Arabic Numerals Recognition based on an Improved Version of the Loci Characteristic

Ouafae EL Melhaoui
Labo LEPAS, FS, University Mohammed 1, Maroc

Mohamed El Hitmy
Labo LEPAS, FS, University Mohammed 1, Maroc

Fairouz Lekhal
Labo LEPAS, FS, University Mohammed 1, Maroc

ABSTRACT
The present work is concerned with handwritten and printed numeral recognition based on an improved version of the loci characteristic method (CL) for extracting the numeral features. After a preprocessing of the numeral image, the method divides the image into four equal parts and applies the traditional CL to each of the parts. The recognition rate obtained by this method is improved indicating that the numeral features extracted contain more details. Numeral recognition is carried out in this work through k nearest neighbors and multilayer perceptron techniques.

Keywords
Handwritten and printed numeral recognition, loci characteristic, features extraction, preprocessing, k nearest neighbors, multilayer perceptron.

1. INTRODUCTION
For the last three decades, the recognition of handwritten and printed letters has played a crucial and highly beneficial role in tedious and hard tasks such as postal sorting, bank check reading, order form processing, robotics etc [9].
Several numerals recognition schemes have been proposed in the literature [9, 11]. They are all concerned with finding ways of how to differentiate between the different numerals. They are also interested in finding techniques for better numerals classification. Methods proposed in this context include dynamic programming, hidden Markov modeling, neural networks, support vector machines, k nearest neighbors, etc [9, 11].
Numeral features extraction is a delicate process and is crucial for a good numeral recognition. It consists of transforming the image into an attribute vector which contains a set of characteristics making the numeral recognition an easy task. The classifier performance is highly correlated with a pertinent choice of the attribute vector. The classification results may be very poor if the attributes describing the numerals are not as discriminate as they should be. The most representative attributes for numerals must fulfill requirements such as, small intra class and high inter class variances, small number of attributes, scale, and translation and rotation invariants. Previous works have used invariant moments [4] Zernike moments [15], freeman coding [3], horizontal, vertical and different orientation profiles, Fourier and Gabor descriptors [6, 10], water reservoir [9]…
Today numeral features extraction is still an open research work. The big variability of handwriting, the different styles and fonts used for printed numerals and the conditions of the writers, make the recognition process a complex task. We have used in this work a database of 600 numerals, printed and handwritten, provided by various categories of writers.
In section 2 numeral features extraction method is presented. The k nearest neighbors and the multilayer perceptron techniques of classification are discussed in section 3. The proposed system is presented in section 4, section 5 gives experimental results and discusses ways for reducing the attribute vector size. Finally, we give a conclusion in section 6.

2. NUMERAL FEATURES EXTRACTION
Loci characteristic method has initially been proposed by Gluksman in 1967 [1, 2, 7]. The method has been widely used because of its many advantages. It is insensitive to small perturbations, robust to variations in styles and fonts, and has obtained good results in recognizing handwritten numerals [7].
The method is based on the pixels background rather than the numeral pixels themselves. The loci characteristics are generally defined on horizontal and vertical directions. They are computed with respect to the number of white to black transitions in the right, down, left and up directions for each background pixel of the image. For each background pixel, a number of four digits is obtained, called a loci number (figure.1)
A scanning of all the image background pixels is carried out, an attribute vector element is constituted by the total number of background pixels having the same loci number [5, 8].
We begin this approach by cropping the numeral from its background, which only leaves the interesting part of the image.
Traditional CL limits for each background pixel, the number of transitions to two, the size of the attribute vector is 3^4=81, and the numeral recognition rate is generally relatively small. Other studies have extended the number of transitions to three, this has obtained more details for the CL vector and the size of the attribute vector has however increased to 256. In other studies, along with three transitions considered for the pixel, four pixel background borders around the numeral image have been used (figure.2). With these modifications, improved recognition rates as compared to the traditional CL were obtained.

