Advanced Supervised Spectral Classifiers for Hyperspectral Images: A Review

Pedram Ghamisi, *Member, IEEE,* Javier Plaza, *Senior Member, IEEE,* Yushi Chen, *Member, IEEE,* Jun Li, *Senior Member, IEEE,* Antonio Plaza, *Fellow, IEEE*

Abstract—**Hyperspectral image classification has been a vibrant area of research in recent years. Given a set of observations, i.e., pixel vectors in a hyperspectral image, classification approaches try to allocate a unique label to each pixel vector. However, the classification of hyperspectral images is a challenging task due to a number of reasons such as the presence of redundant features, or the imbalance between the limited number of available training samples, as well as the high dimensionality of the data. The aforementioned issues (among others) make the commonly used classification methods designed for the analysis of gray scale, color, or multispectral images inappropriate for hyperspectral images. To this end, several spectral classifiers have been specifically developed for hyperspectral images or carried out on such data. Among those approaches, support vector machines, random forests, neural networks, deep approaches, and logistic regression-based techniques have gained a great interest in the hyperspectral community. This paper reviews most existing spectral classification approaches in the literature. Then, it critically compares the most powerful hyperspectral classification approaches from different points of view, including their classification accuracy, and computational complexity. The paper also provides several hints for readers about the logical choice of an appropriate classifier based on the application at hand.**

Index Terms—**Hyperspectral Image Classification, Support Vector Machines, Random Forests, Neural Networks, Extreme Learning Machine, Deep Learning, Multinomial Logistic Regression.**

I. INTRODUCTION

Imaging spectroscopy (also known as hyperspectral imaging) is an important technique in remote sensing. Hyperspectral imaging sensors often capture data from the visible through the near infrared wavelength ranges, thus providing hundreds of narrow spectral channels from the same area on the surface of the Earth.

These instruments collect data consisting of a set of pixels represented as vectors, in which each element is a measurement corresponding to a specific wavelength. The size of each vector is equal to the number of spectral channels or bands. Hyperspectral images usually consist of several hundred spectral data channels for the same area on the surface of the Earth, while in multispectral data the number of spectral channels are usually up to tens of bands [1]. The detailed spectral information collected by hyperspectral sensors increases the capability of discriminating between different land-cover classes with increased accuracy. A number of operational hyperspectral imaging systems are currently available, providing a large volume of image data that can be used for a wide variety of applications such as ecological science, geological science, hydrological science, precision agriculture and military applications.

Due to the detailed spectral information available from the hundreds of (narrow) bands collected by hyperspectral sensors, accurate discrimination of different materials is possible. This fact makes hyperspectral data a valuable source of information to be fed to advanced classifiers. The output of the classification step is known as *classification map*.

Fig. 1 categorizes different groups of classifiers with respect to different criteria, followed by a brief description. Since classification is a wide field of research and it is not feasible to investigate all those approaches in a single paper, we tried to narrow down our description by excluding the green parts in Fig. 1, which have been extensively covered in other contributions. We reiterate that our main goal in this paper is to provide a comparative assessment and best practice recommendations for the remaining contributions in Fig. 1.

With respect to the availability of training samples, classification approaches can be split into two categories, supervised and unsupervised classifiers. Supervised approaches classify input data using a set of representative samples for each class, known as *training samples*. Training samples are usually collected either by manually labeling a small number of pixels in an image or based on some field measurements [2]. In contrast, unsupervised classification (also known as clustering) does not consider training samples. This type of approaches classify the data only based on an arbitrary number of initial "cluster centers" that may be either user-specified or may be quite arbitrarily selected. During the processing, each pixel is associated with one of the cluster centers based on a similarity criterion [1, 3]. Therefore, pixels that belong to different clusters are more dissimilar to each other compared to pixels within the same clusters [4, 5].

Pedram Ghamisi is with German Aerospace Center (DLR), Remote Sensing Technology Institute (IMF) and Technische Universität München (TUM), Signal Processing in Earth Observation, Munich, Germany (corresponding authors e-mail: p.ghamisi@gmail.com).

Javier Plaza and Antonio Plaza are with the Department of Technology of Computers and Communications, University of Extremadura, Spain

Yushi Chen is with the Department of Information Engineering, Harbin Institute of Technology (e-mail: chenyushi@hit.edu.cn).

Jun Li is with the Guangdong Provincial Key Laboratory of Urbanization and Geo-simulation, Center of Integrated Geographic Information Analysis, School of Geography and Planning, Sun Yat-sen University, Guangzhou, 510275, China. E-mail: lijun48@mail.sysu.edu.cn.

This research has been supported by the Chinese 1000 people program B under project 41090427 and Guangdong Provincial Science Foundation under project 42030397. This work is also partly supported by the Alexander von Humboldt Fellowship for postdoctoral researchers.

There is a vast amount of literature on unsupervised classification approaches. Among those methods, Kmeans [6], ISODATA [7] and Fuzzy Cmeans [8] rank amongst the most popular unsupervised approaches. This set of approaches is known for being very sensitive to its initial cluster configuration and may be trapped into sub-optimal solutions [9]. To address this issue, researchers have tried to improve the resilience of the Kmeans (and its family) by optimizing it with bio-inspired optimization techniques [3]. Since supervised approaches consider class specific information, which is provided by training samples, they lead to more precise classification maps than unsupervised approaches. In addition to unsupervised and supervised approaches, semi-supervised techniques have been introduced [10, 11]. In this type of methods, the training is based on both labeled training samples as well as unlabeled samples. In the literature, it has been shown that the classification accuracy obtained with semi-supervised approaches can outperform that obtained by supervised classification.

In this paper, our focus is on supervised classification approaches. The remainder of this section is organized as follows: First, we present the concept of supervised classification by setting some notations. Then, we discuss parametric versus nonparametric classification and address some specific challenges for classification of hyperspectral data. Next, we provide a detailed literature review followed by a brief comment on strategies for classification accuracy assessment. The section concludes with a summary of the main contributions of the paper as a prelude to the description of relevant techniques in subsequent sections.

A. Supervised Classification of Hyperspectral Data

A hyperspectral data set can be seen as a stack of many pixel vectors, here denoted by $\mathbf{x} = (x_1, ..., x_d)^T$, where d represents the number of bands or the length of the pixel vector. A common task when interpreting remote sensing images is to differentiate between several land cover classes. A *classification algorithm* is used to separate between different types of patterns [5]. In remote sensing, classification is usually carried out in a feature space [12]. In general, the initial set of features for classification contains the spectral information, i.e., the wavelength information for the pixels [1]. In this space, each feature is presented as one dimension and pixel vectors can be represented as points in this d-dimensional space. A classification approach tries to assign unknown pixels to one of y classes $\Omega = \{y_1, y_2, ..., y_K\}$, where K represents the number of classes, based on a set of representative samples for each class referred to as training samples. The individual classes are discriminated based either on the similarity to a certain class or by decision boundaries, which are constructed in the feature space [5].

B. Parametric versus Nonparametric Classification

From another perspective, classification approaches can be split into parametric and non-parametric. For example, the widely used supervised maximum likelihood classifier (MLC) is often applied in the parametric context. In that manner, the MLC is based on the assumption that the probability density function for each class is governed by the Gaussian distribution [13]. In contrast, nonparametric methods are not constrained by any assumptions on the distribution of the input data. Hence techniques such as SVMs, neural networks, decision trees, and ensemble approaches (including random forests) can be applied even if the class conditional densities are not known or cannot be estimated reliably [1]. Therefore, for hyperspectral data with a limited number of available training samples, such techniques may lead to more accurate classification results.

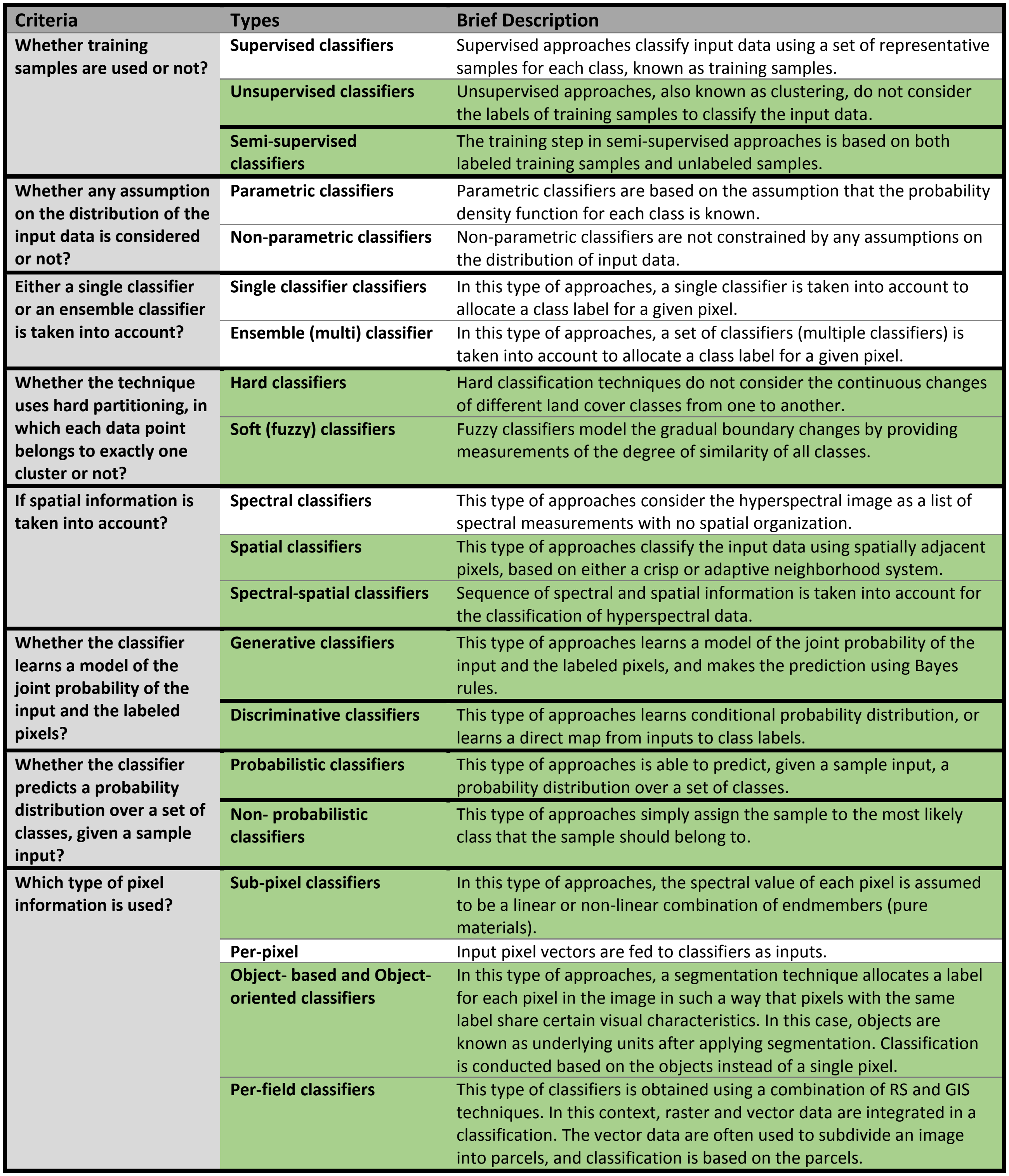
C. Challenges for the Classification of Hyperspectral Data

In this section, we discuss on some specific characteristics of hyperspectral data, which make the classification step challenging.

1) Curse of Dimensionality: In [14–16], researchers have reported some distinguishing geometrical, statistical, and asymptotical properties of high-dimensional data through some experimental examples such as: (1) as dimensionality increases, the volume of a hypercube concentrates in corners, or (2) as dimensionality increases, the volume of a hypersphere concentrates in an outside shell. With respect to these examples, the following conclusions have been drawn:

- A high-dimensional space is almost empty, which implies that multivariate data in $\mathbb{R}$ is usually in a lower dimensional structure. In other words, high-dimensional data can be projected into a lower subspace without sacrificing considerable information in terms of class separability [1].
- Gaussian distributed data have a tendency to concentrate in the tails while, uniformly distributed data have a tendency to be concentrated in the corners, which makes the density estimation of high-dimensional data for both distributions more difficult.
- Fukunaga [13] showed that there is a relation between the required number of training samples and the number of dimensions for different types of classifiers. The required number of training samples is linearly related to the dimensionality for linear classifiers and to the square of the dimensionality for quadratic classifiers (e.g., Gaussian MLC [13]).
- In [17], Landgrebe showed that too many spectral bands might be undesirable in terms of expected classification accuracy. When dimensionality (the number of bands) increases, with a constant number of training samples, a higher dimensional set of statistics must be estimated. In other words, although higher spectral dimensions increase the separability of the classes, the accuracy of the statistical estimation decreases. This leads to a decrease in classification accuracies beyond a number of bands. For the purpose of classification, these problems are related to the *curse of dimensionality*.

