

Design of a Hybrid Classifier for Natural Textures in Images from the Bayesian and Fuzzy Paradigms

¹María Guijarro, ¹Gonzalo Pajares, ²Raquel Abreu, ¹Luis Garmendia, ³Matilde Santos

¹Dpto. Ingeniería del Software e Inteligencia Artificial
Facultad Informática. Universidad Complutense. Madrid. Spain
28040 Madrid, SPAIN
Phone: +34.1.3 94 75 46
Fax: +34.1.3 94 75 29
E-mail: pajares@fdi.ucm.es

²Dpto. Informática y Automática, Escuela Técnica Superior de Informática. UNED. Madrid. Spain

³Dpto. Arquitectura Computadores y Automática, F. Informática, U. Complutense. Madrid. Spain

Abstract – One objective for classifying textures in natural images is to achieve the best performance possible. The combination of multiple classifiers has been tested as a suitable technique. This is because the individual behaviors of the different classifiers are exploited by joining performances. Two main problems are to be addressed for combining different classifiers. First, which set of classifiers are to be selected? Second, which is the best combination method? Different strategies have been proposed focusing on both problems. We propose a new automatic approach based on two well tested classifiers, the non-parametric Parzen's windows and the Fuzzy clustering. A probability density function (PDF) is estimated through the Parzen's windows estimator to be embedded in the Bayesian framework. The decision is made by combining a likelihood value provided by the PDF and the a priori probability supplied by the Fuzzy clustering strategy through the membership degree. Experimental results for aerial images demonstrate the performance of the proposed design.

Keywords – Hybrid classifier, natural textures, Parzen windows, Bayesian, Fuzzy k-Means.

I. INTRODUCTION

Nowadays the increasing technology of aerial images is demanding solutions for different image-based applications. The natural texture classification is one of such applications because of the high image spatial resolutions. The areas where textures are suitable include agricultural crop ordination, forest areas determination, urban identifications and damages evaluation in catastrophes or dynamic path planning during rescue missions or intervention services also in catastrophes (fires, floods, etc.) among others.

A line of research in texture classification has been directed towards the performance analysis for different classifiers including Bayesian, K-Nearest, Neural Networks or Learning Vector Quantization [1,2,3] among others. It is commonly accepted that the combination of classifiers performs better than if they are used individually. Two main strategies are commonly accepted for combining classifiers, namely: selection and fusion [4]. In selection, each individual classifier provides a decision and a unique classifier is selected as the best according to a maximum value or any previously fixed criterion. In fusion the classifiers are combined. Different strategies have been proposed for combining classifiers [5,6,7], including majority voting, max, min or median rules.

The behavior of different features has been also studied in texture classifications, where the set of features describes each pattern [4,8,9]. There are pixel-based and region-based approaches. A pixel-based approach tries to classify each pixel as belonging to one of the classes. The region-based identifies patterns of textures within the image and describes each pattern by applying filtering (laws masks, Gabor filters, Wavelets, etc.), it is assumed that each texture displays different levels of energy allowing its identification at different scales [8,10,11].

The aerial images used in our experiments do not display texture patterns. This implies that textured regions cannot be identified. This paper is focused on the fusion of two classifiers under a pixel-based approach. The performance of features is out of its scope.

One of the best classification approaches is the Bayesian one [8]. This classifier is based on the Bayesian decision theory. Given an observed value, a posterior probability is

computed based on this observation and a prior knowledge. The decision is made according to the posterior probability. The main drawback in the computation of the posterior probability is the prior ignorance in many applications, i.e. the mapping of the prior knowledge. This implies that the prior probability is unknown.

Some Bayesian strategies solve this problem assuming that given an observation, it has the same prior probability to be classified as belonging to any class. On the contrary, we propose a new approach assuming that this prior probability can be supplied by the fuzzy classifier based on the computation of a membership degree. This makes the main finding of this paper, where two classifiers are combined under the Bayesian paradigm obtaining the proposed hybrid classifier. It performs better than the simple classifiers. We have chosen the fuzzy classifier because our textures display some degree of uncertainty (fuzziness). Indeed, pixels which are spatial neighbors could have spectral signatures.

The paper is organized as follows. Section II describes the design of the automatic classifier, where the combination of the Fuzzy and Bayesian classifiers is explained. Section III shows experimental and comparative results. Finally, in the section IV some concluding remarks are presented.

