A MULTIPLE CASCADE-CLASSIFIER SYSTEM FOR A ROBUST AND PARTIALLY UNSUPERVISED UPDATING OF LAND-COVER MAPS

Lorenzo Bruzzone and Roberto Cossu

2002

Technical Report # DIT-02-0026

A Multiple Cascade-Classifier System for a Robust and Partially Unsupervised Updating of Land-Cover Maps

Lorenzo Bruzzone, *Member, IEEE,* and Roberto Cossu, *Student Member, IEEE*

Department of Information and Communication Technologies - University of Trento
Via Sommarive, 14, 38050, Povo, Trento, Italy
Telephone: +39-0461-882056; Fax: +39-0461-882672; e-mail: lorenzo.bruzzone@ing.unitn.it

Abstract - **A system for a regular updating of land-cover maps is proposed that is based on the use of multitemporal remote-sensing images. Such a system is able to face the updating problem under the realistic but critical constraint that, for the image to be classified (i.e., the most recent of the considered multitemporal data set), no ground truth information is available. The system is composed of an ensemble of partially unsupervised classifiers integrated in a multiple classifier architecture. Each classifier of the ensemble exhibits the following novel peculiarities: i) it is developed in the framework of the cascade-classification approach to exploit the temporal correlation existing between images acquired at different times in the considered area; ii) it is based on a partially unsupervised methodology capable to accomplish the classification process under the aforementioned critical constraint. Both a parametric maximum-likelihood classification approach and a non-parametric radial basis function (RBF) neural-network classification approach are used as basic methods for the development of partially unsupervised cascade classifiers. In addition, in order to generate an effective ensemble of classification algorithms, hybrid maximum-likelihood and RBF neural network cascade classifiers are defined by exploiting the peculiarities of the cascade-classification methodology. The results yielded by the different classifiers are combined by using standard unsupervised combination strategies. This allows the definition of a robust and accurate partially unsupervised classification system capable of analyzing a wide typology of remote-sensing data (e.g., images acquired by passive sensors, SAR images, multisensor and multisource data). Experimental results obtained on a real multitemporal and multisource data set confirm the effectiveness of the proposed system.**

I. Introduction

One of the major problems in geographical information systems (GISs) consists in defining strategies and procedures for a regular updating of land-cover maps stored in the system databases. This crucial task can be carried out by using remote-sensing images regularly acquired by space-born sensors in the specific investigated areas. Such images can be analyzed with automatic classification techniques in order to derive updated land-cover maps. The classification process can be performed by considering either the information contained in a single image [1] or the information contained in a multitemporal series of images of the same area [2] (i.e., by exploiting the temporal correlation between images acquired at different times). The latter approach is called "cascade classification" and allows one to increase the categorization accuracy. However, at the operating level, both aforementioned approaches are usually based on supervised classification algorithms. Consequently, they require the availability of ground truth information for the training of the classifiers. Unfortunately, in many real cases, it is not possible to rely on training data for all the images necessary to ensure an updating of land-cover maps that is as frequent as required by applications. This prevents all the remotely sensed images acquired in the investigated area from being used to update land-cover maps. For these reasons, the process of temporal updating of land-cover maps results in a complex and challenging problem.

In previous works [3], [4], the authors have already addressed the aforementioned problem. In particular, partially unsupervised classification approaches have been defined and developed. (The term "partially unsupervised" is used here to point out that, on the one hand, no ground truth information is assumed to be available for the specific image to be classified, but, on the other hand, a training set exists related to an image of the same geographical area acquired before the one to be classified). In [3], a partially unsupervised classification methodology is proposed that is able to update the parameters of an already trained parametric maximum-likelihood classifier on the basis of the distribution of a new image for which training data are not available. In [4], in order to take into account the temporal correlation between series of remote-sensing images, the partially unsupervised maximum-likelihood classification approach is reformulated in the framework of the Bayesian rule for cascade classification. This allows an increase in the robustness of the unsupervised retraining process.

Although the aforementioned approaches have proved effective on several data sets, they exhibit some limitations. Firstly, given the intrinsic complexity of the problem addressed, these approaches result in classifiers that are less reliable and less accurate than the corresponding supervised classifiers. Secondly, the parametric nature of the proposed classifiers prevents the approaches from being used for the analysis of multisensor and multisource remote-sensing images. This can be critical in complex classification problems, in which multisource and/or multisensor information may play a fundamental role.

In this paper, a novel classification system aimed at obtaining an accurate and robust partially unsupervised updating of land-cover maps is proposed. Such a system extends the approaches proposed in [3] and [4], defining an effective classification framework based on a multiple cascade-classifier system (MCCS), which is able to overcome the main limitations of the aforementioned methods. The ensemble of classifiers used in the MCCS architecture is derived from maximum-likelihood (ML) and radial basis function (RBF) neural-network classification approaches. Three important methodological novelties are associated with the presented system: i) all the partially unsupervised classifiers of the ensemble are defined in the framework of cascade classification; ii) a new non-parametric partially unsupervised cascade classifier based on RBF neural networks is proposed; iii) hybrid maximum-likelihood and RBF neural classifiers are defined by exploiting the peculiarities of the cascade-classification approach in order to generate an effective ensemble of classifiers. It is worth noting that, thanks to the non-parametric nature of the RBF neural-network cascade classifiers, the proposed system is able to analyze multisensor and multisource data.

Experimental results obtained on a multitemporal and multisource data set related to the Island of Sardinia, Italy, confirm the effectiveness of the proposed system.

The paper is organized into seven sections. Section II reports the formulation of the problem and describes the general architecture of the proposed system. Section III presents the partially unsupervised classification problem in the framework of the cascade-classification approach for both the ML and RBF neural-network classification techniques. Section IV addresses the problem of defining suitable ensembles of cascade classifiers, and describes the proposed hybrid ML and RBF classifiers. Section V deals with the unsupervised strategies used for the combination of the results yielded by the cascade classifiers included in the considered ensemble. Experimental results are reported in Section VI. Finally, in Section VII, discussion is provided and conclusions are drawn.

II. Problem Formulation and Description of the System Architecture

A. Problem Formulation and Simplifying Assumptions

Let $\mathbf{X}_1 = \{x_1^1, x_2^1, .., x_B^1\}$ and $\mathbf{X}_2 = \{x_1^2, x_2^2, .., x_B^2\}$ denote two multispectral images composed of B pixels and acquired in the area under analysis at the times t_1 and t_2, respectively. Let x_j^1 and x_j^2 be the $1 \times d$ feature vectors associated with the j-th pixels of the images (where d is the dimensionality of the input space), and $W = \{w_1, w_2, ..., w_C\}$ be the set of C land-cover classes that characterize the geographical area considered at both t_1 and t_2. Let l_j^2 be the classification label of the j-th pixel at the time t_2. Finally, let X_1 and X_2 be two multivariate random variables representing the pixel values (i.e., the feature vector values) in $\mathbf{X}_1$ and $\mathbf{X}_2$, respectively.

In the formulation of the proposed approach, we make the following assumptions:

1) the same set Ω of C land-cover classes characterize the area considered over time (only the spatial and spectral distributions of such classes are supposed to vary);

2) a reliable training set $\mathbf{Y}_1$ for the image $\mathbf{X}_1$ acquired at t_1 is available;

3) a training set $\mathbf{Y}_2$ for the image $\mathbf{X}_2$ acquired at t_2 is not available.

It is worth noting that assumption 1), even if not verified in all possible applications, is reasonable in a wide range of real problems.

In the aforementioned assumptions, the proposed system aims at performing a robust and accurate classification of $\mathbf{X}_2$ by exploiting the image $\mathbf{X}_1$, the training set $\mathbf{Y}_1$, and the image $\mathbf{X}_2$, as well as the temporal correlation between the classes at t_1 and t_2.

B. System Architecture

The proposed system is based on a multiple classifier architecture composed of N different classification algorithms (see Fig.1). The choice of this kind of architecture is due to the complexity of the problem addressed. In particular, the intrinsic difficulty of the partially unsupervised classification problem results in classifiers that are less reliable and less accurate than the corresponding supervised ones, especially for complex data sets. Therefore, by taking into account that, in general, ensembles of classifiers are more accurate and more robust than the individual classifiers that make them up [5], we expect that a multiple-classifier approach may increase the reliability and the accuracy of the global classification system. A further step aimed at improving the performance of the system consists in implementing each partially unsupervised classification algorithm of the ensemble in the framework of a cascade-classifier approach, thus exploiting also the temporal correlation between the multitemporal images in the updating process.

The following sections address the individual components of the presented system. In particular, the proposed partially unsupervised cascade classifiers, the strategy adopted to define the ensemble of cascade classifiers, and the combination methods will be described in detail.

III. Partially Unsupervised Classification Techniques: a Cascade-Classifier Approach

Let us focus our attention on the choice of each partially unsupervised classifier to be included in the multiple-classifier architecture. In order to obtain robust and accurate classifiers, we propose to consider classification strategies defined in the context of the cascade-classifier approach [2], [6]. The standard supervised cascade-classifier approach (proposed by Swain [2]) exploits the correlation between multitemporal images in order to increase the classification accuracy in the cases in which training data are available for all the images considered. In our method, we extend the application of the standard supervised cascade-classifier approach to partially unsupervised classification problems. In particular, we exploit the temporal dependence between land-cover classes to increase the reliability and the accuracy of the unsupervised estimation of the parameters related to the image **X**2.