It is expected that, as dimensionality increases, more information is demanded in order to detect more classes with higher accuracy. At the same time, the aforementioned characteristics demonstrate that conventional techniques developed



Criteria	Types	Brief Description
Whether training samples are used or not?	**Supervised classifiers**	Supervised approaches classify input data using a set of representative samples for each class, known as training samples.
	Unsupervised classifiers	Unsupervised approaches, also known as clustering, do not consider the labels of training samples to classify the input data.
	Semi-supervised classifiers	The training step in semi-supervised approaches is based on both labeled training samples and unlabeled samples.
Whether any assumption on the distribution of the input data is considered or not?	**Parametric classifiers**	Parametric classifiers are based on the assumption that the probability density function for each class is known.
	Non-parametric classifiers	Non-parametric classifiers are not constrained by any assumptions on the distribution of input data.
Either a single classifier or an ensemble classifier is taken into account?	**Single classifier classifiers**	In this type of approaches, a single classifier is taken into account to allocate a class label for a given pixel.
	Ensemble (multi) classifier	In this type of approaches, a set of classifiers (multiple classifiers) is taken into account to allocate a class label for a given pixel.
Whether the technique uses hard partitioning, in which each data point belongs to exactly one cluster or not?	**Hard classifiers**	Hard classification techniques do not consider the continuous changes of different land cover classes from one to another.
	Soft (fuzzy) classifiers	Fuzzy classifiers model the gradual boundary changes by providing measurements of the degree of similarity of all classes.
If spatial information is taken into account?	**Spectral classifiers**	This type of approaches consider the hyperspectral image as a list of spectral measurements with no spatial organization.
	Spatial classifiers	This type of approaches classify the input data using spatially adjacent pixels, based on either a crisp or adaptive neighborhood system.
	Spectral-spatial classifiers	Sequence of spectral and spatial information is taken into account for the classification of hyperspectral data.
Whether the classifier learns a model of the joint probability of the input and the labeled pixels?	**Generative classifiers**	This type of approaches learns a model of the joint probability of the input and the labeled pixels, and makes the prediction using Bayes rules.
	Discriminative classifiers	This type of approaches learns conditional probability distribution, or learns a direct map from inputs to class labels.
Whether the classifier predicts a probability distribution over a set of classes, given a sample input?	**Probabilistic classifiers**	This type of approaches is able to predict, given a sample input, a probability distribution over a set of classes.
	Non- probabilistic classifiers	This type of approaches simply assign the sample to the most likely class that the sample should belong to.
Which type of pixel information is used?	**Sub-pixel classifiers**	In this type of approaches, the spectral value of each pixel is assumed to be a linear or non-linear combination of endmembers (pure materials).
	Per-pixel	Input pixel vectors are fed to classifiers as inputs.
	Object- based and Object-oriented classifiers	In this type of approaches, a segmentation technique allocates a label for each pixel in the image in such a way that pixels with the same label share certain visual characteristics. In this case, objects are known as underlying units after applying segmentation. Classification is conducted based on the objects instead of a single pixel.
	Per-field classifiers	This type of classifiers is obtained using a combination of RS and GIS techniques. In this context, raster and vector data are integrated in a classification. The vector data are often used to subdivide an image into parcels, and classification is based on the parcels.

Fig. 1. A terminology of classification approaches based on different criteria. In order to narrow down the research line of the paper, we intentionally avoid elaborate on the green parts.

for multispectral data may not be suitable for the classification of hyperspectral data.

The aforementioned issues related to the high-dimensional nature of the data have a dramatic influence on supervised classification techniques[18]. These techniques demand a large number of training samples (which is almost impossible to obtain in practice) in order to make a precise estimation. This problem is even more severe when dimensionality increases. Therefore, classification approaches developed on hyperspectral data need to be capable of handling high dimensional data when only a limited number of training samples is available.

2) Uncertainties: Uncertainties generated at different stages of data acquisition and classification procedure can dramatically influence the classification accuaries and the quality of the final classification map [19–22]. There are many reasons for such uncertainties, including atmospheric conditions at data acquisition time, data limitation in terms of radiometric and spatial resolutions, mosaicing several images and many others. Image registration and geometric rectification cause position uncertainty. Furthermore, algorithmic errors at the time of calibrating either atmospheric or topographic effects may lead to radiometric uncertainties [23].

3) Influence of Spatial Resolution: Classification accuracies can be highly influenced by the spatial resolution of the hyperspectral data. A higher spatial resolution can significantly reduce the mixed-pixel problem and detect more details of the scene. In [24], it was mentioned that classification accuracies are the result of a tradeoff between two aspects. The first aspect refers to the influence of boundary pixels on classification results. In this case, as spatial resolution becomes finer, the number of pixels falling on the boundary of different objects will decrease. The second aspect refers to the increased spectral variance of different land-covers associated with finer spatial resolution.

When we deal with low or medium spatial resolution optical data, the existence of many mixed pixels between different land-cover classes is the main source of uncertainties, which influence on classification results dramatically.

Fine spatial resolution can provide detailed information about shape and structure of different land-covers. Such information can also be fed to the classification system to further increase classification accuracy values and improve the quality of classification maps. The consideration of spatial information into the classification system is a vibrant research topic in the hyperspectral community, and it has been investigated in many works like [1, 25–29]. As mentioned, the consideration of spatial information in the classification system is out of the scope of this work, which focuses on supervised spectral classifiers. However, the use of high resolution hyperspectral images introduces some new problems, especially those caused by the presence of shadows, which leads to high spectral variations within the same land-cover class. These disadvantages may decline classification accuracy if classifiers cannot effectively handle such effects [30].

D. Literature Review

In this subsection, we briefly outline some of the most popular supervised classification methods for hyperspectral imagery. Some of these methods will be further detailed in subsequent sections of the paper.

1) Probabilistic approaches: A common subclass of classifiers is based on probabilistic approaches. This group of classifiers use statistical terminologies to find the best class for a given pixel. In contrast with those algorithms, which simply allocate a label with respect to a ”best” class, probabilistic algorithms output a probability of the pixel being a member of each of the possible classes [5, 13, 31]. The best class is normally then selected as the one with the highest probability.

For instance, the multinomial logistic regression (MLR) classifier [32], which is able to model the posterior class distributions in a Bayesian framework, supplies (in addition to the boundaries between the classes) a degree of plausibility for such classes [33]. Sparse MLR (SMLR), by adopting a Laplacian prior to enforce sparsity, leads to good machine generalization capabilities in hyperspectral classification [34, 35], though with some computational limitations. The logistic regression via splitting and augmented Lagrangian (LORSAL) algorithm opened the door to processing of hyperspectral images with median or big volume and a very large number of classes, using a high number of training samples [36, 37]. More recently, a subspace-based version of this classifier, called MLRsub [38], has also been proposed. The idea of applying subspace projection methods relies on the basic assumption that the samples within each class can approximately lie in a lower dimensional subspace. The exploration of MLR, SMLR, LORSAL and MLRsub for hyperspectral model present two important advantages. On the one hand, with the advantages of good algorithm generalization and fast computation, MLR has beenh1 widely aq used to model the spectral information of hyperspectral data [39–48]. On the other hand, as the structure of MLR classifiers is very open and flexible, composite kernel learning [49, 50] and multiple feature learning [51, 52] become active topics under the MLR model and lead to very competitive results for hyperspectral image classification problems.

2) Neural networks: The use of neural networks in complex classification scenarios is a consequence of their successful application in the field of pattern recognition [53]. Particularly in the 1990s, neural network approaches attracted many researchers in the area of classification of hyperspectral images [54, 55]. The advantage of such approaches over probabilistic methods are mainly resulting from the fact that neural networks do no need prior knowledge about the statistical distribution of the classes. The attractiveness of such increased due to the availability of feasible training techniques for nonlinearly separable data [56], although their use has been traditionally affected by their algorithmic and training complexity [57] as well as by the number of parameters that need to be tuned.

Several neural network-based classification approaches have been proposed in the literature considering both supervised and unsupervised, non-parametric approaches [58–62], being feedforward neural network (FN)-based classifiers the most commonly adopted ones. FNs have been well studied and widely used since the introduction of the well-known backpropagation algorithm (BP) [63], a first order gradient method for parameter optimization, which presents two main problems: slow convergence and the possibility of falling in local minima, especially when the parameters of the network are not properly fine-tuned. With the aim of alleviating the disadvantages of the original BP algorithm, several second order optimization-based strategies, which are faster and need less input parameters, have been proposed in the literature [64, 65]. Recently, the extreme learning machine (ELM) learning algorithm has been proposed to train single hidden layer feedforward neural networks (SLFN) [66, 67]. Then, the concept has been extended to multi-hidden-layer networks [68], radial basis function networks (RBF) [69], and kernel learning [70]. The main characteristic of the ELM is that the hidden layer (feature mapping) is randomly fixed and need not to be iteratively tuned. ELM based networks are remarkably efficient in terms of accuracy and computational complexity and has been successfully applied as nonlinear classifier for hyperspectral data providing comparable results with the state-of-the-art methodologies [71–74].

3) Kernel methods including support vector machines (SVMs): SVMs are another example of a supervised classification approach, which has been widely used for the classification of hyperspectral data due to their capability to handle high dimensional data with a limited number of training samples [1, 75, 76]. SVMs were originally introduced to classify linear classification problems. In order to generalize the SVM for non-linear classification problems, the so-called *kernel trick* was introduced [77]. The sensitivity to the choice of the kernel and regularization parameters are the most important disadvantages of a kernel SVM. For the former, the Gaussian radial basis function (RBF) is widely used in remote sensing [77]. The latter is classically addressed using cross-validation techniques using training data [78]. Gomez *et. al* proposed an approach by combining both labeled and unlabeled pixels using clustering and mean map kernel to increase the classification accuracy and reliability of SVM [79]. In [80], a local *k*-nearest neighbor adaption was taken into account to formulate localized variants of SVMs. Tuia and Camps-Vallas proposed a regularization approach to tackle the issue of kernel predetermination. The method was based on the identification of kernel structures through analysis of unlabeled pixels [81]. In [82], a so-called bootstrapped SVM was proposed as a modification of the SVM. The training strategy of the approach is as follows: an incorrectly classified training sample in a given learning step is removed from the training pool, re-assigned a correct label, and re-introduced into the training set in the subsequent training cycles.

In addition to the SVM, a composite kernel framework for the classification of hyperspectral images has been recently investigated. In [83], a linearly weighted composite kernel framework with SVMs has been used for the classification of hyperspectral data. However, classification using composite kernels and SVMs demands convex combination of kernels and a time-consuming optimization process. To overcome these limitations, a generalized composite kernel framework for spectral-spatial classification has been developed in [83]. The MLR [84–86] has been also investigated as an alternative to the SVM classifier for the construction of composite kernels, and a set of generalized composite kernels, which can be linearly combined without any constraint of convexity, were proposed.

4) Decision trees: Decision trees represent another subclass of nonparametric approaches, which can be used for both classification and regression. Safavian and Landgrebe [87] provided a good description of such classifiers. During the construction of a decision tree, the training set is progressively split into an increasing number of smaller, more homogeneous groups. This unique hierarchical concept is different from other classification approaches, which generally use the entire feature space at once and make a single membership decision per class [88]. The relative structural simplicity of decision trees as well as the relatively short training time required (compared to methods that can be computationally demanding) are the main advantages of such classifiers [1, 89, 90]. Moreover, decision tree classifiers make it possible to directly interpret class membership decisions with respect to the impact of individual features [5]. Although a standard decision tree may be deteriorated under some circumstances, its general concept is of interest and the classifier performance can be further improved in terms of classification accuracies by classifier ensembles or multiple classifier systems [91, 92].