II. AUTOMATIC HYBRID CLASSIFIER

Our system works in two stages: 1) performing a training process with a set of patterns; 2) performing a classification process, where the incoming new patterns are classified as belonging to one class. Figure 1 shows both processes.

In both processes (training and classification) each pattern is characterized by a feature vector \mathbf{x} . As mentioned before, in this paper we use a pixel-based approach and taking into account that we are classifying multiespectral textured images, we use as the attribute vector the spectral components, i.e. the Red, Green and Blue. The RGB map performs better than other colour representations [12]. So, \mathbf{x} is a 3-dimensional vector representing each pixel, where its components are the red, green and blue values respectively.

A. The training process

During the training phase, we start with the observation of a set X of n training patterns, i.e. $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\} \in \mathbb{R}^3$. Each sample is to be assigned to a given class, where the number of possible classes is c . Each class is identified as w_j , where $j = 1, 2, \dots, c$.

Now, the problem is to assign each pattern sample to a class and compute the cluster prototypes (centers). For such purpose we have chosen the well-tested fuzzy clustering framework [5,7,9], which has been customized and tailored for working in an unsupervised fashion according to the criterion described in [13].

This makes an important contribution with respect to the original fuzzy clustering approach because it becomes an automatic approach performing favourably for the set of images tested.

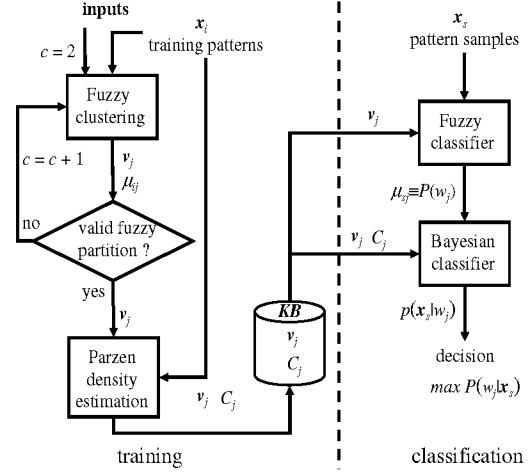


Fig. 1. Classifier combination: training and classification

Fixed the number of clusters c and following [13,14,15], the fuzzy k-means clustering algorithm (FkM) is based on the minimization of the objective function J ,

$$J(U; \mathbf{v}) = \sum_{i=1}^n \sum_{j=1}^c \mu_{ij}^m d_{ij}^2 \quad (1)$$

subject to

$$\mu_{ij} \in [0,1]; \sum_{j=1}^c \mu_{ij} = 1; \sum_{i=1}^n \mu_{ij} < n; 1 \leq j \leq c, 1 \leq i \leq n \quad (2)$$

where $\mathbf{v} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c\}$, $\mathbf{v}_j \in \mathbb{R}^3$ is the j -th cluster center. These cluster centers are to be determined. The $n \times c$ matrix $U = [\mu_{ij}]$ contains the membership grade of pattern i with cluster j ; $d_{ij}^2 = d^2(\mathbf{x}_i, \mathbf{v}_j)$ is the squared Euclidean distance. The number m is called the exponent weight [14]. In order to minimize the objective function (1), the cluster centers and membership grades are chosen so that high memberships occur for samples close to the corresponding cluster center. The higher the value of m , the less those samples whose memberships are low contribute to the objective function. Consequently, such samples tend to be ignored in determining the cluster centers and membership degrees [15].

The original FkM computes for each \mathbf{x}_i at the iteration k its membership grade and updates the cluster centers according to equations (3) and (4),

$$\mu_{ij}(k) = \frac{1}{\sum_{r=1}^c (d_{ir}(k)/d_{ir}(k))^{2/(m-1)}} \quad (3)$$

$$\mathbf{v}_j(k+1) = \frac{\sum_{i=1}^n \mu_{ij}^m(k) \mathbf{x}_i}{\sum_{i=1}^n \mu_{ij}^m(k)} \quad (4)$$

The stopping criterion of the iteration process is achieved when $\|\mu_{ij}(k+1) - \mu_{ij}(k)\| < \epsilon \forall ij$ or a number N of iterations is reached. The number of classes is initially set to 2. Nevertheless, if we have some previous knowledge about this number it can be set to the known value.