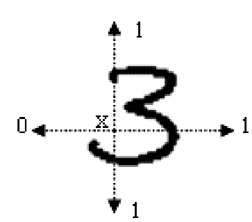

Fig 1: Loci number for background pixel's 'x', the pixel code is (1,1,0,1)

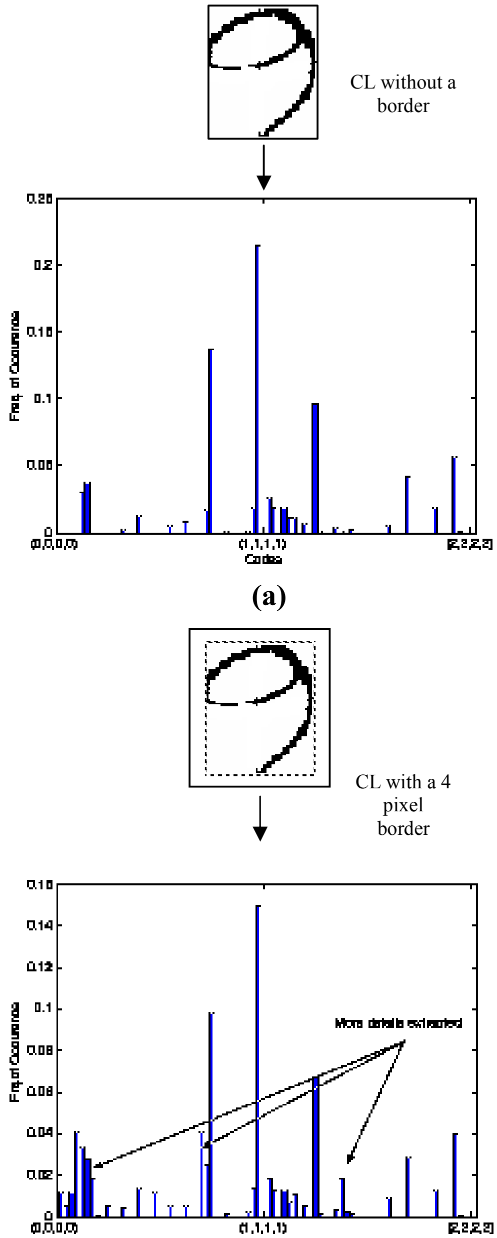

Fig 2: The characteristic loci obtained: (a) without a border. (b) with 4 pixels border.

3. RECOGNITION METHODS

3.1. K nearest neighbors (*KNN*)

K nearest neighbors is a widely used method for data classification. Proposed in 1967 by Cover et al. [12], it has been widely used in handwritten numerals recognition [9] for its conceptual simplicity, theoretical elegance and its robustness [14]. *KNN* is a method which was inspired from the closest neighbor rule, it is based on the hypothesis that only the closest neighbor has most influence on the classification decision approach. *KNN* is based on computing the distance between the test data and the different learning data samples [9] and then attribute the test sample to the k nearest neighbors.

This method is apparently simple and easy to program when it is compared to neural network and support vector machine methods. It requires however a large amount of memory to store and examine the data. Since several decades, researches have tried to improve the method, by reducing the size of data in the hope of reducing the processing time (condensing). Data to be discarded concerns the outliers and biased data and also if possible to remove overlapping between classes which give rise to a more sure decision for data classification task (editing) [13].

3.2. Multilayer Perceptron (*MLP*)

3.2.1. Network structure

Multilayer networks are defined as mathematical models corresponding to a very much simplified version of the biological reality. They have been inspired from a long study of human brain and how the human neurons function [14].