5) Ensemble methods (multiple classifiers): Traditionally, a single classifier was taken into account to allocate a class label for a given pixel. However, in most cases, the use of an *ensemble of classifiers* (*multiple classifiers*) can be considered in order to increase the classification accuracies [1]. In order to develop an efficient multiple classifier, one needs to determine an effective combination of classifiers that is able to benefit each other while avoiding the weaknesses of them [91]. Two highly used multiple classifiers are boosting and bagging [91, 93, 94], which were elaborated in detail in [1].

6) Random forests: Random forests (RFs) were first introduced in [95], and they represent a popular ensemble method for classification and regression. This classifier has been widely used in conjunction with hyperspectral data since it does not assume any underlying probability distribution for input data. Moreover, it can provide a good classification result in terms of accuracies in an ill-posed situation when there is no balance between dimensionality and number of available training samples. In [96], *Rotation Forest* is proposed based on the idea of RFs to encourage simultaneously both member diversities and individual accuracy within a classifier ensemble. For a detailed description of this approach please see [1, 92, 95, 97, 98].

7) Sparse representation classifiers (SRCs): Another important development has been the use of SRCs with dictionary-based generative models [99, 100]. In this case, an input signal is represented by a sparse linear combination of samples (*atoms*) from a dictionary [99], where the training data is generally used as the dictionary. The main advantage of SRCs is that it avoids the heavy training procedure which a supervised classifier generally conducts, and the classification is performed directly on the dictionary. Given the availability of sufficient training data some researchers have also developed discriminative as well as compact class dictionaries to improve classification performance [101].

8) Deep learning: Deep learning is a kind of neural network with multi-layers, typically deeper than three layers, which tries to hierarchically learn the features of input data. Deep learning is a fast-growing topic, which has shown usefulness in many research areas, including computer vision and natural language processing [102]. In the context of remote sensing, some deep models have been proposed for hyperspectral data feature extraction and classification [103]. Stacked auto-encoder (SAE) and auto-encoder with sparse constrain were proposed for hyperspectral data classification [104, 105]. Later, another deep model, deep belief network (DBN), was proposed for the classification of hyperspectral data [106]. Very recently, an unsupervised convolutional neural network (CNN) was proposed for remote sensing image analysis, which uses greedy layer-wise unsupervised learning to formulate a deep CNN model [107].

E. Classification Accuracy Assessment

Accuracy assessment is a crucial step to evaluate the efficiency and capability of different classifiers. There are many sources of errors such as: errors caused by the classification algorithm, position errors caused by the registration step, mixed pixels and unacceptable quality of training and test samples. In general, it is assumed that the difference between the classified image and reference data is because of the errors caused by the classification algorithm itself [23]. A considerable number of works and reviews on classification accuracy assessment have been conducted in the area of remote sensing [1, 108–113].

F. Contributions of the Paper

The main aim of this paper is to critically compare representative spectral-based classifiers (such as those outlined in subsection I-D) from different perspectives. Without any doubt, classification plays an important role for the analysis of hyperspectral data. There are many papers dealing with advanced classifiers but, to the best of our knowledge, there is no contribution in the literature that critically reviews and compares advanced classifiers with each other, providing recommendations on best practice when selecting a specific classifier for a given application domain.

To make our research line more specific, in this paper we only consider spectral and per-pixel based classifiers. In other words, spatial classifiers, fuzzy approaches, sub-pixel classifiers, object-based approaches, and per-field RS-GIS approaches are considered to be out of the scope of the paper.

Compared to previous review papers such as [114] published in 2009, which provides a general review on the advances in techniques for hyperspectral image processing till that date, this paper is specifically on spectral classifiers, which includes the most recent and advance spectral classification approaches in the hyperspectral community (with many new developments since the previous publication of that paper).

In addition, we believe that a few specific classifiers have gained great interest in the hyperspectral community due to their capability to handle high dimensional data with a limited number of training samples. Among those approaches, neural networks, random forests, MLR, SVM, deep convolutional neural network-based classifiers are the most widely used ones at present. As a result, we first elaborate on these approaches and then, we further compare them based on different scenarios, such as the capability of the methods in terms of having different number of training samples, spatial resolution, stability, complexity and automation of the considered classifiers. The aforementioned approaches are applied to three widely used hyperspectral images (e.g., Indian Pines, Pavia University, and Houston) and the obtained results are critically compared with each other. In order to make the equations easier to follow, Table I details all the notations, which have been used in this paper.

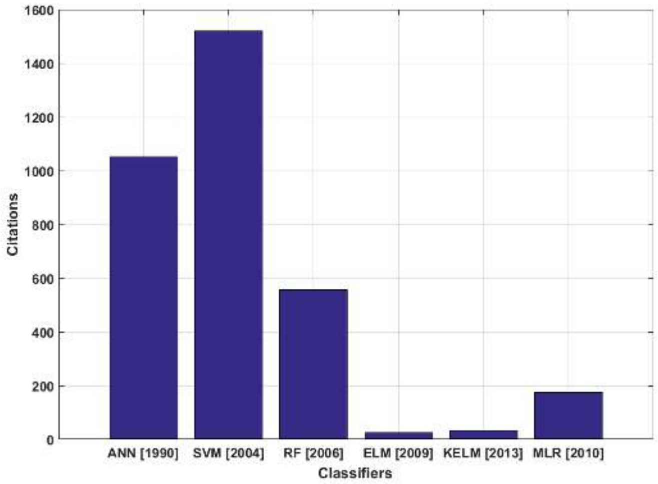


Fig. 2. The number of citations associated to each classifier.

Fig. 2 shows the classification approaches investigated in this paper along with their publication year and the number of obtained citations so far. However, it should be noted that in each paper, authors cited different papers as the original one. Here, we use the most cited paper of the corresponding classifier used in the remote sensing community. Here, we used [58] for neural network, [92] for RF, [84] for MLR, [115] for SVM, [116] for ELM, and [117] for KELM. Since CNN has very recently been published, it was not shown in Fig. 2.

II. Neural Networks

Artificial neural networks (ANNs) have been traditionally used in multi-hyperspectral data classification. Particularly, FNs have been extensively applied due to their ability to approximate complex nonlinear mappings directly from the input samples using one single hidden layer [118]. Traditional learning techniques are based on the original BP algorithm [63]. The most popular group is gradient descent-based learning methods, which are generally slow and may easily converge to a local minima. These techniques adjust the weights in the steepest descent direction (negative of the gradient), which is the direction in which the performance function decreases most rapidly, but this does not necessarily produce the fastest convergence [64]. In this sense, several conjugate gradient algorithms have been proposed to perform

TABLE I
THE LIST OF NOTATIONS AND ACRONYMS.

Notations	Definition	Notations	Definition	Notations	Definition	Notations	Definition
$\mathbf{x}$	Pixel vector	d	Number of bands	b	Bias	λ	Regularization parameter
Φ	Transformation	C	Regularization parameter	υ	Stack variable	k	Kernel
$\|\|.\|\|$	Euclidean norm	w	Normal vector	L	Number of hidden nodes	K	Number of classes
y	Classification label	$\mathbf{w}$	Input Weight	n	Number of training samples	$p(y_i\|\mathbf{x}_i)$	Probability of pixel i
α	Lagrange multiplier	β	Output weight	$\mathbf{v}$	Visible units	$\mathbf{h}$	Hidden units

a search along conjugate directions, which generally result in faster convergence. These algorithms usually require high storage capacity and are widely used in networks with large number of weights. Last, Newton's based learning algorithms generally provide better and fast optimization than conjugate gradient methods. Based in the Hessian matrix (second derivatives) of the performance index at the current values of the weight and biases, their convergence is faster although their complexity usually introduce an extra computational burden for the calculation of the Hessian matrix.

Recently, the ELM algorithm has been proposed to train single hidden layer feedforward neural networks [66, 67], which has emerged as an efficient algorithm that provides accurate results in much less time. Traditional gradient-based learning algorithms assume that all the parameters (weight and bias) of the feedforward networks need to be tuned, establishing a dependency between different layers of parameters and fostering very slow convergence. In [119, 120], it was first shown that a SLFN (with N hidden nodes) with randomly chosen input weights and hidden layer biases can exactly learn N distinct observations, which means that it may not be necessary to adjust the input weights and first hidden layer biases in applications. In [66], it was proved that the input weights and hidden layer biases of a SLFN can be randomly assigned if the activation function of the hidden layer is infinitely differentiable, which allow to determinate the rest of parameters (weights between hidden and output layers) analytically, being the SLFN a linear system. This fact leads to a significative decrease of the computational complexity of the algorithm, making it much faster than its predecessors, and turning ELM in the main alternative specially in the analysis of large amount of data.

Let $(\mathbf{x}_i\mathbf{t}_i)$ be n distinct samples where $\mathbf{x}_i = [x_{i1}, x_{i2}, ..., x_{id}]^T \in \mathbb{R}^d$ and $\mathbf{t}_i = [t_{i1}, t_{i2}, ..., t_{iK}]^T \in \mathbb{R}^K$, where d is the spectral dimensionality of the data and K the number of spectral classes. A SLFN with L hidden nodes and activation function $f(x)$ can be expressed as:

$$\sum_{i=1}^{L} \beta_i f_i(\mathbf{x}_j) = \sum_{i=1}^{L} \beta_i f(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{o}_j, j = 1, ..., n, \quad (1)$$

where $\mathbf{w}_i = [w_{i1}, w_{i2}, ..., w_{id}]^T$ is the weight vector connecting the *i*th hidden node and the input nodes, $\beta_i = [\beta_{i1}, \beta_{i2}, ..., \beta_{iK}]^T$ is the weight vector connecting the *i*th hidden node and the output nodes, b_i is the bias of the *i*th hidden node and $f(\mathbf{w}_i \cdot \mathbf{x}_j + b_i)$ is the output of the *i*th hidden node regarding the input sample $\mathbf{x}_i$. The above equation can be rewritten compactly as

$$\mathbf{H} \cdot \beta = \mathbf{Y}, \quad (2)$$

$$\mathbf{H} = \begin{bmatrix} f(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \dots & f(\mathbf{w}_L \cdot \mathbf{x}_1 + b_L) \\ \vdots & \dots & \vdots \\ f(\mathbf{w}_1 \cdot \mathbf{x}_n + b_1) & \dots & f(\mathbf{w}_L \cdot \mathbf{x}_n + b_L) \end{bmatrix}_{L \times L}, \quad (3)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times K}, \mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^T \\ \vdots \\ \mathbf{y}_L^T \end{bmatrix}_{n \times K} \quad (4)$$

where $\mathbf{H}$ is the output matrix of the hidden layer and β is the output weight matrix. The objective is to find specific $\hat{\mathbf{w}}_{\mathbf{i}}, \hat{b}_i, \hat{\beta}$ $(i = 1, ..., L)$ so that:

$$||\mathbf{H}(\hat{\mathbf{w}}_{\mathbf{i}}, \hat{b}_i)\hat{\beta} - \mathbf{Y}||^2 = \\ \min_{\mathbf{w}_{\mathbf{i}}, b_i, \beta} ||\mathbf{H}(\mathbf{w}_1, \dots, \mathbf{w}_L, b_1, \dots, b_L)\beta - \mathbf{Y}||^2. \quad (5)$$

As mentioned before, traditionally, the minimum of $||\mathbf{H}\beta - \mathbf{Y}||^2$ is calculated using gradient-based learning algorithms. The main issues related with these traditional methods are:

1) First and foremost, all gradient-based learning algorithms are very time-consuming in most applications. This became an important problem when classifying hyperspectral data.
2) The size of the learning rate parameter strongly affects the performance of the network. Too small values generate very slow convergence process while too large scores in η make the learning algorithm became unstable and to diverge.
3) The error surface generally presents local minima. Gradient-based learning algorithms can get stuck at a local minima. This can be an important issue if this local minima is far above a global minima.
4) FNs can be overtrained using BP-based algorithms, thus obtaining worse generalization performance. The effects of overtraining can be alleviated using regularization or early stopping criteria [121].