The cascade-classifier decision strategy associates a generic pixel x_j^2 of the image **X**2 with a land-cover class according to the following decision rule [2]:

$$l_j^2 = \boldsymbol{w}_m \ \hat{I}\ W \quad \textit{if and only if} \quad P\left(\boldsymbol{w}_m / x_j^1, x_j^2\right) = \max_{w_h \in W} \left\{ P\left(\boldsymbol{w}_h / x_j^1, x_j^2\right) \right\} \qquad (1)$$

where $P\left(\boldsymbol{w}_h / x_j^1, x_j^2\right)$ is the value of the probability that the j-th pixel of the image belongs to the class $\boldsymbol{w}_h$ at t_2, given the observations x_j^1 and x_j^2. Under the conventional assumption of class-conditional independence [2], [6], the decision rule (1) can be rewritten as [4]:

$$l_j^2 = \boldsymbol{w}_m \ \hat{I}\ W \qquad \textit{if and only if}$$

$$\sum_{n=1}^{C} p\left(x_j^1 / w_n\right) p\left(x_j^2 / w_m\right) P\left(w_n, w_m\right) = \max_{w_h \in W} \left\{ \sum_{n=1}^{C} p\left(x_j^1 / w_n\right) p\left(x_j^2 / w_h\right) P\left(w_n, w_h\right) \right\} \tag{2}$$

where $p\left(x_j^i / w_r\right)$ is the value of the conditional density function for the pixel x_j^i, given the class $w_r \in W$, and $P\left(w_n, w_h\right)$ is the prior joint probability of the pair of classes (w_n, w_h). The latter term takes into account the temporal correlation between the two images.

We propose to integrate the partially unsupervised classification of the image $\mathbf{X}_2$ in the context of the above-described classification rule. As the training set $\mathbf{Y}_2$ is not available, the density functions of the classes at the time t_1 (i.e., $p\left(X_1 / w_n\right)$, $w_n \in W$) are the only statistical terms of (2) that we can estimate in a completely supervised way. This means that, in order to accomplish the classification task, we should estimate both the density functions of the classes at t_2 ($p\left(X_2 / w_h\right)$, $w_h \in W$) and the prior joint probabilities of the classes ($P\left(w_n, w_h\right)$, $w_n \in W, w_h \in W$) in an unsupervised way. It is worth noting that usually the estimation of $p\left(X_i / w_r\right)$ ($w_r \in W$, $i=1,2$) involves the computation of a parameter vector. The number and nature of the vector components depend on the specific classifier used. Consequently, the procedure to be adopted to accomplish the unsupervised estimation process depends on the technique used to carry out the cascade classification, in particular, on the vector of parameters required by the classifier.

The possibility of establishing a relationship between the classifier parameters and the statistical terms involved in (2) is a basic constraint that each classification technique should satisfy in order to permit the use of the cascade-classification decision rule. To meet this requirement, we propose to use two suitable classification methods. The first is a parametric approach based on the maximum-likelihood (ML) classifier [3]; the second consists of a non-parametric technique based on radial basis function (RBF) neural networks [7], [8]. The specific architectures of the ML and RBF cascade classifiers and the procedures for the partially unsupervised estimation of the related parameters are described in the following two subsections.

A. Maximum-Likelihood Cascade Classifier

The formulation of the partially unsupervised classification problem in the framework of the ML cascade approach has already been addressed in [4]. Therefore, here we briefly recall the basic issues described in that paper.

For simplicity, let us assume that the probability density function of the generic class w_r at the time t_i (i.e., $p(X_i / w_r), w_r \in W$, i=1,2) can be described by a Gaussian distribution (i.e., by a mean vector m_r^i and a covariance matrix S_r^i). Accordingly, hyper-quadrics decision surfaces can be modeled. Under this common assumption (widely adopted for multispectral image classification problems), the mean vectors and the covariance matrices that characterize the conditional density functions of the classes at t_1 can be easily computed by a standard procedure using the training set $\mathbf{Y}_1$. Concerning the parameter vector J of the classifier to be estimated in a partially unsupervised way, it consists of the following components:

$$J = \left[m_1^2, S_1^2, P(w_1, w_1), \ldots, m_C^2, S_C^2, P(w_C, w_C)\right] \tag{3}$$

where the superscript “2” denotes the parameters of the conditional density functions of the classes at the time t_2. To carry out the partially unsupervised estimation process, we propose to adopt a procedure based on the observation that, under the assumption of class-conditional independence over time, the joint density function of the images $\mathbf{X}_1$ and $\mathbf{X}_2$ (i.e., $p(X_1, X_2)$) can be described as a mixture density with $C \times C$ components (i.e., as many components as possible pairs of classes):

$$p(X_1, X_2) \cong \sum_{n=1}^{C} \sum_{h=1}^{C} p(X_1 / w_n) p(X_2 / w_h) P(w_n, w_h). \tag{4}$$

In this context, the estimation of the above terms becomes a mixture-density estimation problem, which can be solved via the EM algorithm [9]-[12]. By applying such an algorithm, we can derive the following iterative equations to estimate the components of the vector J necessary to accomplish the cascade-classification process [4]:

$$\left[m_h^2\right]^{t+1} = \frac{\sum_{j=1}^{B}\left\{\sum_{n=1}^{C} P^t\left(w_n, w_h / x_j^1, x_j^2\right)\right\} x_j^2}{\sum_{j=1}^{B}\left\{\sum_{n=1}^{C} P^t\left(w_n, w_h / x_j^1, x_j^2\right)\right\}} \tag{5}$$

$$\left[S_h^2\right]^{t+1} = \frac{\sum_{j=1}^{B}\left\{\sum_{n=1}^{C} P^t\left(w_n, w_h / x_j^1, x_j^2\right)\right\} \left(x_j^2 - \left[m_h^2\right]^{t+1}\right)^T \left(x_j^2 - \left[m_h^2\right]^{t+1}\right)}{\sum_{j=1}^{B}\left\{\sum_{n=1}^{C} P^t\left(w_n, w_h / x_j^1, x_j^2\right)\right\}} \tag{6}$$

$$P^{t+1}\left(w_n, w_h\right) = \frac{1}{B}\sum_{j=1}^{B} P^t\left(w_n, w_h / x_j^1, x_j^2\right) \tag{7}$$

where the superscripts t and $t+1$ refer to the values of the parameters at the current and next iterations, respectively, the superscript T refers to the vector transpose operation, and the joint posterior probabilities of the classes are approximated by:

$$P^{t}\left(w_n, w_h / x_j^1, x_j^2\right) \cong \frac{p\left(x_j^1 / w_n\right) p^t\left(x_j^2 / w_h\right) P^t\left(w_n, w_h\right)}{\sum_{g=1}^{C}\sum_{f=1}^{C} p\left(x_j^1 / w_g\right) p^t\left(x_j^2 / w_f\right) P^t\left(w_g, w_f\right)}. \tag{8}$$

It is worth noting that all the previous equations implicitly depend on J. Concerning the initialization of the components of the vector J, the initial values of the parameters of the density functions of classes at t_2 are obtained by considering the corresponding values estimated at time t_1 by supervised learning, whereas all the prior joint probabilities of classes are assumed to have the same values. It is possible to prove that, at each iteration, the estimated parameters evolve from their initial values to the final ones by maximizing the following log-likelihood function (the convergence to a local maximum can be proven) [9]:

$$L(\mathbf{X}_1,\mathbf{X}_2 / \boldsymbol{J}) = \sum_{j=1}^{B} log\left\{\sum_{n=1}^{C}\sum_{h=1}^{C} p\left(x_j^1 / \boldsymbol{w}_n\right)p\left(x_j^2 / \boldsymbol{w}_h\right)P\left(\boldsymbol{w}_n, \boldsymbol{w}_h\right)\right\}. \qquad (9)$$

The estimates of the parameters obtained at convergence and those achieved by the classical supervised procedure at the time t_1 are then substituted into (2) in order to accomplish the ML cascade-classification process. We refer the reader to [4] for greater details on the ML partially unsupervised cascade classifier and on alternative initialization conditions on the iterative estimation algorithm.

B. RBF Neural Network Cascade Classifier

The problem of partially unsupervised cascade classification by using RBF neural networks is much more complex than the one associated with the ML parametric cascade classifier. The increased complexity mainly depends on the non-parametric nature of RBF neural networks. In our case, we have to resolve two critical issues in order to develop the cascade classifier in the framework of RBF neural networks: i) we should define a specific architecture that is able to implement the cascade-classification decision rule; ii) we should devise a partially unsupervised procedure for the training of the proposed architecture.

