International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.8, No.10 (2015), pp.241-256
http://dx.doi.org/10.14257/ijsip.2015.8.10.26

Printed Noisy Greek Characters Recognition Using Hidden Markov Model, Kohonen Network, K Nearest Neighbours and Fuzzy Logic

Salouan, S. Safi and B. Bouikhalene

Department of Mathematic and Informatic, Polydisciplinary Faculty, Sultan Moulay Slimane University, Beni Mellal, MOROCCO
rachidsalouan@gmail.com

Abstract

In this paper, we present for printed multi-oriented, multi-scaled and noisy Greek characters recognition a comparison in terms of precision, rapidity and stability between several classifiers which the first one is a probabilistic that is hidden Markov model, the second is a neuronal that is Kohonen network or self-organizing maps while the rest of other classifiers are based on a combination between these both classifiers and even more a statistical method that is K nearest neighbors in their tree different versions which are majority voting, weighted distances and fuzzy. For this purpose we have for pre-processed each character image by the median filter and the thresholding technique, then in order to extract efficiently their features, we have exploited the Krawtchouk invariant moments.

Keywords: *Printed multi-oriented, multi-scaled and noisy Greek characters, Median filter, Thresholding, Krawtchouk invariant moments, Hidden Markov model, Kohonen network, K nearest neighbors, Fuzzy logic*

1. Introduction

Optical Character Recognition (OCR) has been undoubtedly a subject of extensive research for more than five decades due to its various applications in several areas such as archiving documents, automatic verification of bank checks and postal processing, etc.

In fact, there are many studies and researches have been done towards the recognition of Arabic, Latin characters and numerals by using invariant moments[1-6], hidden Markov model [7-9], Kohonen network [10-13] or K nearest neighbours [14-16]. In this context, we have interested in this research to perform for printed multi-oriented, multi-scaled and noisy Greek characters a comparison in terms of precision, rapidity and stability between several classifiers which are Hidden Models Markov (HMM) [17], Kohonen network (Self-Organizing Maps (SOM)) [18] then each one of them is combined with K Nearest Neighbours (KNN) classifier in their tree different versions which are majority voting, weighted distances [19] and fuzzy [20-21]. For this goal, in order to carry out this comparison, we have used for pre-processing each character image the median filter and the thresholding technique, afterwards we have employed for extracting their features, the Krawtchouk Invariant Moment (KIM) [22].

Moreover, this paper is formed by seven sections which the first one presents an introduction, the second contains the methodology opted in order to realize the desired recognition systems. Features extraction is explained in fourth section. Recognition phase includes learning and classification is presented in fifth section, then experimental results are discussed in sixth Section. Finally, in seventh section the paper is ended by a conclusion.

ISSN: 2005-4254 IJSIP

2. The Methodology

The recognition systems that we have used are presented as follows:

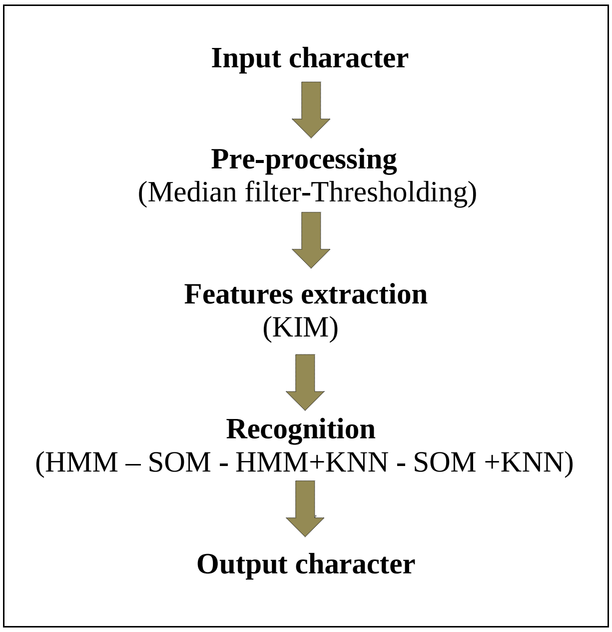

Figure 1. The Proposed Recognition Systems

3. Pre-processing

Pre-processing is the first phase in each OCR system, it is used for removing noise and needless information presented in character image in reason to render its quality in a best shape which will enable accordingly to facilitate the extraction of their features or primitives in an efficient manner. For this aim we have adopted both techniques of pre-processing which are the median filter used to filtrate each character image then the thresholding exploited to convert it to a binary form that is to say contains nothing that black and white colors according to a fixed threshold previously.

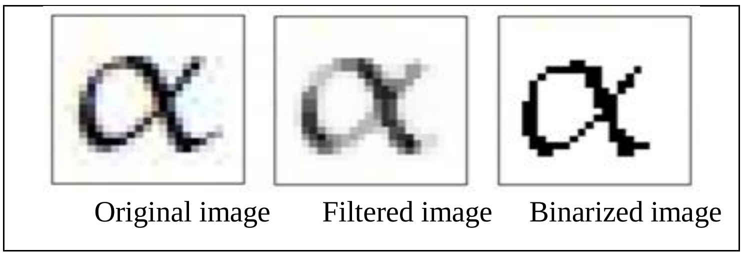

Figure 2. Different Techniques Used in Pre-processing Phase

4. Features Extraction

Features extraction is undoubtedly considered as the most important operation in each OCR; furthermore the desired aim of this step is to achieve a conversion of each character image which is presented firstly as a matrix to a vector and carry out a great discrimination between different classes of characters for rendering its recognition easier. In this framework, we have used the Krawtchouk invariant moment which is a very powerful tool used for extracting efficiently significant features from characters.