The method requires the initialization of the cluster centers, so that the equation (3) can be applied at the iteration $k = 1$. This is achieved through the pseudorandom procedure described in [13], as follows:

- 1) Perform a linear transform $Y = f(X)$ of the training pattern values so that they range in the interval [0,1].
- 2) Initialize $\mathbf{v} = 2D\bar{\mathbf{M}} \circ \mathbf{R} + D\bar{\mathbf{m}}$, where $\bar{\mathbf{m}}$ is the mean vector for the transformed training pattern values in Y and $\bar{\mathbf{M}} = \max(\text{abs}(Y - \bar{\mathbf{m}}))$, both of size 1×3 ;

$D = [1 \dots 1]^T$ with size $c \times 1$; \mathbf{R} is a $c \times 3$ matrix of random numbers in [0,1]; the operation \circ denotes the element by element multiplication.

After the fuzzy clustering process, a partition of the input training patterns is obtained, where each cluster j has associated its center \mathbf{v}_j . Also, for each sample i its corresponding membership grades (μ_{ij}) of belonging to each cluster j , are computed.

The next step consists in the cluster validation. This is carried out by computing the *partition coefficient* for the number of classes specified as follows [15,16],

$$PC(U; c) = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^c (\mu_{ij})^2 \quad (5)$$

The maximum value of PC for different values of c determines the best partition, i.e. the best number of classes for the set of training samples available. Values of PC near the unity indicate that the partition is acceptable. This is because the PC is upper bounded by the unity. Following the scheme in figure 1 and based on the PC value, if the partition is rejected, then increase the number of clusters ($c = c + 1$) and try a new partition; otherwise if it is accepted, each training pattern is assigned to a class according to the maximum membership grade; the cluster centers \mathbf{v}_j are inversely transformed to range in the original input values.

The partition (centers and clusters) is transferred to the Parzen's windows density estimation procedure. Other different validation coefficients could be used [11,14,15], even if we have verified that the classical coefficient proposed in this paper suffices for the set of experiments carried out.

The Parzen's windows density estimation procedure receives the partition, i.e. the number of clusters, the training samples belonging to the clusters and the centers representing the classes.

Given a pattern sample \mathbf{x}_s the goal is to assign it to the class w_j . Under the Bayesian framework [7], the problem is reduced to compute the probability of belonging to w_j given the sample \mathbf{x}_s .

The above is expressed as the *a posterior* probability as follows,

$$P(w_j | \mathbf{x}_s) = \frac{p(\mathbf{x}_s | w_j)P(w_j)}{\sum_{k=1}^c p(\mathbf{x}_s | w_k)P(w_k)} \quad (6)$$

Two main problems to be solved are the estimations of the class-conditional probability density functions $p(\mathbf{x}_s | w_j)$ and the *a priori* probabilities $P(w_j)$ for each class w_j .

Two kinds of approaches are proposed for estimating $p(\mathbf{x}_s | w_j)$, namely [7]: parametric and non-parametric. The parametric assumes a known distribution (generally Gaussian), the non-parametric does not need this assumption. This justifies our choice.

The Parzen estimate of the class-conditional probability density function at \mathbf{x} is given by [17,18,19],

$$p(\mathbf{x} | w_j) = \frac{1}{n_j} \sum_{k=1}^{n_j} K_h(\mathbf{x} - \mathbf{x}_k) \quad (7)$$

where $K_h(z) = K(z/h)/V_h$, $K(\cdot)$ is a window or kernel function with $\int_{-\infty}^{+\infty} K(t)dt = 1$ (to ensure the area under the probability density function is 1 and h is the window-width or smoothing parameter, determining the spread of the kernel function, $(h > 0)$ [20]. Typically, it is a monotonically decreasing function g of the distance of its argument to the sample \mathbf{x}_j . That is, $K(\mathbf{x} - \mathbf{x}_k) = g(d(\mathbf{x}, \mathbf{x}_k))$, where $d(\mathbf{x}, \mathbf{x}_j)$ is a metric distance. The statistical properties of this estimator can be found in [7].

The equation (7) shows that the contribution of each training pattern \mathbf{x}_k , toward the density estimate at a point \mathbf{x} is determined by the width and the shape of the kernel. For example, if the window is Gaussian then patterns that fall close to \mathbf{x} contribute more toward the estimate of the density at \mathbf{x} than the patterns that are far away from \mathbf{x} . Such equations can be interpreted as centring the kernel K over each training pattern and summing the overlapping functions at each point \mathbf{x} in the space.

