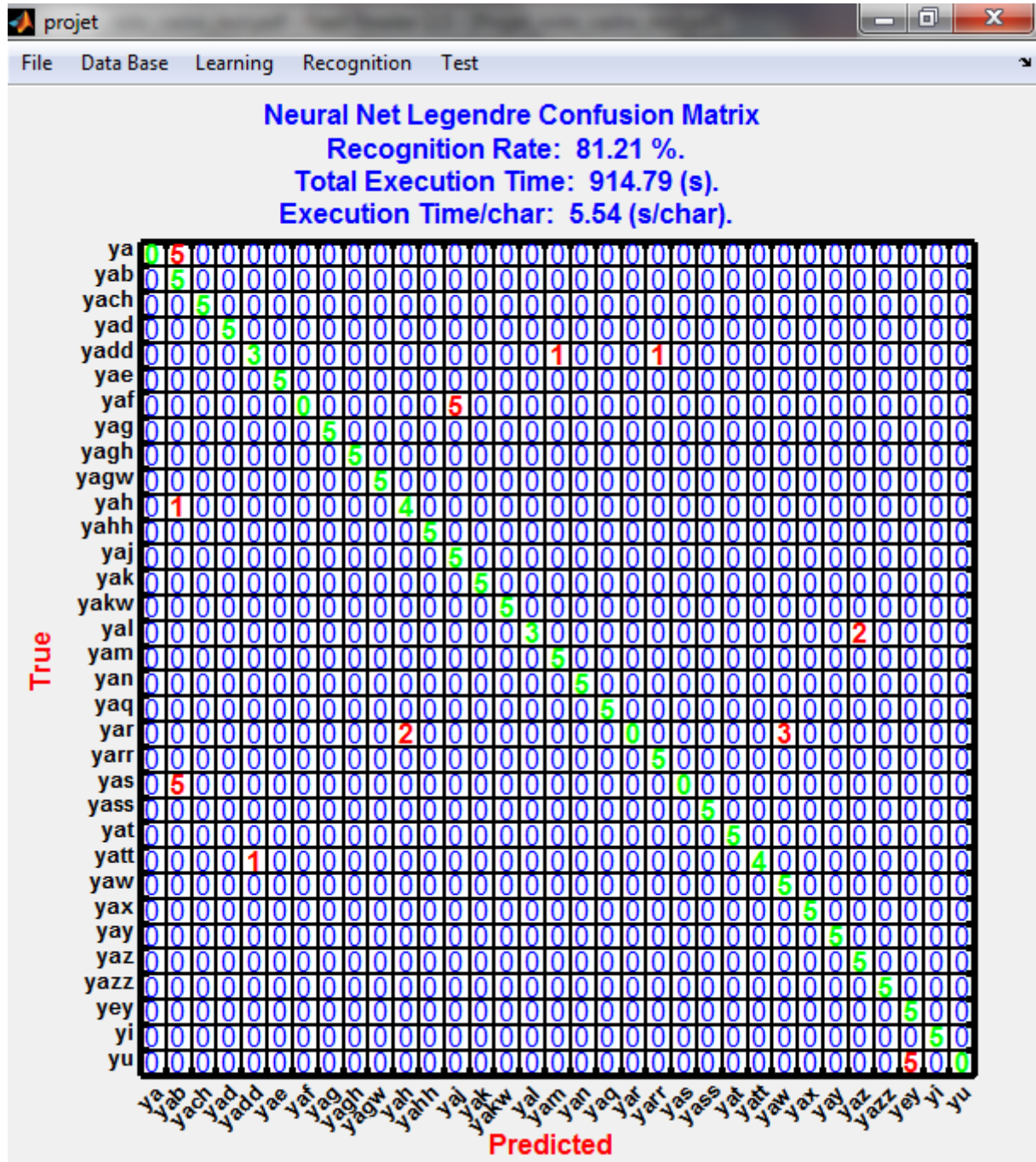
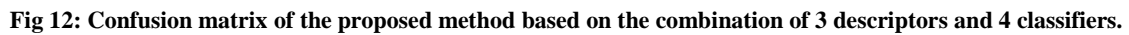


Fig 11: Confusion matrix of Legendre moments descriptor and neural network classifier.



other descriptors like Walsh transform, texture and GIST descriptors with addition also of other classifiers like Bayesian networks, the recognition rates are ameliorated significantly while the execution time is perhaps increased.

