Combining Neural Networks for Arabic Handwriting Recognition

Chergui Leila[1], Kef Maamar[2], and Chikhi Salim[3]
[1]Department of Computer Sciences, University Larbi Ben Mhidi, Algeria
[2]Department of Computer Sciences, University Hadj Lakhdar, Algeria
[3]Department of Computer Sciences, University Mentouri, Algeria

Abstract: *Combining classifiers is an approach that has been shown to be useful on numerous occasions when striving for further improvement over the performance of individual classifiers. In this paper we present a Multiple Classifier System (MCS) for off-line Arabic handwriting recognition. The MCS combines three neuronal recognition systems based on Fuzzy ART network used for the first time in Arabic OCR, multi layer perceptron and radial basic functions. We use various feature sets based on Tchebichef, Hu and Zernike moments. For deriving the final decision, different combining schemes are applied. The best combination ensemble has a recognition rate of 90,10 %, which is significantly higher than the 84,31% achieved by the best individual classifier. To demonstrate the high performance of the classification system, the results are compared with three research using IFN/ENIT database.*

Keywords: *MCS, Arabic recognition, neural networks, tchebichef moments, hu moments, and Zernike moments.*

Received December 20, 2010; accepted March 1, 2011

1. Introduction

The research domain of Multiple Classifier Systems (MCS) examines how several classifiers can be applied together to obtain better classification systems. MCS methods may be used to increase the speed of the system or to reduce the time taken for the design of the classification system. There are three basic structures of classifier combination: serial, parallel and conditional. In a serial combination the output of a classifier is the input of the next classifier. In conditional combination some classifiers are only applied if the output of other classifiers meets a certain condition.

In a parallel combination the output of all classifiers is combined in a last step to a single output. This combination is very popular, because each classifier that solves the classification problem may be used. There are many ways to combine in parallel the results of a set of classifiers, depending on the type of the classifiers' output. If the output is only the best ranked class then majority voting can be applied [10]. More sophisticated voting schemes also look at the probability of the classification error for a specific class (Bayesian Combination Rule [13]) and the dependencies between the classifiers (Behavior-Knowledge Space [4]). Some classifiers have a ranked list of classes as output. For them often Borda count [12] or related methods are used. Some classifiers also generate a score value for each class. In this case the sum, product, maximum, minimum, or the median of the scores of all classifiers can be calculated [12] and the class with the highest value is regarded as the combined result. Another approach is to use the score values output by the individual classifiers as input for a trainable classifier, e.g., a neural-network [4], which acts as the combiner.

Neural network classifiers exhibit powerful discriminative properties and they have been used in Optical Character Recognition implementation. MLP is usually a common choice because they are very easy to train and very fast to use in classification decision process. Radial Basis Function (RBF) networks, on the other hand, offer better features for recognition applications, in addition to highly efficient training, no design details are required since the network automatically adjusts its single hidden layer. The Fuzzy ART architecture exhibits considerable stability and can easily integrates with other hierarchical theories of recognition.

In this case study, our research aims at developing an automatic recognition system for off-line Arabic handwritten words using a parallel combination scheme between a Fuzzy ART, MLP and RBF networks, we focus on three combining methods; majority vote, max-rule, and sum-rule. We also applied three types of moments as features.

This paper is organized as follows: section 2 briefly reviews the Arabic script characteristics. In section 3 related works concerning MCS implemented for Arabic script are given. Used neural networks are detailed in section 4. In section 5 we introduce the three types of moments chosen as futures for our application. Section 6 reports the experimental results conducted on the publicly available IFN/ENIT database of handwritten Tunisian town names including system parameters and architecture, combining methods and comparison with other systems. Finally, we conclude the work and discuss the future directions in the last section.

2. Arabic Script

Arabic, one of the six United Nations official languages, is the mother tongue of more than 300 million people, The Arabic script evolved from the Nabataean Aramaic script. It has been used since the 4th century AD the most features of Arabic language are:

- The Arabic script is cursive and it is written from right to left [20].
- As illustrated in Figure 1, the Arabic alphabet contains 28 letters, and most letters change form depending on whether they appear at the beginning, middle, or end of a word, or on their own.