Artificial neural networks in general and *MLP*s in particular have been widely used for data classification, pattern recognition, prediction, signal processing etc…

MLPs are forward propagation networks where the two closest layers are fully connected. The structure of an *MLP* is an input, an output and a certain number of hidden layers. We have been limited, in this work to one hidden layer, this have shown to be sufficient after many simulations carried out (figure3). The number of neurons in the hidden layer has generally been determined heuristically or by trial and error [17, 21, 22]. The input layer has n_1 neurons that we call ne_k, $1 \le k \le n_1$, the hidden and output layers contain n_2 and n_3 neurons that we respectively call nc_j and ns_i, where $1 \le j \le n_2$, $1 \le i \le n_3$.

$z_k^{(1)}$ is the realization of the attribute vector component for a numeral image *I*, with $k = 1,2,\ldots n_1$.

$W^{(1)} = \left\{ w_{jk}^{(1)} \right\}$, $j = 1 \ldots n_2$, $k = 1 \ldots n_1$, $w_{jk}^{(1)}$ are the synaptic weights connecting neurons in the input to the neurons in the hidden layers.

$W^{(2)} = \left\{ w_{ij}^{(2)} \right\}$, $i = 1 \ldots n_3$, $j = 1 \ldots n_2$, $w_{ij}^{(2)}$ are the synaptic weights connecting neurons in the hidden to the neurons in the output layers. The output of the neuron *j*, nc_j of the hidden layer is:

$$z_j^{(2)} = f\left(y_j^{(2)}\right) \tag{1}$$

where

$$y_j^{(2)} = \sum_{k=1}^{n_1} w_{jk}^{(1)} z_k^{(1)} \tag{2}$$

and

$$f\left(y_j^{(2)}\right) = \frac{1}{1 + e^{-y_j^{(2)}}} \tag{3}$$

for $j = 1,2,\ldots,n_2$

f is the activation function which we choose to be of sigmoid type.

The output of the neuron *i* , ns_i of the output layer is:

$$z_i^{(3)} = f\left(y_i^{(3)}\right) \tag{4}$$

Where $$y_i^{(3)} = \sum_{j=1}^{n_2} w_{ij}^{(2)} z_j^{(2)} \quad (5)$$

for $i = 1,2,\ldots,n_3$.

Notice that the superscripts (1), (2) and (3) represent respectively input, hidden and output layers.

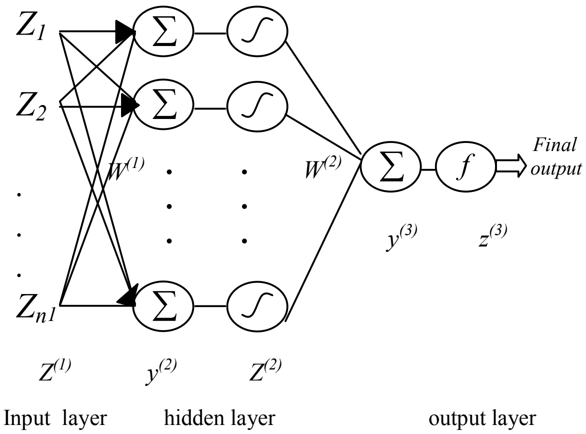

Fig3: Structure of a multilayer perceptron neural network.

The *MLP* is characterized by its ability to learn and gradually improve its performance through a learning process [18]. Learning is a phase where the behavior of the network is modified by modifying the synaptic weights until a desired output pattern is obtained. The type of learning is determined by the way how the network synaptic weights are updated. There are three principal learning types, gradient backpropagation algorithm, hebb rules and competitive learning.

In this work, we have used the gradient backpropagation algorithm, the objective is to minimize the squared error between the desired and computed output of the *MLP*. Figure 4 shows the different steps of the gradient backpropagation algorithm.

1. Randomly initialize the synaptic weights between -1 and 1.
2. Randomly apply a realization vector of an object to the input layer and its corresponding known output to the output layer.
3. Compute the network output and error E between computed and desired outputs.
4. Adjust the weights by the gradient method :

   $$W^{(r)}(t+1) = W(t)^{(r)} - \eta \frac{\partial E}{\partial W^{(r)}}$$

   η is the learning rate, which is in general a value between 0.1 and 0.9. r = 1, 2.
5. Go to 2 as long as the network does not show satisfactory performances.

