Published in IET Computer Vision
Received on 2nd April 2008
Revised on 19th April 2009
doi: 10.1049/iet-cvi.2008.0023

ISSN 1751-9632

Combining classifiers through fuzzy cognitive maps in natural images

G. Pajares[1] *M. Guijarro*[2] *P.J. Herrera*[1] *A. Ribeiro*[3]

[1]*Departamento de Ingeniería del Software e Inteligencia Artificial, Facultad Informática, Universidad Complutense, 28040 Madrid, Spain*
[2]*Centro Superior de Estudios Felipe II, Ingeniería Técnica en Informática de Sistemas, 28300 Aranjuez, Madrid, Spain*
[3]*Instituto de Automática Industrial, Consejo Superior de Investigaciones Científicas, Arganda del Rey, Madrid, Spain*
E-mail: pajares@fdi.ucm.es

Abstract: A new automatic hybrid classifier for natural images by combining two base classifiers through the fuzzy cognitive maps (FCMs) approach is presented in this study. The base classifiers used are fuzzy clustering (FC) and the parametric Bayesian (BP) method. During the training phase, different partitions are established until a valid partition is found. Partitioning and validation are two automatic processes based on validation measurements. From a valid partition, the parameters of both classifiers are estimated. During the classification phase, FC provides for each pixel the supports (membership degrees) that determine which cluster the pixel belongs to. These supports are punished or rewarded based on the supports (probabilities) provided by BP. This is achieved through the FCM approach, which combines the different supports. The automatic strategy and the combined strategy under the FCM framework make up the main findings of this study. The analysis of the results shows that the performance of the proposed method is superior to other hybrid methods and more accurate than the single usage of existing base classifiers.

1 Introduction

One of the key applications for aerial images is the identification of natural textures. This task can be carried out by applying classification methods. There are many classical base classifiers. To improve the performance of the classification, this paper proposes a new automatic hybrid classifier where two base classifiers can be conveniently combined. The two base classifiers are fuzzy clustering (FC) [1, 2] and the probabilistic parametric Bayesian (BP) method [1]. Combining the two classifiers through the fuzzy cognitive maps (FCMs) [3–7] strategy makes up the main finding of this work. The goal of this paper is the classification of the natural textures existing in aerial images. The method can be applied to other types of images and also with different base classifiers. In this section, we provide more details about the classification of textures in images and justify the use of the fusion strategy for classification under the FCM paradigm.

1.1 Classification of textures in natural images

The increasing applications of aerial imaging demand technological improvements. One such application is natural texture classification. Results for texture in high image spatial resolution are suitable to study crop placement, forest area determination, urban identification, catastrophic damage evaluation, dynamic path planning during rescue operations and intervention services in natural disasters (fires, floods etc.). The proposed approach is applied with data from textured images. The first step is to select features. The behaviour of different features has been studied in texture classifications, with a set of features describing each pattern [8–10]. There are pixel-based [11–13] and region-based approaches [9, 14–17]. Pixel-based approaches attempt to classify each pixel as belonging to one of the clusters. Region-based approaches identify patterns of textures within the image and describe each pattern by applying filtering (laws masks, Gabor filters,

Wavelets etc.). As each texture displays different levels of energy, the texture can be identified at different scales. The aerial images used in our experiments do not display texture patterns, so textured regions cannot be identified. In this paper, we focus on the pixel-based category. Since we are classifying multi-spectral textured images, the spectral components, that is, the red, green and blue (RGB) colour mapping, are used as the attribute vectors. In our experiments we have verified that the RGB map performs better than other colour representations [18], justifying its choice.

1.2 Combination of classifiers

One important conclusion reported in the literature is that the combination of classifiers performs better than simple classifiers [8, 19–23]. Specifically, the studies carried out in [24] and [25] report the advantages of using combined classifiers rather than simple ones. This is because each classifier produces errors on a different region of the input pattern space [26].

Nevertheless, the main problem is: what is the best strategy for combining simple classifiers? This is still an open question. Indeed in [20] it is stated that there is no best combination method. In [22] and [27] a review of different approaches is reported that includes the way in which the classifiers are combined. Some important conclusions are: (a) if only labels are available, a majority vote should be suitable; (b) if continuous outputs such as posterior probabilities are supplied, an average or some other linear combinations are suggested; (c) if the classifier outputs are interpreted as fuzzy membership values, fuzzy approaches could be used; (d) also it is possible to train the output classifier separately using the outputs of the input classifiers as new features. As regards (a), a selection criterion is applied; for (b) and (c), a fusion strategy is carried out; and, in (d) a hierarchical approach is used [8].

Because we have continuous outputs, we propose a new fusion approach that combines the FC and BP base classifiers. As usual, the hybrid classifier involves two phases: training and decision. During the training phase, these two base classifiers are trained and an optimal partition is established from the available data through the FC classifier. This makes the approach automatic. During the decision or classification phase, the base classifiers make individual decisions classifying each pixel in the image as belonging to a kind of texture. The decisions are made according to the supports provided by each classifier (FC gives membership degrees and BP probabilities). Also, during the classification phase, our proposed hybrid approach combines these supports by applying the FCM paradigm and makes decisions based on a new combined support. FCMs are well-suited methods for dynamic systems [28]; they have been studied in terms of stability [29, 30] and applied to different areas [31].

The paper is organised as follows. In Section 2, the design of the automatic classifier is explained, where the most significant details are given for both base classifiers during the training and classification phases. The strategy for estimating the best partition is provided based on a validation criterion. The combination of the base classifiers through the FCM is also explained. In Section 3, we give details about the performance of the proposed strategy applied to the classification of textures in natural images. In Section 4, conclusions are presented.

2 Design of the automatic hybrid classifier

As mentioned above, the system works in two phases: training and classification. Fig. 1 displays its architecture. The training patterns are supplied to the FC classifier, which automatically establishes a partition assuming a number of clusters c, starting from $c = 1$ and until validation of the partition. This makes the process automatic [1]. For each cluster, FC computes the cluster centres and BP estimates a probability density function with the mean and covariance matrix as parameters. The centres, means and covariance matrices are stored in the Knowledge Base (KB) and then recovered during the classification phase. Given a pattern $\boldsymbol{x}_s$, it is classified as belonging to a cluster w_j by combining, through the FCM approach, its membership degrees (provided by FC) and probabilities (supplied by BP).

2.1 Training phase

This phase consists of the following steps:

1. We start with the observation of a set X of n training samples, that is, $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\} \in \Re^d$, where d is the data dimensionality. Each sample is to be assigned to a given cluster w_j, where the number of possible clusters is c, that is, $j = 1, 2, \ldots, c$. Each training sample vector $\boldsymbol{x}_i$ represents an image pixel, where its components are the three RGB values of that pixel at the image location (x,y). This means that in our experiments the data dimensionality is $d = 3$. The RGB is the colour space used, as mentioned in Section 1.1.

2. We start by assuming that initially the number of clusters is $c = 1$; under this assumption the training samples are supplied to the FC clustering approach, which determines the unique cluster centre $\boldsymbol{v}_1 \in \Re^d$ according to the training procedure described below (Section 2.1.1).