The shape of the kernel is not as important as its width in density estimation. There are different types of kernels functions [7]. We have chosen the commonly used Gaussian kernel by the following reasons. The first is because with such a kernel, we have information about the cross-correlation between the attributes of the training patterns through the embedded covariance matrix C_j , where the sub-index j identifies the cluster. The second reason is its convenient analytical properties. Hence, we get the following density estimate,

$$p(\mathbf{x} | w_j) = \frac{1}{n_j} \sum_{k=1}^{n_j} \left\{ \frac{\exp\{-D(\mathbf{x}; \mathbf{x}_k, h_k)\}}{(2\pi)^{d/2} h_j^{n_j} |C_j|^{1/2}} \right\} \quad (8)$$

where $D(\cdot) = (\mathbf{x} - \mathbf{x}_k)^t C_j^{-1} (\mathbf{x} - \mathbf{x}_k) / 2h_j^2$; d is the dimension of the pattern space (which in this paper is 3) as \mathbf{x} is a 3-dimensional vector; t denotes transpose. Each kernel function considers that the samples are around a theoretical pattern \mathbf{x}_j where the average vector for each kernel is the pattern itself. The smoothing parameter h is often expressed as a function of the number of patterns,

$$h_j = hn_j^{-\frac{r}{d}} \text{ for } 0 \leq r \leq 0.5 \quad (9)$$

The choice of the bandwidth h is very critical in Parzen density estimation [19]. An overlay small h gives a spiky or noisy estimate of $p(\mathbf{x}/w_j)$, with each spike corresponding to the kernel itself at the training patterns. When h is very large, each training pattern provides essentially the same contribution toward density estimation at every point \mathbf{x} and the result is an over smoothed estimate of $p(\mathbf{x}/w_j)$. As shown in [19], the window-width parameter appears to be fixed according to the set of data from the images processed. A more in-depth discussion of window width may be found in [7,18]. Indeed, in [18] we can find a study in the bi-dimensional case, as in this paper, of Parzen's window estimates of a Gaussian kernel for both n_j and h values, where acceptable results are obtained with $n_j = 256$ and $h = 1$, although the best results are obtained for $n_j \rightarrow +\infty$ for any h . From this study we have established the optimal h_j value through the equation (9) with $n_j = 256$, $h = 1$ and for each r in steps of 0.1 ranging in the interval $[0,0.5]$. Then, during the *training* process n_j is known for each set of training samples, hence we can compute h from the equation (9) by varying r also in steps of 0.1 in the range $[0,0.5]$ and using the h_j obtained above as a function of r . We have tested different values for r , obtaining the best performance for $r = 0.4$.

Finally, the computation of the covariance matrix associated to each cluster is established,

$$C_j = \frac{1}{n_j - 1} \sum_{k=1}^{n_j} (\mathbf{x}_k - \mathbf{v}_j)(\mathbf{x}_k - \mathbf{v}_j)^t \quad (10)$$

where t denotes transpose.

The \mathbf{v}_j and C_j are stored in the Knowledge Base (*KB*), Fig. 1, so that they can be recovered during the classification process.

B. The classification process

During the classification process new images and consequently new texture patterns are to be processed by the system. With such purpose, we recover the \mathbf{v}_j cluster centers stored in *KB* during the training process.

Now, given a new pattern sample \mathbf{x}_s , the problem is to decide which the class it belongs is. This is carried out by applying the Bayes rule given in the equation (6). The decision is made through the equation (11),

$$\mathbf{x}_s \in w_j \text{ if } P(w_j | \mathbf{x}_s) > P(w_k | \mathbf{x}_s) \quad \forall w_k | w_k \neq w_j \quad (11)$$

The equation (11) can be re-written avoiding the denominator in (6), as it appears in both members of the inequality,

$$\mathbf{x}_s \in w_j \text{ if } p(\mathbf{x}_s | w_j)P(w_j) > p(\mathbf{x}_s | w_k)P(w_k) \quad (12)$$

The probability density functions are estimated according to the equation (8) and the a priori probabilities are assumed to be the membership grades supplied by the Fuzzy classifier, which are computed through the following expression,

$$P(w_j) = \mu_{sj} = \frac{1}{\sum_{p=1}^c (d_{sj}/d_{sp})^{2/(m-1)}} \quad (13)$$

where d_{sj}^2 , d_{sp}^2 are the squared Euclidean distances between \mathbf{x}_s and the corresponding cluster centers \mathbf{v}_j , \mathbf{v}_p .

