2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)

New Hybrid Arabic Handwriting Recognizer

Leila Chergui
Department of Computer Sciences
University Larbi Ben Mhidi
Oum el Bouaghi, Algeria
pgleila@yahoo.fr

Maâmar Kef
Department of Computer Sciences
University Hadj Lakhdar
Batna, Algeria
lm_kef@yahoo.fr

Salim Chikhi
Department of Computer Sciences
University Mentouri
Constantine, Algeria
slchikhi@yahoo.com

Abstract—**Recently, there is a popular belief that classifier combination of different architecture could complement each other for improving results performance. In this paper we introduce a framework to combine results of multiple classifiers for offline Arabic handwriting recognition, by introducing a new scheme of combination of Multi Layer Perceptron and ART1 networks. Besides using two different recognition architectures (MLP and ART1 networks), we exploit various feature sets calculated from the contour of image; the Hu moments and features obtained with sliding windows. The implementation results on IFN/ENIT database show a high degree of accuracy by applying the majority vote method.**

Keywords- Key words: ART1 network, Combining classifiers, Hu moments, Multi Layer Perceptron.

I. INTRODUCTION

The recognition of Arabic script has many applications such as mail sorting, bank check reading, and, more recently, the recognition of historical manuscripts. Arabic writing is naturally cursive, and it has 28 characters. Letter shapes are context sensitive according to their position within the word (beginning, middle, and end) or when the character is isolated, resulting in 100 different shapes. Additional marks (hamza, shadda, etc.), dots, and diacritical marks change the letter and the word meaning or indicate vowels.

The study of Multiple Classifier Systems (MCS) has become an area of intensive research in pattern recognition. Many combination methods have been proposed, a recent survey [13] categorizes the methods into parallel (horizontal) and sequential (vertical, cascaded) ones. In the parallel combination, all the classifiers run on the same data in order to produce a decision on the class corresponding to the unknown pattern. The overall classification is obtained by an appropriate combination of the decisions provided by single classifiers. In the serial combination, the activation of some classifiers is conditioned by the results of classifiers that are taken into account first. Parallel combination is more often adopted for improving the classification accuracy, whereas sequential combination is mainly used for accelerating the classification of large category set.

Our approach is based on parallel combination of an Multi Layer Perceptron and Adaptive Resonance Theory classifiers.

The contribution of the paper is two-fold. Firstly, two classifiers for Arabic handwritten word recognition with different architectures are introduced and results of their combination are presented. Secondly, two different methods of feature extraction are used. The remaining part of the paper is organized as follows. In section 2 we will present previous work in Multiple Classifier Offline Arabic System. Section 3 focuses on summarize the proposed combination methods. Section 4 describes the architecture of the proposed MCS. Experiments and results are discussed in section 5 and conclusions are drawn in the last section of this paper.

II. RELATED WORKS

Compared to Latin script where a lot of research work is done, the number of work for the combination of Arabic script is quite limited. One of the first works in this field was given by Farah [7] who introduced a system based on the combination of three MLPs for the recognition of Arabic literal amount with a recognition rate of 94% on a small test database containing 4800 words. El-Hajj [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 [16], 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 [15] 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 presented in [15] a hybrid approach to the recognition of literal amounts on bank-checks. Three classifiers ran in parallel: neural networks, K-nearest-neighbor, and Fuzzy K-nearest-neighbor. 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 [3] Burrow applied one K-Nearest Neighbors 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.

III. ARCHITECTURE OF MCS

The architecture of our system is shown in figure1:

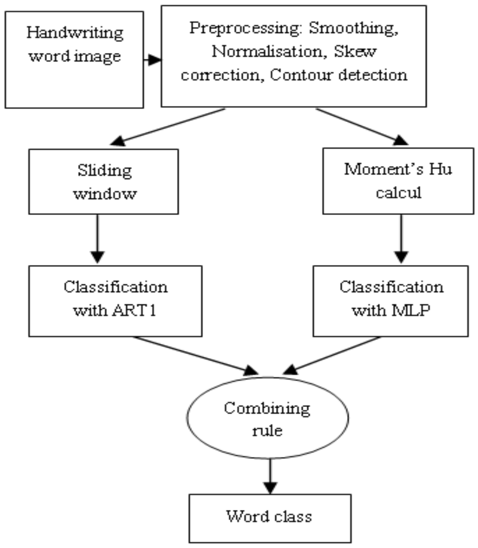


Figure 1. The overall structure of the MCS

A. *preprocessing*

Each person has a different writing style with its own characteristics. This fact makes the recognition of handwriting complicated. To reduce variations in the handwritten words as much as possible, a number of pre-processing operations are applied.