3. The partition established in Step 2 is submitted to a validation process according to the criterion described in Section 2.1.2. If it is not validated, we set $c = c + 1$ and go to Step 2 where c new centres, $\boldsymbol{v}_j \in \Re^d$, are computed. Steps 2 and 3 are repeated until we achieve a valid partition.

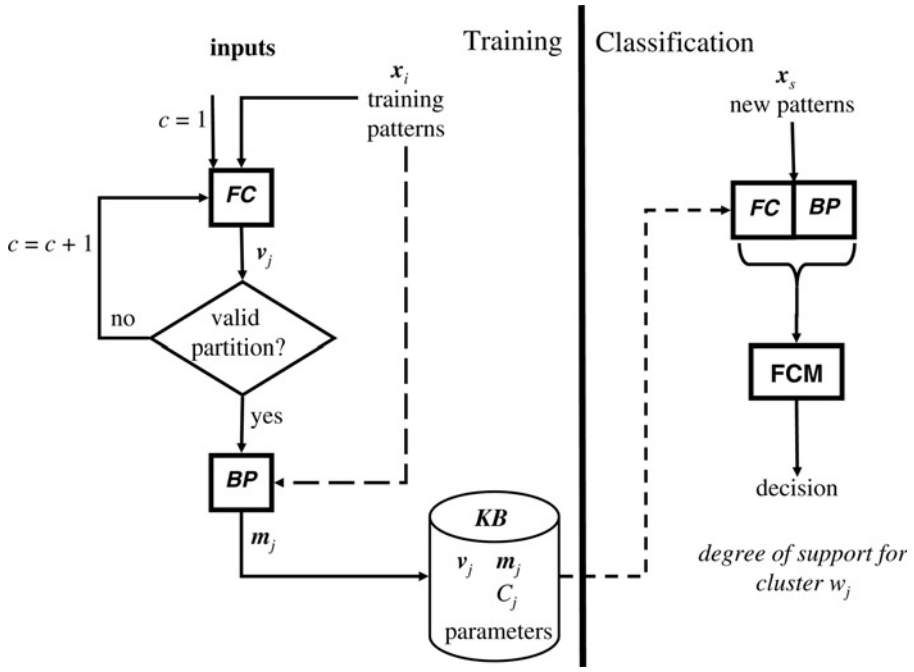

Figure 1 *Architecture of the hybrid classifier: training and classification phases*

4. Once the partition is validated, it is transferred to the BP to perform its training process, where a probability density function is estimated for each cluster w_j as described in Section 2.1.3. Each function involves the mean $\boldsymbol{m}_j \in \Re^d$ and the covariance matrix $\boldsymbol{C}_j$ of the corresponding cluster.

5. All parameters, $\boldsymbol{v}_j$, $\boldsymbol{m}_j$ and C_j, are stored in a Knowledge Base (KB), to be recovered later during the classification phase.

2.1.1 Training through FC: This process receives the input training patterns $\boldsymbol{x}_i$ and establishes a partition, assuming the number of clusters c [2, 27, 32] is known. FC computes for each $\boldsymbol{x}_i$ at iteration t, its degree of membership in cluster $w_j(\mu_i^j)$ and updates the cluster centres $\boldsymbol{v}_j$ as follows

$$\mu_i^j(t+1) = \frac{1}{\sum_{r=1}^{c} (d_{ij}(t)/d_{ir}(t))^{2/(b-1)}},$$
$$\boldsymbol{v}_j(t+1) = \frac{\sum_{i=1}^{n} [\mu_i^j(t)]^b \boldsymbol{x}_i}{\sum_{i=1}^{n} [\mu_i^j(t)]^b} \quad (1)$$

$d_{ij}^2 \equiv d^2(\boldsymbol{x}_i, \boldsymbol{v}_j)$ is the squared Euclidean distance. The number b is called the exponential weight [1, 33], $b > 1$. The stopping criterion of the iteration process is achieved when $\|\mu_i^j(t+1) - \mu_i^j(t)\| < \varepsilon \; \forall ij$ or a number $t_{\max}$ of iterations is reached.

The method requires the initialisation of the cluster centres, so that (1) can be applied at the iteration $t = 1$. For this purpose, we apply the pseudorandom procedure described in [34]:

1. Perform a linear transform $Y = f(X)$ of the training sample values so that they range in the interval [0, 1].

2. Initialise $\boldsymbol{v} = 2D\bar{\boldsymbol{M}} \circ \boldsymbol{R} + D\bar{\boldsymbol{m}}$, where $\bar{\boldsymbol{m}}$ is the mean vector for the transformed training samples values in Y and $\bar{\boldsymbol{M}} = \max(\text{abs}(Y - \bar{\boldsymbol{m}}))$, both of size $1 \times d$; $D = [1 \ldots 1]^{\text{T}}$ with size $c \times 1$; $\boldsymbol{R}$ is a $c \times d$ matrix of random numbers in [0, 1]; the operation $\circ$ denotes the element by element multiplication.

Once the FC process is carried out, a partition of the input training samples is obtained, where each cluster w_j has associated its centre $\boldsymbol{v}_j$.

As described later in Section 3.1, 12 images were available, which were used for setting different parameters following the same procedure. From these images we selected randomly three images; two were used for training and one for validation. We used the percentage of error during the classification process as the validation criterion. The errors were computed taking into account the incorrect classifications as compared to the reference ground truth image (see Section 3.2.1). We varied ε from 0.01 to 0.1 and computed the percentage of error for each value. The maximum percentage of error was obtained with $\varepsilon = 0.1$ with ten iterations and the minimum with $\varepsilon = 0.01$ with 47 iterations. The difference between both percentages was 0.2%. Because of this small difference we consider that $\varepsilon = 0.1$ suffices, as the convergence is faster than with $\varepsilon = 0.01$. With $t_{\max}$ set to 50, we assume a wide margin for the convergence. The above parameters were obtained with the exponential weight $b = 2$. With $\varepsilon = 0.1$ and $t_{\max} = 50$, we varied b from 1.1 to 4.0 in steps of 0.1 for

doi: 10.1049/iet-cvi.2008.0023

the same data sets, verifying that the best performance in terms of percentage of error was obtained for $b = 2.0$.

2.1.2 Validation of the partition: This topic has been widely studied (see [35] and related references). We found acceptable performance was achieved through the sum-of-squared error criterion (SE) [1], which uses the cluster centres $\boldsymbol{v}_j$ obtained by FC as follows

$$\mathrm{SE}(\boldsymbol{v}_j, \boldsymbol{x}, c) = \sum_{j=1}^{c} \sum_{\boldsymbol{x} \in w_j} \left\| \boldsymbol{x} - \boldsymbol{v}_j \right\|^2 \tag{2}$$

SE decreases rapidly until reaching the best partition (equivalent to the best number of clusters $c = \hat{c}$), decreasing much more slowly thereafter until it reaches zero when the number of clusters is equal to the number of samples, that is, $c = n$. Given a number of clusters c, we compute the delta function $\Delta^{\mathrm{SE}}(c) = \mathrm{SE}(c+1) - \mathrm{SE}(c)$, $c = 1, \ldots, G-1$ which represents the difference in value for two consecutive numbers of clusters, where G is the last number of clusters evaluated. We then normalise these differences to range between [0, 1], as follows,

$$\hat{\Delta}^{\mathrm{SE}}(c) = \frac{\Delta^{\mathrm{SE}}(c)}{\sum_{i=1}^{G-1} \Delta^{\mathrm{SE}}(c)} \tag{3}$$

Starting from $c = 1$, a partition is validated for a given number of clusters $\hat{c}$ when we find $\hat{\Delta}^{\mathrm{SE}}(\hat{c}) < T$; T has been obtained after experimentation with the following categories of data: (a) nine data sets from the Machine Learning Repository [32] (bupa, cloud, glass, imageSegm, iris, magi4, thyroid, pimaIndians and wine); (b) three synthetic data sets manually generated with different number of clusters and (c) four data sets coming from outdoor natural images, also with different number of clusters. Because we know the true number of clusters for each category, we compute the coefficient according to (c) observing that for the true number of clusters and for each category its value is less than 0.1, hence T is set to this value.