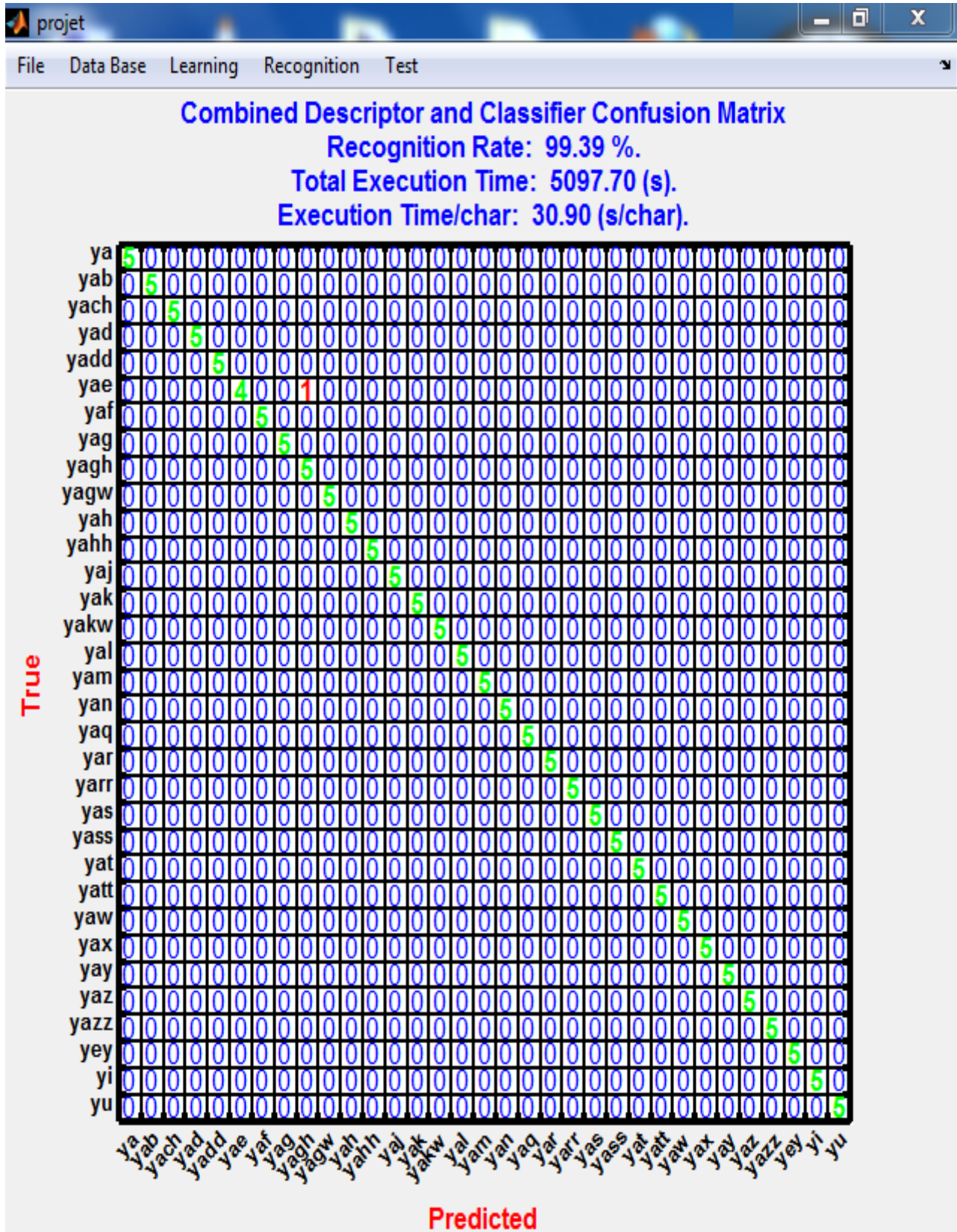


Fig 13: Confusion matrix of the proposed method based on the combination of 6 descriptors and 5 classifiers.