4.1 The Krawtchouk Moment

4.1.1 The Krawtchouk Polynomial: The Krawtchouk polynomial of order n is given by:

$$K_n(x,p,N) = \sum_{k=0}^{N} a_{k,n,p}x^k = {}_2F_1\left(-n,-x;-N;\frac{1}{p}\right) \quad (1)$$
$$x,n = 0,1,2 \ldots .M, N > 0, \qquad p \in [0,1]$$

${}_2F_1$ is the hyper geometric function defined as:

$${}_2F_1(a,b;c;x) = \sum_{k=0}^{\propto} \frac{(a)_k(b)_k}{(c)_k}\frac{x^k}{k!} \quad (2)$$

And $(a)_k$ is the Pochhammer symbol (called also rising factorial) defined by:

$$(a)_k = a(a+1)\ldots(a+k-1) = \frac{\Gamma(a+k)}{\Gamma(a)} \quad (3)$$

The Γ function is defined by:

$$\Gamma(x) = \int_0^{\propto} t^{x-1}e^{-t}dt \quad (4)$$
$$\forall\, n \in N \ , \ \Gamma(n+1) = n! \quad (5)$$

The set of (N+1) Krawtchouk polynomial $\{K_n\,(x;\ p,\ N)\}$ forms a complete set of discrete basis functions with the weight function:

$$w(x,p,N) = \binom{N}{x}p^x(1-p)^{N-x} \quad (6)$$

And satisfies the orthogonally condition:

$$\sum_{x=0}^{N} w(x,p,N)\, K_n(x,p,N)K_m(x,p,N) = \binom{N}{x}p^x(1-p)^{N-x} \quad (7)$$

$$m,n = 0,1,\ldots.,N$$

$\rho(n;p,N)$ is the squared norm defined by:

$$\rho(n,p,N) = (-1)^n\left(\frac{1-p}{p}\right)^n\frac{n!}{(-N)_n} \quad (8)$$

And δ_{nm} is the Kronecker symbol defined by:

$$\delta_{nm} = \begin{cases} 1 & \text{if} \quad n = m \\ 0 & \text{elsewhere} \end{cases} \quad (9)$$

4.1.2 The Krawtchouk Moment: The Krawtchouk moment have the interesting property of being able to efficiently extract local features of an image this moment of order of (n+m) of an image $f(x,y)$ is given by:

$$Q_{nm} = \sum_{x=0}^{n-1}\sum_{y=0}^{M-1} \overline{K_n}(x,p_1,N-1)\overline{K_m}(y,p_2,M-1)f(x,y) \quad (10)$$

The NxM is the number of pixels of an image $f(x,y)$.The set of weighted Krawtchouk polynomials is:

$$\overline{K_n}(x,p,N) = \overline{K_n}(x,p,N)\sqrt{\frac{w(x,p,N)}{\rho(x,p,N)}} \quad (11)$$

4.1.3 The Krawtchouk Invariant Moment

The geometric moment of an image $f(x,y)$ is given by:

$$M_{nm} = \sum_{x=0}^{M-1} \sum_{y=1}^{N-1} x^n y^m f(x,y) \quad (12)$$

The standard set of the geometric invariant moments that are independent to rotation, scaling, translation is:

$$V_{nm=}M_{00}^{-y} \sum_{x-1}^{N-1} \sum_{y=0}^{M-1} [(x-\bar{x})\cos\theta + (y-\bar{y})\sin\theta]^n [(y-\bar{y})\cos\theta\,(x-\bar{x})\sin\theta]^m f(x,y) \quad (13)$$

$$\gamma = \frac{p+q}{2} + 1, \quad \bar{x} = \frac{M_{10}}{M_{00}} \quad , \quad \bar{y} = \frac{M_{01}}{M_{00}} \quad \theta = \frac{1}{2} \arctang \frac{2\mu_{!1}}{\mu_{20} - \mu_{01}} \quad (14)$$

And μ_{nm} is the central moment defined by :

$$\mu_{nm} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (x-\bar{x})^n (y-\bar{y})^m f(x,y) \quad (15)$$

Finally the Krawtchouk invariant moment is given by:

$$\tilde{\Omega}_{nm} = \Omega_{nm} \sum_{i=0}^{n} \sum_{j=0}^{m} a_{i,n,p_1}\, a_{j,m,p_2} \tilde{V}_{ij} \quad (16)$$

$$\Omega_{nm} = [\rho(n, p_1, N-1).\, \rho(m, p_2, M-1)]^{-1/2} \quad (17)$$

$$\tilde{V}_{ij} = \sum_{p=0}^{i} \sum_{q=0}^{j} \binom{i}{p} \binom{j}{q} \left(\frac{N^2}{2}\right)^{\frac{p+q}{2}+1} \left(\frac{N}{2}\right)^{i+j-p-q} V_{pq} \quad (18)$$

$$\binom{x}{y} = \frac{x!}{y!(x-y)!} \quad (19)$$

The coefficients $a_{i,n,p}$ are determined in equation (1).

5. Recognition

5.1 Simple Classifiers

5.1.1 The Hidden Markov Model

HMM is been proven as a one of the most powerful tools in pattern recognition. In fact, this probabilistic model offers really many important properties for modeling characters or words. Among these properties is the existence of efficient algorithms allowing to automatically learn the models without any need of labeling pre-segmented data.

The HMM is based on a doubly stochastic processes whose the first of them is hidden, while the second is observable. The transition of the process from the actual state to the next is based on this underlying process. The observable outputs or the observations are generated by other stochastic process which is given by a set of probabilities.

The HMM with a discrete observation symbol is defined by $\lambda = (A, B, \pi)$, where A is the matrix of the probabilities of transitions, B is the matrix of the probabilities of observations, and π is the vector probability of initial states, where:

N is the number of states s_1 , s_2 ....s_N .
T is the number of observations.
q_t is the state of the process at the time t , $q_t = \{ s_1, s_2 \ldots\ldots s_N \}$.

o_t is the observation at the time t , $o_t = \{v_1, v_2 \dots\dots v_M\}$.
M is the size of observations v_1 , v_2 ....v_M .