- Smoothing; in our approach we have applied a filter to reduce the noise. For a given point P, we deduct its new value (0 or 1) according to its eight direct neighbours.
- Normalisation; the original word image can be normalized and encoded in a canonical form. The normalization task will give a uniformed size of 400x100 pixels for the entire words image.
- Skew Correction; the word is horizontally aligned, i.e. rotated, so that the baseline is parallel to the *x*-axis of the image [4]. We use this operation to make the features robust against different writing styles.
- Contour representation; Feature extraction from stroke contour has been widely adopted because the contour length is nearly independent of stroke-width variation.

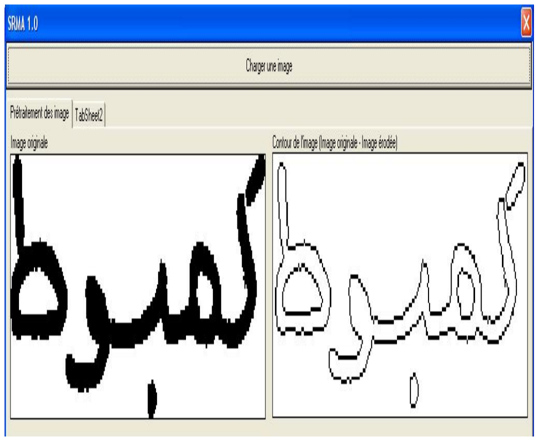


Figure 2. Contour detection of a word image.

B. *Feature extraction*

In feature extraction stage, each word is represented as a feature vector, which becomes its identity. The major goal of feature extraction is to extract a set of features, which maximizes the recognition rate with the least amount of elements.

In our implementation, moment invariants used by Hu [14] have been utilized to build the feature space for MLP classifier. Using nonlinear combinations of geometric moments, we 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:

$$M_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\bar{x})^p (y-\bar{y})^q f(x,y)dxdy \qquad (1)$$

Where:

$$\bar{x} = \frac{\overline{M}_{10}}{\overline{M}_{00}}, \qquad \bar{y} = \frac{\overline{M}_{01}}{\overline{M}_{00}} \qquad and$$
$$\overline{M}_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)dxdy \qquad (2)$$

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 (1) and (2) are approximated by the summations:

$$M_{pq} = \sum_{x=0}^{m}\sum_{y=0}^{n} (x-\bar{x})^p (y-\bar{y})^q f(x,y)dxdy \qquad (3)$$

$$\overline{M}_{pq} = \sum_{x=0}^{m}\sum_{y=0}^{n} x^p y^q f(x,y)dxdy \qquad (4)$$

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

$$\phi_1 = M_{20} + M_{02} \tag{5}$$

$$\phi_2 = (M_{20} - M_{02}) + 4M_{11}^2 \tag{6}$$

$$\phi_3 = (M_{30} - 3M_{12})^2 + (3M_{21} - M_{03})^2 \tag{7}$$

$$\phi_4 = (M_{30} + M_{12})^2 + (M_{21} + M_{03})^2 \tag{8}$$

$$\begin{aligned}\phi_5 = &(M_{30} - 3M_{12})(M_{30} + M_{12}) \\ &[(M_{30} + M_{12})^2 - 3(M_{21} + M_{03})^2] + (3M_{12} - M_{03}) \\ &(M_{21} + M_{03})[3(M_{30} + M_{12})^2 - (M_{21} + M_{03})^2]\end{aligned} \tag{9}$$

$$\begin{aligned}\phi_6 = &(M_{20} - M_{02})[(M_{30} + M_{12})^2 - (M_{21} + M_{03})^2] + \\ &4M_{11}(M_{30} + M_{12})(M_{21} + M_{03})\end{aligned} \tag{10}$$

$$\begin{aligned}\phi_7 = &(3M_{21} - M_{03})(M_{30} + M_{12})[(M_{30} + M_{12})^2 - \\ &3(M_{21} + M_{03})^2] + 3(M_{21} - M_{03})(M_{21} + M_{03})* \\ &[3(M_{30} + M_{12})^2 - (M_{21} + M_{03})^2]\end{aligned} \tag{11}$$