The Fig. 13 gives the confusion matrix of this second combination of the descriptors and classifiers while the Fig. 14 shows the confusion matrix of the proposed method based on the combination of selected descriptors and classifiers.

Only the descriptors and classifiers having recognition rate higher than 80%. With those selected descriptors and classifiers, the best recognition rate is achieved.



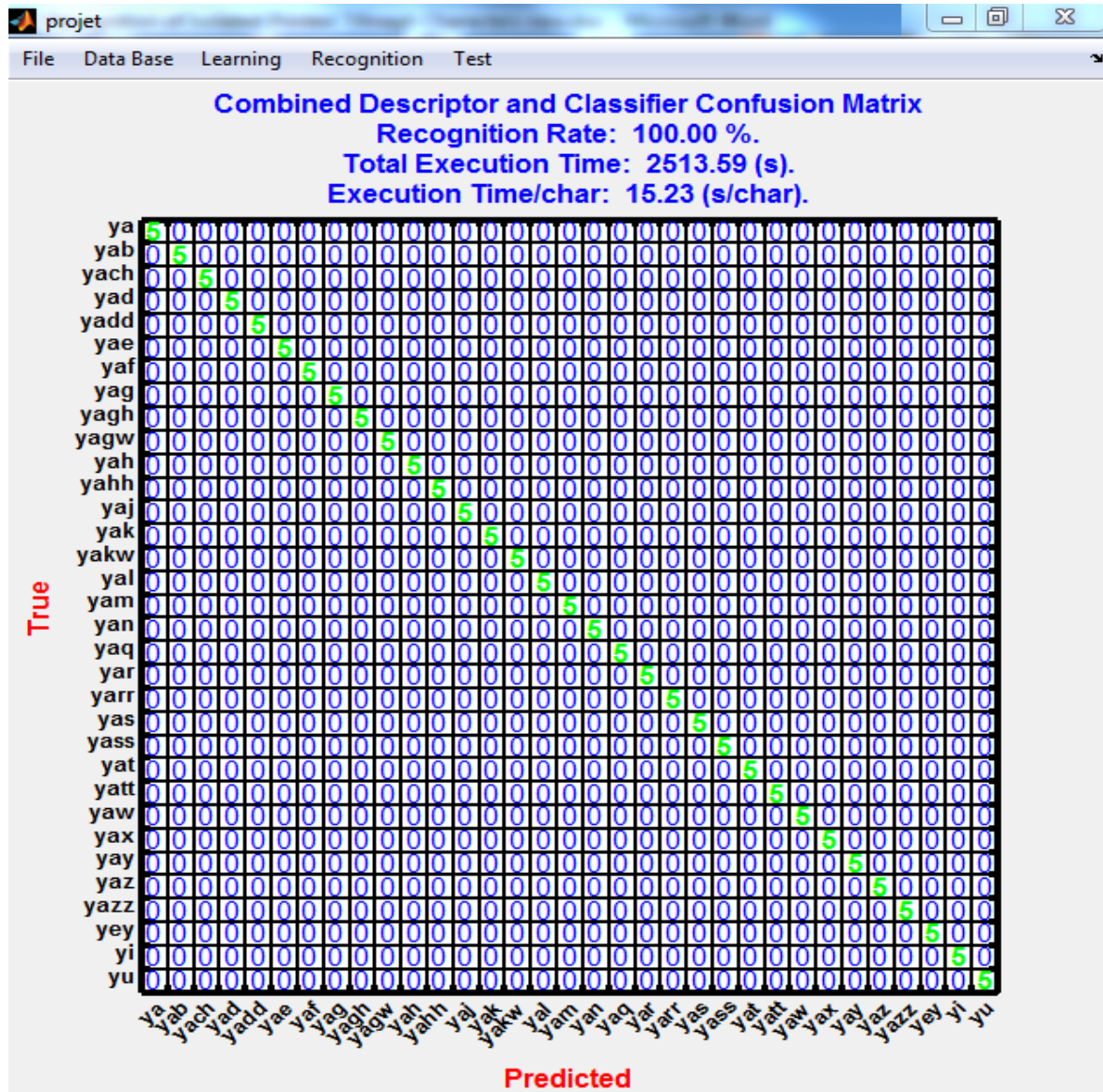


Fig 14: Confusion matrix of the proposed method based on the combination of descriptors and classifiers with a recognition rate higher than 80%.