Fig 4: gradient backpropagation algorithm

Algorithm convergence is achieved when the synaptic weights remain almost steady and the computed outputs are close enough to the desired ones.

- Optimization of the $w_{ij}^{(2)}$:

In order to be able to update $w_{ij}^{(2)}$ synaptic weights for different indices i and j, consider the network error to be:

$$E(t) = \frac{1}{2}\sum_{i=1}^{n3} (z_i^{(3)} - d_i(t))^2 \quad (6)$$

where $d_i(t)$ is the known desired output at neuron ns_i. t represents the current iteration. Differentiating $E(t)$ with respect to $w_{ij}^{(2)}$ gives:

$$\frac{\partial E(t)}{\partial w_{ij}^{(2)}} = (z_i^{(3)}(t) - d_i(t)) \frac{\partial z_i^{(3)}}{\partial w_{ij}^{(2)}} \quad (7)$$

Where $$\frac{\partial z_i^{(3)}}{\partial w_{ij}^{(2)}} = z_j^{(2)} \times f(y_i^{(3)}) \times (1 - f(y_i^{(3)})) \quad (8)$$

and $$\frac{\partial y_i^{(3)}}{\partial w_{ij}^{(2)}} = z_j^{(2)} \quad (9)$$

And $$\frac{\partial f(y_i^{(3)})}{\partial y_i^{(3)}} = f(y_i^{(3)}) \times (1 - f(y_i^{(3)})) \quad (10)$$

The synaptic weight $w_{ij}^{(2)}$ may now be updated.

$$w_{ij}^{(2)}(t+1) = w_{ij}^{(2)}(t) - \eta \delta_i^{(3)}(t) z_j^{(2)}(t) \quad (11)$$

for $i = 1 \ldots n_3$ and $j = 1 \ldots n_2$

where

$$\delta_i^{(3)}(t) = (z_i^{(3)}(t) - d_i(t)) f(y_i^{(3)}) \times (1 - f(y_i^{(3)})) \quad (12)$$

- Optimization of the $w_{jk}^{(1)}$

The derivation in this case is similar to that presented previously, so we go to the updating of the weights according to:

$$w_{jk}^{(1)}(t+1) = w_{jk}^{(1)}(t) - \eta \delta_j^{(2)}(t) z_k^{(1)}(t) \quad (13)$$

for $j = 1 \ldots n_2$ and $k = 1 \ldots n_1$

where

$$\delta_j^{(2)}(t) = \sum_{i=1}^{n_3} \delta_i^{(3)}(t) w_{ij}^{(2)}(t) \frac{\partial f(y_j^{(2)}(t))}{\partial y_j^{(2)}(t)} \quad (14)$$

3.2.2. Determination of number of neurons in the hidden layer.

The recognition performance of backpropagation network will highly depend on the structure of the network and training algorithm [16], especially number of neurons in the hidden layer. The learning rate used for the backpropagation algorithm is 0.4.

The attribute vector obtained is composed of four vectors of size 81 each, i.e, an attribute vector of size 324. The structure of the *MLP* used is, an input layer with 324 neurons, an output layer with 10 neurons where each of them represents a numeral from 0 to 9. Therefore only the number of neurons in the hidden layer needs to be determined.

Several experiments were carried out to determine this number that obtains the best performance of the network, table 1 illustrates the influence of this number on the recognition rate.

Table 1: number of neurons in the hidden layer with respect to the recognition rate of numerals

Number of neurons in the hidden layer	Recognition rate (%)
15	98.5
20	**99**
30	98
40	98
80	98
100 . . 140	98 . . 97.5

Table 1 highlights the fact that a number of 20 neurons in the hidden layer obtains good recognition rate performance of isolated numerals. The number 20 was also adopted in other studies published in the literature [5, 19, 20].