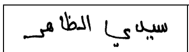
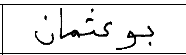
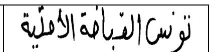
These functions can be normalized to make them invariant under a scale change by using the normalized central moments instead of the central moments. The normalized central moments are defined by:

$$M_{pq} = \frac{M_{pq}}{M_{00}^a} \quad where \quad a = \frac{(p+q)}{2} + 1 \tag{12}$$

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

The ϕ_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 ϕ_s [14] thus; the seven moment invariants used in the proposed system are replaced by their logarithmic values. Table 1 shows the rounded values of ϕ obtained for some of the words in the training set. Each of the words is a 400x100 binary image.

TABLE I. MOMENT INVARIANT VALUES FOR THREE WORDS

			
Φ_1	1.1510	0.9937	1.2045
Φ_2	2.2821	1.9367	2.3945
Φ_3	1.0679	1.1754	1.5419
Φ_4	0.3176	0.4587	1.5390
Φ_5	0.5082	1.1193	3.0794
Φ_6	-0.7184	1.0512	2.7312
Φ_7	0.9877	1.1312	1.0762

To extract a sequence of feature vectors from a word for the ART1 classifier a sliding window is used. The width of the window is forty pixels and its high is equal to the word's height. The window is moved from right to left over each word; we suppose that there is no overlap between two consecutive window positions. Seven geometrical quantities are computed at each window position. A graphical window technique is shown in figure 3.

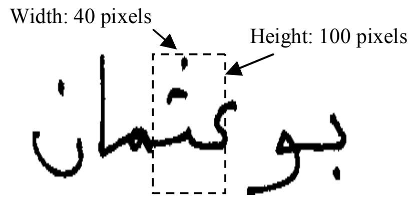


Figure 3. Illustration of the sliding window technique.

The first two features are the weight of the window and its center of gravity. This set characterizes the window from global point of view. The other features represent additional information about the writing. Feature three and four define the position of the upper and the lower contour in the window. The next two features, number five and six, give the orientation of the upper and the lower contour in the window by the gradient of the contour of the window's position. Finally feature number seven gives the number of black pixels between the upper and lower contour.

To summarize, the output of the feature extraction phase for the ART1 classifier is a sequence of 7-dimentional feature vectors. For each word to be recognized there exists one such vector per forty pixels along the horizontal extension of the considered word.

C. Classification

In this section, we briefly describe the single classifiers based on Multilayer Layer Perceptrons and on ART1 network.

1) MLP-based classifiers: 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:

$$\bar{y} = \max_{i=1,\ldots,n} \{y_i\} < RM \tag{13}$$

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) [11]. This algorithm performs a gradient descent in the connection weight space on an error surface defined by:

$$E = \frac{1}{p}\sum_{p} E_p \tag{14}$$

Where:

$$E_p = \frac{1}{2}\sum_{K} (t_{pk} - y_{pk})^2 \tag{15}$$

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:

$$w_{jk}(t+1) = w_{jk}(t) - \eta \frac{\partial E_p(t)}{\partial w_{jk}(t)} + \alpha \Delta w_{jk}(t-1) \tag{16}$$

$$w_{ij}(t+1) = w_{ij}(t) - \eta \frac{\partial E_p(t)}{\partial w_{ij}(t)} + \alpha \Delta w_{ij}(t-1) \tag{17}$$

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 w_{jk} at time t–1. $\eta(>0)$ and $\alpha(0<\alpha<1)$ are respectively called the learning rate and moment factor.

2) ART1 network: The ART network is an unsupervised vector classifier that accepts input vectors that are classified according to the stored pattern they most resemble. It also provides for a mechanism allowing adaptive expansion of the output layer of neurons until an adequate size is reached based on the number of classes, inherent in the observation [5].

