

# Recognition of Tifinaghe Characters Using a Multilayer Neural Network

## Rachid EL Ayachi

*Faculty of Sciences and Techniques/  
Computer Sciences Department  
Sultan Moulay Slimane University  
Béni-Mellal, BP:523, Morocco*

*rachidieea@yahoo.fr*

## Mohamed Fakir

*Faculty of Sciences and Techniques/  
Computer Sciences Department  
Sultan Moulay Slimane University  
Béni-Mellal, BP:523, Morocco*

*medfaki@yahoo.fr*

## Belaid Bouikhalene

*Faculty of Sciences and Techniques/ Information  
processing and telecommunications teams  
Sultan Moulay Slimane University  
Béni-Mellal, BP:523, Morocco*

*bbouikhalene@yahoo.fr*

---

## Abstract

In this paper, we present an off line Tifinaghe characters recognition system. Texts are scanned using a flatbed scanner. Digitized text are normalised, noise is reduced using a median filter, baseline skew is corrected by the use of the Hough transform, and text is segmented into line and lines into words. Features are extracted using the Walsh Transformation. Finally characters are recognized by a multilayer neural network.

**Keywords:** Tifinaghe Characters, Baseline Skew Correction, Segmentation, Walsh Transform, Hough Transform, Neural Network, Recognition.

---

## 1. INTRODUCTION

Optical Character Recognition (OCR) is one of the most successful applications of automatic pattern recognition. It is a very active field of research and development.

Several studies have been conducted on Latin, Arabic and Chinese characters [1, 2, 3, 4, 5, 6, 7, 8, 9]. However, for Tifinaghe characters system few works was done [13, 14, 15, 16].

Succession of operations in most digital image recognition system can be divided into three stages. First stage is a pre-processing including thresholding improving image quality, segmentation and son on. Second, features extraction for avoiding data abundance and reducing its dimension. Third stage is a classification. During this stage classes name is joint with unknown image by extracted features analyses and matching its representatives of the class, which the classifier has trained at a stage of training.

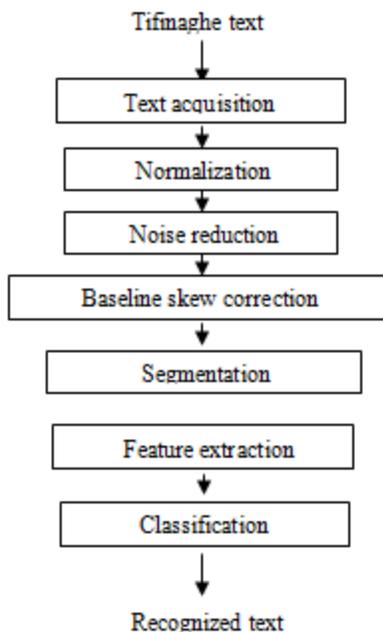
In this study a recognition system (Figure 1) for the recognition of Tifinaghe characters issued from an image scanner is presented. Initially, an image that contains Tifinaghe characters is normalized and segmented to produce a data base. Then, we applied the approach of Walsh Transform to extracted features which are used in the classification phase with a multilayer neural network.

The organisation of this paper is as follows. In section 2 characteristics of Tifinagh characters are given. In section 3 pre-processing process is described. Features extraction step is described in

section 4. Section 5 deals with the recognition step. Experimental results are given in section 6. Finally, this work is ended by a conclusion.