And:

$$A = \{a_{ij} = Prob(s_j / s_i)\} \quad (20)$$

$$\sum_{j=1}^{N} a_{ij} = 1 \quad (21)$$

$$\pi = \{ \pi_i = Prob(s_i)\} \quad (22)$$

$$\sum_{i=1}^{N} \pi_i = 1 \quad (23)$$

$$B = \{b_j (k) = Prob(o_t = \mathbf{v}_k / o_t = \mathbf{s}_j)\} \quad (24)$$

$$\sum_{k=1}^{M} b_j(k) = 1 \quad (25)$$

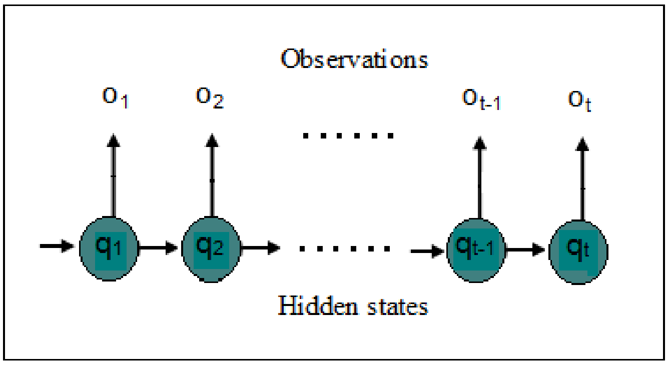

Figure 3. Example of Hidden Markov Model

The HMM with a continuous observation symbol is defined by $\lambda = (A, B, \sigma_i, \mu_i)$ where μ_i and σ_i are respectively the mean and the standard deviation associated to a state i of the Gaussian function that is used to generate the probability of observation:

$$b_i(\upsilon_k) = P(O_t = \upsilon_k | q_t = i) = \frac{1}{(2\Pi)^{\frac{1}{2}} \sigma_i} e^{\frac{(O_t-\mu_i)^2}{2\sigma_i^2}} \quad (26)$$

In learning phase, each character image that is converted to a vector in the features extraction phase by KIM; this vector is used as an observation vector of an initial own HMM of this character in order to determine the probability that generated this observation. Then this model is trained for maximizing as much as possible this probability by using the Baum-Welch algorithm. Afterwards, all these trained models (optimal models) of all characters are saved for forming a learning base.

In the classification phase, an unknown character (test character) is presented as a vector of observation, and then the probability generated by this observation is calculated by all the optimal models already recorded in the learning base by the forward algorithm. The recognition will be finally given to the class of character which presented the highest probability.

5.1.2 The Kohonen Network

The Kohonen network is composed of two layers; the first one has I nodes that are the input of network, the second has J nodes that are its output. These layers are connected by IxJ coefficients called weights W.

The topological maps of Kohonen (self-organizing maps) weighed a special structuring to its neurons (nodes).This structure binds the neurons and have forced them to respect a certain topology during the learning phase .Thus the near data in the input space have an very closest representations in topological Kohonen map.

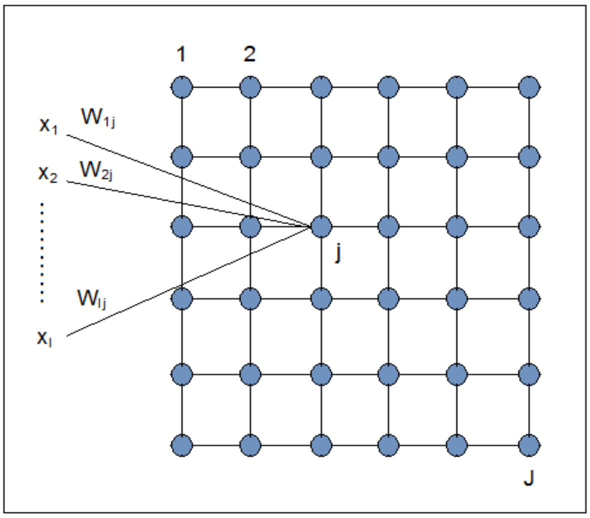

Figure 4. The Kohonen Network (Self-Organizing Maps)

The learning algorithm of SMO contains the following steps:

• Initialize the weights randomly: W_j^0 : $j \in [1 , J]$.

• Presentation of vector (character) $X = (x_1, x_2 \ldots\ldots x_T)^t$ in input to the current iteration n and calculating its distance from each of the vectors W_j^n : $j \in [1 , J]$.

• Selection of winner neuron j^* that is nearest to the input X by computing the distances:

$$d_j^2 = \sum_{i=1}^{I}(x_i(t) - w_{ij}(t))^2 \quad (27)$$

• Update he weights W_j^n .

The result obtained W^* after the training (learning) phase is a memory containing a set of an optimal weight vectors that are very nearest to each input vector $X = (x_1, x_2 \ldots\ldots x_T)^t$.

Concerning the classification phase, the Euclidean distance is calculated between the test character and each one of optimal weight vectors. The recognition will be attributed to label of neuron which its weight vector is nearby to test character.

5.1.3 The k Nearest Neighbors

5.1.3.1 The Majority Voting k Nearest Neighbors

K-Nearest-Neighbors (KNN) is an efficient method used in classification problems without any need of learning. It works by calculating the distances between an unknown vector (character of test) and a set of vectors (characters of learning base) whose each of them its class is known. A K-nearest-neighbor classifier takes into account only the K nearest prototypes to the unknown character, and the majority of class values of the K neighbors determine the decision. In the K-nearest neighbor classification, the unknown character is assigned to the class which is most represented, for example the figure 5 shows that the class of unknown vector is the class 2. In this context, it must be noted that the importance of choosing of the number K is indisputable.