It has been proved in [66] that the input weights $\mathbf{w}_i$ and the hidden layer biases b_i do not need to be tuned so that the output matrix of the hidden layer $\mathbf{H}$ can remain unchanged after a random initialization. Fixing the input weights $\mathbf{w}_i$ and the hidden layer biases b_i means that training an SLFN is equivalent to find a least-squares solution $\hat{\beta}$ of the linear system $\mathbf{H}\beta = \mathbf{Y}$. Different from the traditional gradient-based learning algorithms, ELM aims to reach not only the smallest training error but also the smallest norm of output weights.

$$\text{Minimize:} \quad ||\mathbf{H}\beta - \mathbf{Y}||^2 \quad \text{and} \quad ||\beta||^2. \tag{6}$$

Let $\mathbf{h}(\mathbf{x}) = [f(\mathbf{w}_1 \cdot \mathbf{x} + b_1), ..., f(\mathbf{w}_L \cdot \mathbf{x} + b_L)]$, if we express equation (9) from the optimization theory point of view

$$\min_\beta \tfrac{1}{2}||\beta||_2^2 + C\tfrac{1}{2}\textstyle\sum_{i=1}^n \xi_i^2, \tag{7}$$

$$s.t. \quad \mathbf{h}(\mathbf{x}_i)\beta = \mathbf{y}_i^T - \xi_i^2, i = 1, ..., n, \tag{8}$$

where ξ_i^2 is the training error of training sample $\mathbf{x}_i$ and C is a regularization parameter. The output of ELM can be analytically expressed as

$$\mathbf{h}(\mathbf{x})\beta = \mathbf{h}(\mathbf{x})\mathbf{H}^T(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T)^{-1}\mathbf{Y}. \tag{9}$$

This expression can be generalized to kernel version of ELM using the kernel trick [71]. The inner product operation considered in $\mathbf{h}(\mathbf{x})\mathbf{H}^T$ and $\mathbf{H}\mathbf{H}^T$ can be replaced by a kernel function: $\mathbf{h}(\mathbf{x}_i) \cdot \mathbf{h}(\mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j)$. Both the regularized and kernel extensions of the traditional ELM algorithm require the setting of the needed parameters (C and all kernel-dependent parameters). When compared with traditional learning algorithms, ELM has the following advantages:

1) There is no need to iteratively tuning the input weights $\mathbf{w}_i$ and the hidden layer biases b_i using slow gradient-based learning algorithms.
2) Derived from the fact that ELM tries to reach both the smallest training error and the smallest norm of output weights, this algorithm exhibits better generalization performance in most cases when compared with traditional approaches.
3) The learning speed of ELM is much faster than in the traditional gradient-based learning algorithms. Depending on the application, ELM can be tens to hundreds of times faster [66].
4) The use of ELM avoids inherent problems to gradient-descent methods such as getting stucked in a local minima or overfitting the model [66].

III. Support Vector Machines

Support vector machines (SVMs) [115] have been often used for the classification of hyperspectral data due to their capability to handle high dimensional data with a limited number of training samples. The goal is to define an optimal linear separating hyperplane (the class boundary) within a multidimensional feature space that differentiates training samples of two classes. The best hyperplane is the one that leaves the maximum margin from both classes. The hyperplane is obtained using an optimization problem that is solved via structural risk minimization. In this way, in contrast with statistical approaches, SVMs minimize classification error on unseen data without any prior assumptions made on the probability distribution of the data [122].

The SVM tries to maximize the margins between the hyperplane and the closest training samples [75]. In other words, in order to train the classifier only samples that are close to the class boundary are needed to locate the hyperplane vector. This is why the closest training samples to the hyperplane are called *support vector*. More importantly, since only the closest training samples are influential on placing the hyperplane in the feature space, SVM can classify the input data efficiently even if only a limited number of training samples is available [2, 115, 123, 124]. In addition, SVMs can efficiently handle the classification of noisy patterns and multimodal feature spaces.

With regards to a binary classification problem in a d-dimensional feature space $\mathbb{R}^d$, $\mathbf{x}_i \in \mathbb{R}^d$, i =1, …, n is a set of n training samples with their corresponding class labels $y_i \in \{1, +1\}$. The optimal separating hyperplane $f(\mathbf{x})$ is determined by a normal vector $w \in \mathbb{R}^d$ and the bias b, where $|b|/||w||$ is the distance between the hyperplane and the origin, with $||w||$ as the Euclidean norm from w:

$$f(\mathbf{x}) = w\mathbf{x} + b. \tag{10}$$

The support vectors lie on two canonical hyperplanes $w\mathbf{x}$ + b = ± 1 that are parallel to the optimal separating hyperplane. The margin maximization leads to the following optimization problem:

$$\min \frac{w^2}{2} + C\sum_i^n \upsilon_i, \tag{11}$$

where the slack variables υ_i and the regularization parameter C are considered to deal with misclassified samples in a non separable cases, i.e., cases that are not linearly separable. The regularization parameter is a constant used as a penalty for samples that lie on the wrong side of the hyperplane. It is able to efficiently control the shape of the solution of the decision boundary. Thus, it affects the generalization capability of the SVM (e.g., a large value of C may cause the approach to overfit the training data) [97].

The SVM described above is a linear classifier, while decision boundaries are often nonlinear for classification problems. To tackle this issue, kernel methods are required to extend the linear SVM approach to nonlinear cases. In such cases, a nonlinear mapping is used to project the data into a high-dimensional feature space. After the transformation, the input pattern $\mathbf{x}$ can be described by $\Phi(\mathbf{x})$.

$$(\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)) = k(\mathbf{x}_i, \mathbf{x}_j). \tag{12}$$

The transformation into the higher-dimensional space can be computationally intensive. The computational cost can be decreased using a positive definite kernel k, which fulfills the so-called Mercer's conditions [77, 97]. When the Mercer's conditions are met, the final hyperplane can be defined by

$$f(\mathbf{x}) = (\sum_{i=1}^n \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}_j) + b), \tag{13}$$

where α_i denotes the Lagrange multipliers. For a detailed derivation of (13) we refer readers to [125]. In the new feature space, an explicit knowledge of Φ is not needed. The only required knowledge lies on the kernel function k. Therefore, one needs to estimate the parameters of the kernel function as well as the regularization parameter. To solve this issue, an automatic model selection based on a cross-validation have been introduced [126]. In [127], a genetic algorithm-based approach was used to regulate hyperplane parameters of an SVM while it finds efficient features to be fed to the classifier.

In terms of kernels, the Gaussian radial basis function (RBF) kernel may be the most widely used one in remote sensing [77, 97]. This kernel can handle more complex, nonlinear class distributions in comparison with a simple linear kernel, which is just a special case of the Gaussian RBF kernel [1, 128].

SVMs were originally developed for binary classification problems. In general, one needs to deal with multiple classes in remote sensing [1]. To address this, several *multiclass strategies* have been introduced in the literature. Among those approaches, two main strategies are best-known, which are based on the separation of the multiclass problem into several binary classification problems [129]. These are the one-against-one strategy and the one-against-rest strategy [97]. Some important points are listed bellow:

1) The capability of the SVM in handling a limited number of training samples, self-adaptability, a swift training stage and easiness of the use are considered as the main advantages of this classifier. In addition, SVMs are resilient to getting trapped in local minima since the convexity of the cost function enables the classifier to consistently identify the optimal solution [122]. More precisely, SVM deals with quadratic problems and as a result, it guarantees to the global minimum. Furthermore, the result of the SVM is stable for the same set of training samples and there is no need to repeat the classification step as this is a case for many approaches such as neural networks. Last but not least, SVMs are non-parametric, and do not assume a known statistical distribution of the data to be classified. This is considered as an important advantage due to the fact that the data acquired from remotely sensed imagery usually have unknown distributions [122].
2) One drawback of the SVM lies on the setting of the key parameters. For example, choosing a small value for the kernel width parameter may cause overfitting while a large value may cause oversmoothing, which is a common drawback of all kernel-based approaches. Moreover, the choice of the regularization parameter C, which controls the trade-off between maximizing the margin and minimizing the training error, is highly important.

For further reading, a detailed introduction of SVM is given by Burges [125], Cristianini and Shawe-Taylor [130], and Scholkopf and Smola [77].

IV. Multinomial Logistic Regression (MLR)

The MLR models the posterior densities $p(y_i|\mathbf{x}_i,\boldsymbol{\omega})$ as follows [32]

$$p(y_i = k|\mathbf{x}_i,\boldsymbol{\omega}) = \frac{\exp(\boldsymbol{\omega}^{(k)^T}\Phi(\mathbf{x}_i))}{\sum_{k=1}^{K}\exp(\boldsymbol{\omega}^{(k)T}\Phi(\mathbf{x}_i))}, \qquad (14)$$

where $\boldsymbol{\omega} = [\boldsymbol{\omega}^{(1)^T},...,\boldsymbol{\omega}^{(K-1)^T}]^T$ are the logistic regressors. Again, y_i is the class label of pixel $\mathbf{x}_i \in \mathbb{R}^d$ and d is the number of bands, K is the number of classes. Since the density in (14) does not depend on translations of the regressors $\boldsymbol{\omega}^{(k)}$, we take $\boldsymbol{\omega}^{(K)} = \mathbf{0}$. The term $\Phi(\mathbf{x}) = [\phi_1(\mathbf{x}),...,\phi_l(\mathbf{x})]^T$ is the fixed functions of the input, often termed *features*. The open structure of $\Phi(\mathbf{x})$ leads to the flexible selection of the input features, *i.e*, it can be linear, kernel and nonlinear functions. In order to control the algorithm complexity and its generalization capacity, the regressor $\boldsymbol{\omega}$ is modeled as a random vector with Laplacian density [131]:

$$p(\boldsymbol{\omega}) \propto \exp(-\lambda\|\boldsymbol{\omega}\|_1), \qquad (15)$$

where λ is the regularization parameter controlling the degree of sparsity of $\boldsymbol{\omega}$.

In the present problem, under a supervised scenario, learning the class density amounts to estimating the logistic regressors $\boldsymbol{\omega}$, which can be done by computing the maximum a posterior (MAP) estimate of $\boldsymbol{\omega}$:

$$\widehat{\boldsymbol{\omega}} \quad = \quad \arg\max_{\boldsymbol{\omega}} \quad \ell(\boldsymbol{\omega}) + \log p(\boldsymbol{\omega}), \qquad (16)$$

where $\ell(\boldsymbol{\omega})$ is the log-likelihood function over the labeled training samples. For supervised learning, it is given by

$$\ell(\boldsymbol{\omega}) \quad \equiv \quad \sum_{i=1}^{n}\log p(y_i = k|\mathbf{x}_i,\boldsymbol{\omega}), \qquad (17)$$

where n is the number of training samples. Problem (16), although convex, it is difficult to compute because the term of $\ell(\boldsymbol{\omega})$ is non-quadratic and the term $\log p(\boldsymbol{\omega})$ is non-smooth. Following [32], $\ell(\boldsymbol{\omega})$ can be estimated by a quadratic function. However, the problem is still difficult as $\log p(\boldsymbol{\omega})$ is non-smooth. This optimization problem (16) can be solved by the SMLR in [131] and by the fast SMLR (FSMLR) in [35]. However, most hyperspectral data sets are beyond the reach of these algorithms, as their processing becomes unbearable when the dimensionality of the input features increases. This is even more critical in the frameworks of composite kernel learning and multiple feature learning. In order to address this issue, the LORSAL algorithm is proposed in [36, 37] to deal with high-dimensional features and leads to good success in hyperspectral classification. For more information about the LORSAL algorithm, please see [33, 37].

The advantages of MLR are finally listed as follows:

1) MLR classifiers are able to learn directly the posterior class distributions and deal with the high dimensionality of hyperspectral data in a very effective way. The class posterior probability plays a crucial role in the complete posterior probability under the Bayesian framework to include the spectral and spatial information.
2) The sparsity inducing prior on the regressors leads to sparse estimates, which allows us to control the algorithm complexity and their generalization capacity.
3) The open structure of the MLR results in a good flexibility for the input functions, which can be linear, kernel-based and nonlinear.

V. Random Forests (RFs)

RFs were proposed in [95] as an ensemble method for classification and regression. Ensemble classifiers get their name from the fact that several classifiers, i.e., an ensemble of classifiers, are trained and their individual results are then combined through a voting process [132, 133]. In other words, the classification label is allocated to the input vector ($\mathbf{x}$) through $y_{rf}^{B} = majority\ vote\{y_b(\mathbf{x})\}_1^B$, where $y_b(\mathbf{x})$ is the class prediction of the *b*th tree and B shows the total number of trees. RFs can be considered as a particular case of decision trees. However, since RFs are composed of many classifiers, it infers special characteristics that make it completely different from a traditional classification trees and, therefore, it should be understood as a new concept of classifiers [134].