2.1.3 Training through the parametric Bayesian classifier: Following Duda *et al.* [1], the Bayesian's classifier makes the decision about any sample $\boldsymbol{x}$, which is to be classified, according to the following rule,

$$\boldsymbol{x} \in w_j \text{ if } p(\boldsymbol{x}|w_j)P(w_j) > p(\boldsymbol{x}|w_h)P(w_h), \quad \forall h \neq j \tag{4}$$

$P(w_j)$ and $P(w_h)$ are the prior probabilities that $\boldsymbol{x} \in w_j$ and $\boldsymbol{x} \in w_h$, respectively; $p(\boldsymbol{x}|w_j)$ and $p(\boldsymbol{x}|w_h)$ are the likelihoods that $\boldsymbol{x} \in w_j$ and $\boldsymbol{x} \in w_h$, respectively. Because we do not have prior information about samples $\boldsymbol{x}$, which are to be classified, we assume that $P(w_j) = P(w_h)$. This means that the decision is made based only on the likelihoods, generally modelled as normal (Gaussian) probability density functions, where a function is obtained for each cluster w_j as follows,

$$p(\boldsymbol{x}|w_j) = \frac{1}{(2\pi)^{d/2}\left|C_j\right|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{m}_j)^{\mathrm{T}} C_j^{-1} (\boldsymbol{x} - \boldsymbol{m}_j)\right] \tag{5}$$

d is the data dimensionality, that is, $d = 3$ in this work.

The goal of the training process for this classifier is to estimate both parameters: the mean $\boldsymbol{m}_j$ and the covariance C_j, both for each cluster w_j with n_j samples. This estimation is carried out through maximum likelihood estimation from the validated partition supplied by FC, Fig. 1, as follows

$$\boldsymbol{m}_j = \frac{1}{n_j} \sum_{k=1}^{n_j} \boldsymbol{x}_k, \; C_j = \frac{1}{n_j - 1} \sum_{k=1}^{n_j} \left(\boldsymbol{x}_k - \boldsymbol{m}_j\right)\left(\boldsymbol{x}_k - \boldsymbol{m}_j\right)^{\mathrm{T}} \tag{6}$$

where T denotes transpose.

2.2 Classification phase

After the training phase and according to the scheme on the right part of Fig. 1, a new sample $\boldsymbol{x}_s \in \Re^d$ must be classified as belonging to a cluster w_j. This sample, like each training sample, represents a pixel at the image location (x, y) with the R, G, B components. BP computes the probabilities through (5); these probabilities can be used for classifying $\boldsymbol{x}_s$ according to the rule in (4). This allows comparing the performance of BP with FCM. FC computes the membership degrees for $\boldsymbol{x}_s$ to each cluster according to (1) and classifies the pixel according to the following rule: $\boldsymbol{x}_s \in w_j$ if $\mu_i^j > \mu_i^h$ for all $h \neq j$. So, we can also compare the performance of FC with FCM (see Table 2). Given an image to be classified, we compute these probabilities and membership degrees for each pixel in the image before the FCM is started. The membership degrees are used for initialising the networks as described in the next section.

2.2.1 Network topology of the FCMs: Now, we will describe the FCM approach which combines the information provided by the base classifiers, Fig. 1. As before, given the new sample $\boldsymbol{x}_s$, the problem is to classify it as belonging to a cluster w_j. The rule for making this decision is given.

FCMs are networks used to create models as collections of concepts and the various causal relations that exist between these concepts [3–7]. The concepts are represented by nodes and the causal relationships by directed arcs between the nodes. Each arc is accompanied by a causal weight that defines the type of causal relation between the two nodes.

For each cluster w_j, we build a network of nodes, net_j, where the topology of this network is established by the spatial distribution of the pixels in the image to be classified with size $M \times N$. Each node i in net_j is associated to the pixel location (x, y) in the image, that is

$i \equiv (x, y)$. Hence, the number of nodes in net$_j$ is $q = M \times N$. For simplicity, instead of using sample $\boldsymbol{x}_s$ with its R, G, B components at (x, y), we say that node i is to be classified. Node i in net$_j$ is initialised with the membership degree μ_i^j provided by FC according to (1), but mapped linearly for ranging in $[-1, +1]$ instead of $[0, +1]$. The network states (activation levels) are the membership degrees associated to the nodes. Through the FCM these network states are reinforced or punished iteratively based on the influences exerted by their neighbours. The goal is to make the best decision based on more stable state values. The causal weights s_{ik}^j are the values of the causal relations between nodes i and k in the network net$_j$, taking values in the fuzzy causal interval $[-1, +1]$; they indicate the influence exerted by node k over i, increasing or decreasing the network state in i, depending on positive or negative causality values, respectively. $s_{ik}^j = 0$ means no causality. In FCMs no feedback from a node to itself is allowed [5, 6], so $s_{ii}^j = 0$. Given an image of size $M \times N$, the goal is to classify the pixel (node) i located at (x, y) as belonging to a cluster w_j. The initial states at $t = 0$, $\mu_i^j(0) \equiv \mu_i^j$, are updated based on the causal influences at each iteration t. Every node is positively or negatively activated to a certain degree. At the end of the iterative process, the decision about the cluster to which it belongs is made based on maximum state values considering all j networks.