III. COMPARATIVE ANALYSIS AND PERFORMANCE EVALUATION

We have used a set of 36 digital aerial images acquired during May in 2006 from the Abadia region located at Lugo (Spain). They are multispectral images with 512x512 pixels in size. The images are taken at different days from an area with several natural textures. The initial training patterns are extracted from 20 images selected randomly from the set of 36. The remainder 16 images are used for testing and four sets, S0, S1 S2 and S3 of 4 images each one, are processed during the test according to the strategy described below. The images assigned to each set are randomly selected from the 16 images available.

A. Design of a test strategy

In order to assess the validity and performance of the proposed approach we have designed a test strategy with two purposes: 1) to verify the performance of our approach as compared against some existing strategies; 2) to study the behaviour of the method as the training (i.e. the learning) increases.

To compare the performance of our Bayesian and Fuzzy (BF) combined method against single approaches, we select the following non-fusion (without combination) strategies: a) the Fuzzy Clustering (FC) approach based on the membership grade computed as described through the equation (13); b) the Bayesian framework with the Parzen's windows (BP) estimation based on maximum a posterior probability according to equation (6) assuming identical a priori probabilities for all classes and c) the Learning Vector Quantization (VQ) [7].

The performance of our combined fusion strategy is also compared against the following existing classical combination approaches [5,6]: a) max rule (MA); b) min rule (MI) and mean rule (ME). When the number of classifiers to be fused is

greater than two, the median is used instead of the mean because of its robustness [6]. We have not tested the classical majority vote because we are only using two classifiers and majority is only suitable for a number of classifiers greater than two.

The FC method gives a membership grade, μ_{sj} , for each class $j = 1, 2, \dots, c$ and the BP a probability density value $P(x_s | w_j)$ also for each class, both ranging in $[0, 1]$.

Hence, given the pattern sample x_s , we assign it to w_j according to the decisions given in equations (14), (15) and (16) for max, min and median rules respectively,

$$\text{Max rule: } \max_{j=1}^c \max\{\mu_{sj}, P(x_s | w_j)\} \quad (14)$$

$$\text{Min rule: } \min_{j=1}^c \max\{\mu_{sj}, P(x_s | w_j)\} \quad (15)$$

$$\text{Mean rule: } \max_{j=1}^c \frac{1}{2} [\mu_{sj} + P(x_s | w_j)] \quad (16)$$

The test is carried out according to the following steps:

STEP 0 (initial training): for each image (from the 20 available) we perform a downsampling by 4, i.e. we obtain $20 \times 128 \times 128$ training samples.

For $j = 1$ to $c = 6$ (maximum number of classes allowed) validate the partition by computing the partition coefficient PC through the equation (5) and determine the best partition (number of classes and centers). These classes are used for classifying the pattern samples during the next steps.

STEP 1: given the images in S_0 and S_1 , classify each pixel as belonging to a textured class, which has been identified previously, according to the BF, FC, BP, VQ, MA, MI and ME methods. Compute the percentage of successes according to the ground truth defined for each class at each image. The classified pattern samples from S_1 are added to the previous training samples and a new training process is carried out. The set S_0 is used as a pattern set in order to verify the performance of the training process as the learning increases.

Perform the same process for STEPs 2 and 3 but using the sets S_2 and S_3 respectively instead of S_1 . Note that S_0 is also processed as before.

As one can see the number of training samples added at each STEP is $4 \times 512 \times 512$ because this is the number of pixels classified during the STEPs 1 to 3 belonging to the sets S_1 , S_2 and S_3 .

B. Analysis of Results

Table I shows the partition coefficient values against the number of classes used during the initial training process in the STEP 0. One can see that the best number of classes is 4 because of the maximum value is obtained for this number. Hence, this is the number of clusters used.

Figure 2 displays: (a) a representative original image to be classified belonging to the set S_0 ; (b) the ground truth for the

cluster number two; (c) the cluster correspondence between the natural color for each cluster center and the labels assigned each cluster based on a previous fixed color map; (d) the labeled image after the classification with our BF method.

TABLE I.
PARTITION COEFFICIENT (PC) AGAINST THE NUMBER OF CLUSTERS

PC	Number of Clusters				
	2	3	4	5	6
	0.36	0.68	0.87	0.77	0.61

Each ground truth is built by applying the fuzzy classifier and then modifying the results manually according to the human expert criterion.