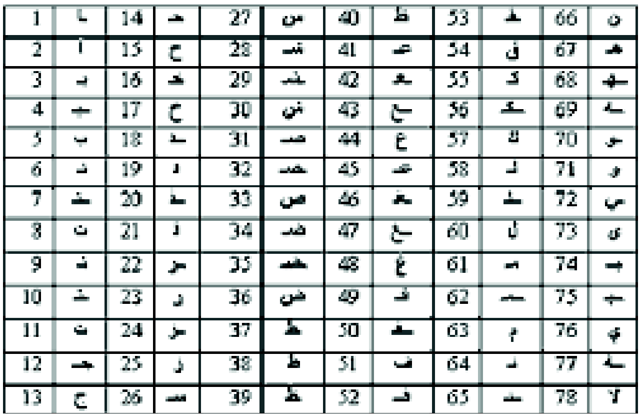


Figure 1. Arabic alphabet.

- The long vowels /a:/, /i:/ and /u:/ are represented by the letters *'alif*, *yā'* and *wāw*, respectively.
- The Arabic characters of a word are connected along a baseline. A baseline is the line with the highest density of black pixels.
- In addition to connectivity, vowel diacritic signs are an essential part of written Arabic. Vowels are represented in the form of overscores or underscores
- Many Arabic characters have dots, which are positioned above or below the letter body. Dots can be single, double or triple. Different Arabic letters can have the same body and differ in the number of dots identifying them.
- Characters in an Arabic script usually overlap with their neighbouring letters depending on their position in the word. The degree of overlapping varies depending on the size and style of the character.
- Certain character combinations form new ligature shapes which are often font dependent.
- Spacing may separate not only words but also certain characters within a word forming sub-words.
- Some characters contain closed loops.

3. Related Works

Compared to Latin script where a lot of research work is done, the number of work for MCS of Arabic script is quite limited. One of the first works in this field was given by Farah *et al*. [7] who introduced a system based on the combination of three Multi Layer Perceptrons for the recognition of Arabic literal amount with a recognition rate of 94% on a small test database containing 4800 words. El-Hajj *et al*. [6] have used Neural Networks to combine three homogeneous HMM-based classifiers, which have different features as input, they used the IFN/ENIT database achieving a recognition rate of 94,44%. In [15], a strategy for Arabic handwritten word recognition has been proposed by Miled, the idea is based on a sequential hierarchical cooperation of three classifiers, all of a Markovian type. The first classifier is based on a global description of the word using sequential visual indices. The second classifier is associated with an analytic approach that models the characters deprived of their diacritic dots. The third classifier is associated to the sub-word. Different types of combinations methods were tested (measure, rank, hierarchy, class). The rates of recognition of the system exceed 89%. This represents an increase of about 8% with respect to the best performing classifier taken individually. Alma'adeed [14] combined a rule based recognizer with a set of HMMs to recognize words in a bank-check lexicon of 47 words. The rule-based engine used ascenders, descenders, and other structural features to separate the data into groups of words, and an HMM classifier for each group used frame-based features to determine the word. The system was tested on about 4700 words collectively written by 100 writers, excluding about 10% of the words due to errors in baseline detection and pre-processing. A near 60% recognition rate was achieved.

Souici-Meslati and Sellami [14] presented a hybrid approach to the recognition of literal amounts on bank-checks. Three classifiers ran in parallel: neural networks, K-nearest-neighbour, and Fuzzy K-nearest-neighbour. The outputs were combined by word-level score summation. 1200 word by 100 writers were used for training and 3600 words for testing. The recognition rate was 96%. In [2] Burrow applied one K-Nearest Neighbours classification approach to each sub-word, a majority vote is taken on its overall class and repeated for each sub-word sample. First results result in 47% accuracy. By refining the scoring system and adding some features, including the number of dots, the author scores at 74% for sub-word on correctly represented classes.

4. Neural Network

Neural networks have been a much-publicized topic of research in recent years and are now beginning to be used in a wide range of subject areas. One of the strands of interest in neural networks is to explore possible models of biological computation. Neural network structure is a collection of parallel processors connected together in the form of a directed graph, organized such that the network structure lends itself to the problem being considered, they are specified by the net topology, node characteristics, and training or learning rules. In our system we have implemented three kind of neural network.

4.1. Adaptive Resonance Theory

Adaptive Resonance Theory (ART) is a family of algorithms for incremental unsupervised learning developed by Carpenter and Grossberg [9]. The ART neural network is a typical representative of competitive networks. The Fuzzy ART neural network has two advantages; one is the ability to handle both binary and analogy vectors and the other is faster implementation.

Fuzzy ART is a generalized ART-1 method which is restricted to continuous binary data in the interval of [0, 1]. It is similar to many iterative clustering algorithms where each pattern is processed by finding the nearest cluster and then updating that cluster to be closer to the pattern. However, in the framework of Fuzzy ART is a little changed by introducing the concept of resonance so that each case is processed by first finding the nearest cluster seed that resonates with the case, and then updating that cluster seed. Resonance is just a matter of being within a certain threshold of a second similarity measure. Fuzzy ART takes three input parameters: choice parameter ($\beta>0$), vigilance parameter ($0\leq p\leq 1$) and learning rate ($0\leq \lambda\leq 1$). The detail of training Fuzzy ART algorithm is given in [3].