4. PROPOSED SYSTEM

The system proposed contains three main steps, preprocessing, features extraction and classification. The full system is shown in figure 5.

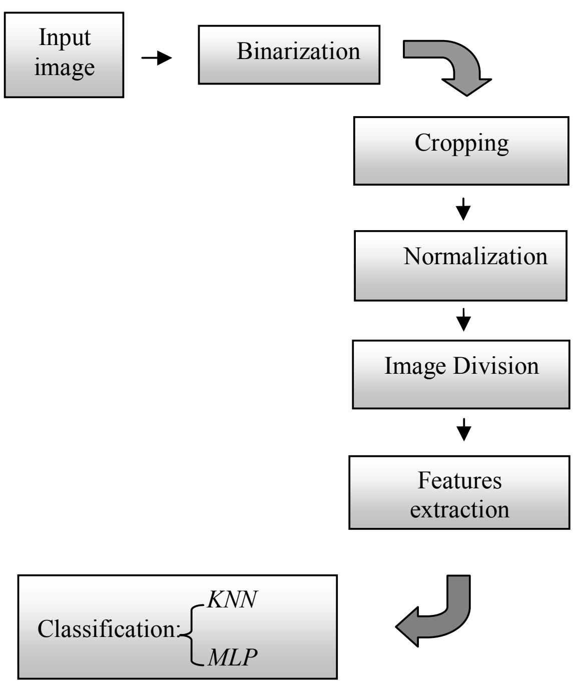

Fig 5: Proposed scheme for numerals recognition.

The preprocessing stage consists of initially binarizing the numeral input image which is presented in grey level, setting out a threshold value against which the image pixels are compared, if the pixel value is greater than the threshold, it is set to one, it is set to zero otherwise.

The next stage is to only preserve the numeral position in image by cropping it (figure 6). Final stage is to carry a transformation on the cropped numeral image in order that it gets a fixed size chosen here to be 40×40 pixels. The numeral image is then divided into four equal parts of size 20×20. We add two lines to each side of the subimage making its size to be 24×24 instead of 20×20. This will make it possible to account for the numeral pixels positioned in the subimage edge. We apply the traditional CL with two transitions for each part of the image. KNN and *MLP* methods are used for classification task.

Figure 6 shows the extraction of just the useful zone, figure 7 shows an image divided into four parts of equal size.

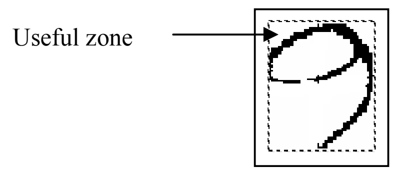

Fig 6: Useful zone of number 9.

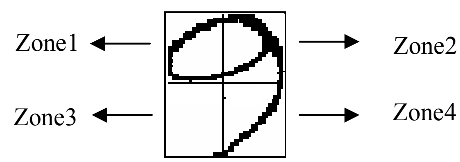

Fig 7: Image division into 4 zones.

5. EXPERIMENTAL RESULTS AND COMPARATIVE STUDY.

5.1. Experimental results.

We have used a database formed of 600 Arabic numerals, both printed and handwritten by different writers. A sample of the database is shown in figure 8. The database is divided into two sets, one set of 400 numerals is used for learning and the remaining 200 numerals are used for the test stage. The classes are equipropables.

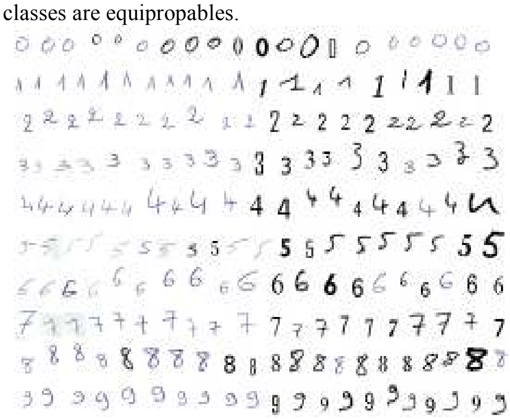

Fig 8: A sample of handwritten and printed Arabic numerals.