## 2. TIFINAGHE CHARACTERS

The Tifinaghe script is used by approximately 20 million people who speak varieties of languages commonly called Berber or Amazigh. The three main varieties in Morocco are known as Tarifite, Tamazighe, and Tachelhite. In Morocco, more than 50% of the population speaks Berber. In accordance with recent governmental decisions, the teaching of the Berber language, written in the Tifinaghe script, will be generalized and compulsory in Tifinaghe is an alphabetic writing system. It uses spaces to separate words and makes use of Western punctuation. The earliest variety of the Berber alphabet is Libyan. Two forms exist: a Western form and an Eastern form. The Western variety was used along the Mediterranean coast from Kabylia to Morocco and most probably to the Canary Islands. The Eastern variety, old Tifinaghe, is also called Libyan-Berber or old Tuareg. It contains signs not found in the Libyan variety and was used to transcribe Old Tuareg. A number of variants of Neo-Tifinaghe exist, the first of which was proposed in the 1960s by the Académie Berbère. That variant has spread in Morocco and in Algeria, especially in Kabylia. Other Neo-Tifinaghe systems are nearly identical to the Académie Berbère system. The encoding in the Tifinaghe block is based on the Neo-Tifinaghe systems. Historically, Berber texts did not have a fixed direction. Early inscriptions were written horizontally from left to right, from right to left, vertically (bottom to top, top to bottom); boustrophedon directionality was also known.



**FIGURE 1:** Tifinaghe recognized system.

Modern-day Berber script is most frequently written in horizontal lines from left to right; therefore the bidirectional class for Tifinaghe letters is specified as strong left to right. The encoding consists of four Tifinaghe character subsets: the basic set of the " Institut Royal de la Culture Amazighe (IRCAM) ", the extended IRCAM set, other Neo-Tifinaghe letters in use, and modern Tuareg letters. The first subset represents the set of characters chosen by IRCAM to unify the orthography of the different Moroccan modern day Berber dialects while using the historical Tifinaghe script. The alphabet Tifinaghe adopted by IRCAM [9] is composed of thirty-three characters representing consonants and vowels as shown in Table 1.

The table displays the Tifinaghe alphabet adopted by IRCAM. It is organized into two main sections. The top section shows the letters Σ, I, H, C, I with their corresponding Tifinaghe symbols. Below this, there are two columns of symbols, each with a corresponding Latin letter underneath. The symbols are: Σ, I, H, C, I, O, Q, Y, O, E, O, C, T, E, U, O, L, H, X, Z. A legend is provided at the bottom right of the table.

TABLE 1: Tifinaghe characters adopted by IRCAM.

### 3. PREPROCESSING

Pre-processing is the first part of Tifinaghe characters recognition system which covers four functions to produce a cleaned up version of the original image so that it can be used directly and efficiently by the feature extraction components of the OCR. These functions are: scanning the text and digitizing it into a digital image and cleaning it (by medium filter for example), converting the grey-scale image into binary image, normalizing the text, detecting and correcting Baseline Skew, and segmenting the text into lines and the lines into characters.

#### 3.1 Normalization of the position

The position normalization is designed to eliminate unwanted areas and reduce the processing time. In this operation, firstly, we compute the horizontal and vertical histograms, secondly, we scan the horizontal histogram in two directions: from top to bottom and bottom to top respectively until the first meeting of black pixels, finally, we scan the vertical histogram in two directions: from left to right and right to left respectively until the first meeting of black pixels. After obtaining the positions of first black pixels, unwanted areas are eliminated in the image as shown in (Fig. 2).