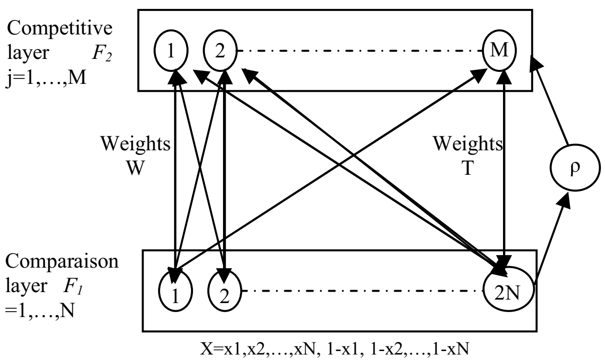


Figure 2. Fuzzy ART network.

4.2. Radial Basis Function

RBF is a combination of unsupervised and supervised learning. The basis function is radial and symmetric around the mean vector, which is the centroid of the clusters formed in the unsupervised learning stage, hence the name radial basis function. The RBF networks are two-layer networks in which the first layer nodes represent radial functions (usually Gaussian). The second layer weights are used to combine linearly the individual radial functions, and the weights are adapted via a linear least squares algorithm during the training by supervised learning. The activation function of output nodes can be either linear or sigmoidal; to facilitate minimum error training by gradient descent and make the output values approximate a posteriori probabilities, we use sigmoid functions in the output layer. For an RBF network with d input signals, m hidden units, and M output units, assume spherical Gaussian functions for the hidden nodes:

$$\varphi_j(x) = exp\left(-\frac{\|x - \mu_j\|^2}{2\sigma_j^2}\right), j = 1, \dots, m \quad (1)$$

where μ_j is the center Gaussian, and σ_j^2 is the variance. The linear combination of Gaussian functions:

$$\begin{aligned} y_k(x) &= \upsilon_k(x) = \sum_{j=1}^{m} w_{kj}\phi_j(x) + w_{k0} \\ &= \sum_{j=0}^{m} w_{kj}\phi_j(x), \qquad k = 1, \dots, M \end{aligned} \quad (2)$$

Under supervised training of all parameters, the RBF network can achieve a higher classification performance with much fewer hidden nodes. This can compete or even exceed the Multi-Layer Perceptron. The training algorithm is described in [4].

4.3. Multi Layer Perceptron

MLP networks are widely used in handwritten recognition systems because they are very easy to train and very fast to use in classification decision processes. This popularity is related to the use of the gradient back-propagation algorithm [9] in the training process. MLPs generally achieve good performance in terms of correct recognition rate in handwritten classification. When adopting MLPs, the word image is pre-processed so as to yield a feature vector, which is used for feeding the neural network. The class membership is commonly given by exclusive coding of the output (one-hot coding). During the recognition phase, the network is fed with the input pattern which is propagated through forward steps to the outputs. The expected class is simply given by the output unit with the highest value. The classifier can reject patterns whose membership cannot be clearly established.

A typical classification criterion which is used consists of rejecting a pattern if:

$$\bar{y} = \max_{i=1,\dots,n}\{y_i\} < RM \quad (3)$$

where n is the number of classes, $y_i \in (0,1)$ is the i[th] output of the network, and RM is a proper threshold. Unfortunately, there are limits when using MLPs in classification tasks: First, there is no theoretic relationship between the MLP structure (ex: hidden layers number and nodes number per layer) and the classification task. The second limitation is due to the fact that MLP derives separating hyper plane surfaces, in feature representation space, which are not optimal in terms of the margin area between the examples of two-different classes. To train the MLP network we have used Back Propagation algorithm (BP) [9]. This algorithm performs a gradient descent in the connection weight space on an error surface defined by:

$$E = \frac{1}{p}\sum_{p} E_p \tag{4}$$

Where:

$$E_p = \frac{1}{2}\sum_{K} (t_{pk} - y_{pk})^2 \tag{5}$$

here P is the total number of patterns in the training set and, $\{t_{pk}\}, \{y_{pk}\}$ are respectively, the target and output vectors corresponding to the p-th input pattern. The quantity E is called the system error. In BP algorithm, weight updating rules are given by:

$$w_{jk}(t+1) = w_{jk}(t) - \eta\frac{\partial E_p(t)}{\partial w_{jk}(t)} + \alpha\Delta w_{jk}(t-1) \tag{6}$$

$$w_{ij}(t+1) = w_{ij}(t) - \eta\frac{\partial E_p(t)}{\partial w_{ij}(t)} + \alpha\Delta w_{ij}(t-1) \tag{7}$$

where $w_{jk}(t)$ is the weight connecting a hidden node j with an output node k while $w_{ij}(t)$ is the weight connecting an input node i with a hidden node j at time t. $\Delta w_{jk}(t-1)$ is the modification amount to w_{jk} at time t-1. $\eta(>0)$ and $\alpha(0<\alpha<1)$ are respectively called the learning rate and moment factor.

5. Invariant Moments

Moments and functions of moments have been employed as pattern features in numerous applications to recognize two-dimensional image patterns. These pattern features extract global properties of the image such as the shape area, the center of the mass, the moment of inertia, and so on. Moments used in this application are:

5.1. Hu moments

In our implementation, moment invariants used by Hu [19] have been utilized to build the feature space for MLP classifier. Using nonlinear combinations of geometric moments, Hu derived a set of invariant moments which has the desirable property of being invariant under image translation, scaling and rotation. The central moments, which are invariant under any translation, are defined as:

$$M_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x-\bar{x})^p(y-\bar{y})^q f(x,y)dxdy \tag{8}$$

Where:

$$\bar{x} = \frac{\overline{M}_{10}}{\overline{M}_{00}}, \quad \bar{y} = \frac{\overline{M}_{01}}{\overline{M}_{00}} \quad and$$
$$\overline{M}_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}x^p y^q f(x,y)dxdy \tag{9}$$

however, for images, the continuous image intensity function f(x, y) is replaced by a matrix, where x and y are the discrete locations of the image pixels. The integrals in equations 8 and 9 are approximated by the summations:

$$M_{pq} = \sum_{x=0}^{m}\sum_{y=0}^{n}(x-\bar{x})^p(y-\bar{y})^q f(x,y)dxdy \tag{10}$$

$$\overline{M}_{pq} = \sum_{x=0}^{m}\sum_{y=0}^{n}x^p y^q f(x,y)dxdy \tag{11}$$

where m and n are dimensions of the image. The set of moment invariants that has been used by Hu are given by:

$$\phi_1 = M_{20} + M_{02} \tag{12}$$

$$\phi_2 = (M_{20} - M_{02}) + 4M_{11}^2 \tag{13}$$

$$\phi_3 = (M_{30} - 3M_{12})^2 + (3M_{21} - M_{03})^2 \tag{14}$$

$$\phi_4 = (M_{30} + M_{12})^2 + (M_{21} + M_{03})^2 \tag{15}$$

$$\begin{aligned}\varphi_5 &= (M_{30}-3M_{12})(M_{30}+M_{12})\\ &[(M_{30}+M_{12})^2 - 3(M_{21}+M_{03})^2] + (3M_{12}-M_{03})\\ &(M_{21}+M_{03})[3(M_{30}+M_{12})^2 - (M_{21}+M_{03})^2]\end{aligned} \tag{16}$$

$$\begin{aligned}\varphi_6 &= (M_{20}-M_{02})[(M_{30}+M_{12})^2 - (M_{21}+M_{03})^2] +\\ &4M_{11}(M_{30}+M_{12})(M_{21}+M_{03})\end{aligned} \tag{17}$$

$$\begin{aligned}\varphi_7 &= (3M_{21}-M_{03})(M_{30}+M_{12})[(M_{30}+M_{12})^2 -\\ &3(M_{21}+M_{03})^2] + 3(M_{21}-M_{03})(M_{21}+M_{03}) *\\ &[3(M_{30}+M_{12})^2 - (M_{21}+M_{03})^2]\end{aligned} \tag{18}$$

These functions can be normalized to make them invariant under a scale change by using the normalized central moments instead of the central moments. The normalized central moments are defined by:

$$M_{pq} = \frac{M_{pq}}{M_{00}^a} \; where \quad a = \frac{(p+q)}{2} + 1 \tag{19}$$

These, when substituted into the above equations, will give seven moments which are invariant to translation, scale change and rotation.

The $\o_s$ have large dynamic values. Thus, it was found that it was more practical to deal with the logarithms of the absolute value of the $\o_s$ [19] thus; the seven moment invariants used in the proposed system are replaced by their logarithmic values. Table 1 shows the rounded values of $\o$ obtained for some of the words in the training set.

Table1. Moment invariant values for three words.