First of all, let us briefly recall the standard architecture of an RBF neural classifier to be used for the classification of a generic image $\mathbf{X}_i$ (see Fig. 2). This architecture is made up of three layers: an input layer (composed of as many units as input features), a hidden layer (composed of S neurons) and an output layer (composed of as many units as land-cover classes). The input layer just propagates the input features to the hidden layer. Each unit of the hidden layer applies a simple non-linear transformation to the input data according to a symmetric radial basis function $\boldsymbol{j}_s$ (usually a Gaussian function characterized by a mean value $\boldsymbol{p}_s$ and a width $\boldsymbol{s}_s$). The connections between the hidden and output units are associated with a numerical value called weight (let w_s^r denote the weight that connects the s-th hidden neuron to the r-th output neuron). The output neurons apply a linear transformation to the weighted outputs of the hidden neurons. It can be proven that, if the classifier has been properly trained [13], the outputs of an RBF neural network can be related to the conditional densities of the classes, which are expressed as a mixture of the kernel functions associated with the units of the hidden layer. In addition, the statistical terms computed by the neural classifier can be related to the global density function $p(X_i)$ of the image $\mathbf{X}_i$ as follows:

$$p(X_i) = \sum_{r=1}^{C}\sum_{s=1}^{S} p(X_i/\boldsymbol{j}_s)P(\boldsymbol{j}_s)P(\boldsymbol{w}_r/\boldsymbol{j}_s) \tag{10}$$

where $p(X_i \ / \boldsymbol{j}_s)$ is the conditional density of the variable X_i given the kernel function $\boldsymbol{j}_s$, $P(\boldsymbol{w}_r/\boldsymbol{j}_s)$ is the conditional probability of the class $\boldsymbol{w}_r$, given the kernel $\boldsymbol{j}_s$, $P(\boldsymbol{j}_s)$ is the prior probability of the kernel $\boldsymbol{j}_s$, and S is the number of kernels considered. It is worth noting that the statistical terms in (10) can be associated with the parameters of the RBF neural architecture as follows [13]:

$$\boldsymbol{j}_s(X_i) = p(X_i \ / \boldsymbol{j}_s) \tag{11}$$

$$w_s^r = P(\boldsymbol{j}_s)P(\boldsymbol{w}_r/\boldsymbol{j}_s) \tag{12}$$

We refer the reader to [7], [8] for more details on standard RBF neural classifiers.

In order to define a cascade classifier in the context of the RBF neural-network theory, let us approximate the joint density function $p(X_1, X_2)$ of the two images $\mathbf{X}_1$ and $\mathbf{X}_2$ as a mixture of Gaussian kernel functions. To this end, let us consider K kernel functions $\boldsymbol{j}_k^1$ and Q kernel functions $\boldsymbol{j}_q^2$ associated with the statistics of the images $\mathbf{X}_1$ and $\mathbf{X}_2$, respectively. Accordingly, under the assumption of kernel-conditional independence in the temporal domain, we can write:

$$p(X_1, X_2) \cong \sum_{h=1}^{C}\sum_{n=1}^{C}\sum_{k=1}^{K}\sum_{q=1}^{Q} p(X_1 / \boldsymbol{j}_k^1)p(X_2 / \boldsymbol{j}_q^2)P(\boldsymbol{j}_k^1, \boldsymbol{j}_q^2)P(\boldsymbol{w}_n, \boldsymbol{w}_h / \boldsymbol{j}_k^1, \boldsymbol{j}_q^2) \tag{13}$$

where $p(X_i / \boldsymbol{j}_r^i)$ is the value of the conditional density function of the variable X_i, given the kernel $\boldsymbol{j}_r^i$, $P(\boldsymbol{w}_n, \boldsymbol{w}_h / \boldsymbol{j}_k^1, \boldsymbol{j}_q^2)$ is the joint conditional probability of the pair of classes $(\boldsymbol{w}_n, \boldsymbol{w}_h)$ given the pair of kernels $\left(\boldsymbol{j}_k^1,\boldsymbol{j}_q^2\right)$, and $P\left(\boldsymbol{j}_k^1,\boldsymbol{j}_q^2\right)$ is the joint prior probability of the kernels $\left(\boldsymbol{j}_k^1,\boldsymbol{j}_q^2\right)$. In this context, the cascade classification decision rule can be rewritten as:

$$l_j^2 = w_m \ \hat{I}\ W \qquad if\ and\ only\ if$$

$$\sum_{n=1}^{C}\sum_{k=1}^{K}\sum_{q=1}^{Q} p\left(x_j^1 / \boldsymbol{j}_k^1\right) p\left(x_j^2 / \boldsymbol{j}_q^2\right) P\left(\boldsymbol{j}_k^1,\boldsymbol{j}_q^2\right) P\left(w_n, w_m \big/ \boldsymbol{j}_k^1,\boldsymbol{j}_q^2\right) = \tag{14}$$

$$\max_{w_h \in W}\left\{\sum_{n=1}^{C}\sum_{k=1}^{K}\sum_{q=1}^{Q} p\left(x_j^1 / \boldsymbol{j}_k^1\right) p\left(x_j^2 / \boldsymbol{j}_q^2\right) P\left(\boldsymbol{j}_k^1,\boldsymbol{j}_q^2\right) P\left(w_n, w_h \big/ \boldsymbol{j}_k^1,\boldsymbol{j}_q^2\right)\right\}.$$

It is worth noting that the temporal correlation between the two images is taken into account by the terms $P\left(\boldsymbol{j}_k^1,\boldsymbol{j}_q^2\right)$ and $P\left(w_n, w_h \big/ \boldsymbol{j}_k^1,\boldsymbol{j}_q^2\right)$. By analyzing equation (14), we can observe that $p\left(x_j^1 / \boldsymbol{j}_k^1\right)$ and $p\left(x_j^2 / \boldsymbol{j}_q^2\right)$ can be derived by applying two standard RBF neural-network classifiers to the t_1 and t_2 images, respectively. In particular, we can apply an RBF neural-network classifier with K hidden units to the image $\mathbf{X}_1$ and an RBF neural-network classifier with Q hidden units to the image $\mathbf{X}_2$ (see Fig. 3). If a proper training algorithm is used, the terms $p\left(x_j^1 / \boldsymbol{j}_k^1\right)$ and $p\left(x_j^2 / \boldsymbol{j}_q^2\right)$ are given by the outputs of the hidden neurons of the aforementioned neural classifiers. However, in order to implement the cascade classification decision rule, a non-conventional architecture should be considered, which involves the joint statistical terms $P\left(\boldsymbol{j}_k^1,\boldsymbol{j}_q^2\right)$ and $P\left(w_n, w_h \big/ \boldsymbol{j}_k^1,\boldsymbol{j}_q^2\right)$ in the classification process. To this end, the outputs of the hidden neurons of the t_1 and t_2 networks are given as input to a specific block (let us call it “cascade classification” block) that presents as many outputs as land-cover classes (i.e., C outputs). In particular, the output $u_{h,}$ which is associated with the land-cover class w_h, is given by:

$$u_h\left(x_j^1, x_j^2\right) = \sum_{n=1}^{C}\sum_{k=1}^{K}\sum_{q=1}^{Q} P\left(\boldsymbol{j}_k^1,\boldsymbol{j}_q^2\right) P\left(w_n, w_h \big/ \boldsymbol{j}_k^1,\boldsymbol{j}_q^2\right) \boldsymbol{j}_k^1\left(x_j^1\right) \boldsymbol{j}_q^2\left(x_j^2\right). \tag{15}$$

According to equation (14), each pixel is classified as belonging to the land-cover class associated with the maximum output value.

The main problem that remains to be solved is the estimation of all the parameters considered in the proposed architecture in a partially unsupervised way (i.e., by using only the joint density function $p(X_1, X_2)$ and the training set $\mathbf{Y}_1$). Concerning the parameters of the $p\left(X_1 / \boldsymbol{j}_k^1\right)$ (i.e., the centers $\boldsymbol{p}_k^1$ and the widths $\boldsymbol{s}_k^1$ of the Gaussian kernel functions that process the image $\mathbf{X}_1$), they can be estimated in a supervised way according to the statistical procedure described in [7], [8]. Consequently, the parameter vector $\boldsymbol{J}$ that remains to be estimated in a partially unsupervised way is composed of the following terms:

$$\boldsymbol{J} = \left[\boldsymbol{p}_1^2, \boldsymbol{s}_1^2, \ldots, \boldsymbol{p}_Q^2, \boldsymbol{s}_Q^2, P\left(\boldsymbol{j}_1^1, \boldsymbol{j}_1^2\right), \ldots, P\left(\boldsymbol{j}_K^1, \boldsymbol{j}_Q^2\right), P\left(\boldsymbol{w}_1, \boldsymbol{w}_1 / \boldsymbol{j}_1^1, \boldsymbol{j}_1^2\right), \ldots, P\left(\boldsymbol{w}_C, \boldsymbol{w}_C / \boldsymbol{j}_K^1, \boldsymbol{j}_Q^2\right)\right] \qquad (16)$$

where $\boldsymbol{p}_q^2$ and $\boldsymbol{s}_q^2$ are the centers and the widths characterizing the kernel functions $\boldsymbol{j}_q$ that process the image $\mathbf{X}_2$. In order to estimate the components of the parameter vector, we propose to apply the EM algorithm to (13). Accordingly, it is possible to prove that part of the components of the parameter vector can be estimated by using the following iterative equations:

$$\left[\boldsymbol{p}_q^2\right]^{t+1} = \frac{\sum_{j=1}^{B} \left\{ \sum_{k=1}^{K} P^t\left(\boldsymbol{j}_k^1, \boldsymbol{j}_q^2 / x_j^1, x_j^2\right) \right\} x_j^2}{\sum_{j=1}^{B} \left\{ \sum_{k=1}^{K} P^t\left(\boldsymbol{j}_k^1, \boldsymbol{j}_q^2 / x_j^1, x_j^2\right) \right\}} \qquad (17)$$