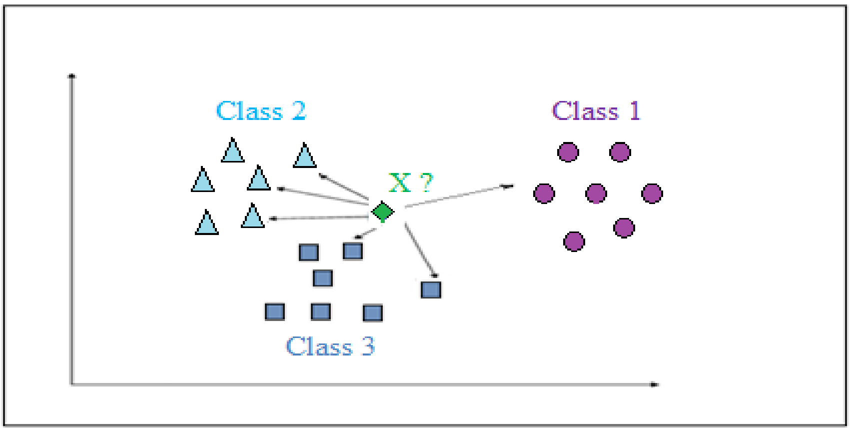

Figure 5. The Majority Voting KNN (K=6).

5.1.3.2 The Weighted Distances k Nearest Neighbors

First of all, x_1 is the first nearest neighbor and x_k is the farthest neighbor of unknown vector X, the idea of this approach is to assign a weight to each of all nearest neighbors defined by:

$$W_i = \begin{cases} \frac{d(X,x_k)-d(X,x_i)}{d(X,x_k)-d(X,x_1)} & \text{if} \quad d(X,x_k) \neq d(X,x_1) \\ 1 & \text{elsewhere} \end{cases} \quad (28)$$

Namely that the distance $W_i \in [0\ 1]$, more exactly, it takes the value 1 for the first nearest neighbors and the value 0 for the farthest neighbor. Whereas after having calculated all the weight W_i, the rule the K-nearest neighbors with weighted distances assigns the unknown character to the class for where in the sum of weighted distances to its representatives in the K-nearest neighbors which presents the greatest value.

5.1.3.3 The Fuzzy K Nearest Neighbors

This method is based on fuzzy logic which consists of calculating all the membership functions $\mu_i(X)$ of an unknown vector X to each of classes C_1, C_2 ...... C_n while using the following formula:

$$\sum_{l=1}^{n} u_i = 1 \quad (29)$$

Where:
j is the est le j^{th} nearest neighbors of X.
k is the total number of all nearest neighbors.
m is a fuzzy parameter.
μ_{ij} is a degree of membership of nearest neighbor to a class C_i which is given by:

$$u_{ij} = \begin{cases} 0{,}51 + 0{,}49\left(\frac{n_j}{K}\right) & \text{if} \quad j = 1 \\ 0{,}49\left(\frac{n_j}{K}\right) & \text{if} \quad j \neq 1 \end{cases} \quad (30)$$

Where:

$$\sum_{j=1}^{k} u_{ij} = 1 \qquad (31)$$

$$u_j(x) = \frac{\sum_{j=1}^{k} u_{ij}\,[d(x, x_j)\}^{\frac{2}{1-m}}}{\sum_{j=1}^{k}[d(x, x_j)\}^{\frac{2}{1-m}}} \qquad (32)$$

The value n_j designates the number of neighbors in the class C_j.
While the recognition of the unknown vector X is assigned to the class that presents the greatest value of μ_i (X).

5.2 Hybrid Classifiers

5.2.1 Hidden Model Markov + k Nearest Neighbors

The idea of this classifier consists to carry out the learning phase by HMM while the classification is will be performed by virtue of KNN. More exactly after having trained each Markovian initial model of every character by the Baum-Welch algorithm for maximizing the likelihood probability that generates the observation modeling this character, all these trained models (optimal models) as well as all likelihood probabilities must stocked for forming a learning base. Afterwards, the classification of each unknown characters will be performed by KNN classifier with their tree different versions which are majority voting, weighted distances and fuzzy.

5.2.2 Self-organizing Maps + K Nearest Neighbors

We follow the same procedure as that of first hybrid classifier, that is to say the learning is performed by SOM while the classification will be carry out by KNN also with their tree different versions.

6. Experiments and Results

Firstly, we present an image of some printed multi-oriented, multi-scaled and noisy Greek characters.

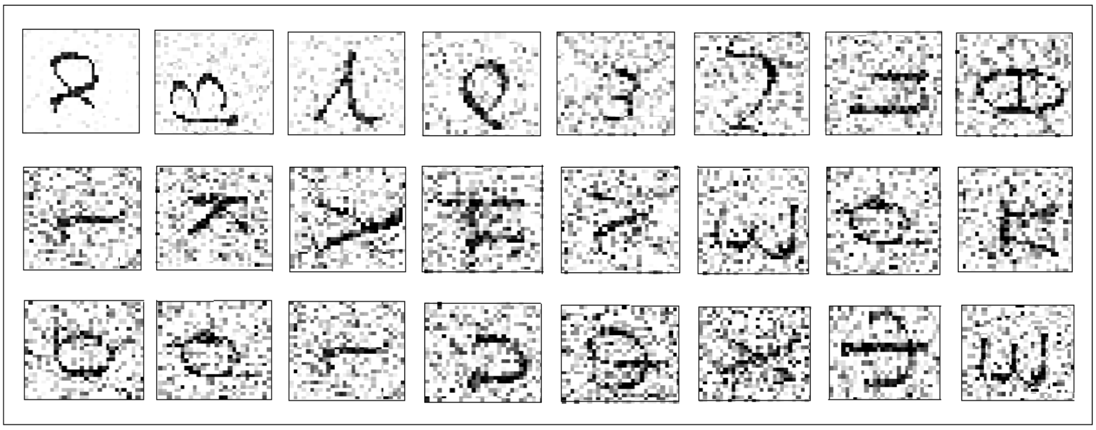

Figure 6. Example of Greek Characters of Test

In fact, the desired goal in this study is to compare between the performances in terms of precision, rapidity and stability of the following classifiers:

- Hidden Model Markov (HMM).
- The Kohonen network or Self-Organizing Maps (SOM).
- Hidden Model Markov combined with K nearest neighbours (HMM + KNN).
- Self-Organizing Maps combined with K nearest neighbours (SOM + KNN).