2.2.2 Iterative updating process: According to [3] and [4], the activation level at the iteration $t+1$ is computed as

$$\mu_i^j(t+1) = f(\mu_i^j(t), \mathcal{A}_i^j) = f\left(\mu_i^j(t), \sum_{k=1}^{q} s_{ik}^j(t)\mu_k^j(t)\right) - d_i^j \mu_i^j(t) \tag{7}$$

where $\mu_i^j(t)$ is the activation of certainty neuron i at iteration t in network net$_j$. Each causal weight s_{ik}^j is defined as a combination of two coefficients representing the mutual influence exerted by k neighbours over i: (a) a regularisation coefficient which computes the consistency between the states of the nodes and the supports provided by BP in a given neighbourhood for each net$_j$; (b) a contextual coefficient which computes the consistency between the clustering labels. $\mathcal{A}_i^j$ is the sum of the weighted influence that certainty neuron i receives at the iteration t from all other neurons. The term $d_i^j \in [0, 1]$ is the decay factor of certainty neuron i. This factor determines the fraction of the current activation level that will be subtracted from the new activation level as a result of the neuron's natural intention to get closer to activation level zero. The bigger the decay factor, the stronger the decay mechanism. Following [3] and [4] function f is that used in the MYCIN expert system for the aggregation of the certainty factors [36, 37], defined as follows,

$$f(x, y) = \begin{cases} x + y(1 - x) & \text{if } x, y \geq 0, \\ x + y(1 + x) & \text{if } x, y < 0, \\ (x + y)/(1 - \min(|x|, |y|)) & \text{else} \end{cases} \tag{8}$$

where $|x|, |y| \leq 1$; $\mu_i^j(t)$ is always in that interval. However, this does not apply for $\mathcal{A}_i^j$, as a concept can be influenced by many concepts and perhaps the sum $\sum_{k=1}^{q} s_{ik}^j \mu_k^j(t)$ can take a value outside the interval $[-1, +1]$. In order to keep $\mathcal{A}_i^j$ within this interval it is passed through the sigmoid function, that is, $\mathcal{A}_i^j = \tanh(\mathcal{A}_i^j)$, as suggested in [4]. So, under the FCM framework the influences are mapped in $\mathcal{A}_i^j$ as consistencies, assuming that they could be non-symmetric; the self-influence is embedded in the $\mu_i^j(t)$ memory term. Taking into account (7), the goal is to compute: (a) the causal weights $s_{ik}^j(t)$ and (b) the decay factor. A detailed analysis of the characteristics of the function in (8) can be found in [38].

2.2.3 Computation of the causal weights: As mentioned before, the causal weight s_{ik}^j is the combination of the regularisation and contextual coefficients. We define both coefficients and then we combine them for obtaining the causal weights. This is described below. Taking into account the mapping between a pixel location (x, y) and node i at each net$_j$, the neighbourhood $\boldsymbol{N}_i^m$ contains m nodes surrounding i, mapped from the image and representing the m-connected spatial region around the pixel (x, y). A typical value of m used in the literature is 8, which defines a 3×3 region; 8 is the value chosen in this paper. We tested other values greater than 8, verifying an over-influence of the neighbourhood in (7). As can be observed, index i varies from 1 to q, that is, this is the number of neighbours explored at each net$_j$. For borders nodes in the image, the neighbourhood only includes the pixels belonging to the image, that is, $m = 3$ in the four corners and $m = 5$ in the remainder borders.

We define the regularisation coefficient at the iteration t as follows

$$r_{ik}^j(t) = \begin{cases} 1 - \left|\mu_i^j(t) - p_k^j\right| & k \in \boldsymbol{N}_i^m, \quad i \neq k \\ 0 & k \notin \boldsymbol{N}_i^m, \quad i = k \end{cases} \tag{9}$$

where p_k^j is supplied by BP, the probability is that node (pixel) k with attributes $\boldsymbol{x}_k$ belongs to cluster w_j, computed through (5), that is $p_k^j \equiv p(\boldsymbol{x}_k | w_j)$. These values are mapped linearly to range between $[-1, +1]$ instead of $[0, +1]$. From (9) we can see that $r_{ik}^j(t)$ ranges between $[-1, +1]$ where the lower/higher limit means minimum/maximum influence, respectively. The contextual coefficient for node i for the cluster j at the iteration t is defined

doi: 10.1049/iet-cvi.2008.0023

taking into account the clustering labels l_i and l_k as follows,

$$c_{ik}(t) = \begin{cases} +1 & l_i = l_k \quad k \in \boldsymbol{N}_i^m,\ i \neq k \\ -1 & l_i \neq l_k \quad k \in \boldsymbol{N}_i^m,\ i \neq k \\ 0 & \quad\ \ k \notin \boldsymbol{N}_i^m,\ i = k \end{cases} \tag{10}$$

where values of -1 and $+1$ mean negative and positive influences, respectively. Labels l_i and l_k are obtained as follows: Given node i, at each iteration t, we know its state at each net$_j$ as given by (7); we determine that node i belongs to cluster w_j if $\mu_i^j(t) > \mu_i^h(t) \forall j \neq h$, so we set l_i to the j value which identifies the cluster, $j = 1, \ldots, c$. Label l_k is set similarly. Thus, this coefficient is independent of the net$_j$, because it is the same for all networks.

Both coefficients are combined as the following averaged sum, taking into account the signs,

$$\begin{aligned} z_{ik}^j(t) &= \gamma r_{ik}^j(t) + (1-\gamma) c_{ik}(t),\ s_{ik}^j \\ &= \left[\operatorname{sgn}\left(z_{ik}^j\right)\right]^v z_{ik}^j,\ \operatorname{sgn}\left(z_{ik}^j\right) = \begin{cases} -1 & z_{ik}^j \leq 0 \\ +1 & z_{ik}^j > 0 \end{cases} \end{aligned} \tag{11}$$

$\gamma \in [0, 1]$ represents the trade-off between both coefficients; *sgn* is the *signum function* and v is the number of negative values in set $C \equiv \left\{W_{ik}^j(t), r_{ik}^j(t), c_{ik}(t)\right\}$, that is, given $S \equiv \{q \in C / q < 0\} \subseteq C$, $v = \operatorname{card}(S)$.

2.2.4 Computation the decay factor: We define the decay factor based on the assumption that high stability in the network states implies that the activation level for node i in network net$_j$ would be to lose some of its activation. We build an accumulator of cells of size $q = M \times N$, where each cell i is associated to the node of identical name. Each cell i contains the number of times, h_i^j, that node i has changed significantly its activation level in the net$_j$. Initially, all h_i^j values are set to zero and then $h_i^j = h_i^j + 1$ if $\left|\mu_i^j(t+1) - \mu_i^j(t)\right| > \varepsilon$. The stability of node i is measured as the fraction of changes accumulated by cell i compared with the changes in its neighbourhood $k \in \boldsymbol{N}_i^m$ and the number of iterations t. The decay factor is computed as follows

$$d_i^j = \begin{cases} 0 & h_i^j = 0 \text{ and } h_k^j = 0 \\ \dfrac{h_i^j}{\left(\bar{h}_k^j + h_i^j\right) t} & \text{otherwise} \end{cases} \tag{12}$$

where h_i^j is defined above and $\bar{h}_k^j$ is the average value accumulated by nodes $k \in \boldsymbol{N}_i^m$. As one can see, from (12), if $h_i^j = 0$ and $h_k^j = 0$, the decay factor takes the null value, this means that no changes occur in the network states, that is, high stability is achieved; if the fraction of changes is small, the stability of node i is also high and the decay term tends towards zero. Even if the fraction is constant the decay term tends to zero as t increases, this means that perhaps initially some changes can occur and then no more changes are detected, this is another sign of stability. The decay factor subtracts from the new activation level a fraction; this implies that the activation level could take values lesser/greater than -1 or $+1$. In these cases, the activation level is set to -1 or $+1$, respectively.

2.2.5 Summary of the full FCM process: adjusting the parameters: The iterative process ends if all nodes in the network fulfil the convergence criterion $\left|\mu_i^j(t-1) - \mu_i^j(t)\right| > \varepsilon$ or a number of iterations, $t_{\max}$, are reached.