$$\left[\boldsymbol{s}_q^2\right]^{t+1} = \frac{\sum_{j=1}^{B} \left\{ \sum_{k=1}^{K} P^t\left(\boldsymbol{j}_k^1, \boldsymbol{j}_q^2 / x_j^1, x_j^2\right) \right\} \left\| x_j^2 - \left[\boldsymbol{p}_q^2\right]^{t+1} \right\|^2}{d \cdot \sum_{j=1}^{B} \left\{ \sum_{k=1}^{K} P^t\left(\boldsymbol{j}_k^1, \boldsymbol{j}_q^2 / x_j^1, x_j^2\right) \right\}} \qquad (18)$$

$$P^{t+1}\left(\boldsymbol{j}_k^1, \boldsymbol{j}_q^2\right) = \frac{1}{B} \sum_{j=1}^{B} P^t\left(\boldsymbol{j}_k^1, \boldsymbol{j}_q^2 / x_j^1, x_j^2\right) \qquad (19)$$

where d is the dimensionality of the input space, the superscripts t and $t+1$ refer to the values of the parameters at the current and next iterations, respectively, and the $P^t\left(\boldsymbol{j}_k^1,\boldsymbol{j}_q^2/x_j^1,x_j^2\right)$ are approximated by:

$$P^t\left(\boldsymbol{j}_k^1,\boldsymbol{j}_q^2/x_j^1,x_j^2\right)\cong\frac{p\left(x_j^1/\boldsymbol{j}_k^1\right)p^t\left(x_j^2/\boldsymbol{j}_q^2\right)P^t\left(\boldsymbol{j}_k^1,\boldsymbol{j}_q^2\right)}{\sum\limits_{z=1}^{K}\sum\limits_{v=1}^{Q}p\left(x_j^1/\boldsymbol{j}_z^1\right)p^t\left(x_j^2/\boldsymbol{j}_v^2\right)P^t\left(\boldsymbol{j}_z^1,\boldsymbol{j}_v^2\right)}. \qquad (20)$$

Concerning the initialization of the aforementioned components of the parameter vector $\boldsymbol{J}$, the initial values of the parameters of the conditional density functions of kernels at t_2 can be obtained by applying a standard unsupervised clustering algorithm to the $\mathbf{X}_2$ image [7], whereas the initial values of prior joint probabilities of the kernels can be easily computed in the assumption of independence between the kernels at two dates (i.e., $P\left(\boldsymbol{j}_k,\boldsymbol{j}_q\right)=P\left(\boldsymbol{j}_k\right)\cdot P\left(\boldsymbol{j}_q\right)$).

As we have already pointed out, the estimation of RBF cascade neural-network classifier parameters is significantly more complex than the estimation of ML cascade-classifier parameters. Despite the parameters $\boldsymbol{p}_q^2$, $\boldsymbol{s}_q^2$ and $P\left(\boldsymbol{j}_k^1,\boldsymbol{j}_q^2\right)$ of the vector $\boldsymbol{J}$ can be estimated in a fully unsupervised way, the estimation of the joint conditional probabilities $P\left(\boldsymbol{w}_n,\boldsymbol{w}_h\big/\boldsymbol{j}_k^1,\boldsymbol{j}_q^2\right)$ requires other information in addition to the one contained in the training set $\mathbf{Y}_1$ (it is worth noting that the terms $P\left(\boldsymbol{w}_n,\boldsymbol{w}_h\big/\boldsymbol{j}_k^1,\boldsymbol{j}_q^2\right)$ express the relationship between kernel functions and land-cover classes). To solve this problem, we propose to exploit some of the information obtained (at convergence) by the ML cascade classifier described in the previous subsection. In particular, a set $\hat{\mathbf{Y}}_2$ of pixels, which is composed of the patterns that are most likely correctly categorized by the ML cascade classifier, is used for the initialization of the $P\left(\boldsymbol{w}_n,\boldsymbol{w}_h\big/\boldsymbol{j}_k^1,\boldsymbol{j}_q^2\right)$ conditional probabilities. These patterns are selected on the basis of the values of the posterior probabilities provided by the ML classifier. In greater detail, pixels associated with values of the posterior probabilities above a predefined threshold ε are chosen. Let $\mathbf{Y}_{n,m}$ be the set of pairs of pixels $\left(x_j^1, x_j^2\right)$ such that $x_j^1 \in \mathbf{Y}_1$ belongs to the land-cover class $\boldsymbol{w}_n$ and $x_j^2 \in \hat{\mathbf{Y}}_2$ is categorized by the ML cascade-classifier as belonging to the class $\boldsymbol{w}_m$. Let $\mathbf{Y}_{n,0}$ be the set of pairs of pixels $\left(x_j^1, x_j^2\right)$ such that $x_j^1 \in \mathbf{Y}_1$ belongs to the land-cover class $\boldsymbol{w}_n$ and $x_j^2 \notin \hat{\mathbf{Y}}_2$. Analogously, let $\mathbf{Y}_{0,m}$ be the set of pairs of pixels $\left(x_j^1, x_j^2\right)$ such that $x_j^1 \notin \mathbf{Y}_1$ and $x_j^2 \in \hat{\mathbf{Y}}_2$ is categorized by the ML cascade-classifier as belonging to the class $\boldsymbol{w}_m$. The iterative equations to be used to estimate the joint conditional probabilities $P\left(\boldsymbol{w}_n, \boldsymbol{w}_h / \boldsymbol{j}_k^1, \boldsymbol{j}_q^2\right)$ are the following:

$$P^{t+1}\left(\boldsymbol{w}_n, \boldsymbol{w}_h / \boldsymbol{j}_k^1, \boldsymbol{j}_q^2\right) = \frac{1}{A}\Bigg\{ \sum_{\left(x_j^1, x_j^2\right) \in \mathbf{Y}_{n,h}} P^t\left(\boldsymbol{j}_k^1, \boldsymbol{j}_q^2 / x_j^1, x_j^2\right) + \sum_{\left(x_j^1, x_j^2\right) \in \mathbf{Y}_{n,0}} \left[\sum_{g=1}^{C} P^t\left(\boldsymbol{w}_g, \boldsymbol{w}_h / \boldsymbol{j}_k^1, \boldsymbol{j}_q^2\right) P^t\left(\boldsymbol{j}_k^1, \boldsymbol{j}_q^2 / x_j^1, x_j^2\right) \right] + + \sum_{\left(x_j^1, x_j^2\right) \in \mathbf{Y}_{0,h}} \left[\sum_{f=1}^{C} P^t\left(\boldsymbol{w}_n, \boldsymbol{w}_f / \boldsymbol{j}_k^1, \boldsymbol{j}_q^2\right) P^t\left(\boldsymbol{j}_k^1, \boldsymbol{j}_q^2 / x_j^1, x_j^2\right) \right] \Bigg\} \tag{21}$$

where the normalizing factor A is equal to:

$$A = \sum_{h=1}^{C} \left[\sum_{\left(x_j^1, x_j^2\right) \in \mathbf{Y}_{0,h}} P^t\left(\boldsymbol{j}_k^1, \boldsymbol{j}_q^2 / x_j^1, x_j^2\right) \right] + \sum_{n=1}^{C} \left[\sum_{\left(x_j^1, x_j^2\right) \in \mathbf{Y}_{n,0}} P^t\left(\boldsymbol{j}_k^1, \boldsymbol{j}_q^2 / x_j^1, x_j^2\right) \right] + + \sum_{n=1}^{C} \sum_{h=1}^{C} \left[\sum_{\left(x_j^1, x_j^2\right) \in \mathbf{Y}_{n,h}} P^t\left(\boldsymbol{j}_k^1, \boldsymbol{j}_q^2 / x_j^1, x_j^2\right) \right]. \tag{22}$$

It is worth noting that this iterative procedure significantly improves the initial estimates biased by the patterns included in $\hat{\mathbf{Y}}_2$.

Analogously to the ML cascade classifier, also in this case the estimated parameters evolve from their initial values to the final ones by maximizing the following log-likelihood function (the convergence to a local maximum can be proven):

$$\begin{aligned} L(\mathbf{X}_1,\mathbf{X}_2/\boldsymbol{J}) = & \sum_{(x_j^1,x_j^2)\in\mathbf{Y}_{0,0}} log\left\{\sum_{k=1}^{K}\sum_{q=1}^{Q} p(x_j^1/\boldsymbol{j}_k^1)p(x_j^2/\boldsymbol{j}_q^2)P(\boldsymbol{j}_k^1,\boldsymbol{j}_q^2)\right\}+ \\ & +\sum_{n=1}^{C}\left\{\sum_{(x_j^1,x_j^2)\in\mathbf{Y}_{n,0}} log\left[\sum_{h=1}^{C}\sum_{k=1}^{K}\sum_{q=1}^{Q} p(x_j^1/\boldsymbol{j}_k^1)p(x_j^2/\boldsymbol{j}_q^2)P(\boldsymbol{j}_k^1,\boldsymbol{j}_q^2)P(\boldsymbol{w}_n,\boldsymbol{w}_h/\boldsymbol{j}_k^1,\boldsymbol{j}_q^2)\right]\right\}+ \\ & +\sum_{h=1}^{C}\left\{\sum_{(x_j^1,x_j^2)\in\mathbf{Y}_{0,h}} log\left[\sum_{n=1}^{C}\sum_{k=1}^{K}\sum_{q=1}^{Q} p(x_j^1/\boldsymbol{j}_k^1)p(x_j^2/\boldsymbol{j}_q^2)P(\boldsymbol{j}_k^1,\boldsymbol{j}_q^2)P(\boldsymbol{w}_n,\boldsymbol{w}_h/\boldsymbol{j}_k^1,\boldsymbol{j}_q^2)\right]\right\}+ \\ & +\sum_{n=1}^{C}\sum_{h=1}^{C}\left\{\sum_{(x_j^1,x_j^2)\in\mathbf{Y}_{n,h}} log\left[\sum_{k=1}^{K}\sum_{q=1}^{Q} p(x_j^1/\boldsymbol{j}_k^1)p(x_j^2/\boldsymbol{j}_q^2)P(\boldsymbol{j}_k^1,\boldsymbol{j}_q^2)P(\boldsymbol{w}_n,\boldsymbol{w}_h/\boldsymbol{j}_k^1,\boldsymbol{j}_q^2)\right]\right\} \end{aligned} \tag{23}$$

where $\mathbf{Y}_{0,0}$ is the set of pairs of pixels (x_j^1, x_j^2) such that $x_j^1 \notin \mathbf{Y}_1$ and $x_j^2 \notin \hat{\mathbf{Y}}_2$.