All the tests are performed using a database containing a set of 33 reference characters and 165 image characters [6]. The proposed system has been implemented and tested on a core 2 Duo personnel computer using Matlab software.

## 6. CONCLUSION

Generally, the recognition system uses one method to extract the parameters of the input image and one approach in the classification phase. The objective of this work is to increase the recognition rates of the amazigh character recognition system by using a combination of many descriptors and classifiers since they can complement each other. A lot of extraction methods are combined with many classification approaches in order to increase the recognition rate. The obtained results show the robustness of the recognition system based on the proposed approach in the recognition field of the isolated printed Tifinagh characters. The result of the combination of some selected descriptors and classifiers are very promoting despite the increase in execution time. The use of other fast descriptors and classifiers may increase the recognition rate of Tifinagh characters and ameliorate the

execution time in order to have a fast online system. The character segmentation algorithm may be also considered for recognition of amazigh documents.

## 7. REFERENCES

- [1] Y. Es Saady, A. Rachidi, M. El Yassa, D. Mammass, « Printed Amazigh Character Recognition by a Syntactic Approach using Finite Automata », ICGST-GVIP Journal, Volume 10, Issue 2, June 2010.
- [2] O. Bencharef, M. Fakir, N. Idrissi, B. Bouikhalen et B. Minaoui, « Application de la géométrie riemannienne à la reconnaissance des caractères Tifinaghe », Agadir-Maroc, 06-07 Mai 2011. pp : 179 :188.
- [3] R. El Ayachi, K. Moro, M. Fakir et B. Bouikhalene, « On The Recognition Of Tifinaghe Scripts ». JATIT, vol. 20, No. 2, pp: 61-66, 2010.
- [4] M. Amrouch, Y. Es saady, A. Rachidi, M. El Yassa and D. Mammass. Printed Amazigh Character Recognition by a Hybrid Approach Based on Hidden Markov Models

and the Hough Transform, 978-1-4244-3757-3/09/\$25.00 ©2009 IEEE, 2009.