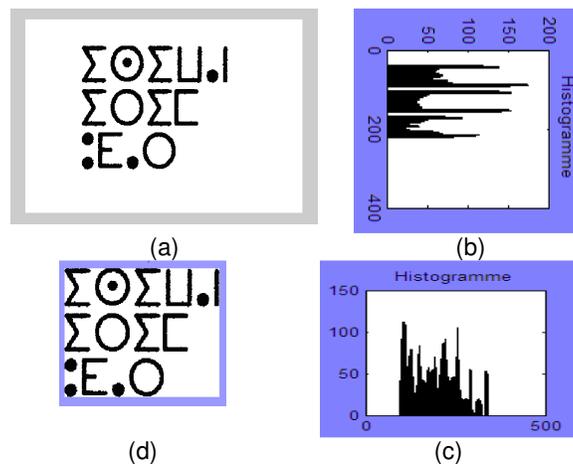


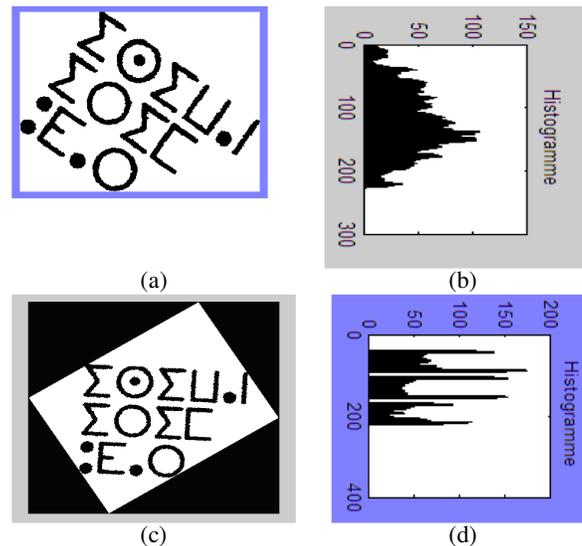
FIGURE 2: (a) Before normalization, (d) After normalization, (b) Horizontal histogram and (c) Vertical histogram

#### 3.2 Baseline Skew Detection and Correction

A skew angle is the angle that the text lines of the document image make with the horizontal direction. The skew correction is necessary for the success of many OCR systems.

There are many methods to use for detecting skew angle, such as: the Trincklin method that uses the process of least squares to estimate skew angle, the Projection method is based on the calculation of horizontal histogram of the image, Hough transform, Fourier transform, Correlation lines, k-nearest neighbors. [17,18,19].

In this paper, the operation of skew correction is to estimate a skew angle  $\theta_s$  using the Hough transform and to rotate the image by  $\theta_s$  in the opposite direction, which gave the good results as showed in (Fig.3).



**FIGURE 3:** (a) Before correction, (c) After correction, (b) Horizontal histogram before correction, (d) Horizontal histogram after correction.

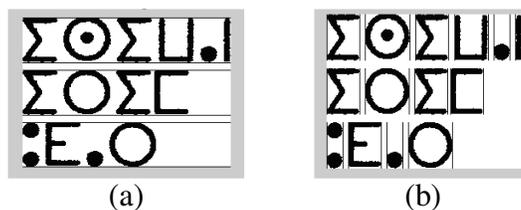
### 3.3 Segmentation

The last function to apply into pre-processing part is the segmentation; it is used to detect lines and characters in the image.

This method covers two steps: firstly, we use the horizontal histogram to detect lines; secondly, we use the vertical histogram to detect characters.

In the horizontal histogram, we browse from top to bottom until the first line containing at least one black pixel, the line is the beginning of the first line of text, then we continue traverse until a line that contains only white pixels, this line corresponds to the end of the first line of text. With the same way, we continue to detect other text lines.

In the vertical histogram, for each line of text, we browses from left to right until the first column containing at least one black pixel, this column is the beginning of the first character, then we continue traverse until a column that contains only white pixels, this column corresponds to the end of the first character. We continue detecting other characters of text with the same way.



**FIGURE 4:** (a) lines segmentation, (b) Characters segmentation

#### 4. FEATURES EXTRACTION

The second phase of Tifinaghe characters recognition system is Features extraction. Several methods can be used to compute the features: invariant momentsm Walsh transformation [20,21] etc.,.

In this recognition system, we use Walsh Transformation to extract features, because this method is independent to translation, rotation and scale change.