The estimates of the parameters obtained at convergence and the ones achieved by the classical supervised procedure are used to accomplish the RBF cascade-classification process.

IV. A Strategy for Generating Ensembles of Partially Unsupervised Cascade Classifiers: Hybrid ML and RBF Neural-network Classifiers

The selection of the pool of classifiers to be integrated into the multiple cascade-classifier architecture is an important and critical task. In the literature, several different strategies for defining a classifier ensemble have been proposed [5], [14]-[17]. From a theoretical viewpoint, necessary and sufficient conditions for an ensemble of classifiers to be more accurate than any of its individual members are that the classifiers should be accurate and different [18]. In our case, we can control only the second condition, since no training set is available to verify the first one.

The main issue to be resolved for the definition of the ensemble concerns the capability of different classifiers to incur uncorrelated errors. In practice, several strategies have been proposed to make up pools of classifiers that incur uncorrelated errors. These strategies involve the selection of different classification algorithms, the choice of different initial training conditions for a given classification algorithm, the use of different architectures for the same kind of classifier (e.g., neural networks), the manipulation of the training examples, the manipulation of the input features, the manipulation of the output targets, the injection of randomness, etc. [18]. In our system, the choice of both a parametric (ML) and a non-parametric (RBF) classifier guarantees the use of two classification algorithms based on significantly different principles. For this reason, we expect these classifiers to incur sufficiently uncorrelated errors. However, two classification algorithms are not enough to define an effective multiple classifier architecture. To increase the reliability of the system, we need to generate a pool of N classifiers (N>2). According to the literature, we could define different RBF neural-network architectures in order to derive different classification algorithms for the ensemble [19]. However, as we are dealing with cascade-classifier techniques, we propose to adopt an alternative, deterministic, and simple strategy for making up the ensemble. This strategy is based on the peculiarities of the cascade-classification approach, in which a set of key parameters, estimated by the partially unsupervised process, is composed of the prior joint probabilities of classes $P(w_n, w_h)$ (they are associated with the temporal correlation between classes). The different cascade classifiers (i.e., ML and RBF neural networks) perform different estimations of the aforementioned probabilities, on the basis of the different classification and estimation principles. According to this observation, we propose to introduce in the ensemble hybrid classifiers obtained by exchanging the estimates of the prior joint probabilities of classes performed by different algorithms. In our case, given an ML cascade classifier and an RBF neural-network cascade classifier, this strategy results in an ensemble composed of the two “original” classifiers and of two hybrid ML and RBF algorithms obtained by exchanging the prior joint probabilities estimated in a partially unsupervised way by the original classifiers. These hybrid classifiers are described in the following.

Let $P^{ML}(\boldsymbol{w}_n,\boldsymbol{w}_h)$, $p^{ML}(X_1/\boldsymbol{w}_n)$ and $p^{ML}(X_2/\boldsymbol{w}_h)$ denote the joint probabilities and the conditional densities of classes estimated by the ML cascade classifier, respectively. Analogously, let $P^{RBF}(\boldsymbol{w}_n,\boldsymbol{w}_h/\boldsymbol{j}_k^1,\boldsymbol{j}_q^2)$, $P^{RBF}(\boldsymbol{j}_k^1,\boldsymbol{j}_q^2)$, $p^{RBF}(X_1/\boldsymbol{w}_n)$ and $p^{RBF}(X_2/\boldsymbol{w}_h)$ denote the joint probabilities of the classes conditioned to the kernels, the joint probabilities of the kernels, and the conditional densities of the classes at the times t_1 and t_2 estimated by the RBF cascade classifier, respectively.

The first hybrid classifier (let us call it ML-hybrid cascade classifier) is obtained by merging the joint probabilities estimated by the RBF cascade classifier with the conditional densities estimated by the ML cascade classifier. Hence, the corresponding classification rule is the following:

$$l_j^2 = w_m \ \hat{\boldsymbol{I}}\ \boldsymbol{W} \ \textit{if and only if}$$

$$\sum_{n=1}^{C} p^{ML}(x_j^1/\boldsymbol{w}_n)p^{ML}(x_j^2/\boldsymbol{w}_m)P^{RBF}(\boldsymbol{w}_n,\boldsymbol{w}_m) = \qquad (24)$$

$$= \max_{w_h \in W}\left\{\sum_{n=1}^{C} p^{ML}(x_j^1/\boldsymbol{w}_n)p^{ML}(x_j^2/\boldsymbol{w}_h)\,P^{RBF}(\boldsymbol{w}_n,\boldsymbol{w}_h)\right\}$$

where:

$$P^{RBF}(\boldsymbol{w}_n,\boldsymbol{w}_h) = \sum_{k=1}^{K}\sum_{q=1}^{Q} P^{RBF}(\boldsymbol{w}_n,\boldsymbol{w}_h/\boldsymbol{j}_k^1,\boldsymbol{j}_q^2)P^{RBF}(\boldsymbol{j}_k^1,\boldsymbol{j}_q^2). \qquad (25)$$

Analogously, the second hybrid classifier (let us call it RBF-hybrid cascade classifier) is obtained by merging the joint probabilities estimated by the ML cascade classifier with the conditional densities $p^{RBF}(x_j^1,x_j^2/\boldsymbol{w}_n,\boldsymbol{w}_h)$ that can be estimated by using the RBF cascade classifier parameters. Hence, the corresponding classification rule is the following:

$$l_j^2 = w_m \in \Omega \ \textit{ if and only if } \ \sum_{n=1}^{C} p^{RBF}(x_j^1,x_j^2/\boldsymbol{w}_n,\boldsymbol{w}_m)P^{ML}(\boldsymbol{w}_n,\boldsymbol{w}_m) =$$

$$= \max_{w_h \in W}\left\{\sum_{n=1}^{C} p^{RBF}(x_j^1,x_j^2/\boldsymbol{w}_n,\boldsymbol{w}_h)P^{ML}(\boldsymbol{w}_n,\boldsymbol{w}_h)\right\} \qquad (26)$$

where the conditional densities $p^{RBF}\left(x_j^1, x_j^2 \big/ w_n, w_h\right)$ can be approximated by:

$$p^{RBF}\left(x_j^1, x_j^2 \big/ w_n, w_h\right) \cong \frac{\sum_{q=1}^{Q}\sum_{k=1}^{K} P^{RBF}\left(w_n, w_h \big/ j_k^1, j_q^2\right) P^{RBF}\left(j_k^1, j_q^2\right) p^{RBF}\left(x_j^1 / j_k^1\right) p^{RBF}\left(x_j^2 / j_q^2\right)}{\sum_{q=1}^{Q}\sum_{k=1}^{K} P^{RBF}\left(w_n, w_h \big/ j_k^1, j_q^2\right) P^{RBF}\left(j_k^1, j_q^2\right)}. \tag{27}$$

The use of these hybrid classifiers allows one to obtain a multiple classifier architecture composed of four classifiers. It is worth noting that it is possible to further increase the number of classifiers by extending the aforementioned procedure to the case of more RBF neural network architectures with different numbers of hidden units.

V. Multiple Cascade Classifier Architecture: Unsupervised Combination Strategies

In the proposed system, the classification results provided by the N members of the considered pool of cascade classifiers are combined by using classical multiple-classifier strategies. In particular, we consider two simple and widely used combination procedures: *Majority Voting* and *Combination by Bayesian Average* [5]. Both procedures exhibit the common peculiarity of requiring no prior training to carry out the combination process. This is a mandatory requirement in our approach, as we have no ground truth information (and hence no training set) for the image $\mathbf{X}_2$.

The *Majority Voting* procedure faces the combination problem by considering the results of each single classifier in terms of the class labels assigned to the patterns. A given input pattern receives N classification labels from the MCCS: each label corresponds to one of the C classes considered. The combination method is based on the interpretation of the classification label resulting from each classifier as a “vote” to one of the C land-cover classes. The data class that receives the largest number of votes is taken as the class of the input pattern.

The *Combination by Bayesian Average* strategy is based on the remark that, given the observations

x_j^1 and x_j^2, the N classifiers considered provide an estimate of the posterior probability $P\left(w_h / x_j^1, x_j^2\right)$ for each class $w_h \in W$. Therefore, a possible strategy for combining these classifiers consists in the computation of the average posterior probabilities, i.e.,

$$P^{ave}\left(w_h / x_j^1, x_j^2\right) = \frac{1}{N}\sum_{g=1}^{N} \hat{P}_g\left(w_h / x_j^1, x_j^2\right) \tag{28}$$

where $\hat{P}_g\left(w_h / x_j^1, x_j^2\right)$ is the estimate of the posterior probability $P\left(w_h / x_j^1, x_j^2\right)$ provided by the g-th classifier. The classification process is then carried out according to the Bayes rule by selecting the land-cover class associated with the maximum average probability.