The training algorithm for RFs applies the general technique of bootstrap aggregating, or bagging, to tree learners [94]. Bootstrap aggregating is a technique used for training data creation by resampling the original data set in a random fashion with replacement (i.e., there is no deletion of the data selected from the input sample for generating the next subset) [134]. The bootstrapping procedure leads to more efficient model performance since it decreases the variance of the model without increasing the bias. In other words, while the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not that sensitive as far as the trees are not correlated [135]. By training many trees on a single training set, strongly correlated trees (or even the same tree many times, if the training algorithm is deterministic) are produced. Bootstrap sampling decorrelates the trees by showing them different training sets. RF uses trees as base classifiers, $\{h(\mathbf{x}, \theta_k), \quad k = 1, \ldots, \}$, where $\mathbf{x}$ and θ_k are the set of input vectors and the independent and identically distributed random vectors [95, 136]. Since some data may be used more than once for the training of the classifier while some others may not be used, greater classifier stability is achieved. This makes the classifier more robust when a slight variations in input data occurs and consequently, higher classification accuracy can be obtained [134, 136]. As mentioned in several studies such as [90, 91, 134, 137], methods based on bagging such as RFs, in contrast with other methods based on boosting, are not sensitive to noise or overtraining.

In RFs, there are only two parameters in order to generate the prediction model: the number of trees and the number of prediction variable. The number of trees is a free parameter, which can be chosen with respect to the size and nature of the training set. One possible way to choose the optimal number of trees is based on cross-validation or by observing the out-of-bag error [95, 133, 138]. For a detailed information regarding RFs and their different implementations please see [1, 132, 133]. The number of prediction variable is referred to the only adjustable parameter to which the forest is sensitive. As mentioned in [1], the "optimal" range of this parameter can be quite wide. However, the value is usually set approximately to the square root of the number of input features [132, 133, 139, 140].

Using RFs, the out-of-bag error, the variable importance, and proximity analysis, can be driven. In order to find detailed information about the RF and its derived parameters, please see [1, 88, 95, 132, 133, 133, 138]. Below, some important points of RFs are listed:

1) RFs are quite flexible and they can handle different scenarios such as large number of attributes, very limited number of training samples, and small or large data sets. In addition, they are easy and quick to evaluate.
2) RFs do not assume any underlying probability distribution for input data and can provide a good classification result in terms of accuracies, and can handle many variables and a lot of missing data. Another advantage of RF classifier is that it is insensitive to noise in the training labels. In addition, RF provides an unbiased estimate of the test set error as trees are added to the ensemble and finally it does not overfit.
3) The generated forest can be saved and used for other data sets.
4) In general, for sparse feature vectors, which is the case in most high dimensional data, a random selection of features may not be efficient all the time since uninformative or correlated features might be selected which downgrades the performance of the classifier.
5) Although RFs have widely been used for classification purposes, a gap still remains between the theoretical understanding of RFs and their corresponding practical use. A variety of RF algorithms have introduced showing promising practical success. However, these algorithms are difficult to analyze, and the basic mathematical properties of even the original variant are still not well understood [141].

VI. Deep Learning-based Approaches

There are some motivations to extract the invariant features from hyperspectral data. First, undesired scattering from neighboring objects may deform the characteristics of the object of interest. Furthermore, different atmospheric scattering conditions and intra-class variability make it extremely difficult to extract the features effectively. Moreover, hyperspectral data quickly increased in volume, velocity and variety, so it is difficult to analyze in the complicated real situation. On the other hand, it is believed that deep models can progressively lead to more invariant and abstract features at higher layers [102]. Therefore, deep models have the potential to be a promising tool. Deep learning involved a number of models including stacked auto-encoders (SAE) [142], deep belief networks (DBN) [143], and deep convolutional neural network (CNN) [144].

A. Stacked Auto-Encoder (SAE)

Auto-encoder (AE) is the basic part of SAE [142]. As shown in Fig. 3, an AE contains one visible layer of *d* inputs, one hidden layer of *L* units, and one reconstruction layer of *d* units. During training procedure, $\mathbf{x} \in \mathbb{R}^d$ is mapped to $\mathbf{z} \in \mathbb{R}^L$ in the hidden layer, and it is called "encoder". Then, $\mathbf{z}$ is mapped to $\mathbf{r} \in \mathbb{R}^d$ by a "decoder", which is called "reconstruction". These two steps can be formulated as:

$$\mathbf{z} = f(\mathbf{w}_z\mathbf{x} + b_z),$$
$$\mathbf{r} = f(\mathbf{w}_r\mathbf{x} + b_r),$$

where $\mathbf{w}_z$ and $\mathbf{w}_r$ denote the input-to-hidden and the hidden-to-output weights, respectively. b_z and b_r denote the bias

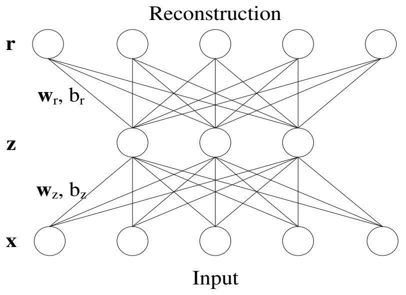


Fig. 3. Single hidden layer auto-encoder. The model learns a hidden feature "**z**" from input "**x**" by reconstructing it on "**r**".

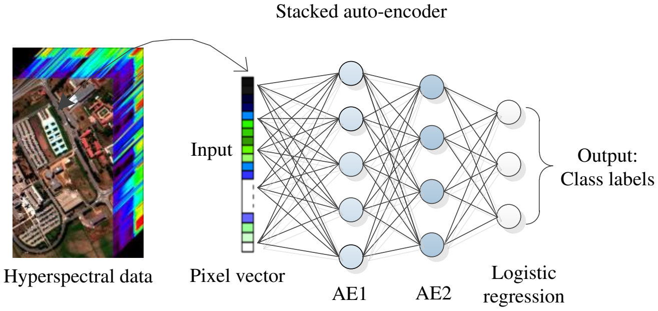


Fig. 4. A typical instance of a SAE connected with a subsequent logistic regression classifier.

of hidden and output units, and $f(.)$ denotes the activation function.

Stacking the input and hidden layers of auto-encoders together layer by layer constructs an SAE. Fig. 4 shows a typical instance of a SAE connected with a subsequent logistic regression classifier. The SAE can be used as a spectral classifier.

B. Deep Belief Networks (DBN)

Restricted Boltzmann machine (RBM) is a layer-wise training model in the construction of a DBN [143]. As shown in Fig. 5, it is a two-layer network with "visible" units $\mathbf{v} = \{0,1\}^d$ and "hidden" units $\mathbf{h} = \{0,1\}^L$. A joint configuration of the units has an energy given by:

$$E(\mathbf{v},\mathbf{h};\theta) = -\sum_{i=1}^{d} b_i v_i - \sum_{j=1}^{L} a_j h_j - \sum_{i=1}^{d}\sum_{j=1}^{L} w_{ij} v_i h_j \qquad (18)$$
$$= -\mathbf{b^T v} - \mathbf{a^T h} - \mathbf{v^T w h}$$

where $\theta = \{b_i, a_j, w_{ij}\}$, in which w_{ij} is the weight between visible unit i and hidden unit *j*; b_i and a_j are bias terms of visible and hidden unit, respectively. The learning of w_{ij} is done by a method called constructive divergence [143].

Due to the complexity of input hyperspectral data, RBM is not the best way to capture the features. After the training of RBM, the learnt features can be used as the input data for the following RBM. This kind of layer-by-layer learning system constructs a DBN. As shown in Fig. 6, a DBN is employed for feature learning and add a logistic regression layer above the DBN to constitute a DBN-logistic regression (DBN-LR) framework.

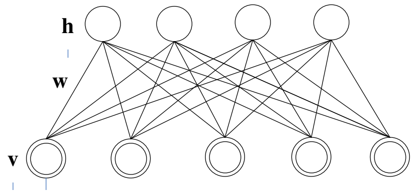


Fig. 5. Graphical illustration of a restricted Boltzmann machine. The top layer represents the hidden units and the bottom layer represents the visible units

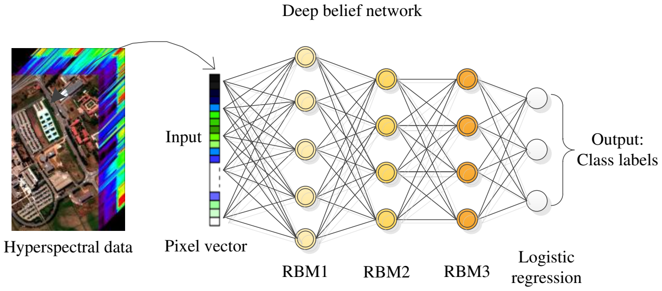


Fig. 6. A spectral classifier based on DBN. The classification scheme shown here has four layers: one input layer, 2 RBMs, and a logistic regression layer.

C. Deep Convolutional Neural Network (CNN)

CNN is a special type of deep learning model which is inspired by neuroscience. A complete CNN stage contains a convolution layer with nonlinear operation and a pooling layer. A convolutional layer is as follows [1]:

$$\mathbf{x}_j^l = f\left(\sum_{i=1}^{M} \mathbf{x}_i^{l-1} * \mathbf{k}_{ij}^l + b_j^l\right),$$

where $\mathbf{x}_i^{l-1}$ is the *i*-th feature map of *(l-1)*-th layer, $\mathbf{x}_j^l$ is the *j*-th feature map of current *(i)*-th layer, and *M* is the number of input feature maps. $\mathbf{k}_{ij}^l$ and b_j^l are the trainable parameters in the convolutional layer. $f(.)$ is a nonlinear function and $*$ is the convolution operation.

Pooling operation offers invariance by reducing the resolution of the feature maps. The neuron in the pooling layer combines a small $N \times 1$ patch of the convolution layer and the most common pooling operation is max pooling.

A convolution layer, nonlinear function and pooling layer are three fundamental parts of CNNs [146]. By stacking several convolution layers with nonlinear operation and several pooling layers, a deep CNN can be formulated. Deep CNN can hierarchically extract the features of inputs, which tend to be invariant and robust [102].

The architecture of a deep CNN for spectral classification is shown in Fig. 7. The input of the system is a pixel vector of hyperspectral data and the output is the label of the pixel to be classified. It consists of two convolutional and two pooling layers as well as a logistic regression layer. After convolution and pooling, the pixel vector can be converted into a feature vector, which captures the spectral information.

[1] It should be noted that we here explain 1D CNN as this paper deals with spectral classifiers. In order to find detailed information about 2D and 3D CNN for the classification of hyperspectral data, please see [145]

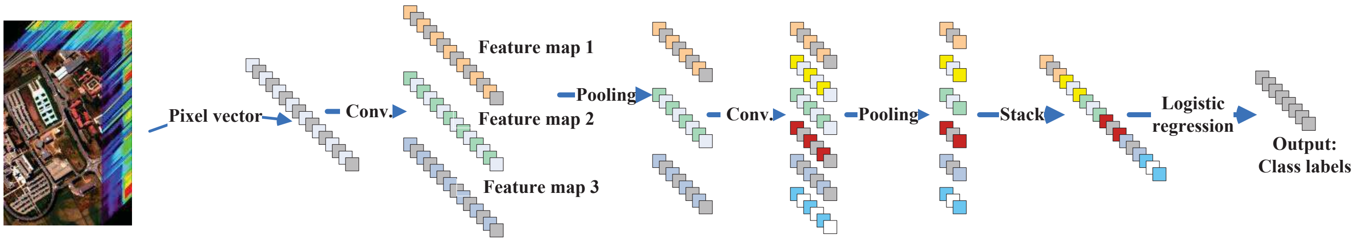


Fig. 7. Spectral classifier based on deep CNN.