Table 2 shows the recognition rate for each numeral from 0 to 9 achieved by the two methods, *kNN* and *MLP*

Table 2: Arabic numeral recognition by *kNN* and *MLP*.

Learning database = 400, test database = 200, attributes number = 324				
Arabic numerals	**Correctly classified samples in the test set by *kNN***	**Recognition rate by *kNN* (%)**	**Correctly classified samples in the test set by *MLP***	**Recognition rate by *MLP* (%)**
0	20	100	20	100
1	19	95	20	100
2	20	100	20	100
3	20	100	19	95
4	19	95	19	95
5	20	100	20	100
6	19	95	20	100
7	20	100	20	100
8	20	100	20	100
9	17	85	20	100
Recognition average rate (%)	97		99	

Table 2 shows that the average recognition rate is high for the two methods, despite that the *MLP* outperforms the *kNN*.

5.2. Comparative study

Table 3 shows the recognition rate results of Arabic numerals, using the proposed system, the traditional *CL* and CL with extension already proposed in the literature [5]. The CL with extension considers three transitions and a 4 pixels border added after cropping.

Table 3 shows that the results obtained by the proposed methods are good, especially with the *MLP* where the recognition rate achieved is 99%. Dividing the image into four equal parts has provided more details for the attribute vector which has become more discriminate and has been able to differentiate between nearly similar numerals such as 3, 8 and 5, 6, 9. A well designed *MLP* algorithm is also crucial for achieving good results.

Analyzing the attribute vectors obtained for different numerals in our database, we noticed that there are some attributes which have zero for their loci values such as (2,2,2,2); (2,2,2,1); (2,2,2,0); (2,2,1,2); (2,2,1,1); …etc and have therefore no effect on system performances. They represent redundant information and may simply be discarded from the attribute vector. Discarding these attribute elements from the attribute vector has reduced its size to 198 while it was initially 324. The recognition rate results obtained by the new attribute vector are exactly the same as those obtained in table 3.

Table 3: Recognition rates using different features extraction methods by *kNN* and *MLP*.

Features extraction methods	**Recognition average rate by *kNN* (%)**	**Recognition average rate by *MLP* (%)**
Traditionnel CL	90	93
CL extended, proposed in the literature	90.5	95,5
proposed system	**97**	**99**

6. CONCLUSION

In this work we have presented a new system for recognizing Arabic numerals, using the characteristics loci (CL). The image representing the numeral is initially preprocessed; it is binarized, cropped, brought to a fixed size, enlarged by two pixels added on each of its sides and divided into four equal parts. Traditional CL is applied to each part of the image. This process provides more information on the numeral and reduces the recognition error rate. The proposed technique was compared with some CL based methods published in the literature, using the k nearest neighbors and multilayer perceptron classifiers. The simulations have obtained good results; a recognition rate of 99 % was achieved with the proposed method and the MLP classifier despite the different styles, fonts, scripts and writers.

7. REFERENCES

[1] Reza Ebrahimpour, Mohammad R. Moradian, Alireza Esmkhani,Farzad M. Jafarlou . 2009. Recognition of Persian handwritten digits using Characterization Loci and Mixture of Experts. International Journal of Digital Content Technology and its Applications. Volume 3, Number 3.

[2] H.Glusksmen. 1967. Classification of mixed-font alphabetic by characteristic loci. Digest of 1st Annual IEEE Comp. Conf, pp.138-141.

[3] Dupré, X. 2003. Contributions à la reconnaissance de l'écriture cursive à l'aide de modèles de Markov cachés. Thèse, Université René Descartes - Paris.

[4] Hamid Reza Boveiri, 2010. Persian Printed Numerals Classification Using Extended Moment Invariants. World Academy of Science, Engineering and Technology 63.