	سيدي الظاهر	بوعثمان	تونس القباضة الأملية
Ø1	1,1510	0,9937	1,2045
Ø2	2,2821	1,9367	2,3945
Ø3	1,0679	1,1754	1,5419
Ø4	0,3176	0,4587	1,5390
Ø5	0,5082	1,1193	3,0794
Ø6	-0,7184	1,0512	2,7312
Ø7	0,9877	1,1312	1,0762

5.2. Zernike Moments

Zernike moments are chosen for RBF network; Zernike [4] defined a complete orthogonal set $\{V_{nm}(x, y)\}$ of complex polynomials over the polar coordinate space inside a unit circle ($x^2+y^2=1$) as follows:

$$V_{nm}(x,y) = V_{nm}(\rho,\theta) = R_{nm}(\rho)e^{jm\theta} \tag{20}$$

where $j=\sqrt{-1}$, $n\geq 0$, m is a positive or negative integer, $|m|\leq n$, $n-|m|$ is even, ρ is the shortest distance from the origin to (x, y) pixel, θ is the angle between vector ρ and x-axis in counter clockwise direction, and $R_{nm}(\rho)$ is the orthogonal radial polynomial given by:

$$R_{nm}(\rho) = \sum_{s=0}^{n-|m|/2} (-1)^s . \frac{(n-s)!}{s!\left(\frac{n+|m|}{2}-s\right)!(\frac{n-|m|}{2}-s)!}\rho^{n-2s} \tag{21}$$

Note that $R_{n-m}(\rho)=R_{nm}(\rho)$. These polynomials are orthogonal and satisfy the following condition:

$$\int\int_{x^2+y^2\leq 1} [V_{nm}(x,y)]^* V_{pq}(x,y)dxdy = \frac{\pi}{n+1}\delta_{np}\delta_{mq} \tag{22}$$

Where:

$$\delta_{ab} = \begin{cases} 1; & if \quad a=b \\ 0; & otherwise \end{cases} \tag{23}$$

Zernike moments are the projection of the image intensity function $f(x, y)$ onto the complex conjugate of the previously defined Zernike polynomial $Vnm(\rho, \theta)$, which is defined only over the unit circle:

$$A_{nm} = \frac{n+1}{\pi}\int\int_{x^2+y^2\leq 1} f(x,y)V^*_{nm}(\rho,\theta)dxdy \tag{24}$$

For a digital image, Zernike moments are given by:

$$A_{nm} = \frac{n+1}{\pi}\sum_x\sum_y f(x,y)V^*_{nm}(\rho,\theta), x^2+y^2\leq 1 \tag{25}$$

5.3. Tchebichef Moments

For Fuzzy ART we have opted for Tchebichef geometric moments. In these moments the basis function is the discrete orthogonal Tchebichef polynomial. For a given positive integer N (normally the image size), the Tchebichef polynomial is given by the following recurrence relation:

$$t_n(x) = \frac{(2n-1)t_1(x)t_{n-1}(x)-(n-1)(1-\frac{(n-1)^2}{N^2})t_{n-2}(x)}{n} \tag{26}$$

with the initial conditions:

$$\begin{aligned} t_0(x) &= 1 \\ t_1(x) &= (2x+1-N)/N \end{aligned} \tag{27}$$

where $n=0,1,\ldots,N-1$. The Tchebichef moment of order $(p + q)$ of an image intensity function is defined as:

$$T_{nm} = \frac{1}{\rho(m,N)\rho(n,N)}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} t_m(x)t_n(y)f(x,y) \tag{28}$$

Where $n,m=0,1,\ldots,N-1$. The Tchebichef polynomial satisfies the property of orthogonallity with:

$$\rho(n,N) = \frac{N(1-\frac{1}{N^2})(1-\frac{2^2}{N^2})\ldots(1-\frac{n^2}{N^2})}{2n+1} \tag{29}$$

6. Experiments Results

We tested the three classifiers on the IFN/ENIT database [17]. IFN/ENIT was produced by the Institute for Communications Technology at the Technical University of Braunschweig (IFN) and the “Ecole Nationale d’Ingénieurs de Tunis”. The total number of binary images of handwritten Tunisian town/village names is 26459. Those names were written by 411 writers, and they were labelled according to 946 name classes. We have used as programming language Delphi 2009 Architect, and implemented our application on Microsoft Windows XP professional platform.

All experiments are performed on an Intel Pentium IV, 3GHz personal computer with 512Mb of RAM. Training of MLP is obtained with 10000 cycle consumed 40h, concerning Fuzzy ART, training is obtained with 2000 iterations taking 20mn, finally RBF tacked 20h with 5000 iterations. An average processing time of 359,14ms per name image on the test set is obtained by the combining rule.