We note that in this framework that each K nearest neighbours classifier is used in their tree different versions which are majority voting (MKNN), weighted distances (WKNN) and Fuzzy (FKNN). For this finality, we have chosen the following data's:

- Each numeral image has a size equal to 50x50 pixels.
- The number of all images of learning and of test that we have used is equal to 5000 images.
- The parameters of Krawtchouk moment are equal to p=0.85 and q=0.70.
- The fuzzy parameter m is equal to 2.
- After several tries, we have chosen the number K of nearest neighbors equal to 10.

Therefore, firstly we present a test Greek character in different situations: translated, rotated or resized and not noisy, then we add increasingly a quantity of noise of type 'Gaussian' just for knowing its effect on the rate recognition of each character and to global rate also. The values of the standard deviation values σ of Gaussian noise are [0, 0.01, 0.02……. 0.29, 0.30] while its mean value is fixed to μ = 0.05.

Hence, we grouped the values that we have obtained of the recognition rate τ_c of each character (C) for each classifier (given in %) and the global time of execution t_g (of all characters) (given in second s) :

Table 1. The Recognition Rate τ_c and Global Time of Execution t_g for Each Classifier

C	τ_C(HMM) Forward	τ_C(HMM + KNN)			τ_C(SOM)	τ_C(SOM + KNN)		
		MKNN	WKNN	FKNN		MKNN	WKNN	FKNN
α	65.20	70.25	71.34	75.25	72.34	74.20	75.34	79.55
β	79.34	84.50	85.00	87.34	85.55	90.32	91.32	93.50
γ	69.55	72.87	72.15	75.17	61.40	64.52	65.25	70.44
δ	84.25	88.75	85.53	87.20	85.60	87.10	86.45	87.80
ε	76.80	78.67	79.55	81.67	81.15	83.87	84.67	86.35
ζ	74.22	75.00	77.42	79.85	75.00	77.42	78.50	81.55
η	60.00	62.80	63.20	65.34	63.75	67.74	70.15	74.12
θ	63.67	67.74	64.88	68.85	68.22	70.97	71.75	76.34
ι	75.55	77.34	75.25	78.20	71.25	74.20	75.70	77.80
κ	80.15	84.85	81.34	83.58	88.74	93.55	92.44	94.85
λ	74.87	75.25	74.35	78.95	78.24	80.65	81.45	83.67
μ	80.20	81.45	80.23	82.34	88.34	90.32	91.40	92.55
ν	70.65	74.20	75.45	78.55	74.67	77.42	78.55	81.75
ξ	85.00	90.00	91.34	92.67	90.55	93.55	91.00	92.83
ο	60.25	61.85	62.87	64.45	61.25	64.52	65.75	70.50
π	58.95	62.34	64.34	67.50	63.00	67.74	69.42	72.45
ρ	79.58	84.75	85.25	87.55	88.45	90.32	90.15	91.20

σ	83.00	86.47	88.67	90.15	91.25	93.55	94.00	94.34
τ	75.55	78.50	79.15	82.30	85.45	87.10	88.27	90.87
υ	77.34	79.65	81.34	83.34	78.00	83.87	84.34	85.25
φ	63.45	67.55	68.67	71.45	67.34	70.97	72.67	73.40
χ	57.75	61.78	63.55	67.58	60.67	64.52	67.05	70.67
ψ	70.00	75.55	77.25	81.55	78.50	83.87	85.67	87.20
ω	67.83	70.20	71.85	75.85	70.55	74.20	75.34	78.15
t_g	**565.25**	**623.81**	**633.95**	**640.12**	**508.24**	**553.88**	**566.36**	**572.20**

The graphical representation of recognition rate of each character τ_c is presented as follow:

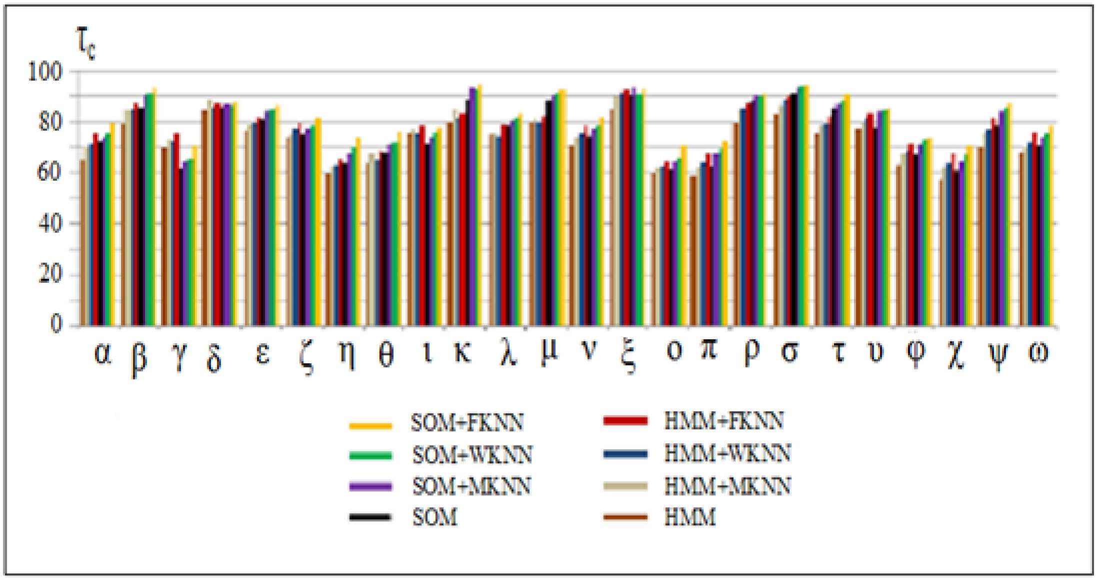

Figure 7. The Graphical Representation of Recognition Rate τ_c of Each Character for Each Classifier

Moreover, the global time of execution t_g for each classifier is presented graphically in the figure bellow:

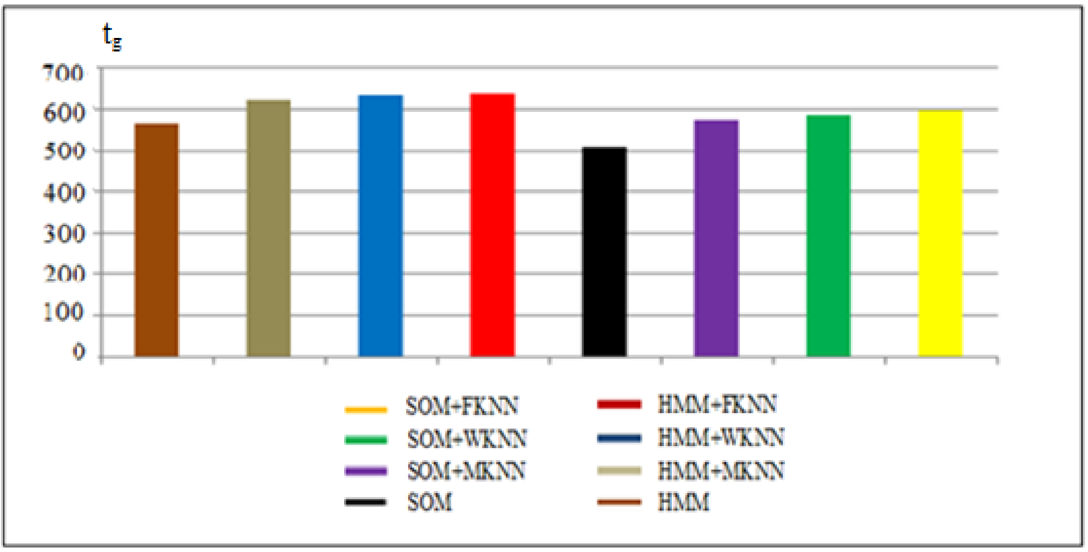

Figure 8. The Graphical Representation of Global Time of Execution τ_g for Each Classifier

We presented also the evolution of the global rate of recognition (of all characters) τ_g (given in %) depending to noise added to each character for each classifier in the following table :

Table 2. The Global Recognition Rate τ_g Depending to Noise Added to Each Character for Each Classifier

Noise	τ_g (HMM) Forward	τ_g (HMM + KNN)			τ_g (SOM)	τ_g (SOM + KNN)		
		MKNN	WKNN	FKNN		MKNN	WKNN	FKNN
0.00	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.01	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.02	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.03	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.04	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.05	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.06	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.07	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.08	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.09	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.10	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.11	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.12	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.13	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.14	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.15	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.16	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.17	96.34	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.18	91.55	100.0	100.0	100.0	100.0	100.0	100.0	100.0
0.19	88.20	95.34	100.0	100.0	94.55	100.0	100.0	100.0

0.20	84.67	90.55	100.0	100.0	89.34	100.0	100.0	100.0
0.21	79.75	83.34	91.67	100.0	83.20	87.50	100.0	100.0
0.22	75.00	80.20	85.25	100.0	78.45	83.34	91.55	100.0
0.23	68.85	71.67	76.80	94.35	70.48	75.00	81.40	100.0
0.24	54.00	58.50	62.45	90.15	58.65	62.50	75.34	95.67
0.25	49.25	53.75	58.70	84.67	54.67	58.34	65.85	87.55
0.26	45.34	50.45	55.34	75.55	50.30	54.18	60.20	78.40
0.27	30.00	35.78	40.67	63.34	35.68	41.68	50.67	69.34
0.28	25.60	30.80	37.85	55.20	30.15	33.34	40.75	60.75
0.29	10.85	15.67	20.35	45.20	15.25	20.84	30.80	54.95
0.30	04.60	07.25	10.70	25.85	07.60	10.00	15.67	30.74

The graphical representation associated to table above is presented in the figure bellow:

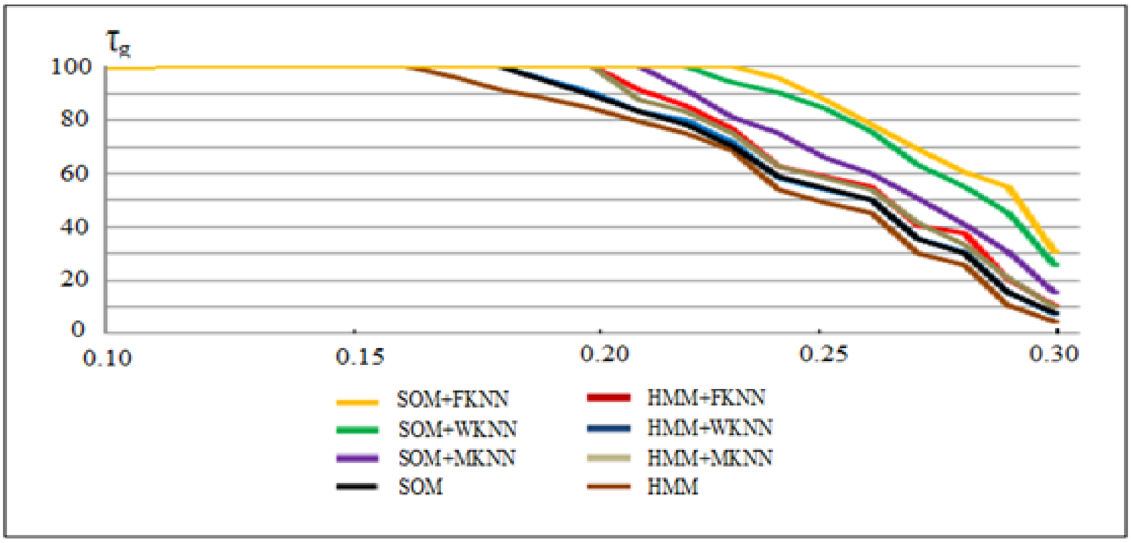

Figure 9. The Graphical Representation of Global Recognition Rate τ_g Depending on Added Noise for Each Classifier

Analysis and comments:

Firstly, in order to better comparing between all these classifiers, we interests to the performances in terms of the precision, the rapidity and the stability, knowing that in this context:

- The precision means to recognizing correctly a certain character translated, rotated or resized but not noisy.
- The rapidity means to recognize in a short time a certain character.
- The stability means to recognizing a certain character correctly but with presence of noise.