Next, we give details about the setting of parameter γ in (11), also ε and $t_{\max}$ for the convergence. We use the images described in Section 3.1 and the same procedure as that used for adjusting the parameters in Section 2.1.1. The experiments here are carried out by testing different combinations of γ (ranging from 0.1 to 0.9 in steps of 0.1) and ε (from 0.01 to 0.1 in steps of 0.01). With $\gamma = 0.8$ and $\varepsilon = 0.05$ we achieve a percentage of error for the classification of 12.2% with 15 iterations. With other combinations and these parameters ranging from $0.75 < \gamma < 0.85$ to $0.01 < \varepsilon < 0.05$ the error is reduced by about 0.1% but at the expense of the number of iterations, which is considerably increased with values close to 50. Based on these experiments, in the end we set γ to 0.80, ε to 0.05 and $t_{\max}$ to 50 (considering a wide margin for the number of iterations).

The FCM process is synthesised as follows:

1. *Initialise:* load each node with $\mu_i^j(t=0)$ as given by (1); set $\varepsilon = 0.05$ and $t_{\max} = 50$. Define nc as the number of nodes that change their state values at each iteration.

2. *FCM process:*

```
t = 0
whereas t < t_max or nc ≠ 0
  t = t + 1; nc = 0;
  for each node i
    update μ_i^j(t) according to (7)
    if |μ_i^j(t) − μ_i^j(t − 1)| > ε then
      nc = nc + 1
    end if;
  end for;
end while
```

3. *Outputs:* the states $\mu_i^j(t)$ for all nodes updated.

2.2.6 Summary of the full process: The method proposed, involving both the training and decision phases, can be summarised in the following steps. In the training phase, a number of clusters c are determined. Based on the distribution of the samples in the c clusters, the cluster centres $\boldsymbol{v}_j$ are computed through FC and the means $\boldsymbol{m}_j$ and covariance matrices $\boldsymbol{C}_j$ through BP, $j = 1, \ldots, c$. All are stored in *KB*. During the classification phase, given an image to be classified, we compute for each pixel i its

membership degree of belonging to each cluster w_j, through (1), by recovering $\boldsymbol{v}_j$ from *KB*. For the same pixel, we compute its probability of belonging to cluster w_j through (5), recovering $\boldsymbol{m}_j$ and C_j from *KB*. The membership degrees are used to initialise the networks. Then the FCM process for adjusting the node values starts. Once this process finishes, a decision is made for each node about which cluster it belongs to. The decision about the classification of node i with attributes $\boldsymbol{x}_i$ as belonging to the cluster w_j is made as: $i \in w_j$ if $\mu_i^j(t) > \mu_i^h(t)$, $\forall j \neq h$ where j and h are identifiers of the clusters and t represents the last iteration. Hence, the decision is made based on the maximum state values considering all j networks.

3 Comparative analysis and performance evaluation

To assess the validity and performance of the proposed approach, we describe the tests carried out according to both processes: training and classification.

3.1 Training: estimating the best partition

We used a set of 36 digital aerial images acquired during May, 2006, from the Abadia region of Lugo (Spain). They are multi-spectral images, 512 × 512 pixels in size. The images were taken during different days from an area with several natural textures. We selected randomly 12 images from the set of 36 available. This set, hereinafter identified as ST, was used for training and the remainder set of images for testing. With this number of images for training and testing, we carried out four experiments interchanging images between both sets, with similar results in the four cases. Each image in the training set is down-sampled by two, obtaining the training samples used in this initial training phase, that is, the number of training samples was $n_0 = 12 \times 256 \times 256 = 786\,432$.

Before the initial training phase started, the free parameters described in Section 2.1.1 had to be adjusted. To do this, we randomly selected three images from ST, where the pixels excluded by the down-sampling in the three images were used for setting these parameters. We knew the distribution of these samples in clusters, because the ground truth was known. We followed the process described in this section, varying the parameters and training with the samples extracted from two of the three images, that is, with 2 × 256 × 256 samples; the samples from the third image were used for computing the classification error. We selected five subsets of three images from ST without affecting the setting of the free parameters. Once the initial process was finished with the above n samples, we adjusted the parameters described in Section 2.2.5 before the FCM process. The samples used for this task were exactly those used for the above-mentioned adjustments of the three images.

Table 1 shows the behaviour of $\hat{\Delta}^{SE}(c)$ against the number of clusters c ranging from $c = 1$ to $G-1$, where G is set to 8 in our experiments. This is because we have not found images with more than 8 clusters. We can see that with $T = 0.1$ (see Section 2.1.2) the condition $\hat{\Delta}^{\mathrm{SE}}(\hat{c}) < T$ is fulfilled for $\hat{c} = 4$, which in the end was selected as the number of clusters estimated.

Once the number of clusters was established, the cluster centres obtained by FC ($\boldsymbol{v}_j$) and BP ($\boldsymbol{m}_j$) and the covariance matrices $\boldsymbol{C}_j$ for BP were stored in *KB*. At this stage, because the number of clusters was already estimated, all the pixels should be classified in one of the four clusters. This implies that the set of images in the initial training phase must be as representative as possible of all images available. To solve this, several sets of training images must be selected. This justifies the experiments carried out with four different sets of 12 images, as described above.

3.2 Classification: comparative analysis

As mentioned previously, the remaining 24 images from the set of 36 were used as images for testing. Four sets, $S0$, $S1$, $S2$ and $S3$ of six images each, were processed during the test according to the strategy described below. The images assigned to each set were randomly selected from the 24 images available.

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

Our combined FCM (FM) method was compared with the base classifiers used for the combination. *FM* was also compared with the following classical hybrid strategies [22]: Mean (*ME*), Max (*MA*) and Min (*MI*). Given node i with supports μ_i^j and p_i^j provided by FC and BP, respectively, *ME* computes the mean value, that is, $(\mu_i^j + p_i^j)/2$; *MA* the maximum value, that is, $\max\{\mu_i^j, p_i^j\}$ and *MI* the minimum value, that is, $\min\{\mu_i^j, p_i^j\}$. They were studied in

Table 1 Behaviour of the $\hat{\Delta}^{\mathrm{SE}}(c)$

C	1	2	3	4	5	6	7
Δ	c1−c2	c2−c3	c3−c4	c4−c5	c5−c6	c6−c7	c7−c8
$\hat{\Delta}^{\mathrm{SE}}(c)$	0.4951	0.2433	0.1221	**0.0728**	0.0611	0.0381	0.0301

IET Comput. Vis., 2009, Vol. 3, Iss. 3, pp. 112–123
doi: 10.1049/iet-cvi.2008.0023

terms of reliability [39]. The final decision was made based on the maximum fused value for all clusters. Yager [40] proposed a multicriteria decision-making approach based on fuzzy sets aggregation. So, FM was also compared against the fuzzy aggregation (FA) defined by (13)

$$\gamma_i^j = 1 - \min\left\{1, \left(\left(1-\mu_i^j\right)^a + \left(1-p_i^j\right)^a\right)^{1/a}\right\}, \quad a \geq 1 \tag{13}$$

This represents the joint support provided by FC and BP; the decision is made based on the maximum support for all clusters. The parameter a is adjusted from the set of images described in Section 3.1 and following the same procedure as that used for setting other parameters (Sections 2.1.1 and 2.2.5). Two images are used for training both classifiers FC and BP and the third for validation. The experiments were carried out by varying the parameter a from 1 to 8. The minimum error rate was achieved with $a = 4$, which in the end was the value used for this parameter. The decision made according to node i with the supports provided by (13) was based on the maximum value, in keeping with the following rule: $i \in w_j$ if $\gamma_i^j > \gamma_i^h, \quad \forall h \neq j$.