6.1. System Architecture

The diagram of the off-line Arabic word recognition system is shown in Figure 3. The recognition process typically consists of three stages: pre-processing, feature extraction, and classification.

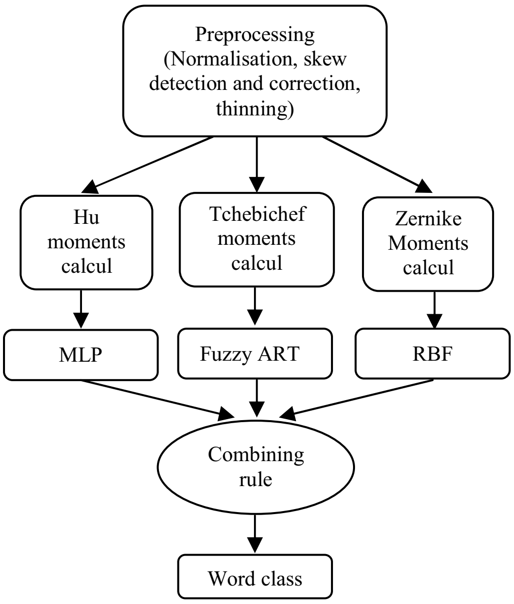


Figure 3. Structure of the proposed OCR system.

In preprocessing phase we have applied a normalisation algorithm [4] giving a uniformed size of 400x100 pixels, Figure 4 illustrates an example.

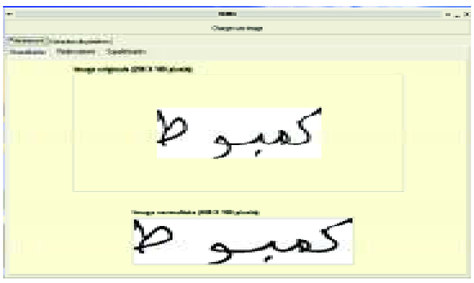


Figure 4. Word normalization.

Skew detection and correction are done by the use of horizontal projection histograms according to eleven different angles of rotation varying from -5° to +5°, the corrector angle of incline will be the one having the denser histogram. When we Calculate the entropy of each obtained histogram, we can determinate the more dense histogram represented by the smallest entropy. Figure 5 indicates a redressing example of one word taken from the database.

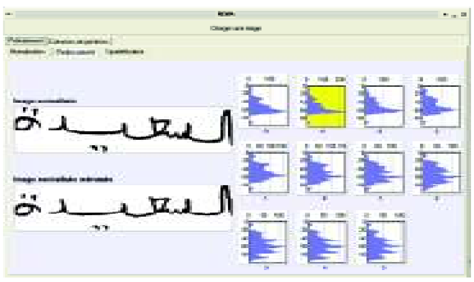


Figure 5. A redressing example.

We have tested four algorithms; of Rutovitz, Zhang and Suen, Deutch and the one of Zhang and Wang. We have taken the one of Zhang-Wang according to its high performance show in Figure 6. Image normalization and skew correction should be used prior to Tchebichef moment extraction.

Figure 6. Thinning example.

6.2. System Parameters

Concerning Fuzzy ART Network, we have used 49 input neurones corresponding to the 49 first Tchebichef moments. Also, we have study the effect of varying vigilance parameter ρ according to the number of class. Figure 7 shows the number of class detected by diverse vigilance parameters $\rho \in [0,1]$. We observed that if the vigilance used is smaller, the number of class=1. If ρ=0.6, the number of classes achieved is 100. Furthermore, if ρ=0.7, the number of classes found surpass one hundred. We use ρ=0.6 as our benchmark for the following experiments due to its performance found.

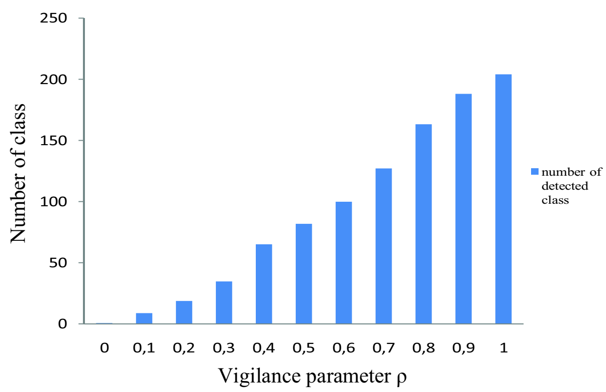


Figure 7. Number of classes detected according to various vigilance parameter.