The Walsh transformation is given by:

$$W(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y)g(x, y, u, v) \quad (2)$$

Where  $f(x, y)$ , is the intensity of the pixel with the coordinates  $(x, y)$  in the original binary image. The size of image  $f$  is  $N*N$ , and  $u, v = 0, \dots, N - 1$ , thus we compute  $N^2$  Walsh transforms,  $g(x, y, u, v)$  is the Kernel function given by the following form:

$$g(x, y, u, v) = (1/N) \prod_{i=0}^{n-1} (-1)^{b_i(x)b_{n-i+1}(u)+b_i(y)b_{n-i+1}(v)} \quad (3)$$

Where  $b_i(x)$  is the  $i$ th bit in the binary expansion of  $x$  (it is equal either 0 or 1).

Table2 represents the seven first elements of the vector Walsh calculated for one character with his four transformations:

$W_i$	$\Sigma$	$\Sigma$	$\Sigma$	$W$
$W_1$	0	0	0	-0.0029
$W_2$	-0.0029	-0.0029	-0.0029	-0.0059
$W_3$	-0.0064	-0.0064	-0.0059	-0.0088
$W_4$	-0.0098	-0.0098	-0.0093	-0.0118
$W_5$	-0.0132	-0.0132	-0.0127	-0.0137
$W_6$	-0.0167	-0.0167	-0.0162	-0.0152
$W_7$	-0.0201	-0.0201	-0.0196	-0.0172

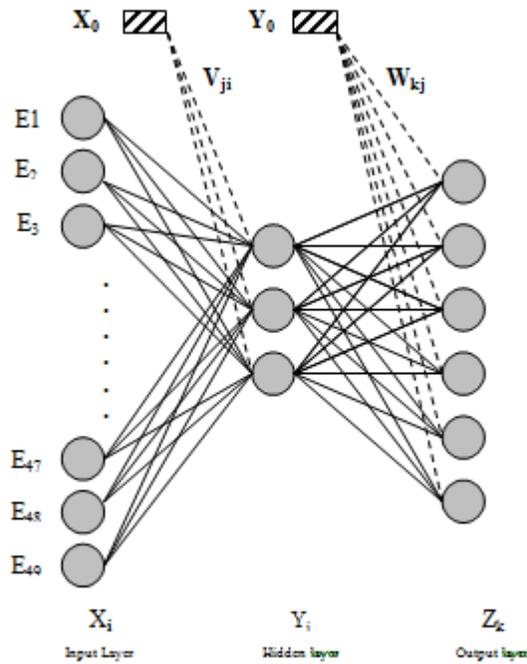
TABLE 2: Walsh Coefficients.

#### 5. CHARACTER RECOGNITION

In the character recognition system, the recognition is the last phase which is used to identify the segmented character. Where we use the Neural Network approach for several reasons: the execution time is reduced and the principle of Neural Network is simple and effective.

In this phase a neural network is used [12], from an analogy with the biological neuron, is a processor that implements simple inputs and can connect with others to form a network that can achieve a relationship any entry-exit.

The Neural Network as shown in (Fig.5) represents an example of Neural Network multilayer which contains one hidden layer. It has:



**FIGURE 5:** Neural Network

- An input layer of 49 (Walsh vector) inputs cells  $E_i = X_i$  (the cells represents the inputs  $E_i$  of Network).
- A hidden layer of 3 activations Neural  $Y_j$ .
- An output layer of 6 activations Neural  $Z_k$ .
- $49 \times 3$  connections between input layer and hidden layer, each weighted by  $V_{ji}$ .
- $3 \times 6$  connections between hidden layer and output layer, each weighted by  $W_{kj}$ .
- $X_0, Y_0$  are initialled values (scalars).

The operation of Neural Network as shown in (Fig.5) contains five steps:

- **Step 1:** (Initializing weights of connexions), the weights are randomly selected.

- **Step 2:** (propagation of inputs)

The inputs  $E_i$  are presented to input layer:  $X_i = E_i$ .

We propagate to hidden layer:

$$Y_j = f \left( \sum_{i=1}^{49} X_i V_{ji} + X_0 \right) \quad (4)$$

After for hidden layer to output layer:

$$Z_k = f\left(\sum_{j=1}^3 Y_j W_{kj} + Y_0\right) \quad (5)$$

The values  $X_0$  and  $Y_0$  are means (scalars).