In order to verify the behaviour of each method as the learning degree increases, we carried out the experiments according to the following three steps, described below:

Step 1: we classify the images in sets $S0$ and $S1$, where each pixel is assigned to one of the four clusters estimated during the initial training phase with set ST. We compute the percentage of error for both sets according to the procedure described in Section 3.2.2. The classified samples from $S1$ are combined with the initial training samples in ST and the resulting set is used for the training of the system. Hence, the number of training samples is $n_1 = n_0 + 6 \times 512 \times 512$, where n_0 is the number of initial training samples coming from ST. This training process is carried out assuming that the number of clusters is known (i.e. $\hat{c} = 4$). Therefore it is a partial process of the defined in Fig. 1 because the number of clusters does not need to be estimated. This implies that only the parameters estimated by each classifier are updated and stored in KB. Set $S0$ is used as a pattern set in order to verify the performance of the training process as the learning increases. Note that it is not considered for training.

Step 2: we classify the images in sets $S0$ and $S2$ according to the four clusters (based on the previous training with sets ST and $S1$) and compute the percentage of error for both sets as before. Now, sets ST, $S1$ and $S2$ are combined and the resulting set is used for the training of the system without estimating the number of clusters, as before. Now, the number of samples is $n_2 = n_1 + 6 \times 512 \times 512$. The new parameters are stored also in KB.

Step 3: we classify the images into sets $S0$ and $S3$ also according to the four clusters (based on the previous training with sets ST, $S1$ and $S2$) and compute the percentage of error for both sets.

To verify the performance for each method we have built a ground truth for each image under the supervision of the expert human criterion. Based on the assumption that the automatic training process determines four classes, we classify each image pixel with the simple classifiers, obtaining a labelled image with four expected clusters. For each class we build a binary image, which is manually touched up until a satisfactory classification is obtained under human supervision.

Fig. 2(*a*) displays an original image belonging to set $S0$; Fig. 2(*b*) displays the correspondence between the classes and the label assigned to the corresponding cluster centre according to a labelling map previously defined; Fig. 2(*c*) labelled image for the four clusters obtained by our proposed FM approach. The labels are artificial grey intensities identifying each cluster. The correspondence between labels and the different textures is: 1 – with forest vegetation; 2 – with bare soil; 3 – with agricultural crop vegetation; 4 – with buildings and man made structures.

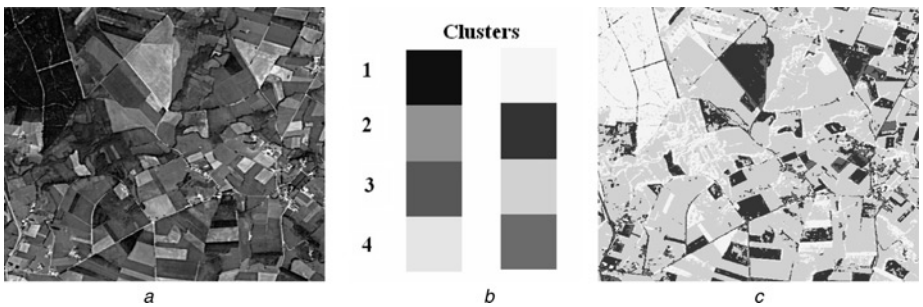

Figure 2 *Mapping between original textures and labels*

a Original image belonging to set $S0$
b Correspondence between labels and clusters
c Labelled image with the four clusters according to (*b*)

Fig. 3 displays the binary ground truth images for each cluster, where the white pixels represent the corresponding cluster; Figs. 3(*a*)–(*d*) represent the clusters numbers 1–4, respectively, identified in Fig. 2.

3.2.2 Results: Table 2 shows the percentage of error during the classification for the different classifiers. For each step, from 1 to 3, we show the results obtained for both sets of tested images *S*0 and either *S*1 or *S*2 or *S*3. These percentages are computed as follows. Let I_N^r be an image $r (r = 1, \ldots, 6)$ belonging to set *SN* $(N = 0, 1, 2, 3)$; i is the node at location (x, y) in I_N^r. An error counter E_N^r is initially set to zero for each image r in set *SN* at each step and for each classifier. Based on the corresponding decision process, each classifier determines the cluster to which node i belongs, $i \in w_j$. If the same pixel location on the corresponding ground truth image is black, then the pixel is incorrectly classified and $E_N^r = E_N^r + 1$. The error rate of the image I_N^r is: $e_N^r = E_N^r/Z$, where Z is the image size, that is, 512×512. The average error rate for set *SN* at each step is: $\bar{e}_N = 1/6 \sum_{r=1}^{6} e_N^r$ and the standard deviation $\bar{\sigma}_N = \sqrt{1/5 \sum_{r=1}^{6} (e_N^r - \bar{e}_N)^2}$. In Table 2, they are displayed as percentages, that is, $\tilde{e}_N = 100\bar{e}_N$ and $\tilde{\sigma}_N = 100\bar{\sigma}_N$. The numbers in square brackets in the row *FM* indicate the rounded and averaged number of iterations required by each set at each step.

3.2.3 Discussion: From the results in Table 2, it is seen that the best performance is achieved by the proposed *FM* strategy. The best performance for the classical hybrid methods is achieved by *ME* and for the simple methods by BP. The best performances are established in terms of the least average percentage of error and the least standard deviation values. For clarity, in Fig. 4 the performance of the proposed *FM* approach for set *SP*0 is displayed against FA and the best methods at each group, that is, *ME* for the classical hybrid approaches and BP for simple methods.

The results show that the hybrid approaches perform favourably for the data sets used. The *MA* and *ME* fusion methods provide better results than the individual ones. This means that fusion strategies are suitable for classification tasks. This agrees with the conclusion reported in [27] about the choice of combiners.

Moreover, as the learning increases through steps 1–3, the performance improves and the number of iterations for *S*0 decreases. This means that the learning phase is important and that the number of samples affects the performance.

The main drawback of the FCM approach is its execution time, which is greater than for the remaining methods and directly proportional to the number of iterations. All tests have been implemented in MATLAB and executed on a Pentium M, 1.86 GHz with 1 GB RAM. On average, the execution per iteration is 17.2 s.