Our MLP parameters are:

1. An input layer formed by 7 neurons, corresponding to Hu moments.
2. An output layer composed of 100 neurons representing the number of classes.
3. Two hidden layers; the first one is composed of 300 neurones and the second one contains 300 neurons.
4. We have chosen the sigmoid function:

$$f(n_i) = \frac{1}{1+e^{-n_i}} \tag{30}$$

where n_i is the total input of neuron I given by the weighted sum of its input-signals. The RBF network consists of three layers, an input layer, a hidden layer and an output layer of linear neurons. There are a total of 100 nodes in the hidden layer. The input layer is composed of 47 input neurons representing the first 47 Zernike moments of order 12.

6.3. Classifier Combination

We investigate basically three different combination strategies for combining our off-line recognizers: majority vote, max-rule, and sum-rule. Majority vote chooses the class receiving the most votes from all classifiers. The max-rule takes the class with the maximum output value among each classifier, while the sum-rule sums up the output for each class and selects the one with the highest sum.

The plots in Figure 8 show the improvement of accuracy rate results by using combining rules. As it is shown, using majority vote rule gives the best recognition rate compared to the other used rules.

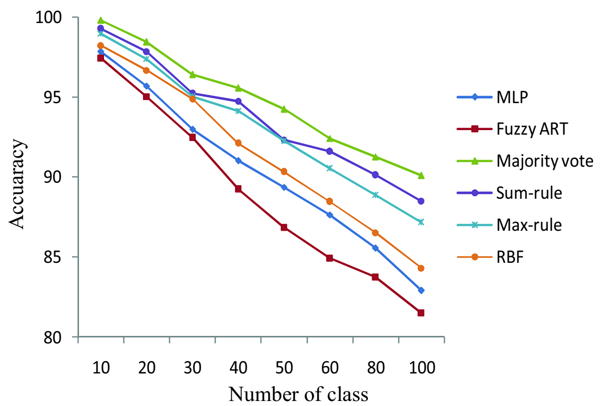


Figure 8. Fuzzy ART, MLP, RBF and Fuzzy ART+MLP+RBF classifier accuracy.

6.4. Comparison with other Systems

The IFN/ENIT database is available to the scientific community and this makes system comparison possible. As mentioned in the abstract, we compared the developed system to three other systems tested also on IFN/EINIT database; the ones of [2, 6, 15], it may be noted from Table 2 that the highest accuracy was obtained by El-Hajj [6], this is due to the use of a segmentation phase, on the other side our system achieves a good accuracy compared with Burrow's and Miled's [2, 15] system which prove the performance of combining an RBF and MLP with a Fuzzy ART network.

Table 2. Comparison results.

	Classifiers	Combining Rule	Accuracy
El-Hajj's System	3 HMMs	Neural Network	94,44%
Miled's System	3HMMs	Majority Vote	89%
Burrow's System	Several K-NN	Majority Vote	74%
Our System	MLP+RBF+ Fuzzy ART	Majority Vote	90,10%

7. Conclusions

Handwritten Arabic word recognition is still a very difficult problem. Classifier combination is considered as a promising way for solving such complex pattern recognition problem. We have shown the efficiency of the presented combining schemes of Fuzzy ART, MLP and RBF network for Arabic cursive word recognition. So all the combination approaches are superior to the best individual classifier, and the majority vote outperforms the other classifier combination scheme by giving 90,10 % accuracy rate.

Hu and Zernike moments are invariant to translation, rotation and scaling which omit the use of some pre-processing operations like normalization and skew detection and correction. Tchebichef moments are fast to implement, present adequate noise tolerance and very accurate image reconstruction. MLPs generally achieve good performances in terms of correct recognition rate in handwritten word classification. The performance of Radial Basis Function Neural Networks on classification is always excellent, compared with other networks. Concerning Fuzzy ART network, it does not demand a priori knowledge of the fixed number of necessary classes.

The obtained results are promising according to the novelty of the idea in spite of the problems finding in the data base concerning the bad writing. An interesting way to improve the performance of the hybrid RBF-MLP-Fuzzy ART recognizer, presented in this paper, consists in adding other kind of classifier to the combining scheme like HMM classifier. Also we can add a post-processing step in our future modifications to the system.

References

[1] Azizi N., Farah N., Sellami N., and Ennaji A., "Using Diversity in Classifier Set Selection for Arabic Handwritten Recognition," *in Proceedings of Multi Classifier System, Lecture Notes in Computer Science*, Berlin, pp.235-244, 2010.