VI. Experimental Results

To assess the effectiveness of the proposed approach, different experiments were carried out on a data set made up of two multispectral images acquired by the Thematic Mapper (TM) sensor of the Landsat 5 satellite. The selected test site was a section (412×382 pixels) of a scene including Lake Mulargias on the Island of Sardinia, Italy. The two images used in the experiments were acquired in September 1995 (t_1) and July 1996 (t_2). Figure 4 shows channels 2 of both images. Five land-cover classes (i.e., urban area, forest, pasture, water body, and vineyard), which characterize the test site at the above-mentioned dates, were considered. The available ground truth was used to derive a training set and a test set for each image (see Table I). To carry out the experiments, we assumed that only the training set associated with the image acquired in September 1995 was available. We used the training set of the July 1996 image only for comparisons with completely supervised classifiers.

Partially unsupervised ML and RBF neural-network cascade classifiers were applied to the September 1995 and July 1996 images. For the ML cascade classifier, the assumption of Gaussian distributions was made for the density functions of the dasses (this is a reasonable assumption as we considered TM images). Concerning the RBF neural cascade classifier, in order to exploit its non-parametric nature, five texture features based on the Gray-Level Co-occurrence matrix (i.e., sum variance, sum average, correlation, entropy and difference variance) [20] were computed and given as input to the classifier in addition to the six TM channels. These features were obtained by using a window size equal to 7X7 and an interpixel distance equal to 1.

As regards the ML cascade classifier, the parameters of the Gaussian density functions of the classes at t_1 were computed in a supervised way by using the available training set for the September 1995 image (i.e., $\mathbf{Y}_1$). These values were also used to initialize the parameters of the conditional density functions of the classes at t_2. Concerning the RBF cascade classifier, several trials were carried out in order to derive an effective number of neurons to be used in the hidden layer. To this end, experiments were carried out using a standard RBF architecture trained by the available set $\mathbf{Y}_1$ and applied to the t_1 test set. The highest accuracy was obtained by an architecture composed of 35 hidden units. On the basis of this result, an architecture composed of 70 hidden units was used for the RBF cascade classifier (i.e., 35 units related to the t_1 image and 35 units related to the t_2 image). It is worth noting that the parameters of the 35 hidden units associated with $\mathbf{X}_1$ were fixed according to the values achieved in a supervised way in the aforementioned experiment. The values of the parameters of the 35 hidden units used to process the image $\mathbf{X}_2$ were initialized by applying an unsupervised clustering to that image.

The parameters of the vectors J related to the ML and RBF cascade classifiers were estimated in an unsupervised way by using the proposed formulations of the iterative EM algorithm (see (5)-(8), and (17)-(22)). Firstly, the ML cascade classifier was trained, and the patterns classified with a posterior probability higher than the threshold value ε=0.98 were used to generate the set $\hat{\mathbf{Y}}_2$ in order to support the RBF training process. The EM algorithms adopted for the ML and RBF partially unsupervised training processes converged in 11 and 25 iterations, respectively. At the end of the iterative process, the resulting estimates were used to perform the classification of the July 1996 image. In addition, from the considered ML and RBF cascade classifiers, the two hybrid ML and RBF neural-network cascade classifiers were derived according to the strategy described in Section IV. Also these hybrid classifiers were applied to the July 1996 image.

The classification accuracies and the kappa coefficients of accuracy exhibited by the aforementioned four partially unsupervised cascade classifiers on the t_2 test set are given in Table II. As one can see, the performances of all the classifiers are very good. In particular, the overall accuracies exhibited by both the RBF and RBF-hybrid dassifiers are very high (i.e., 96.10% and 95.38%, respectively), and also the overall accuracies provided by the ML and ML-hybrid classifiers are satisfactory (i.e., 91.48% and 91.79%, respectively). This confirms the effectiveness of the partially unsupervised training process. Comparisons between standard and hybrid classifiers (i.e., RBF vs. RBF-hybrid and ML vs. ML-hybrid) point out that these classifiers provided very similar overall accuracies. However, a deeper analysis of the results reveals some important differences between the considered classification techniques. For example, the accuracy exhibited by the RBF-hybrid cascade classifier on the vineyard class is significantly higher than the one exhibited by the RBF neural cascade classifier (i.e., 66.67% vs. 61.54%). If one considers the confusion matrices resulting from the aforementioned experiments (see Tables III (a)-(d)), one can verify other significant differences in the behaviors of the classifiers on the different classes. For example, the RBF classifier misclassifies 30 pasture patterns as belonging to urban areas, whereas the RBF-hybrid classifier never incurs such a classification error. This confirms that the assumption that the four classifiers incur quite uncorrelated errors is reasonable.

At this point, the four classifiers were combined by using both the Majority-Voting and the Combination by Bayesian Average strategies (concerning the Majority-Voting strategy, in the case where more than one class received the same number of votes, the class with the maximum posterior probability was chosen). The accuracies obtained on the July 1996 test set are given in Table IV. Both combination strategies provided very high accuracies on all the land-cover classes, with the exception of the vineyard class, which is a minority one. By comparing Tables II and IV, one can conclude that the classification accuracies obtained combining the results of the partially unsupervised cascade classifiers by the two combination strategies considered are significantly higher than the accuracy exhibited by the worst single classifier (i.e., 96.56% and 94.77% vs. 91.48%). In particular, the classification accuracy obtained by applying the majority rule strategy is also higher than those exhibited by all the single classifiers making up the ensemble.

As stated in the methodological part of the paper, the objective of the multiple-classifier approach is not only to improve the overall classification accuracy of the system but also to increase its robustness. In order to investigate this aspect, an experiment was carried out in which the failure of the training process of the RBF neural cascade classifier was simulated. In particular, in order to simulate this situation, the partially unsupervised training of the parameters of the RBF architecture was carried out by replacing the image $\mathbf{X}_2$ with the image $\mathbf{X}_1$. It is worth noting that the resulting incorrect estimation of the RBF parameters also affects the hybrid classifiers. Table V presents the classification accuracies obtained by this experiment. As can be seen, even though the overall accuracies exhibited by both the RBF and the RBF-hybrid cascade classifiers are very poor (i.e, 67.68% and 72.75%, respectively), both combination strategies (i.e., the Combination by Bayesian Average strategy and the majority rule) allow the presented system to achieve classification accuracies (i.e., 92.46% and 95.90%) higher than the ones yielded by all the single classifiers. This confirms that the proposed architecture based on multiple cascade classifiers permits one to increase the robustness of the system versus possible failures of the partially unsupervised training process of single cascade-classification techniques.

Finally, in order to completely assess the effectiveness of the proposed methodology, two additional experiments were carried out using a fully supervised standard RBF classifier. In the first experiment, the RBF classifier was trained on the September 1995 training set and tested on the July 1996 image. The obtained results are given in Table VI. As one can see, the standard supervised RBF neural-network classifier trained on the "old" training set was unable to classify the "new" image with an acceptable accuracy, thus confirming that the use of a more complex classification methodology based on a partially unsupervised training process is mandatory. In the second experiment, the RBF classifier was trained on the July 1996 training set and applied to the test set related to the same image (it is worth noting that this training set was not considered in the previous experiments as we assumed that it was not available). Table VII gives the obtained results. A comparison of these results with the ones provided in Table IV points out that the proposed system outperforms the standard supervised RBF classifier. This surprising result, which mainly depends on the ability of the proposed approach to exploit the temporal correlation between the two images considered, confirms the effectiveness of the presented methodology.

VII. Discussion and Conclusions

In this paper, a novel MCCS for a partially unsupervised updating of land-cover maps has been proposed. This system allows one to update the existing land-cover maps of a given area by exploiting a new remote-sensing image acquired on the investigated site, without requiring the related ground truth. The main features of the proposed system are the following:

a) capability to exploit the temporal correlation between multitemporal images in the process of partially unsupervised updating of land-cover maps;
b) capability to exploit, in a synergical way, the information provided by different classifiers;
c) robustness to the partially unsupervised training process, thanks to the use of different partially unsupervised classifiers;
d) capability to consider multisensor and multisource data in the process of updating of land-cover maps (thanks to the availability of non-parametric classification algorithms in the ensemble).

Concerning the methodological novelties of this work, besides the definition of the global architecture of the system, some specific aspects should be pointed out: the use of cascade classifiers to solve the partially unsupervised classification problem; the original RBF neural-network architecture capable to exploit the temporal correlation between pairs of multitemporal remote-sensing images; the specific formulation of the EM algorithm within the framework of the cascade-classification decision rule for the training of the RBF cascade-classifier; the proposed ML and RBF hybrid cascade classifiers.

Due to the partially unsupervised nature of the proposed cascade classifiers considered in the ensemble, it is not possible to guarantee in all cases the convergence of the estimation process to accurate values of the classifier parameters. The accuracy obtained at convergence depends both on the reliability of the initialization conditions of the partially unsupervised estimation procedures and on the specific classification algorithm considered. However, the use of the multiple cascade-classifier architecture reduces the overall probability that the system may not succeed, thus increasing the robustness of the architecture to the probability of failure of the partially unsupervised training of each single classifier.