- [5] M. OUJAOURA, R. EL AYACHI, O. BENCHAREF, Y. CHIHAB and B. JARMOUNI, application of data mining tools for recognition of tiffinagh characters, (IJACSA) International Journal of Advanced Computer Science and Applications, special issue on selected papers from 3<sup>rd</sup> international symposium on automatic amazigh processing SITACAM13, Vol. 3, No. 2, pp.1–4, 2013. Published by The Science and Information Organization, New York, USA.
- [6] Mustapha OUJAOURA, Brahim MINAOUI and Mohammed FAKIR. Walsh, Texture and GIST Descriptors with Bayesian Networks for Recognition of Tiffinagh characters. (IJCA) International Journal of Computer Applications, Volume 81– No.12, pp. 39-46, November 2013. Published by Foundation of Computer Science, New York, USA.
- [7] M. OUJAOURA, B. MINAOUI and M. FAKIR, Multilayer Neural Networks and Nearest Neighbor Classifier Performances for Image Annotation, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 11, pp.165–171, 2012. Published by The Science and Information Organization, New York, USA.
- [8] Yue Cao, Xiabi Liu, Jie Bing and Li Song, Using Neural Network to Combine Measures of Word Semantic Similarity for Image Annotation, IEEE International Conference on Information and Automation (ICIA), pp. 833 – 837, 2011.
- [9] P.Simard, D. Steinkraus, J. C. Platt, Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis, ICDAR, 2003, pp. 958-962.
- [10] R. Lepage, & B. Solaiman. Les réseaux de neurones artificiels et leurs applications en imagerie et en vision par ordinateur, Ecole de technologie supérieure, 2003.
- [11] R. Rifkin, A. Klautau. In defence of one-versus-all classification. Journal of Machine Learning Research, Vol. 5, pp. 101–141, 2004.
- [12] K.-B. Duan, S.S Keerthi, Which is the best multiclass SVM method? An empirical study. Technical Report CD-03-12, Control Division, Department of Mechanical Engineering, National University of Singapore, 2003.
- [13] Oren Boiman, Eli Shechtman and Michal Irani, In Defense of Nearest-Neighbor Based Image Classification, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2008.
- [14] Ann.Becker, Patrick Naim : les réseaux bayésiens : modèles graphiques de connaissance. Eyrolles.1999.
- [15] J. Pearl, "Bayesian Networks" UCLA Cognitive Systems Laboratory, Technical Report (R-216), Revision I. In M. Arbib (Ed.), Handbook of Brain Theory and Neural Networks, MIT Press, 149-153, 1995.
- [16] Sabine Barrat, Modèles graphiques probabilistes pour la reconnaissance de formes, thèse de l'université Nancy 2, Spécialité informatique, décembre 2009.
- [17] George H. John and Pat Langley. Estimating continuous distributions in bayesian classifiers, the Eleventh Conference on Uncertainty in Artificial Intelligence, 1995.
- [18] Philippe LERAY, Réseaux bayésiens : apprentissage et modélisation de systèmes complexes, Habilitation A Diriger Les Recherches, Spécialité Informatique, Automatique et Traitement du Signal, Université de Rouen, novembre 2006.
- [19] Patrick Naïm, Pierre Henri Wullemmin, Philippe Leray, Olivier pourret, Anna becker, Réseaux bayésiens, Eyrolles, 3ème édition, Paris, 2008.
- [20] Tom .Mitchell: Generative and discriminative classifier: Naïve bayes and logistic regression. Machine learning. Draft 2010.
- [21] M. OUJAOURA, B. MINAOUI, M. FAKIR, B. BOUIKHALENE, R. EL AYACHI and O. BENCHAREF, Invariant Descriptors and Classifiers Combination for Recognition of Isolated Printed Tiffinagh Characters, (IJACSA) International Journal of Advanced Computer Science and Applications, special issue on selected papers from 3<sup>rd</sup> international symposium on automatic amazigh processing SITACAM13, Vol. 3, No. 2, pp.22–28, 2013. Published by The Science and Information Organization, New York, USA.
- [22] F. L. Alt, Digital Pattern Recognition by Moments, J. Assoc. Computing Machinery, Vol. 9, pp. 240-258, 1962.
- [23] M.R. Teague, Image analysis via the general theory of moments, J. Opt. Soc. Amer. 70, pp. 920–930, 1980.
- [24] Chee-Way Chonga, P. Raveendranb and R. Mukundan, Translation and scale invariants of Legendre moments, Pattern Recognition 37, pp. 119 – 129, 2004.
- [25] Sun-Kyoo Hwang, Whoi-Yul Kim, A novel approach to the fast computation of Zernike moments, Pattern Recognition 39, pp. 2065 – 2076, 2006.
- [26] A. Prata, W.V.T. Rusche, Algorithm for computation of Zernike polynomials expansion coefficients, Appl. Opt. 28, pp. 749–754, 1989.
- [27] Mustapha Oujaoura, Brahim Minaoui and Mohammed Fakir. Article: Image Annotation using Moments and Multilayer Neural Networks. IJCA Special Issue on Software Engineering, Databases and Expert Systems SEDEX (1): pp. 46-55, September 2012. Published by Foundation of Computer Science, New York, USA.
- [28] C.W. Chong, P. Raveendran, R. Mukundan, A comparative analysis of algorithms for fast computation of Zernike moments, Pattern Recognition 36 (3) , pp. 731–742, 2003.
- [29] R. Haralick, K. Shanmugan, et I. Dinstein. Textural features for image classification. IEEE Transactions on SMC, 3(6) : pp. 610–621, 1973.
- [30] Aude Oliva and Antonio Torralba. Modeling the shape of the scene : A holistic representation of the spatial envelope. International Journal of Computer Vision, 42 : pp. 145–175, 2001.
- [31] Aude Oliva , Antonio Torralba, Building the gist of a scene: the role of global image features in recognition, Progress in Brain Research, 2006.
- [32] Hans G. Feichtinger, Thomas Strohmer: "Gabor Analysis and Algorithms", Birkhäuser, 1998.
- [33] Olivier Augereau, Nicholas Journet, Jean-Philippe Domenger, "Reconnaissance et Extraction de Pièces d'identité : Une application industrielle à la détection de cartes d'identité et de passeports," 1re soumission à CIFED 2012, le 8 décembre 2011.