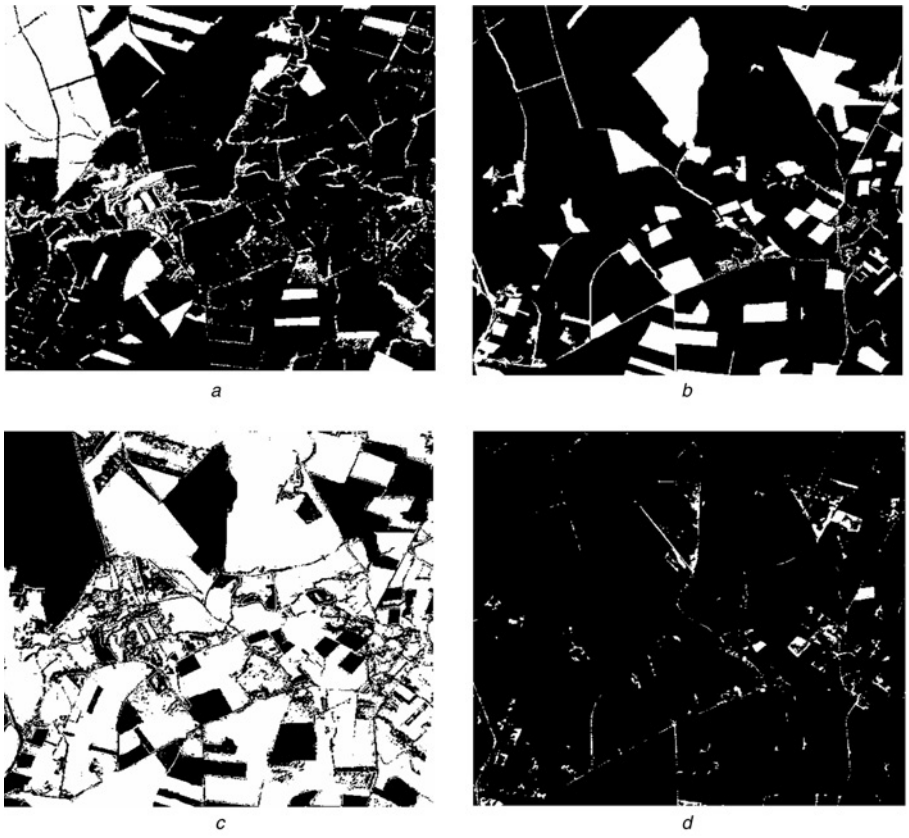

Figure 3 *Ground truth images (a)–(d) for the four clusters 1–4 displayed in Fig. 2*

doi: 10.1049/iet-cvi.2008.0023

Table 2 Average percentages of error and standard deviations at each STEP for the four sets of testing images S0, S1, S2 and S3

$\tilde{e}_N$: average percentage of error $\tilde{\sigma}_N$: standard deviation of error (%)		STEP 1				STEP 2				STEP 3			
		S0		S1		S0		S2		S0		S3	
		$\tilde{e}_0$	$\tilde{\sigma}_0$	$\tilde{e}_1$	$\tilde{\sigma}_1$	$\tilde{e}_0$	$\tilde{\sigma}_0$	$\tilde{e}_2$	$\tilde{\sigma}_2$	$\tilde{e}_0$	$\tilde{\sigma}_0$	$\tilde{e}_3$	$\tilde{\sigma}_3$
iterative and fuzzy hybrid methods	**[iterations] FM** (fuzzy cognitive maps)	[16] 21.7	1.7	[18] 21.6	1.6	[14] 19.2	1.2	[15] 19.9	1.1	[11] 16.1	0.9	[12] 18.5	0.8
	FA (fuzzy aggregation)	25.1	2.2	26.2	2.1	23.8	1.9	24.1	1.8	20.9	1.6	20.1	1.5
classical hybrid methods	**MA** (maximum)	30.4	2.9	29.6	2.7	27.8	2.8	26.9	2.6	26.6	2.1	26.0	1.9
	MI (minimum)	36.4	3.1	36.1	2.9	31.4	3.3	34.3	2.8	29.9	2.4	28.3	2.3
	ME (mean)	28.6	2.6	28.3	2.2	24.8	2.3	26.1	2.2	24.3	1.9	24.2	1.7
simple methods	**FC** (fuzzy clustering)	32.1	2.8	30.2	2.6	27.1	2.3	27.4	2.3	26.0	2.1	25.9	2.0
	BP (Bayesian parametric)	30.2	2.7	29.1	2.5	26.1	2.2	26.4	2.2	25.2	2.0	24.7	1.8

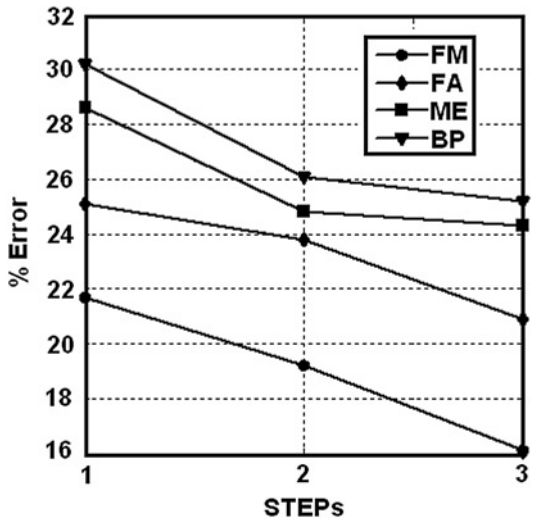

Figure 4 *Percentage of error for FM, FA, ME and BP against the three steps*

4 Conclusions

We have designed an automatic hybrid classifier based on the FCM framework that performs favourably as compared with other existing strategies. This approach makes decisions during the classification by combining the supports provided by two classifiers (FC and BP) under the iterative FCM process. The initial supports obtained by the FC are modified by the influence exerted through the supports provided by BP and also by its own influence. This modification was carried out by considering contextual consistencies. In the future, this approach can be extended in two ways: (a) by introducing mutual influences between both classifiers and updating the supports of both classifiers; (b) by considering more classifiers so that several influences are exerted on each classifier. Additionally, the approach proposed is easily applicable for any kind of classification involving contextual information.

5 Acknowledgments

The authors thank SITGA (Servicio Territorial de Galicia) in collaboration with at the Dimap company (http://www.dimap.es/) for the original aerial images supplied and used in this paper. The authors are also grateful to the referees for their constructive criticism and suggestions.

6 References

[1] DUDA R.O., HART P.E., STORK D.S.: 'Pattern classification' (Wiley, 2000)

[2] ZIMMERMANN H.J.: 'Fuzzy set theory and its applications' (Kluwer Academic Publishers, Norwell, 1991)

[3] TSARDIAS A.K., MARGARITIS K.G.: 'Cognitive mapping and certainty neuron fuzzy cognitive maps', *Inf. Sci.*, 1997, **101**, pp. 109–130

[4] TSARDIAS A.K., MARGARITIS K.G.: 'An experimental study of the dynamics of the certainty neuron fuzzy cognitive maps', *Neurocomputing*, 1999, **24**, pp. 95–116

[5] KOSKO B.: 'Fuzzy cognitive maps', *Int. J. Man Mach. Stud.*, 1986, **24**, pp. 65–75

[6] KOSKO B.: 'Neural networks and fuzzy systems: a dynamical systems approach to machine intelligence' (Prentice-Hall, NJ, 1992)

[7] MIAO Y., LIU Z.Q.: 'On causal inference in fuzzy cognitive maps', *IEEE Trans. Fuzzy Syst.*, 2000, **8**, (1), pp. 107–119