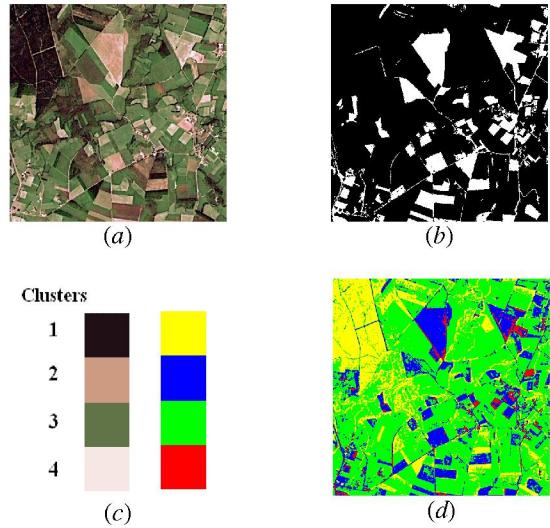


Fig. 2. (a) original image; (b) ground truth for the cluster two; (c) clusters colors and their corresponding labels; (d) labeled image classification obtained by the BF approach

The color for each class, Figure 2(c), matches with the color assigned to the corresponding cluster center according to a previously defined color map. The labels are artificial colors identifying each cluster. The correspondence between labels and the different textures is as follows:

- 1) yellow with forest vegetation
- 2) blue with bare soil
- 3) green with agricultural crop vegetation
- 4) red with buildings and man made structures

Table II shows the percentage of successes in terms of correct classifications obtained for the different methods. For each STEP we show both sets of testing images processed. These percentages are computed taking into account the correct classifications for the four clusters according to the ground truth.

In order to clarify the performance of the BF for the set S_0 against the remainder methods the values in table II are displayed graphically in the figure 3; the percentage of successes of BF against single methods (FC, BP, VQ) is shown in (a) and fusion-based methods (MA, MI, ME) in (b).

TABLE II.

PERCENTAGE OF SUCCESSES AT EACH STEP FOR THE FOUR SETS OF TESTING IMAGES (SP0, SP1, SP2 AND SP3)

% STEP 1	STEP 2		STEP 3			
	SP0	SP1	SP0	SP2	SP0	SP3
BF	79.4	79.2	88.7	86.2	92.1	92.7
FC	68.2	69.5	77.2	74.3	81.2	81.9
BP	69.1	72.3	78.5	76.0	83.2	83.8
VQ	65.3	65.7	71.1	69.2	73.1	74.2
MA	70.2	73.1	79.3	78.4	87.6	88.4
MI	64.4	65.2	68.0	67.3	68.1	73.3
ME	72.5	71.9	82.3	78.8	85.3	86.9

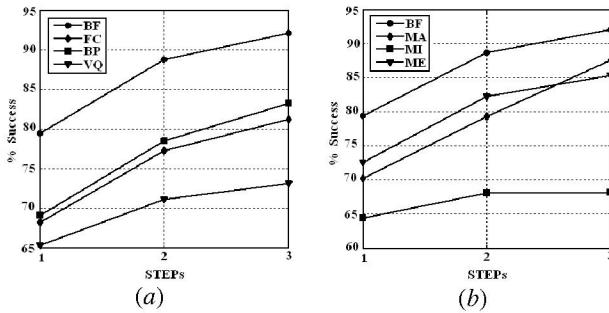


Fig. 3. Percentage of successes for the SP0 in STEPs 1 to 3 (a) BF against single methods (FC, BP, VQ); (b) BF against fusion-based methods (MA, MI, ME)

From results in table II, i.e. figure 3 one can see that the best performance is achieved with our proposed BF approach in both, single and fusion strategies. The MA and ME fusion methods perform favourably, surpassing the performance of single approaches. This means that fusion strategies are suitable. As the learning increases through STEPs 1 to 3 the performance improves. This means that the learning phase is important.

IV. CONCLUSIONS

We propose a new hybrid and automatic making decision process for classifying natural textures. The proposed method is based on the well tested Bayesain framework where the a priori probability is provided through a FkM approach, customized to operate under an unsupervised mode. The posterior probability for making the decision is based on the computation of a probability density function through the Parzen's windows.

The performance of our approach is compared against the single and fusion-based strategies, verifying that it performs favourably in the set of aerial images tested, i.e. that fusion based approaches are suitable.

This approach is applicable to other textured images.