Therefore, taking into account all these obtained results, we have concluded that:

- Precision: in the absence of noise, each classifier is very precise (recognition rates are equal to almost 100 %.).
- Rapidity: the fastest classifier is the Kohonen network followed by HMM with Forward followed by KN+MKNN then KN+WKNN then KN+FKNN then HMM+MKNN then HMM+WKNN followed finally by HMM+FKNN.

- Stability: the KN+FKNN is the most stable classifier followed by KN+WKNN then KN+MKNN then HMM+ FKNN then HMM+WKNN then HMM+MKNN finally followed by Forward HMM.
- For each classifier the global recognition rate τ_g is a decreasing function according to noise added to each character. We speak in this time around of an enforced falling of stability of each recognition system. For better to fix this idea, we denote:
- For n=0 (there will be no noise (each character is multi-oriented and multi-scaled but not noisy)):

$$\Delta P = \tau_{g,c1} - \tau_{g,c2} \quad (33)$$

The difference of precision between a classifier C_1 and an another classifier C_2.
For $n > 0$:

$$\Delta S(n) = \tau_{g,c_1}(n) - \tau_{g,c_2}(n) \quad (34)$$

The difference of stability between a classifier C_1 and a classifier C_2 for a value noise n.
Therefore:

- If ΔS (n) > 0 we will have a gain of stability, in this case ΔS (n) is called the rate of growth of stability.
- If ΔS (n) < 0 we will have a losing of stability, in this case ΔS (n) is called the rate of decay of stability.

$$\Delta R(n) = t_{g,c_1}(n) - t_{g,c_2}(n) \quad (35)$$

- The difference of rapidity between a classifier C_1 and a classifier C_2 for a value noise n. Therefore:
- If ΔR (n) > 0 we will have a advancement of rapidity. In this case ΔR (n) is called the rate of decay of rapidity.
- If ΔR (n) < 0 we will have a delay of rapidity. In this case ΔR (n) is called the rate of growth of rapidity (smaller time of execution).

We summarize the values of difference of stability and its associated difference of rapidity between the hybrid classifier SOM+FKNN and each one of the following hybrid classifiers HMM+MVKNN, HMM+WDKNN and HMM+FKNN in the table below:

Table 2. The Different Values of Precision and Rapidity between Several Classifiers

Noise		0.00	0.20	0.25	0.30
(SOM+FKNN) ; (HMM+MKNN)	ΔS	00.00	09.45	33.80	23.49
	ΔR	-88.45	-87.26	-85.41	-51.61
(SOM+FKNN) ; (HMM+WKNN)	ΔS	00.00	00.00	28.85	20.04
	ΔR	-78.55	-77.63	-75.81	-61.75
(SOM+FKNN) ; (HMM+FKNN)	ΔS	00.00	00.00	02.88	04.89
	ΔR	-71.12	-70.56	-69.34	-67.88

Analysis and comments:

Considering the results obtained in table above, we conclude that for example into the interval of noise [0 0.30]:

- When we substitute HMM+MKNN by SOM+FKNN, we will have in the same time a gain of stability equal to 23.49 % and a gain of rapidity equal to 51.61 seconds.
- When we substitute HMM+WKNN by SOM+FKNN, we will have in the same time a gain of stability equal 20.04 % and a gain of rapidity equal to 61.75 seconds.

7. Conclusion

This paper presents an approach investigating the application of several classifiers which are the hidden Markov model and the Kohonen network, then each one of them is combined with K nearest neighbors with their tree different versions which are the majority voting, the weighted distances and fuzzy for recognition of printed multi-oriented, multi-scaled and noisy Greek characters. In fact the aim of this study is to compare between the performances in regards to the precision, the rapidity and the stability of all these classifiers. For this purpose we have used for pre-processing each character image the median filter and the thresholding technique while in order to extract efficiently their features we have employed the Krawtchouk invariant moment. The simulation result that we obtained provides that the hybridization between the Kohonen network and the K nearest neighbors is more performing than that between this last classifier and the hidden Markov model.

References

[1] R. S. S. Safi and B. Bouikhalene, “A Comparative Study between the Hidden Markov Models and the Support Vector Machines for Noisy Printed Numerals Latin Recognition”, International Journal of Innovation and Scientific Research, ISSN 2351-8014, vol. 5, no. 1, (**2014**) July, pp. 16-24 © 2014 Innovative Space of Scientific Research Journals.

[2] R. S. S. Safi and B. Bouikhalene, “Printed Eastern Arabic Noisy Numerals Recognition Using Hidden Markov Model and Support Vectors Machine”, International Journal of Innovation and Applied Studies, ISSN 2028-9324, vol. 9, no. 3, (**2014**) November, pp. 1032-1042 © 2014 Innovative Space of Scientific Research Journals.

[3] R. S. S. Safi and B. Bouikhalene, “A Comparative Study between the Pseudo Zernike and Krawtchouk Invariants Moments for Printed Arabic Characters Recognition”, JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE, vol. 6, no. 1, (**2014**) February.

[4] R. S. S. Safi and B. Bouikhalene, “Printed Arabic Noisy Characters Recognition Using the Multi-layer Perceptron”, International Journal of Innovation and Scientific Research, ISSN 2351-8014, vol. 9, no. 1, (**2014**) September, pp. 61-69, c 2014 Innovative Space of Scientific Research Journals.

[5] R. S. S. Safi and B. Bouikhalene, “Isolated Handwritten Roman Numerals Recognition using Dynamic Programming”, Naïve Bayes and Support Vectors Machines, International Journal of Computer Applications (0975 – 8887), vol. 113, no. 19, (**2015**), March.