The ART network can adaptively create a new neuron corresponding to an input pattern if it is determined to be "sufficiently" different from existing clusters. This determination, called the vigilance test, is incorporated into the adaptive backward network. Thus, the ART architecture allows the user to control the degree of similarity of patterns placed in the same cluster.

ART1 architecture (figure 4) consists of two layers of neurons called the comparison layer and the recognition layer [9]. The classification decision is indicated by a single neuron in the recognition layer that fires. The neurons in the comparison layer respond to input features in the pattern, analogous to the cell groups in a sensory area of the cerebral cortex.

The synaptic connections (weights) between these two layers are modifiable in both directions, according to two different learning rules. The recognition layer response to an input vector is compared to the original input vector through a mechanism termed vigilance. Vigilance provides a measure of the distance between the input vector and the cluster center corresponding to the firing recognition layer neuron. When vigilance falls below a preset threshold, a new category must be created and the input vector must be stored into that category. That is, a previously unallocated neuron within the recognition layer is allocated to a new cluster category associated with the new input pattern.

The network architecture also consists of three additional modules labelled Gain 1, Gain 2, and reset.

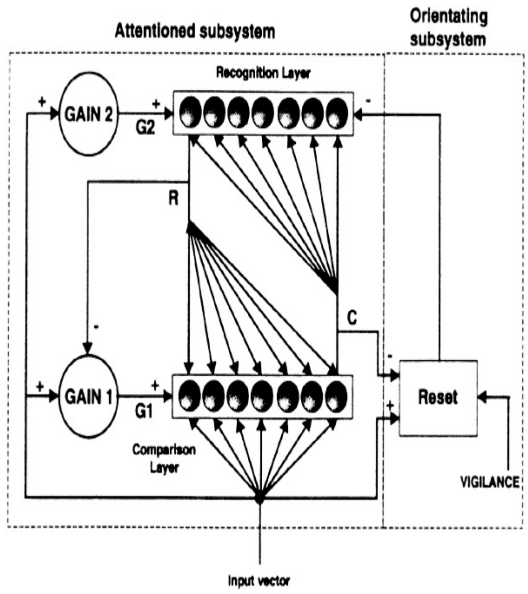


Figure 4. Structure of ART1 network

For training, initially the weights b_{ij} are initialized to the same low value which should be:

$$b_{ij} < \frac{L}{(L-1+m)} \tag{18}$$

Where *m* is the number of components in the input vector and *L* is a constant, typically *L* = 2.

The training algorithm for the ART1 architecture is as follows:

- When an input pattern, **X**, is presented to the network, the recognition layer selects the winner as the maximum of all the net outputs:

$$net_j = \sum_{i=1}^{N} b_{ij} c_i \tag{19}$$

where N is the number of neurons in the comparison layer.

- Perform the vigilance test. A neuron j is declared to pass the vigilance test, if and only if:

$$\frac{net_j}{\sum_{i=1}^{N} x_i} > \rho \tag{20}$$

Where ρ is the vigilance threshold.

If the winner fails the test, mask the current winner and go to step 1 to select another winner. Else, repeat the cycle (steps 1 through a) until a winner is determined that passes the vigilance test; then go to step 4.

- If no neuron passes the vigilance test, create a new neuron to accommodate the new pattern.

- Adjust the feed-forward weights for the winner neuron. Update the feedback weights from the winner neuron to its inputs.

The equations governing the training of the bottom-up and top-down weights are:

$$b_{ij} = \frac{Lc_j}{\left(L - 1 + \sum_{t_{ij}=c_i} c_K\right)} \tag{21}$$

Where c_i is the i^{th} component of the comparison layer vector and j is the index of the winning recognition layer neuron.

IV. EXPERIMENTAL RESULTS

We tested the two classifiers on the benchmark IFN/ENIT database of Arabic city names [18]. It 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 labeled according to 946 name classes.

A. Effect of System parameters

We study the effect of varying vigilance parameter ρ according to the number of classes. Figure 5 shows the number of classes detected by diverse vigilance parameters $\rho \in [0,1]$. We observed that if the vigilance used is smaller, the number of classes=1. If ρ=0.6, the number of classes achieved is 100. Furthermore, if ρ=0.9, 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.