[5] Hatem M.R. Abou-zeid, 2003. Computer Recognition of Unconstrained Handwritten Numerals. The 46th IEEE International Midwest Symposium on Circuits and Systems. Cairo-Egypt.

[6] Farès Menasri, Nicole Vincent, Emmanuel Augustin. "Reconnaissance de chiffres farsi isolés par réseau de

neurones à Convolutions". Actes du dixième Colloque International Francophone sur l'Écrit et le Document.

[7] Reza Ebrahimpour, and Samaneh Hamedi. 2009. Handwritten Digit Recognition by Multiple Classifier Fusion based on Decision Templates Approach. World Academy of Science, Engineering and Technology 57.

[8] M. Soryani, and N. Rafat. 2007. Application of Genetic Algorithms to Feature Subset Selection in a Farsi OCR. International Journal of Engineering and Applied Sciences .

[9] Benne R.G.1, Dhandra B.V.1 and Mallikarjun Hangarge. 2009. Tri-scripts handwritten numeral recognition: a novel approach". Advances in Computational Research, ISSN: 0975–3273, Volume 1, Issue 2, pp: 47-51.

[10] G. G. Rajput, S. M. Mali. 2010. Marathi Handwritten Numeral Recognition using Fourier Descriptors and Normalized Chain Code. IJCA Special Issue on Recent Trends in Image Processing and Pattern Recognition" RTIPPR.

[11] Farès Menasri. 2008. Contributions à la reconnaissance de l'écriture arabe manuscrite. Thèse.

[12] ROBERTO GIL-PITA, XIN YAO. 2008. EVOLVING EDITED *k*-NEAREST NEIGHBOR CLASSIFIERS. International Journal of Neural Systems, Vol. 18, No. 6, pp: 459–467.

[13] Vicente Cerverón and Francesc J. Ferri. 2001. Another Move toward the Minimum Consistent Subset: A Tabu Search Approach to the Condensed Nearest Neighbor Rule". IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, Volume. 31, N°. 3.

[14] Ludmila I. Kuncheva. 1995. Editing for the k-nearest neighbors rule by a genetic algorithm. Elsevier, Pattern Recognition Letters 16, pp: 809-814.

[15] K. Faez, E. Khotanzad, and M. H. Shiralishahreza, 1995. Recognition of Persian Handwritten Letters and Digits Using Pseudo-Zernike Moments and Neural Networks. in Proc. 3rd Iranian Conf. on Electrical Engineering, Tehran, pp. 231-240.

[16] DERDOUR Khedidja. 2009. Reconnaissance de formes du chiffre arabe imprimé : Application au code à barre d'un produit. Mémoire.

[17] Rajashekararadhya, P. Vanaja Ranjan. 2009. Handwritten numeral recognition of Canada script. Proceeding of the international workshop on machine intelligence research labs.

[18] Marc Parizeau, 2004. Réseaux de neurones. Livre

[19] Harifi, A.Aghagolzadeh. 2004. A New Pattern for Handwritten Persian/Arabic, Digit Recognition. International Journal of Information Technology. Vol 1, N° 4.

[20] Alberto Sanfeliu, José Francisco Martínez Trinidad, Jesús Ariel Carrasco Ochoa. 2004. Progress in pattern recognition, image analysis and applications. 9th Iberoamerican Congress on Pattern Recognition, CIARP, Puebla, Mexico. Proceedings. Lecture Notes in Computer Science 3287 Springer 2004, ISBN 3-540-23527-2

[21] J.Pradeep, E.Srinivasan, S.Himavathi. 2010. Diagonal Feature Extraction Based Handwritten Character System Using Neural Network. International Journal of Computer Applications (0975 – 8887). Vol 8, N°.9, pp: 17-22.

[22] Jiawei Han, Micheline Kamber. "Data mining: concepts and techniques". livre