ACKNOWLEDGMENT

The authors would like to thank to SITGA (Servicio Territorial de Galicia) in collaboration with the Dimap company (<http://www.dimap.es/>) for the original aerial images supplied and used in this paper.

REFERENCES

- [1] G. Giacinto, F. Roli and L. Bruzzone, "Combination of neural and statistical algorithms for supervised classification of remote-sensing image," *Pattern Recognition Letters*, vol. 21, no. 5, pp. 385-397, May 2000.
- [2] J.C.W. Chan, N. Laporte and R.S. Defries, "Texture Classification of logged forest in tropical Africa using machine-learning algorithms," *Int. J. Remote Sensing*, vol. 24, n° 6, pp. 1401-1407, March 2003.
- [3] F. del Frate, F. Pacifici, G. Schiavon, C. Solimini, "Use of Neural Networks for Automatic Classification from High-Resolution Images," *IEEE Trans. Geoscience and Remote Sensing*, vol. 45, n° 4, pp. 800-809, April 2007
- [4] R.M. Valdovinos, J. S. Sánchez and R. Barandela, "Dynamic and Static weigthing in classifier fusion," in *Pattern Recognition and Image Analysis*, Lecture Notes in Computer Science (J.S. Marques, N. Pérez de la Blanca and P. Pina, Eds.), Springer-Verlag, Berlin. 2005, pp. 59-66.
- [5] L.I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, Wiley, 2004.
- [6] J. Kittler, M. Hatef, R.P.W. Duin and J. Matas, "On Combining Classifiers" *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226-239, March 1998.
- [7] R.O. Duda, P.E. Hart and D.S. Stork, *Pattern Classification*, Wiley, 2000.
- [8] D. Puig and M.A. García, "Automatic texture feature selection for image pixel classification," *Pattern Recognition*, vol. 39, n° 11, pp. 1996-2009, November 2006.
- [9] M. Hanmandlu, V.K. Madasu and S. Vasikarla, "A Fuzzy Approach to Texture Segmentation," in *Proc. of the IEEE International Conference on Information Technology: Coding and Computing (ITCC'04)*, The Orleans, Las Vegas, Nevada, USA, April. 2004, pp. 636-642.
- [10] T. Randen and J.H. Husøy, "Filtering for Texture Classification: A Comparative Study," *IEEE Trans. Pattern Analysis Machine Intell.*, vol. 21, n° 4, pp. 291-310, April, 1999.
- [11] P. Maillard, "Comparing Texture Analysis Methods through Classification," *Photogrammetric Engineering and Remote Sensing*, vol. 69, n° 4, pp. 357-367, April, 2003.
- [12] A. Drimbarean and P.F. Whelan, "Experiments in colour texture analysis," *Pattern Recognition Letters*, vol. 22, n° 4, pp. 1161-1167, April 2003.
- [13] B. Balasko, J. Abonyi and B. Feil, Fuzzy Clustering and Data Analysis Toolbox for Use with Matlab, Veszprem University, Hungary
[URL:<http://www.fmt.vein.hu/softcomp/fclusttoolbox/FuzzyClusteringToolbox.pdf>](http://www.fmt.vein.hu/softcomp/fclusttoolbox/FuzzyClusteringToolbox.pdf)
- [14] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Kluwer, Plenum Press, New York, 1981.
- [15] H.J. Zimmermann, *Fuzzy Set Theory and its Applications*, Kluwer Academic Publishers, Norwell, 1991.
- [16] D.W. Kim, K.H. Lee and D. Lee, "Fuzzy Cluster validation index based on inter-cluster proximity," *Pattern Recognition Letters*, vol. 24, pp. 2561-2574, November 2003.
- [17] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, pp. 1065-1076, 1962.
- [18] A.K. Jain and M.D. Ramaswami, "Classifier design with Parzen windows," in *Pattern Recognition and Artificial Intelligence toward an integration*, E.S. Geselma and L.N. Kanal (Eds.) Amsterdam, The Netherlands, 1988, pp. 211-218.
- [19] R.P.W. Duin, "On the choice of smoothing parameters for Parzen estimators of probability density functions," *IEEE Trans. Comput.*, C-25, pp. 1175-1179.
- [20] K. Fukunaga and M.D. Hummels, "Bayes error estimation using Parzen and K-NN procedures," *IEEE Trans Pattern Anal. Machine Intelligence*, vol. 9, n° 5, pp. 634-643, May 1987.