Our MLP parameters are:

- An input layer formed by 7 neurons, corresponding to Hu moments.
- An output layer composed of 100 neurons representing the number of classes.
- A hidden layer with 120 neurons.
- We have chosen the sigmoid function:

$$f(n_i) = \frac{1}{1 + e^{-n_i}} \tag{22}$$

Where n_i is the total input of neuron I given by the weighted sum of its input-signals.

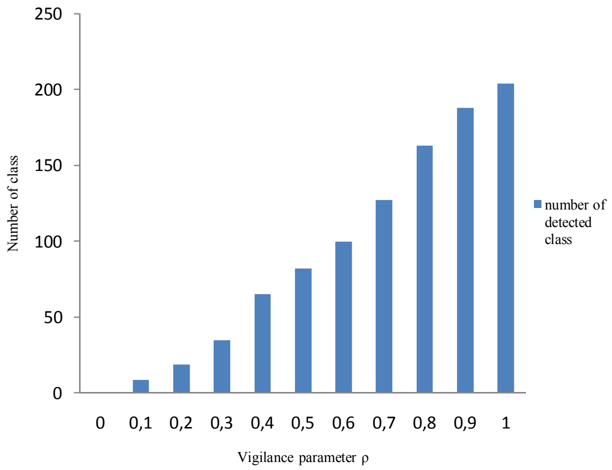


Figure 5. Number of classes detected according to various vigilance parameter.

B. Classifier combination

We have chosen two methods for combining our classifier results; the weighted vote and the majority voting. Figure 6 shows the experimental results about the two classifiers where the performances for individual classifiers are indicated. In these experiments, we considered 100 name classes. Training of word is on sets a, b, c, and d, and test on the e set of IFN/ENIT database. Comparing the individual classifiers, the MLP is the best one with an accuracy of 83.01%. From the results shown in Figure 6 it is obvious that the combination of different classifiers significantly improves performance. Comparing combination strategies, the majority vote outperforms the weighted vote by giving a recognition rate of 88.75%.

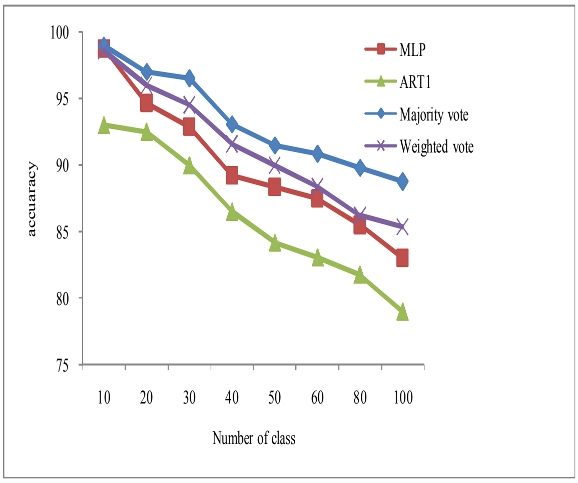


Figure 6. ART1, MLP and ART1+MLP classifier accuracy.

C. *Comparison with other systems*

The IFN/ENIT database is available to the scientific community and this makes system comparison possible. Wherefore we take from the framework given in paragraph 1 only three systems, the ones of El-hajj [6], Miled [16] and Burrow [3], it may be noted from table 2 that the highest accuracy was obtained by El-Hajj following by Miled, this is due to the use of a segmentation phase, on the other side our system achieves a good accuracy compared with Burrow's system which prove the performance of combining an MLP with ART1 network.

TABLE II. 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 + ART1	Majority vote	88.75%

V. CONCLUSION

A new Multi Classifier System for offline Arabic handwritten recognition has been investigated in this paper. The experimental results clearly show the better performance of the proposed architecture with respect to its single components (MLP and ART1 network), and provide evidence that a significant improvement in performance is reached with a small increment of computational complexity.

The main advantages of neural networks lies in the ability to be trained automatically from examples, good performance with noisy data, possible parallel implementation, and efficient tools for learning large databases.

ART 1 is significant for three reasons: first, the issue of plasticity versus stability is both interesting and important, and ART 1 was the first model to clearly identify this problem and propose a solution to it; second, relatives and generalizations of ARTl can handle continuously valued data, and these extensions are as good as any other model you might choose to try, and lastly, the ability to combine this model with other classifiers.