D. Discussion about deep learning approaches

The following aspects are worth being mentioned about deep learning-based approaches:

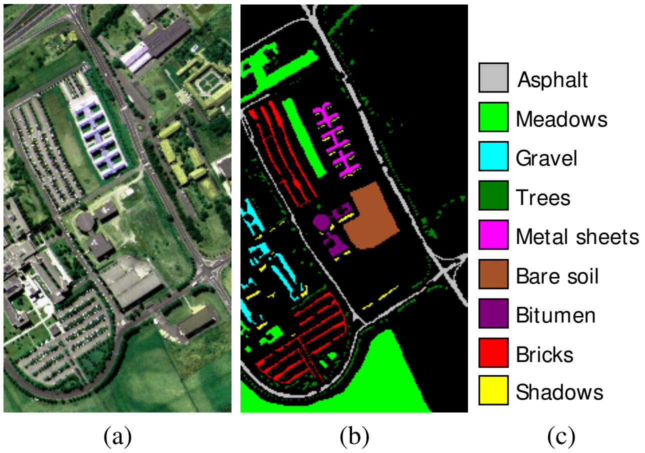
1) Recently, some deep models have been employed into hyperspectral data feature extraction and classification. Deep learning opens a new window for future research, showcasing the deep learning-based methods' huge potential [147].
2) The architecture design is the crucial part of a successful deep learning model. How to design a proper deep net is still an open area in machine learning community, while we may use grid search to find a proper deep model.
3) Deep learning methods may lead to a serious problem called overfitting, which means that the results can be very good on the training data but poor on the test data. To deal with the issue, it is necessary to use powerful regularization methods.
4) Deep leaning methods can be combined with other methods such as sparse coding and ensemble learning, which is another research area in hyperspectral data classification.

VII. EXPERIMENTAL RESULTS

This section describes our experimental results. First, we describe the different hyperspectral data sets used in experiments. Then, we describe the setup for the different algorithms to be compared. We next present the obtained results and provide a detailed discussion about the use of the different classifiers tested in different applications.[2]

A. Data Description

1) Pavia University: This hyperspectral data set has been repeatedly used. This data set was captured on the city of Pavia, Italy by the ROSIS-03 (Reflective Optics Spectrographic Imaging System) airborne instrument. The flight over the city of Pavia, Italy, was operated by the Deutschen Zentrum für Luft- und Raumfahrt (DLR, the German Aerospace Agency) within the context of the HySens project, managed and sponsored by the European Union. The ROSIS-03 sensor has 115 data channels with a spectral coverage ranging from 0.43 to 0.86 μm. Twelve channels have been removed due to noise. The remaining 103 spectral channels are processed. The data have been corrected atmospherically, but not geometrically. The spatial resolution is 1.3 m per pixel. The data set covers the Engineering School at the University of Pavia and consists of different classes including: trees, asphalt, bitumen, gravel, metal sheet, shadow, bricks, meadow and soil. This data set comprises 640 × 340 pixels. Fig. 8 presents a false color image of ROSIS-03 Pavia University data and its corresponding reference samples. These samples are usually obtained by manual labeling of a small number of pixels in an image or based on some field measurements. Thus, the collection of these samples is expensive and time demanding [2]. As a result, the number of available training samples is usually limited, which is a challenging issue in supervised classification.



(a) (b) (c)

Fig. 8. ROSIS-03 Pavia University hyperspectral data. (a) Three band false color composite, (b) Reference data and (c) Color code.

TABLE II
PAVIA UNIVERSITY: NUMBER OF TRAINING AND TEST SAMPLES.

Class		Number of Samples
No	Name	Total
1	Asphalt	6304
2	Meadow	18146
3	Gravel	1815
4	Tree	2912
5	Metal Sheet	1113
6	Bare Soil	4572
7	Bitumen	981
8	Brick	3364
9	Shadow	795
Total		40,002

[2] The sets of training and test samples used in this paper are available on request by sending an email to the authors.

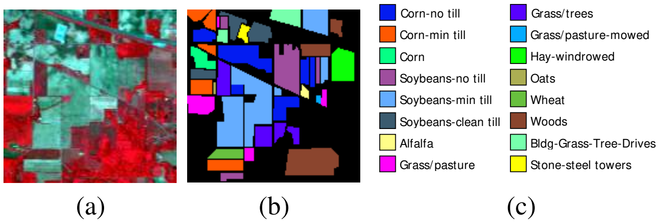


Fig. 9. ROSIS-03 Pavia University hyperspectral data. (a) Three band false color composite, (b) Reference data and (c) Color code.

TABLE III
INDIAN PINES: NUMBER OF TRAINING AND TEST SAMPLES.

Class		Number of Samples
No	Name	Total
1	Corn-notill	1434
2	Corn-mintill	834
3	Corn	238
4	Grass-pasture	497
5	Grass-trees	747
6	Hay-windrowed	489
7	Soybean-notill	968
8	Soybean-mintill	2468
9	Soybean-clean	614
10	Wheat	212
11	Woods	1294
12	Bldg-grass-tree-drives	380
13	Stone-Steel-Towers	95
14	Alfalfa	54
15	Grass-pasture-mowed	26
16	Oats	20
Total		10,366

2) Indian Pines: This data set was acquired by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor over the agricultural Indian Pines test site in northwestern Indiana. The spatial dimensions of this data set are 145 × 145 pixels. The spatial resolution of this data set is 20m per pixel. This data set originally includes 220 spectral channels but 20 water absorption bands (104-108, 150-163, 220) have been removed, and the rest (200 bands) were taken into account for the experiments. The reference data contains 16 classes of interest, which represent mostly different types of crops and are detailed in Table III. Fig. 9 shows a three-band false color image and its corresponding reference samples.

3) Houston Data: This data set was captured by the Compact Airborne Spectrographic Imager (CASI) over the University of Houston campus and the neighboring urban area in June, 2012. The size of the data is 349 × 1905 with the spatial resolution of 2.5m. This data set is composed of 144 spectral bands ranging 0.38-1.05m. This data consists of 15 classes including: Grass Healthy, Grass Stressed, Grass Synthetic, Tree, Soil, Water, Residential, Commercial, Road, Highway, Railway, Parking Lot 1, Parking Lot 2, Tennis Court and Running Track. The "Parking Lot 1" includes parking garages at the ground level and also in elevated areas, while "Parking Lot 2" corresponded to parked vehicles. Table IV demonstrates different classes with the corresponding number of training and test samples. Fig. 10 shows a three-band false color image and its corresponding already-separated training and test samples.

TABLE IV
HOUSTON: NUMBER OF TRAINING AND TEST SAMPLES.

Class		Number of Samples	
No	Name	Training	Test
1	Grass Healthy	198	1053
2	Grass Stressed	190	1064
3	Grass Synthetic	192	505
4	Tree	188	1056
5	Soil	186	1056
6	Water	182	143
7	Residential	196	1072
8	Commercial	191	1053
9	Road	193	1059
10	Highway	191	1036
11	Railway	181	1054
12	Parking Lot 1	192	1041
13	Parking Lot 2	184	285
14	Tennis Court	181	247
15	Running Track	187	473
Total		2,832	12,197

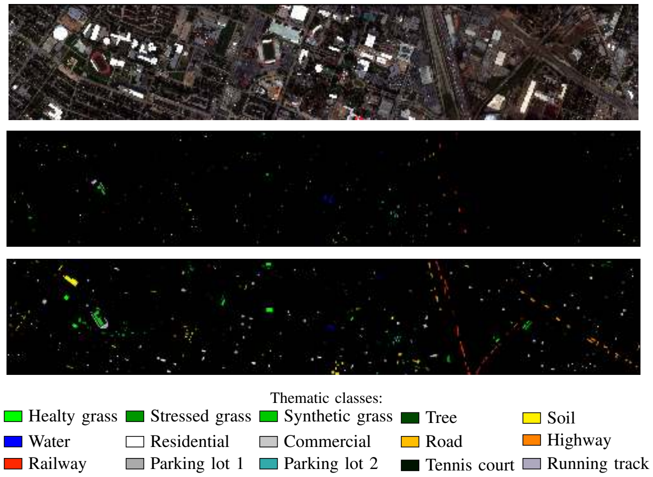


Fig. 10. Houston - From top to bottom: A color composite representation of the hyperspectral data using bands 70, 50, and 20, as R, G, and B, respectively; Training samples; Test samples; and legend of different classes.

B. Algorithm Setup

In this paper two different scenarios are defined in order to evaluate different approaches. For the first scenario, training samples have been chosen with different percentages from the available reference data. For this scenario, only Indian Pines and Pavia University are taken into consideration. In this paper, 1, 5, 10, 15, 20, and 25 percents of the whole samples have been randomly selected as training, except for classes *alfalfa*, *grass-pasture-mowed* and *oats*. These classes contain only a small number of samples in the reference data. Therefore, only 15 samples for each of these classes were chosen at random as training samples and the rest as the test samples. For Pavia University, 1, 5, 10, 15, and 20 percents of the whole samples have been randomly selected as training and the rest as test samples. The experiments have been repeated 10 times, and the mean and the standard deviation of the obtained overall accuracy (OA) have been reported in the paper.

For the second scenario, the Houston data is taken into account. The training and test samples of this data have been separated (Table IV). Results have been evaluated using OA, AA, K, and class specific accuracies.

The following classifiers have been investigated and compared in two different scenarios, discussed above:

- SVM (Support Vector Machine),
- RF (Random Forest),
- BP (Back Propagation Neural Network, also known as Multilayer Perceptron),
- ELM (Extreme Learning Machine),
- KELM (Kernel Extreme Learning Machine),
- 1D CNN (1-dimensional Convolutional Neural Network),
- MLR (Multinomial Logistic Regression).

For the MLR classifier, which is executed by LORSAL algorithm [36, 37], we use a Gaussian Radial Basis Function (RBF) kernel given by $K(\mathbf{x}, \mathbf{z}) = \exp(-\|\mathbf{x} - \mathbf{z}\|^2/2\sigma^2)$, which is widely used in hyperspectral image classification problems [148]. For the parameters involved in the algorithm, we use the default settings provided in the online demo[3], where it illustrates that the MLR classifier is insensitive to the parameter settings, which also can be observed in the following experiments.

In terms of the SVM, the RBF kernel is taken into account. The optimal hyperplane parameters C (parameter that controls the amount of penalty during the SVM optimization) and γ (spread of the RBF kernel) have been traced in the range of $C = 10^{-2}, 10^{-1}, ..., 10^{4}$ and $\gamma = 10^{-3}, 10^{-2}, ..., 10^{4}$ using five-fold cross validation.

In terms of the RF, the number of trees is set to 300. The number of the prediction variable is set approximately to the square root of the number of input bands. The same parameters have been used for all experiments stating that the RF is insensitive to the parameter initialization.

Regarding the BP-based neural network classifier (also known as Multilayer Perceptron, MLP), the network has only one hidden layer and the number of hidden nodes has been empirically set within the range $\frac{(n+K)\times 2}{3} \pm 10$. The number of input nodes equals the number of spectral bands of the image while the number of output nodes equals the number of spectral clasess. Hidden nodes have *sigmoid* activation functions while output nodes implement *softmax* activation function. The implemented learning algorithm is scaled conjugate gradient backpropagation [64]. During the experiments, we empirically adjust the early stopping parameters to achieve reasonable performance goals.

In the case of ELM, the network has also one single hidden layer. The number of nodes L and the regularization parameter C [149] have been traced in the ranges of $L = 400, 600, 800, ..., 2000$ and $C = 10^{-3}, 10^{-2}, ..., 10^{4}$ using five-fold cross validation.

For the KELM, the RBF kernel is considered. Again, the regularization parameter C and the kernel parameter γ have been searched in the ranges $C = 10^{-3}, 10^{-1}, ..., 10^{4}$ and $\gamma = 2^{-3}, 2^{-2}, ..., 2^{4}$ also using five-cross validation.

Layer Name		*I1*	*C2* *S3*	*C4* *S5*	*C6* *S7*	*C8* *S9*	*C10* *S11*	*F12*	*O13*
Kernel Size	Indian Pines	1×200	1×5 1×2	1×5 1×2	1×4 1×2	1×5 1×2	1×4 1×1	Fully connected	1×16
	Pavia University	1×103	1×8 1×2	1×7 1×2	1×8 1×2	-	-	Fully connected	1×9
	Houston	1×144	1× 5 1×2	1×5 1×2	1×6 1×2	1×5 1×2	-	Fully connected	1×15
Number of feature map/ number of neurons			6	12	24	48	96	256	

Fig. 11. The Architectures of 1D CNN on Three Data Sets.

For 1D CNN, the important parameters are the kernel size, number of layers, number of feature maps, number of neurons in hidden layer, and learning rate. Figure 11 shows the architectures of the deep 1D CNN used for the experimental part. As an example, for the Indian Pines data set there are 13 layers, denoted as I1, C2, S3, C4, S5, C6, S7, C8, S9, C10, S11, F12, and O13 in sequence. I1 is the input layer. C refers to the convolution layers and S refers to pooling layers. F12 a fully-connected layer, and O13 is the output layer of the whole neural network. The input data are normalized into [-1 1]. The learning rate is set to 0.005, and the training epoch is 700 for Indian Pines data set. For Pavia University data set, we set the learning to 0.01, and the number of epochs to 300. For the Houston data set, the learning is 0.01 with 500 epochs.