$f$  is the activation function which is given by

$$f(x) = 1/(1 + \exp(-x)) \quad (6)$$

- **Step 3:** (Error back propagation)

For each example of applied learning base input of the network, we calculate the error at output layers, i.e. the difference between the desired output  $S_k$  and  $Z_k$  actual output:

$$E_k = Z_k (1 - Z_k)(S_k - Z_k) \quad (7)$$

We propagate this error on the hidden layer; the error of each neuron of the hidden layer is given by:

$$F_j = Y_j (1 - Y_j) \sum_{k=1}^6 W_{kj} . E_k \quad (8)$$

- **Step 4:** (Correction of connections weights )

We change the weights of connections:

- Between input layer and hidden layer:

$$\Delta V_{ji} = \eta . X_i . F_j \quad \text{And} \quad \Delta Y_0 = \eta . F_j \quad (9)$$

- Between hidden layer and output layer:

$$\Delta W_{kj} = \eta . Y_j . E_k \quad \text{And} \quad \Delta X_0 = \eta . E_k \quad (10)$$

Where  $\eta$  is the learning rate comprised between 0 and 1. This is experimentally determined ( $\eta = 0.9$ )

- **Step 5:** (Loop)

Loop in step tow to a criterion to define.

(Error threshold = 0.0001, Number of iterations = 50000)

After the learning of Network and the execution of Tifinaghe Characters Recognition System to recognize a Text, we use the Euclidian distance to identify the characters of Text.

$$d(t_k, o) = \left(\sum_{i=1}^6 (t_{ki} - o_i)^2\right)^{1/2} \quad (11)$$

Where,  $t_k$  is a desired output and  $o$  is the output of Network.

## 6. EXPERIMENTALS RESULTS

A Data Base used in this system contains 360 images which represents the Tifinaghe characters. All tests are applied on 158 characters.

Tests applied on several images gave the good results, which demonstrate the performance of the recognition system. Table 3 illustrated some recognized words.

Text to be recognized	recognition Results
ΣΟΣΣ	ΣΟΣΣ
⊙ΣΣ.	⊙ΣΣ.
Σ⊙ΣΠ.Ι	Σ⊙ΣΠ.Ι
∴E.O	∴E.O
+X.	+G.
Π.Ι	Π.Ι

TABLE 3: Examples of some words used for the test of the recognition method.

Number of hidden layer	Recognition rates	Error rates	Computing time
1	93.52%	6.48%	21.23s
2	86.71%	13.29%	28.65s
3	87.34%	12.66%	36.21s

TABLE 4: Recognition rates, Error rates and Computing times

A close inspection of Table 4 show that the recognition rate using one hidden layer is higher than those obtained by tow hidden layers, but error rates and computing time using two hidden layers are less than to those obtained by one hidden layer. Table5 illustrates the misrecognised characters. These error are caused by noise or rotation. The method has been implemented in Matlab software on a core (TM) Duo CPU T5870 @ 2.00 GHz

Noise	Rotation
	 
	 
	 

TABLE 5: Misrecognised characters

## 7. CONCLUSION

The subject of the work developed in this paper is to achieve system recognition of Tifinaghe characters. This system consists of three phases applied on the input image: pre-processing features extraction and recognition. Pre-processing phase includes normalisation baseline skew correction and segmentation. The features extraction phase is used to compute the characters features using Walsh Transformation for the reasons of invariance to translation, rotation and scale change. In the recognition phase a multilayer neural network is used to classify characters. Experimental results showed that this method give good recognition rate in a final conclusion, neural network seems to be better than other techniques used for recognition

## 8. REFERENCES

[1] R. M. Bozinovic and S. N. Shihari, Off Line Cursive Script Word Recognition, IEEE Trans.Pattern Anal. Mach. Intell. PAMI 11, 1989, pp. 68- 83.