In the experiments carried out on different remote-sensing data sets, the proposed system proved effective, providing both high classification accuracy and high roboustness. Consequently, it seems a very promising tool to be integrated into a GIS system for a regular updating of land-cover maps. It is worth noting that, in the case where an "old" ground truth is not available, the land-cover map itself can be considered as the training set $\mathbf{Y}_1$ required for the partially unsupervised training process of the proposed system (however, in this situation, the possible errors present in the original land-cover map may affect the accuracy of the system).

The future developments of this work will be oriented in two different directions:

1) developing a procedure that, given the two images $\mathbf{X}_1$ and $\mathbf{X}_2$ and the training set $\mathbf{Y}_2$, may identify the probability of a failure of the partially unsupervised training of each cascade classifier and consequently prevent such a situation;
2) extending the partially unsupervised cascade-classification approach to other kinds of classification techniques to be integrated into the classifier ensemble.

ACKNOWLEDGMENT

This research was supported by the Italian Space Agency (ASI).

REFERENCES

[1] J. A. Richards, *Remote sensing digital image analysis*. New York: 2nd edn. Springer-Verlag, 1993.

[2] P.H. Swain, "Bayesian classification in time-varying environment," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 8, pp. 880-883, 1978.

[3] L. Bruzzone and D. Fernández Prieto, "Unsupervised retraining of a maximum-likelihood classifier for the analysis of multitemporal remote-sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, pp. 456-460, 2001, 2001.

[4] L. Bruzzone and D. Fernández Prieto, “A partially unsupervised approach to the automatic classification of multitemporal remote-sensing images,” *Pattern Recognition Letters*, in press, 2002.

[5] J. Kittler, A. Hojjatoleslami, and T. Windeatt, “Strategies for combining classifiers employing shared and distinct pattern representations,” *Pattern Recognition Letters*, vol. 18, pp. 1373-1377, 1997.

[6] L. Bruzzone and S.B. Serpico, “An iterative technique for the detection of land-cover transitions in multitemporal remote-sensing images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 35, pp. 858-867, 1997.

[7] L. Bruzzone and D. Fernández Prieto, “A technique for the selection of kernel-function parameters in RBF neural networks for classification of remote-sensing images,” *IEEE Transactions on Geoscience and Remote-Sensing*, vol. 37, pp. 1179-1184, 1999.

[8] C.M. Bishop, *Neural networks for pattern recognition.* Oxford: Clarendon Press, 1995.

[9] A.P. Dempster, N.M. Laird, and D.B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of Royal Statistic. Soc.*, vol. 39**,** pp. 1-38, 1977.

[10] T.K. Moon, “The Expectation-Maximization algorithm,” *Signal Processing Magazine*, vol. 13, pp. 47-60, 1996.

[11] B.M. Shahshahani and D. Landgrebe, “The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon,” *IEEE Transactions on Geoscience and Remote-Sensing*, vol. 32, pp. 1087-1095, 1994.

[12] L. Bruzzone, D. Fernández Prieto, and S.B. Serpico, “A neural statistical approach to multitemporal and multisource remote-sensing image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, pp. 1350-1359, 1999.

[13] D.J. Miller and S.U. Hasan, “Combined learning and use for a mixture model equivalent to the RBF classifier,” *Neural Computation*, vol. 10, pp. 281-293, 1998.

[14] L. Lam and C.Y. Suen, “Application of majority voting to pattern recognition: An analysis of its behavior and performance,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 27, pp. 553-568, 1997.

[15] J. Kittler, M. Hatef, R.P.W. Duin, and J. Mates, “On combining classifiers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 126-239, 1998.

[16] J.A. Benediktsson and P.H. Swain, “Consensus theoretic classification methods,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, pp. 688-704, 1992.

[17] J.A. Benediktsson, J.R. Sveinsson, O.K. Ersoy, and P.H. Swain, “Parallel consensual neural networks,” *IEEE Transactions on Neural Networks*, vol. 8, pp. 54-64, 1997.

[18] T.G. Dietterich, “Ensemble methods in machine learning,” *in Proc. of 1st International Workshop on Multiple Classifier Systems*, Cagliari, Italy, pp. 1-15, 2000.

[19] A.J.C. Sharkey, N.E. Sharkey, U. Gerecke, and G.O. Chandroth, “The test and select approach to ensemble combination,” *in Proc. of 1st International Workshop on Multiple Classifier Systems*, Cagliari, Italy, pp. 30-44, 2000.

[20] R.M. Haralick, K. Shanmugan, and I. Dinstein, “Textural features for image classification,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3, pp. 610-621, 1973.

FIGURE CAPTIONS

Fig. 1. General architecture of the proposed system.

Fig. 2. Standard architecture of a supervised RBF neural-network classifier.

Fig. 3. Architecture of the proposed partially unsupervised RBF neural cascade classifier (solide line). The architecture of the standard RBF neural network used for the supervised estimation of the t_1 statistical parameters is also shown (dashed line).

Fig. 4. Bands 5 of the Landsat-5 TM images utilized for the experiments: (a) image acquired in September 1995; (b) image acquired in July 1996.

TABLE CAPTIONS

Table I. Number of patterns in the training and test sets for both the September 1995 and July 1996 images.

Table II. Classification accuracies obtained by the four partially unsupervised cascade classifiers included in the proposed multiple classifier architecture (July 1996 test set).

Table III. Confusion matrices that resulted from the classification of the July 1996 test set by using the proposed partially unsupervised techniques: a) ML cascade classifier; b) RBF neural cascade classifier; c) ML-hybrid cascade classifier; d) RBF-hybrid neural cascade classifier.

Table IV. Overall classification accuracies exhibited by the proposed multiple cascade classifier system.

Table V. Overall classification accuracies exhibited by the four partially unsupervised cascade-classifiers included in the proposed multiple classifier architecture (July 1996 test set). The results are related to the case in which a failure in the partially unsupervised training of the RBF cascade-classifier was simulated. The overall accuracy obtained after combining the proposed classifiers is also given.

Table VI. Classification accuracies exhibited by a standard supervised RBF neural classifier trained on the September 1995 image and tested on the July 1996 image.

Table VII. Classification accuracies exhibited by a standard supervised RBF classifier trained and tested on the July 1996 image.

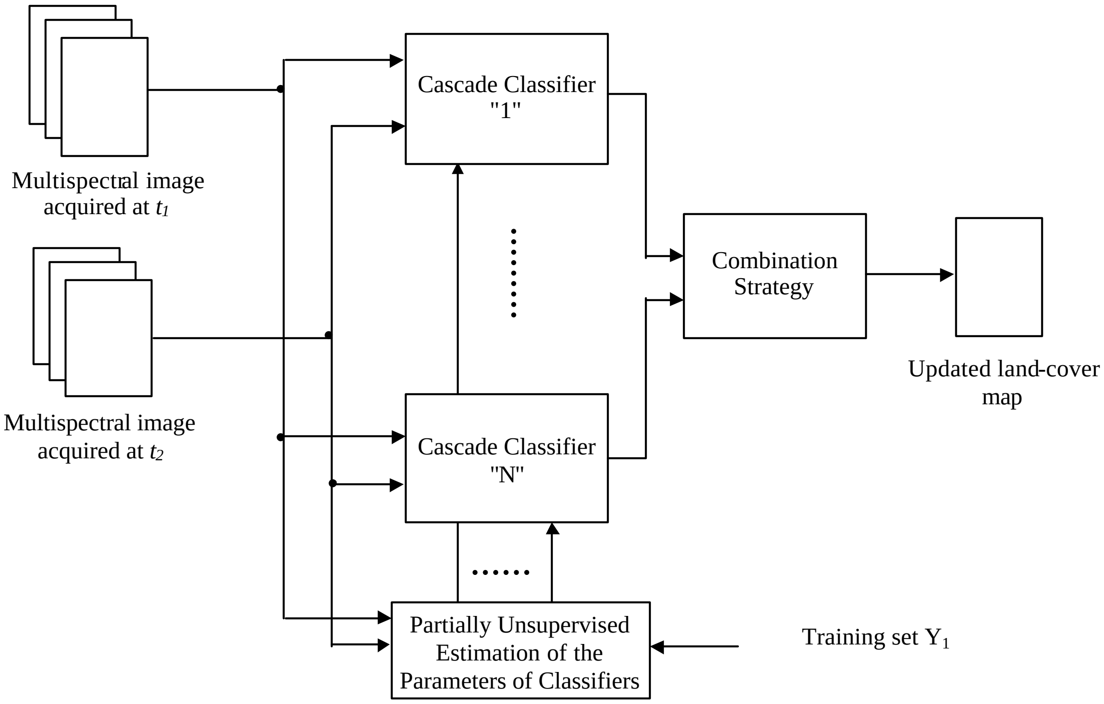
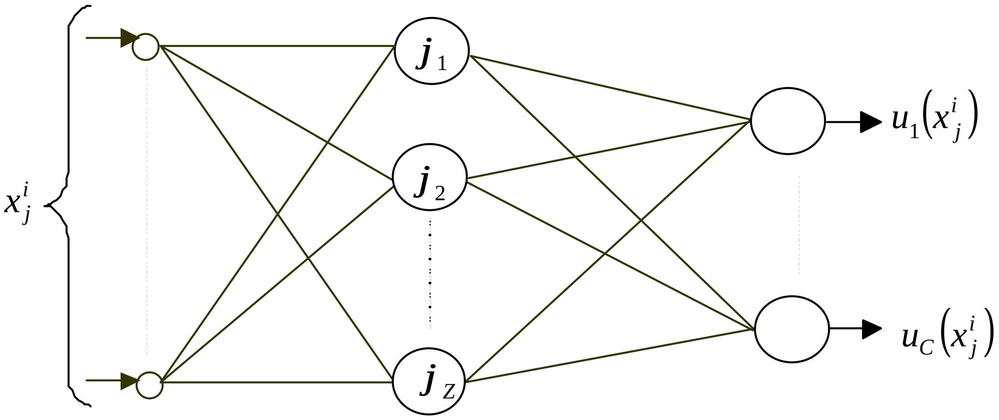