[8] VALDOVINOS R.M., SÁNCHEZ J.S., BARANDELA R.: 'Dynamic and static weighting in classifier fusion' in MARQUES J.S., PÉREZ DE LA BLANCA N., PINA P. (EDS.): 'Pattern recognition and image analysis' (*Lecture notes in computer science* Springer-Verlag, Berlin, 2005), pp. 59–66

[9] PUIG D., GARCÍA M.A.: 'Automatic texture feature selection for image pixel classification', *Pattern Recognit.*, 2006, **39**, (11), pp. 1996–2009

[10] HANMANDLU M., MADASU V.K., VASIKARLA S.: 'A fuzzy approach to texture segmentation'. Proc. IEEE Int. Conf. Information Technology: Coding and Computing (ITCC'04), The Orleans, Las Vegas, Nevada, USA, 2004, pp. 636–642

[11] RUD R., SHOSHANY M., ALCHANATIS V., COHEN Y.: 'Application of spectral features' ratios for improving classification in partially calibrated hyperspectral imagery: a case study of separating Mediterranean vegetation species', *J. Real-Time Image Process.*, 2006, **1**, pp. 143–152

[12] KUMAR S., GHOSH J., CRAWFORD M.M.: 'Best-bases feature extraction for pairwise classification of hyperspectral data', *IEEE Trans. Geosci. Remote Sens.*, 2001, **39**, (7), pp. 1368–1379

[13] YU H., LI M., ZHANG H.J., FENG J.: 'Color texture moments for content-based image retrieval'. Proc. Int. Conf. Image Processing, 2002, vol. 3, pp. 24–28

[14] MAILLARD P.: 'Comparing texture analysis methods through classification', *Photogram. Eng. Remote Sens.*, 2003, **69**, (4), pp. 357–367

[15] RANDEN T., HUSØY J.H.: 'Filtering for texture classification: a comparative study', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1999, **21**, (4), pp. 291–310

[16] WAGNER T.: 'Texture analysis' in JÄHNE B., HAUβECKER H., GEIβLER P. (EDS.): 'Handbook of computer vision and applications' (Academic Press, 1999, vol. 2), (Signal Processing and Pattern Recognition)

[17] SMITH G., BURNS I.: 'Measuring texture classification algorithms', *Pattern Recognit. Lett.*, 1997, **18**, pp. 1495–1501

[18] DR.IMBAREAN A., WHELAN P.F.: 'Experiments in colour texture analysis', *Pattern Recognit. Lett.*, 2003, **22**, (4), pp. 1161–1167

[19] KONG Z., CAI Z.: 'Advances of research in fuzzy integral for classifier's fusion'. Proc. Eighth ACIS Int. Conf. Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2007, vol. 2, pp. 809–814

[20] KUNCHEVA L.I.: '"Fuzzy" vs "non-fuzzy" in combining classifiers designed by boosting', *IEEE Trans. Fuzzy Syst.*, 2003, **11**, (6), pp. 729–741

[21] KUMAR S., GHOSH J., CRAWFORD M.M.: 'Hierarchical fusion of multiple classifiers for hyperspectral data analysis', *Pattern Anal. Appl.*, 2002, **5**, pp. 210–220

[22] KITTLER J., HATEF M., DUIN R.P.W., MATAS J.: 'On combining classifiers', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1998, **20**, (3), pp. 226–239

[23] CAO J., SHRIDHAR M., AHMADI M.: 'Fusion of classifiers with fuzzy integrals'. Proc. Third Int. Conf. Document Analysis and Recognition (ICDAR'95), 1995, vol. 1, pp. 108–111

[24] PARTRIDGE D., GRIFFITH N.: 'Multiple classifier systems: software engineered, automatically modular leading to a taxonomic overview', *Pattern Anal. Appl.*, 2002, **5**, pp. 180–188

[25] DENG D., ZHANG J.: 'Combining multiple precision-boosted classifiers for indoor-outdoor scene classification', *Inf. Technol. Appl.*, 2005, **1**, (4–7), pp. 720–725

[26] ALEXANDRE L.A., CAMPILHO A.C., KAMEL M.: 'On combining classifiers using sum and product rules', *Pattern Recognit. Lett.*, 2001, **22**, pp. 1283–1289

[27] KUNCHEVA L.I.: 'Combining pattern classifiers: methods and algorithms' (Wiley, 2004)

[28] STACH W., KURGAN L., PEDRYCZ W., REFORMAT M.: 'Evolutionary development of fuzzy cognitive maps'. Proc. IEEE Conf. Fuzzy Syst., 2005, pp. 619–624

[29] CHENG Q., FANG Z.T.: 'The stability problem for fuzzy bidirectional associative memories', *Fuzzy Sets Syst.*, 2002, **132**, pp. 83–90

[30] MARTCHENKO A.S., ERMOLOV I.L., GROUMPOS P.P., PODURAEV J.V., STYLIOS C.D.: 'Investigating stability analysis issues for fuzzy cognitive maps'. 11th Mediterranean Conf. Control and Automation, 2003, pp. 619–624

[31] AGUILAR J.: 'A survey about fuzzy cognitive maps papers', *Int. J. Comput. Cogn.*, 2005, **3**, (2), pp. 27–33

[32] ASUNCION A., NEWMAN D.J.: 'UCI machine learning repository' (University of California, School of Information and Computer Science, Irvine, CA, 2008), available on-line http://www.ics.uci.edu/~mlearn/MLRepository.html

[33] BEZDEK J.C.: 'Pattern recognition with fuzzy objective function algorithms' (Kluwer, Plenum Press, New York, 1981)

[34] BALASKO B., ABONYI J., FEIL B.: 'Fuzzy clústering and data analysis toolbox for use with Matlab' (Veszprem University, Hungary, 2008), (available from

doi: 10.1049/iet-cvi.2008.0023

URL: http://www.fmt.vein.hu/softcomp/fclusttoolbox/FuzzyClústeringToolbox.pdf)

[35] VOLKOVICH Z., BARZILY Z., MOROZENSKY L.: 'A statistical model of cluster stability', *Pattern Recognit.*, 2008, **41**, (7), pp. 2174–2188

[36] BUCHANAN B.G., SHORLIFFE E.H. (EDS.): 'Rule-based expert systems. The MYCIN experiments of the Stanford Heuristic Programming Project' (Addison-Wesley, Reading, MA, 1984)

[37] SHORLIFFE E.H.: 'Computer-based medical consultations: MYCIN' (Elsevier, New York, NY, 1976)

[38] TSARDIAS A.K., MARGARITIS K.G.: 'The MYCIN certainty factor handling as uniform operator and its use as threshold function in artificial neurons', *Fuzzy Sets Syst.*, 1998, **93**, pp. 263–274

[39] CABRERA J.B.D.: 'On the impact of fusion strategies on classification errors for large ensambles of classifiers', *Pattern Recognit.*, 2006, **39**, pp. 1963–1978

[40] YAGER R.R.: 'On ordered weighted averaging aggregation operators in multicriteria decision making', *IEEE Trans. Syst. Man Cybern.*, 1988, **18**, (1), pp. 183–190