Fig. 12 shows the overall accuracy of different approaches (i.e., the average value over 10 runs) on different percentages of training samples on Indian Pines and Pavia University. In order to evaluate the stability of different classifiers on the change of training samples, the standard deviation value over 10 runs for each percentage is estimated and shown in Fig. 13.

For the Houston hyperspectral data, since the training and test sets have been already separated, we performed the classifiers on the standard set of training/test samples. The classification accuracies (i.e., overall accuracy (OA), average accuracy (AA), kappa coefficient (Kappa), and class specific accuracies) are reported in Table V. The classification maps of this data set are shown in Fig. 14.

C. Results and Discussion

The main observations obtained from our experimental results are listed systematically as follows:

- SVM *vs.* RF: Although both classifiers have the same number of hyperparameters to tune (i.e., RBF SVM has γ and C, and RFs have the number of trees and the depth of the tree), RFs' parameters are easier to set. In practice, the more trees we have the higher classification accuracy of RFs can be obtained. RFs are trained faster than kernel SVM. A suggested number of trees can be varied from 100 to 500 for the classification of hyperspectral data. However, with respect to our experiments, the SVM established higher classification accuracies than RFs.
- SVM *vs.* BP: the SVM classifier presents the series of advantages over the BP classifier. The SVM exhibits less

[3] http://www.lx.it.pt/~jun/demo_LORSAL_AL.rar.

computational complexity even when the kernel trick is used and, usually, provides better results when a small number of training is available. However, if BP configuration is properly tuned, both classifiers can provide comparable classification accuracies. Last but not least, the BP is much more complex from a computational point of view. Actually, in this work we use the *scaled conjugate gradient* backpropagation algorithm which presents a practical complexity of $O((n((dLK)+L+K))^2)$ (the square of the number of weights of the network), where n the number of training patterns, d the number of spectral bands, L the number of hidden nodes and K the number of classes) [64].

- SVM *vs.* ELM: From an optimization point of view, ELM presents the same optimization cost function as least square SVM (LS-SVM) [150] but much less computational complexity. In general terms, ELM training is tens or hundreds of times faster than traditional SVM. Regarding the classification accuracy, it can be seen that ELM achieves comparable results.
- SVM *vs.* KELM: still in the case of kernel version of ELM, the computational complexity of SVM is much bigger than KELM. It can be seen that KELM slightly outperforms SVM in terms of classification accuracy. Experimental validation shows that the kernel used in KELM and SVM is more efficient than the activation function used in ELM.
- BP *vs.* ELM *vs.* KELM: at the light of the results, it can be seen how the three versions of the single layer feedforward neural network provides competitive results in terms of accuracy. However, it should be noticed that both ELM and KELM are in the order of hundreds or even thousands of times faster than BP. Actually, ELM and KELM have a practical complexity of $O(L^3+L^2n+(K+d)Ln)$ and $O(2n^3+(K+d)n^2)$ respectively [151].
- SVM *vs.* 1D CNN: The main advantage of 2D and 3D CNNs is that they use local connections to handle spatial dependencies. In this work, however, 1D CNN is taken into account to have a fair comparison with other spectral approaches. In general, SVM can obtain higher classification accuracies in a faster way than 1D CNN, so the use of SVMs over 1D CNN is recommended. In terms of CPU processing time, deep learning methods are time-consuming in the training step. Compared to SVM, the training time of 1D deep CNN is about 2 or 3 times longer than RBF-SVM. On the other hand, the advantage of deep CNN is that it is extremely fast on the testing stage.
- Last but not least, some advantages of MLR (executed via LORSAL) in comparison with other methods are listed as follows.
  - It converges very fast and is relatively insensitive to parameter settings. In our experiments, we use the same settings for all data sets and received very competitive results in comparison with those obtained from other methods.
  - It has very low computational cost, with a practical complexity of $O(d^2(K-1))$.

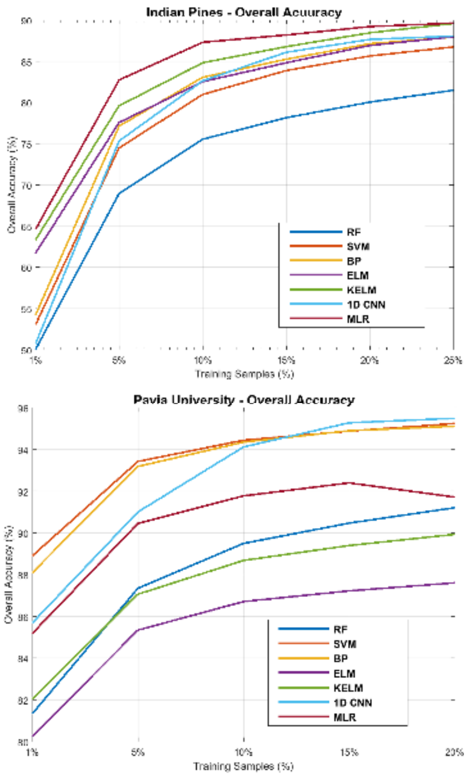


Fig. 12. Scenario 1 - Overall Accuracy: The overall accuracy of different approaches (i.e., the average value over 10 runs) on different percentages of training samples on Indian Pines and Pavia University obtained by different classification approaches.

For illustrative purposes, Fig. 12 provides a comparison of the different classifiers tested in this work with the Indian Pines and Pavia University scenes (in terms of overall accuracy). As shown by Fig. 12, different classifiers provide different performances for the two considered images, indicating that there is no classifier consistently providing the best classification results for different scenes. The stability of the different classifiers with the two considered scenes is illustrated in Fig. 13, which demonstrate how much a classifier is stable with respect to some changes on the available training sets. Furthermore, Table V gives detailed information about classification accuracies obtained by different approaches in a different application domain, represented by the Houston data set. In this case, the optimized classifiers also perform similarly in terms of classification accuracy, so ultimately the choice of a given classifier is more driven by the simplicity of tuning the parameters and configurations rather than the obtained classification results. This is an important observation, as it is felt that the hyperspectral community has reached a point in which many classifiers are able to provide very high classification accuracies. However, the competitive differences between existing classifiers is more related to their simplicity and tuning configurations. In this regard, our assessment of the characteristics of different algorithms and their tuning is believed to provide helpful insights regarding the choice of a given classifier in a certain application domain

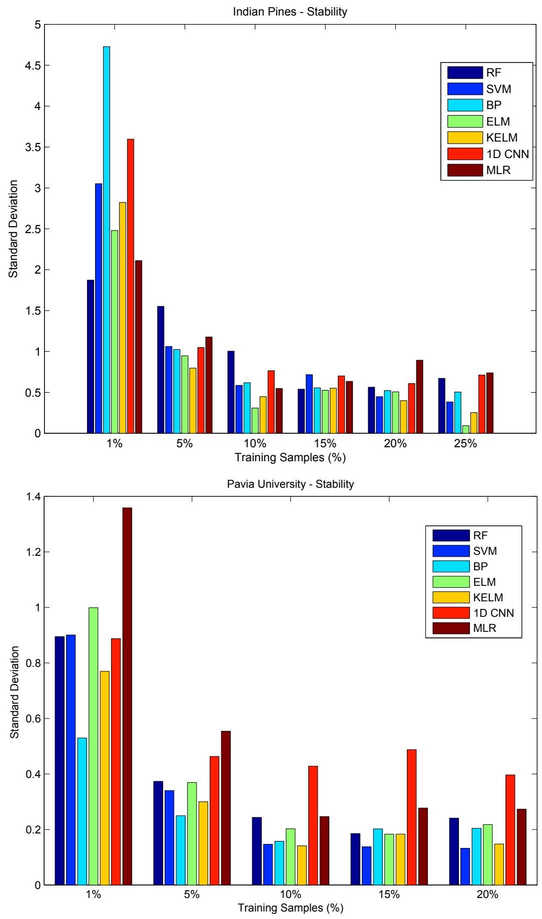


Fig. 13. Scenario 1 - Stability: The standard deviation value over 10 runs on different percentages of training samples on Indian Pines and Pavia University obtained by different classification approaches.

TABLE V
SCENARIO2: CLASSIFICATION ACCURACIES OBTAINED BY DIFFERENT CLASSIFICATION APPROACHES ON THE HOUSTON HYPERSPECTRAL DATA.

Class	**SVM**	**RF**	**BP**	**ELM**	**KELM**	**1D CNN**	**MLR**
1	82.24	82.62	81.86	97.25	95.37	82.91	82.62
2	82.99	83.46	85.63	98.39	98.75	83.65	83.55
3	99.80	97.62	99.90	100.00	100.00	99.8	99.80
4	92.33	92.14	90.11	96.09	99.49	90.06	92.23
5	98.30	96.78	98.08	96.80	97.84	97.82	98.39
6	99.30	99.30	86.43	99.03	100.00	99.3	95.10
7	79.10	74.72	79.64	53.26	73.63	85.63	78.73
8	50.62	32.95	51.80	66.04	76.18	41.41	53.46
9	79.13	68.65	77.26	76.81	73.88	79.41	79.79
10	57.92	43.15	57.46	71.39	76.08	53.38	58.10
11	81.31	70.49	85.76	82.25	67.28	70.49	82.44
12	76.08	55.04	81.76	72.21	59.74	72.72	76.36
13	69.82	60.00	74.42	42.65	41.74	63.86	68.42
14	100.00	99.19	99.31	89.81	90.41	99.6	98.78
15	96.83	97.46	98.08	94.15	94.34	98.52	97.88
OA	80.18	72.99	80.98	79.55	80.64	78.21	80.60
AA	83.05	76.9	83.17	82.4	82.98	81.23	83.04
Kappa	0.7866	0.7097	0.7934	0.7783	0.7901	0.7846	0.7908

TABLE VI
PERFORMANCE EVALUATION OF DIFFERENT SPECTRAL CLASSIFIERS IN TERMS OF CLASSIFICATION ACCURACIES, SIMPLICITY AND SPEED, BEING CLOSER TO AUTOMATIC, AND STABILITY. ONE BULLET INFERS THE WORST PERFORMANCE WHILE FOUR BULLETS INFER THE BEST PERFORMANCE.

Techniques	Accuracy	Automation	Simplicity and Speed	Stability
RF	●	●●●●	●●●●	●●
SVM	●●●●	●●●	●●●	●●●
BP	●●●●	●●	●●	●●
ELM	●●	●●	●●●	●●●
KELM	●●●●	●●	●●●	●●●
1D CNN	●●	●	●	●●
MLR	●●●●	●●●●	●●●●	●●

With the aforementioned observations in mind, we can interpret the results provided in Table VI in more details. In this table, One bullet refers to the worst performance while four bullets refer to the best performance. It can be observed that the KELM can provide high classification accuracies in a short period of time, while the obtained results are also stable with respect to some changes of the input training samples. SVM and MLR also show a fair balance between the accuracy, automation (i.e., can be obtained with respect to the number of parameters needs to be adjusted), speed (i.e., it was evaluated based on the demanded CPU processing time of different classifiers), and stability, which can be advantageous for applications where a trade-off between these elements are needed. In contrast, 1D CNN does not show enough advantages neither in terms of classification accuracy and stability nor speed and automation.

VIII. CONCLUSIONS

In this paper, we have provided a review and critical comparison of different supervised hyperspectral classification approaches from different points of view, with particular emphasis on the configuration, speed and automation capacity of algorithms. The compared techniques include popular approaches such as support vector machines, random forests, neural networks, deep approaches, logistic regression-based techniques and sparse representation-based classifiers, which have been widely used in the hyperspectral analysis community but never investigated systematically using a quantitative and comparative approach. The critical comparison conducted in this work leads to interesting hints about the logical choice of an appropriate classifier based on the application at hand. The main conclusion that can be obtained from the present study is that there is no classifier that consistently provides the best performance among the considered metrics (particularly, from the viewpoint of classification accuracy), but rather different solutions that depend on the complexity of the analysis scenario (i.e., availability of training samples, processing requirements, tuning parameters, speed of the algorithm, etc.) and on the considered application domain. Combined, the insights provided in this paper may facilitate the selection of a specific classifier by an end-user depending on his/her expentations and/or exploitation goals.