A method for choosing the relevant features and the set of non redundant characteristics is necessary as a future work in order to insure a better description of the processed form. The use of tools such as information criteria or hidden Markov models enables the selection of the most discriminative set of features. Moreover, the usage of genetic algorithms has equally achieved encouraging results for the selection of primitives.

REFERENCES

[1] N. Azizi, N. Farah, M. Sellami, and A. Ennaji, "Using diversity in classifier set selection for Arabic handwritten recognition," Springer, Multi Classifier System, pp.235-244 , 2010.

[2] H. Bunke, and A. Kandel, Hybrid Methods in Pattern Recognition. World Scientific, Singapore, 2002.

[3] P. Burrow, Arabic Handwriting Recognition. Report of Master of Science School of Informatics, University of Edinburgh, 2004.

[4] L. Chergui, and M. Benmohammed, "ART network for Arabic handwritten recognition system", ACIT. The 9 th International Arab Conference on Information Technology, Tunisia, 2008.

[5] M. Cheriet, N. Kharma, C.L. Liu, and C.Y. Suen, Character Recognition Systems, Willey, USA, 2007.

[6] R. El-Hajj, C. Mokbel, and L. Likforman-Sulem, "Combination of HMM-Based classifiers for the recognition of Arabic handwritten words," Document Analysis and Recognition, 2007.

[7] N. Farah, M.T. Khadir, and M. Sellami, "Artificial neural network fusion: Application to Arabic words recognition," European Symposium on Artificial Neural Networks, Bruges, pp. 151–156, 2005.

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

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

[10] M. Haidl, and J. Kittler, Multiple Classifier Systems. Springer, Germany, pp. 72–82, 2007.

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

[12] Y. Kessentini, T. Burger, and T. Paquet, "Evidential combination of multiple HMM classifiers for multi-script handwriting recognition," Springer, IPMU, pp. 445–454, 2010.

[13] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience. New Jersey. USA, pp. 112–129, 2004.

[14] M. Liwicki, and H. Bunke, Recognition of Whiteboard Notes Online, Offline and Combination. World Scientific, Singapore, pp. 28–30, 2008.

[15] L. M. Lorigo, and V. Govindaraju, "Off-line Arabic handwriting recognition: a survey," IEEE on Pattern Analysis and Machine Intelligence, vol.28, no. 5, pp. 712-724, 2006.

[16] H. Miled, C. Olivier, M. Cheriet, and Y. Lecourtier, "Coupling observation/letter for a Markovian modilization applied to the recognition of Arabic handwriting," Proc. Fourth Int'l Conf. Document Analysis and Recognition, pp. 580–583, 1997.

[17] J.J. Oliveira, M.N. Kapp, C.O. Freitas, J.M. Carvallo, and R. Sabourin, "Handwritten recognition with multiple classifiers for restricted lexicon," IEEE. Symposium on Computer Graphics and Image Processing, pp. 1530–1534, 2004.

[18] M. Pechwitz , S.S. Maddouri, V. Margner, N. Ellouze, and H.Amiri, "IFN/ENIT Database of handwritten Arabic words," in Proc. CIFED, pp. 129-136, 2002.

[19] C.D. Ruberto, and A. Morgera, "Moment-based technologies for image retrieval," IEEE, 19th International Conference on Database and Expert Systems Applications, pp. 155-159, 2008.

[20] [SAR 05] M. Sarfraz, Computer Aided, Intelligent Recognition, Techniques and Applications. John Wiley & Sons, England, 2005.

[21] V. Subbiah Bharathi, M.A. Leo Yijilious, and L. Ganesam, "Orthogonal moment based texture analysis of CT Liver images," IEEE, International Conference on Computational Intelligence and Multimedia Applications, pp. 249-255, 2007.

[22] S. Theodoidis, and K. Kontrommbas, Pattern Recognition, Elsevier, San Diego, USA, 2003.

[23] Q. Wang, and S. Ju, "A mixed classifier based on combination of HMM and KNN," IEEE Trans. Fourth International Conference on Natural Computation, vol. 4, pp. 32–42, 2008.