[2] Burrow P., "Arabic Handwriting Recognition," *Report of Master of Science School of Informatics*, University of Edinburgh, pp. 28-34, 2004.

[3] Chergui L. and Benmohammed M., "ART Network for Arabic Handwritten Recognition System," *in Proceedings of the 9th International Arab Conference on Information Technology*, Tunisia, 2008.

[4] Cheriet M., Kharma N., Liu C., and Suen C., *Character Recognition Systems*, John Wiley and Sons, USA, 2007.

[5] Duda R., Hart P., and Stork D., *Pattern Classification*, New York, 1999.

[6] El-Hajj R., Mokbel C., and Likforman-Sulem L., "Combination of HMM-Based Classifiers for the Recognition of Arabic Handwritten Words," *in Proceedings of the 9th International Conference on Document Analysis and Recognition*, Parana, pp. 959-963, 2007.

[7] Farah N., Khadir M., and Sellami M., "Artificial Neural Network Fusion: Application to Arabic Words Recognition," *in Proceedings of European Symposium on Artificial Neural Networks*, Bruges, pp. 151-156, 2005.

[8] Govindaraju V. and Sethur S., *Guide to OCR for Indic Scripts, Document Recognition and Retrieval*, Springer, New-York, 2009.

[9] Graupe D., *Principles of Artificial Neural Networks*, World Scientific, Singapore, 2007.

[10] Haidl M. and Kittler J., *Multiple Classifier Systems*, Springer, Germany, 2007.

[11] Kecman V., *Learning and Soft Computing, Support Vector Machines, Neural Networks, and Fuzzy Logic Models*, The MIT Press, USA, 2001.

[12] Kuncheva L., *Combining Pattern Classifiers: Methods and Algorithms*, Wiley-Interscience, USA, 2004.

[13] Liwicki M. and Bunke H., *Recognition of Whiteboard Notes Online, Offline and Combination*, World Scientific, Singapore, 2008.

[14] Lorigo L. and Govindaraju V., "Off-Line Arabic Handwriting Recognition: a Survey," *IEEE on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 712-724, 2006.

[15] Miled H., Olivier C., Cheriet M., and Lecourtier Y., "Coupling Observation/Letter for a Markovian Modilization Applied to the Recognition of Arabic Handwriting," *in Proceedings of 4th International Conference Document Analysis and Recognition*, Algiers, pp. 580-583, 1997.

[16] Mukundan R., "Radial Tchebichef Invariants for Pattern Recognition," *in Proceedings of the IEEE Tencon*, Melbourne, pp. 114-115, 2005.

[17] Pechwitz M., Maddouri S., Margner V., Ellouze N., and Amiri H., "IFN/ENIT Database of Handwritten Arabic Words," *in Proceedings of the 7th Colloque International Francophone sur l'Ecrit Document*, Tunis, pp. 129-136, 2002.

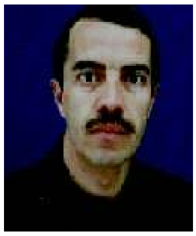
[18] Sarfraz M., *Computer Aided, Intelligent Recognition, Techniques and Applications*, John Wiley and Sons, England, 2005.

[19] Theodoidis S. and Kontrommbas K., *Pattern Recognition*, Elsevier, USA, 2003.

[20] Touj S., Ben-Amara N., and Amiri H., "Arabic Handwritten Words Recognition Based on a Planar Hidden Markov Model," *The International Arab Journal of Information Technology*, vol. 2, no. 4, pp. 318-325, 2005.

Leila Chergui received her MSc degree in computer sciences in 2009 from the University of Larbi Ben Mhidi, Oum El Bouaghi, Algeria. From 2007 to 2009 she occupied the post of pic programmer in a hollandaise society of solar energy. Currently, she is an assistant professor in the Department of Computer Sciences at the same University. Her current research interests include Handwriting recognition, neural networks, pattern recognition and document processing.

Maâmar Kef received his MSc degree in computer sciences from the University of Larbi Ben Mhidi, Algeria, in 2008. From 2004 to 2009 he was IT manager at a National Company. Since 2008 he has been at the University of Constantine, where he is preparing his PhD degree. Currently, he is an assistant professor in the Department of Computer Sciences at Hadj Lakhdar University, Algeria. His current research interests include Ad-hoc networks, artificial intelligence techniques with neural networks, pattern recognition and document processing.



Salim Chikhi received his MSc degree in computer systems from University of Constantine, Algeria, in collaboration with Glasgow University, UK. He received his PhD degree in computer science at the same University in 2005. Currently, he is a full professor at the University of Constantine. His research areas include artificial life, soft computing and complex systems.