[2] M. K. Brown, pre-processing techniques for cursive word recognition, Pattern Recognition, Vol.13, N°5, pp: 447-451, 1983.

- [3] M. Fakir and C. Sodeyama, Recognition of Arabic printed Scripts by Dynamic Programing Matching Method, IECICE Trans. Inf & Syst, Vol. E76- D, No.2 Feb. [4]. M. FAKIR, Reconnaissance des Caractères Arabes Imprimés, Thesis, 2001, pp : 28-36, semlalia faculty of science Morocco, 2001.  
pp: 31-37,1993.
- [4] N.Mezghani A.Cheret N.Mitiche, Bayes classification of online arabic characters by Gibbs modeling of class conditional densities, IEEE Trans PAMI Vol 30, issue 7, pp: 1121-1131, july 2008.
- [5] Asiri, A. and M.S. Khorsheed, 2005. Automatic processing of handwritten Arabic forms using neural networks. Proceeding of the World Academy of Science, Engineering and Technology, Aug. 2005, pp: 313-317.
- [6] Sarhan A.M. and O. Helalat, 2007. Arabic character recognition using ann networks and statistical analysis. Proceeding of European and Mediterranean Conference on Information Systems, June 24-26, Polytechnic University of Valencia, pp: 1-9.
- [7] Alnsour, A.J. and L.M. Alzoubady, 2006. Arabichandwritten characters recognized by neocognitron artificial neural network. J. Pure Appl. Sci., 3: 1- 17.
- [8] Hadjar, K. and R. Ingold, 2003. Arabic newspaper segmentation. Proceeding of 7th International Conference on Document Analysis and Recognition, Aug. 3-6, IEEE Computer Society, pp: 895-899.
- [9] Hamza, Ali A. Back Propagation Neural Network Arabic Characters Classification Module Utilizing Microsoft Word; Journal of Computer Science 4 (9): 744-751, 2008.
- [10] Ch. Choisy and A. Belaid, Cross- learning in analytic word recognition without segmentation, in Int. Journal on documentAnal. & Recognition IJDAR, 4(4): 281-289, 2002.
- [11] Y.X. Gu et al, Application of a multilayer tree in computer recognition of Chinese character, IEEE Trans. On PAMI-5, N°.1, pp: 83-89, 1983.
- [12] Standardisation Amazighe (Actes du organisé par le centre de l'Aménagement Linguistique, Rabat, 8-9, décembre 2003
- [13] 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.
- [14] Y. Es saady, M.Amrouch, A. Rachidi, M. El Yassa and D. Mammass, Reconnaissance de caractères Amazighes Imprimés par le Formalisme des Automates à états finis, SITCAM'09, Agadir-Maroc.
- [15] M. Fakir, B. Bouikhalene and K. Moro, Skeletonization Methods Evaluation for the Recognition of PrintedTifinaghe characters, SITCAM'09, Agadir-Maroc.
- [16] M. Blumenstein & C. K. Cheng & X. Y. Liu, 2002, New Preprocessing Techniques for Handwritten Word Recognition, in Proceedings of the Second IASTED International Conference on Visualization, Imaging and Image Processing (VIIP 2002), ACTA Press, Calgary, pp. 480-484.
- [17] D. J. Burr, A Normalizing Transform For Cursive Script Recognition,Proc. 6th Int. J. Conf. on Pattern Recognition Munich (1982), pp. 1027–1030

- [18] A. Sehad, L. Mezai, M.T. Laskri, M. Cheriet, Détection de l'inclinaison des documents arabes imprimés.
- [19] Attila Fazekas and Andras Hajdu Recognizing Type set Documents using Walsh , JCIT-CIT 9, 2-2001, 101-112.
- [20] Ibrahim S. I. Abuhaiba, Arabic Font Recognition Using Decision Trees Built From Common Words, JCIT-CIT 13, 3-2005, 211-223.