Cascade Classifier "1"
Multispectral image acquired at t_1
Multispectral image acquired at t_2
Combination Strategy
Updated land-cover map
Cascade Classifier "N"
Partially Unsupervised Estimation of the Parameters of Classifiers
Training set Y_1

Fig. 1



$$u_r\left(X_i\right)=\sum_{s=1}^{S} w_s^r j_s\left(X_i\right)$$

Fig. 2

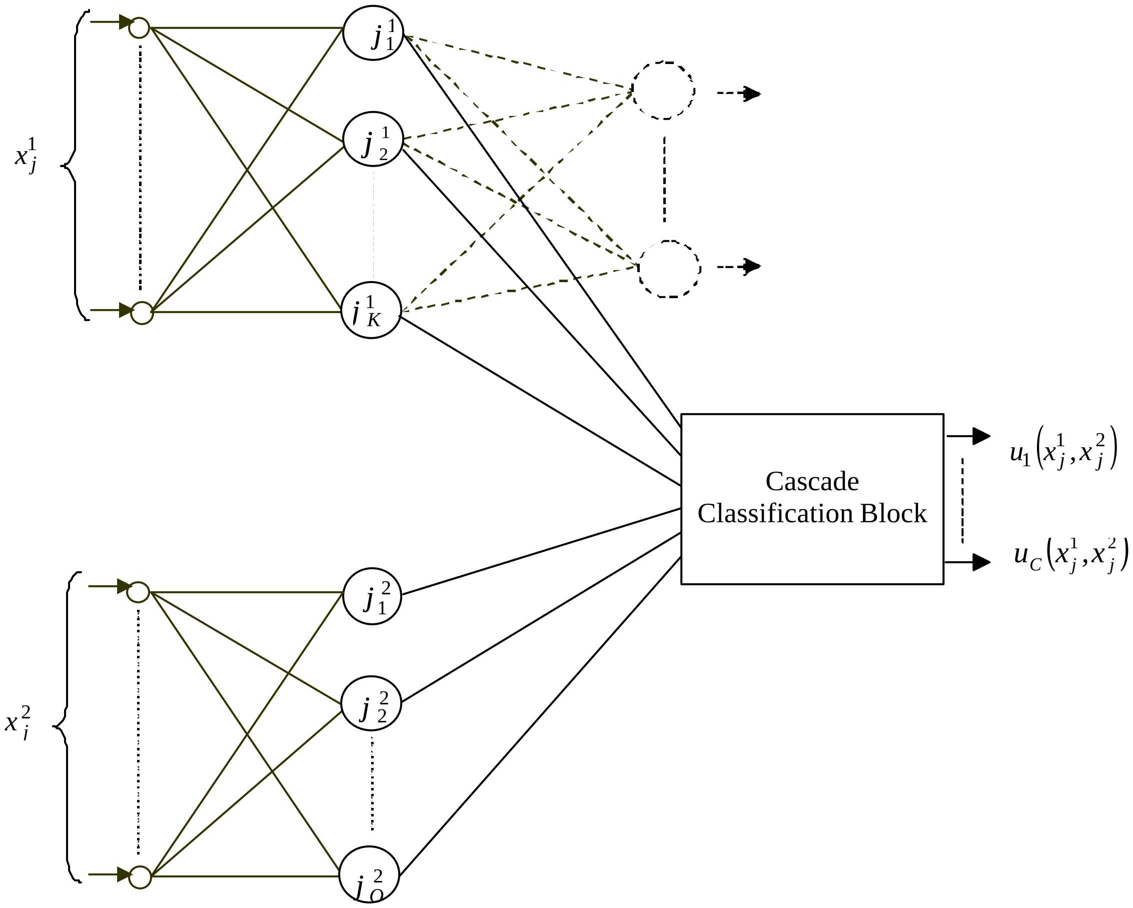
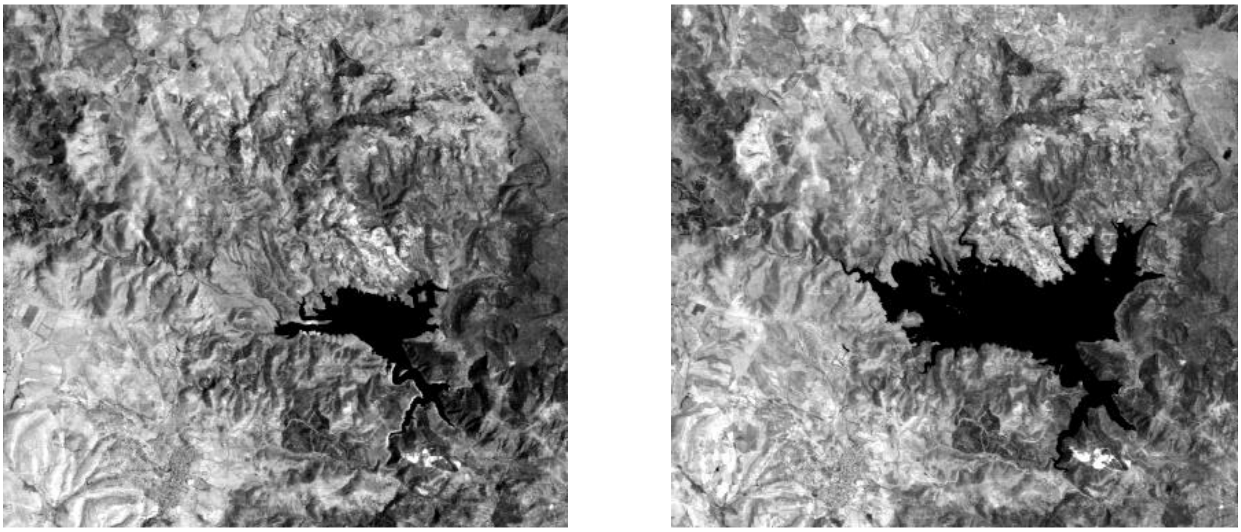

x_j^1
j_1^1
j_2^1
j_K^1
Cascade Classification Block
$u_1(x_j^1, x_j^2)$
$u_C(x_j^1, x_j^2)$
x_i^2
j_1^2
j_2^2
j_Q^2

Fig. 3



(a) (b)

Fig. 4

TABLE I

Land-cover class	Number of patterns	
	Training set	Test set
Pasture	554	589
Forest	304	274
Urban area	408	418
Water body	804	551
Vineyard	179	117
Overall	2249	1949

TABLE II

Land-cover class	Classification accuracy (%)			
	ML	RBF	ML-hybrid	RBF-hybrid
Pasture	83.53	94.91	85.23	94.40
Forest	97.45	100.00	97.45	98.91
Urban area	95.69	99.76	94.98	96.41
Water body	100.00	100.00	100.00	100.00
Vineyard	62.39	61.54	61.54	66.67
Overall	91.48	96.10	91.79	95.38
Kappa coefficient	0.88	0.94	0.89	0.93

TABLE III

	Pasture	Forest	Urban area	Water body	Vineyard
Pasture	492	12	85	0	0
Forest	2	267	2	0	3
Urban area	5	5	400	0	8
Water body	0	0	0	551	0
Vineyard	23	11	10	0	73

(a)

	Pasture	Forest	Urban area	Water body	Vineyard
Pasture	559	0	30	0	0
Forest	0	274	0	0	0
Urban area	0	0	417	1	0
Water body	0	0	0	551	0
Vineyard	31	11	3	0	72

(b)

	Pasture	Forest	Urban area	Water body	Vineyard
Pasture	502	15	72	0	0
Forest	2	267	2	0	3
Urban area	5	7	397	0	9
Water body	0	0	0	551	0
Vineyard	21	11	13	0	72

(c)

	Pasture	Forest	Urban area	Water body	Vineyard
Pasture	556	23	0	10	0
Forest	0	271	0	2	1
Urban area	15	0	403	0	0
Water body	0	0	0	551	0
Vineyard	21	0	3	15	78

(d)

TABLE IV

Land-cover class	Classification accuracy (%)	
	Bayesian Average	Majority rule
Pasture	91.51	94.06
Forest	99.27	99.64
Urban area	98.09	99.28
Water body	100.0	100.0
Vineyard	64.10	76.06
Overall	94.77	96.56
Kappa coefficient	0.93	0.95

TABLE V

Overall classification accuracy (%)					
ML	RBF	RBF-hybrid	ML-hybrid	Bayesian average	Majority rule
91.48	67.68	72.75	91.74	92.46	95.90

TABLE VI

Land-cover class	Classification accuracy (%)
Pasture	47.70
Forest	94.16
Urban area	66.27
Water body	100.00
Vineyard	45.30
Overall	72.85
Kappa coefficient	0.65

TABLE VII

Land cover class	Classification accuracy (%)
Pasture	89.64
Forest	99.27
Urban area	88.28
Water body	100.00
Vineyard	67.52
Overall	92.30
Kappa coefficient	0.89