[6] R. Salouan, S. Safi and B. Bouikhalene, “A Novel Comparison between the Performances of Several Invariant Moments for Printed Multi-Oriented, Multi-Scaled and Noisy Eastern Arabic Numerals Recognition”, British Journal of Mathematics & Computer Science, vol. 9, no. 6, pp. 516-536, (**2015**), Article no.BJMCS.2015.222 ISSN: 2231-0851 science domain international.

[7] S. A. Mahmoud and S. M. Awaida, “Recognition of off-line handwritten Arabic (Indian) numerals using multiscale features and supports vectors machines vs. hidden Markov model”, The Arabian Journal for Science and Engineering, vol. 34, no. 2B, (**2009**), October.

[8] M. Mohamed and P. Gader, “Handwritten word recognition using segmentation-free hidden Markov modeling and segmentation-based dynamic programming techniques”, IEEE Trans. Pattern Anal.Mach. Intell., vol. 18, no. 5, (**1996**) May, pp. 548–554.

[9] A. H. Hassin, X. Tang, J. Liu and W. Zhao, “Printed Arabic character recognition using HMM”, J. Comput. Sci. Technol., vol. 19, no. 4, (**2004**) July, pp. 538–543.

[10] R. S. S. Safi and B. Bouikhalene, “A Comparison between the Self-Organizing Maps and the Support Vector Machines for Handwritten Latin Numerals Recognition”, International Journal of Innovation and Scientific Research, ISSN 2351-8014, vol. 7, no. 1, (**2014**) August, pp. 50-56© 2014 Innovative.

[11] N. Mezghani1 and A. Mitiche, “A Gibbsian Kohonen Network for Online Arabic Character Recognition”, G. Bebis et al. (Eds.): ISVC 2008, Part II, LNCS 5359, pp. 493–500, 2008.c Springer-Verlag Berlin Heidelberg 2008.

[12] P. Agarwal, "Hand-written character recognition using Kohonen Network", IJCSYT, vol. 2, no. 3, (**2011**) September.
[13] R. I. Gandhi and A. K. Iyakutti, "Attempt to Recognize Handwritten Tamil Character Using Kohonen SOM", Int. J. of Advance d Networking and Applications, vol. 01, no. 03, (**2009**), pp. 188-192.
[14] B. El Kessab, C. Daoui, B. Bouikhalene and R. Salouan, "A Comparative Study between the Support Vectors Machines and the K-Nearest Neighbors in the Handwritten Latin Numerals Recognition", International Journal of Signal Processing, Image Processing and Pattern Recognition, vol. 8, no. 2, (**2015**), pp. 325-336.
[15] B. El Kessab, C. Daoui, B. Bouikhalene and R. Salouan, "A Comparative Study Between the K-Nearest Neighbors and the Multi-Layer Perceptron for Cursive Handwritten Arabic numerals Recognition", International Journal of Computer Applications (0975 – 8887) vol. 2014.
[16] B. El Kessab, C. Daoui, B. Bouikhalene and R. Salouan, "A Comparison between the Performances of Several Distances for Isolated Handwritten Arabic Numerals Recognition", International Journal of Signal Processing, Image Processing and Pattern Recognition, vol. 8, no. 6, (**2015**), pp. 9-14.
[17] P. Dymarski, "Hidden Markov models, theory and applications", InTech Janeza Trdine 9, 51000 Rijeka, Croatia, Copyright © 2011 In Tech.
[18] T. Kohonen, "Self-Organizing Maps", 3rd edition, Springer, Berlin, (**2001**).
[19] S. A. Dudani, "The Distance-Weighted k-Nearest-Neighbor Rule", IEEE Transactions on Systems, Man, and Cybernetics, vol. 6, no. 4, (**1976**), pp. 325-327.
[20] J. M. Keller, M. R. Gray, and J. A. Givens, "A Fuzzy K-Nearest Neighbor Algorithm", IEEE Transactions on Systems, Man, and Cybernetics, vol. 15, no. 4, (**1985**), pp. 580-585.
[21] A Jozwik, "A learning scheme for a fuzzy k-nn rule", Pattern Recognition Letters, vol. 1, (**1985**), pp. 287–289.
[22] P.-T. Yap and R. Paramesran, "Senior Member IEEE, and Seng-Huat Ong. Image analysis by Krawtchouk moments, IEEE transactions on image processing, vol. 12, no. 11, (**2003),** November.

Authors

R. Salouan, he received his Master's degree in 2010 from Faculty of Sciences and Technology University Sultan Moulay Slimane Beni Mellal Morocco, currently working on his Ph. D in Sultan Moulay Slimane University. His current research interests include pattern recognition, image analysis, document processing and automatic processing of natural languages using hidden Markov models and neural networks.

S. Safi, he received the B.Sc. degree in physics (option electronics) from Cadi Ayyad University, Marrakech, Morocco in 1995, M.Sc. and Doctorate degrees from Chouaib Doukkali University and Cadi Ayyad University, Morocco, in 1997 and 2002, respectively. He has been a professor of information theory and telecommunication systems at the National School for Applied Sciences, Tangier Morocco, from 2003 to 2005. Since 2006, he is a professor of applied mathematics and programming at the Faculty of Science and Technics, Beni Mellal Morocco. In 2008 He received the Ph.D. degree in telecommunication and informatics from the Cadi Ayyad University. His general interests span the areas of communications and signal processing, estimation, time-series analysis, and system identification – subjects on which he has published 10 journal papers and more than 40 conference papers. His current research topics focus on transmitter and receiver diversity techniques for single-and multi-user fading communication channels, and wide-band wireless communication systems.

B. Bouikhalene, he received his PhD degree on mathematics in 2001 and Master's degree on Science of Computer and Telecommunications in 2007 from the University Ibn Tofail Kenitra. He is currently is a professor in the Sultan Moulay Slimane University Beni Mellal Morocco. His research topics are: the pattern recognition, artificial intelligence and mathematics and its applications.