IX. ACKNOWLEDGMENT

The authors would like to thank the National Center for Airborne Laser Mapping (NCALM) for providing the Houston data set. The ROSIS Pavia University and Indian Pines data and the corresponding reference information were kindly

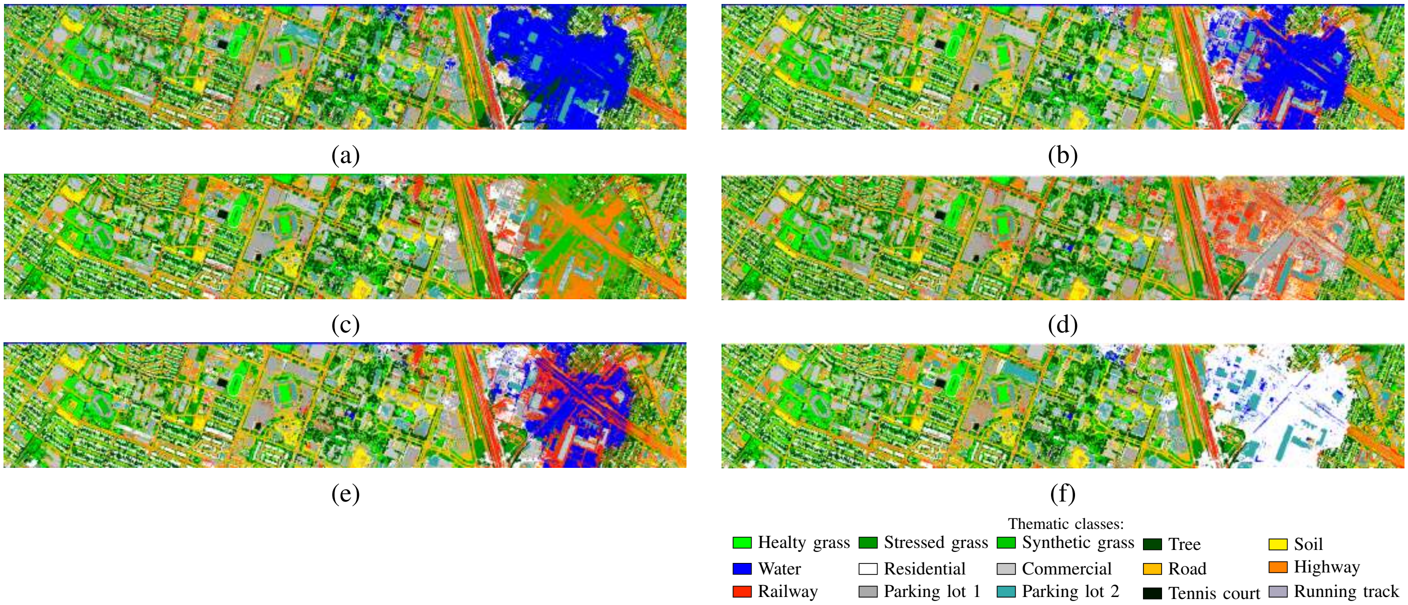


Fig. 14. Scenario 2 - Classification maps for Houston data: The classification map of (a) RF, (b) SVM, (C) BP, (d) KELM, (e) MLR, (f) 1D CNN

provided by Prof. P. Gamba from the University of Pavia, Italy and Prof. D. Landgrebe from Purdue University, respectively.

References

[1] J. A. Benediktsson and P. Ghamisi, *Spectral-Spatial Classification of Hyperspectral Remote Sensing Images*. Artech House Publishers, INC, Boston, USA, 2015.

[2] P. Ghamisi and J. A. Benediktsson, "Feature selection based on hybridization of genetic algorithm and particle swarm optimization," *IEEE Geos. Remote Sens. Lett.*, vol. 12, no. 2, pp. 309–313, 2015.

[3] P. Ghamisi, A.-R. Ali, M. Couceiro, and J. Benediktsson, "A novel evolutionary swarm fuzzy clustering approach for hyperspectral imagery," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. 8, no. 6, pp. 2447–2456, 2015.

[4] A. K. Jain, R. P. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pat. Analysis Mach. Intel.*, vol. 22, no. 1, pp. 4–37, 2000.

[5] B. Waske and J. A. Benediktsson, *Pattern Recognition and Classification, Encyclopedia of Remote Sensing*, E. G. Njoku, Ed. Springer Verlag, Berlin, 2014.

[6] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proc. 5th Berkeley Symp. Math. Stat. Prob.*, pp. 281–297, 1967.

[7] G. Ball and D. Hall, "ISODATA, a novel method of data analysis and classification," *Tech. Rep. AD-699616*, Stanford Univ., Stanford, CA, 1965.

[8] J. C. Bezdek and R. Ehrlich, "FCM: The fuzzy c-means clustering algorithm," *Comp. Geos.*, vol. 10, no. 22, pp. 191–203, 1981.

[9] W. Wang, Y. Zhang, Y. Li, and X. Zhang, "The global fuzzy c-means clustering algorithm," *Intel. Cont. Aut.*, vol. 1, pp. 3604–3607, 2006.

[10] B. M. Shahshahani and D. A. Landgrebe, "The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon," *IEEE Trans. Geos. Remote Sens.*, vol. 32, no. 5, pp. 4–37, 1995.

[11] Q. Jackson and D. Landgrebe, "Adaptive Bayesian contextual classification based on Markov random fields," *IEEE Trans. Geos. Remote Sens.*, vol. 40, no. 11, pp. 2454–2463, 2002.

[12] X. Jia and J. A. Richards, "Cluster-space representation for hyperspectral data classification," *IEEE Trans. Geos. Remote Sens.*, vol. 40, no. 3, pp. 593–598, 2002.

[13] K. Fukunaga, *Introduction to Statistical Pattern Recognition.*, 2nd ed. Academic Press, Inc., San Diego, CA, 1990.

[14] D. W. Scott, *Multivariate Density Estimation*, John Wiley & Sons, New York, NY, 1992.

[15] E. J. Wegman, "Hyperdimensional data analysis using parallel coordinates," *Jour. American Stati. Assoc.*, vol. 85, no. 411, pp. 664–675, 1990.

[16] L. Jimenez and D. Landgrebe, "Supervised classification in high-dimensional space: geometrical, statistical, and asymptotical properties of multivariate data," *IEEE Trans. Sys., Man, Cyber., Part C: Applications and Reviews*, vol. 28, no. 1, pp. 39–54, 1998.

[17] D. A. Landgrebe, *Signal Theory Methods in Multispectral Remote Sensing*. John Wiley & Sons, Hoboken, NJ, 2003.

[18] Y. Qian, F. Yao, and S. Jia, "Band selection for hyperspectral imagery using affinity propagation," *IET Comput. Vis.*, vol. 3, no. 4, p. 213, 2009. [Online]. Available: http://dx.doi.org/10.1049/iet-cvi.2009.0034

[19] F. Canters, "Evaluating the uncertainty of area estimates derived from fuzzy landcover classification," *Photo. Eng. Remote Sens.*, vol. 63, pp. 403–414, 1997.

[20] J. L. Dungan, *Toward a comprehensive view of uncertainty in remote sensing analysis, in, Uncertainty in Remote Sensing and GIS*, 2nd ed. G. M. Foody and P. M. Atkinson (Eds), Chichester: John Wiley Sons, 2002.

[21] M. A. Friedl, K. C. McGwire, and D. K. Mciver, *An overview of uncertainty in optical remotely sensed data for ecological applications*. C. T. Hunsaker, M. F. Goodchild, M.A. Friedl and T.J. Case (Eds), New York: Springer-Verlag, 2001.

[22] X. Wang, "Learning from big data with uncertainty editorial," *Journal of Intelligent & Fuzzy Systems*, vol. 28, no. 5, pp. 2329–2330, 2015.

[23] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *Int. Jour. Remote Sens.*, vol. 28, no. 5, pp. 823–870, 2007.

[24] C. E. Woodcock and A. H. Strahler, "The factor of scale in remote sensing," *Remote Sens. Env.*, vol. 21, no. 3, pp. 311–332, 1987.

[25] P. Ghamisi, M. Dalla Mura, and J. A. Benediktsson, "A survey on spectral–spatial classification techniques based on attribute profiles," *IEEE Trans. Geos. Remote Sens.*, vol. 53, no. 5, pp. 2335–2353, 2015.

[26] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 652–675, 2013.

[27] C. Xu, H. Liu, W. Cao, and J. Feng, "Multispectral image edge detection via clifford gradient," *Science China Information Sciences*, vol. 55, no. 2, pp. 260–269, jan 2012. [Online]. Available: http://dx.doi.org/10.1007/s11432-011-4540-0

[28] Z. Su, X. Luo, Z. Deng, Y. Liang, and Z. Ji, "Edge-preserving texture suppression filter based on joint filtering schemes," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 535–548, apr 2013. [Online]. Available: http://dx.doi.org/10.1109/TMM.2012.2237025

[29] Z. Zhu, S. Jia, S. He, Y. Sun, Z. Ji, and L. Shen, "Three-dimensional Gabor feature extraction for hyperspectral imagery classification using a memetic framework," *Information Sciences*, vol. 298, pp. 274–287, 2015.

[30] J. L. Cushnie, "The interactive effect of spatial resolution and degree of internal variability within land-cover types on classification accuracies," *Int. Jour. Remote Sens.*, vol. 8, pp. 15–29, 1987.

[31] Y. Zhong, Q. Zhu, and L. Zhang, "Scene classification based on the multifeature fusion probabilistic topic model for high spatial resolution remote sensing imagery," *IEEE Trans. Geosci. Remote Sensing*, vol. 53, no. 11, pp. 6207–6222, nov 2015. [Online]. Available: http://dx.doi.org/10.1109/TGRS.2015.2435801

[32] D. Böhning, "Multinomial logistic regression algorithm," *Ann. Inst. Statist. Math.*, vol. 44, pp. 197–200, 1992.

[33] J. Li, J. Bioucas-Dias, and A. Plaza, "Semi-supervised hyperspectral image segmentation using multinomial logistic regression with active learning," *IEEE Trans. Geosci. and Remote Sens.*, vol. 48, pp. 4085–4098, 2010.

[34] P. Zhong, P. Zhang, and R. Wang, "Dynamic learning of smlr for feature selection and classification of hyperspectral data," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 2, pp. 280–284, April 2008.

[35] J. S. Borges, J. M. Bioucas-Dias, and A. R. S. Marcal, "Bayesian hyperspectral image segmentation with discriminative class learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 6, pp. 2151–2164, June 2011.

[36] J. Bioucas-Dias and M. Figueiredo, "Logistic regression via variable splitting and augmented Lagrangian tools," Instituto Superior Técnico, TULisbon, Tech. Rep., 2009.

[37] J. Li, J. Bioucas-Dias, and A. Plaza, "Hyperspectral image segmentation using a new Bayesian approach with active learning," *IEEE Trans. Geosci. and Remote Sens.*, vol. 49, no. 19, pp. 3947–3960, 2011.

[38] ——, "Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and markov random fields," *IEEE Trans. Geosci. and Remote Sens.*, vol. 50, no. 3, pp. 809–823, 2012.

[39] P. Zhong and R. Wang, "Learning conditional random fields for classification of hyperspectral images," *IEEE Transactions on Image Processing*, vol. 19, no. 7, pp. 1890–1907, July 2010.

[40] Y. Qian, M. Ye, and J. Zhou, "Hyperspectral image classification based on structured sparse logistic regression and three-dimensional wavelet texture features," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 4, pp. 2276–2291, April 2013.

[41] P. Zhong and R. Wang, "Jointly learning the hybrid crf and mlr model for simultaneous denoising and classification of hyperspectral imagery," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 7, pp. 1319–1334, July 2014.

[42] M. Khodadadzadeh, J. Li, A. Plaza, and J. M. Bioucas-Dias, "A subspace-based multinomial logistic regression for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 12, pp. 2105–2109, Dec 2014.

[43] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral-spatial classification of hyperspectral data using loopy belief propagation and active learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 2, pp. 844–856, Feb 2013.

[44] M. Khodadadzadeh, J. Li, A. Plaza, H. Ghassemian, J. M. Bioucas-Dias, and X. Li, "Spectral-spatial classification of hyperspectral data using local and global probabilities for mixed pixel characterization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 10, pp. 6298–6314, Oct 2014.

[45] L. Sun, Z. Wu, J. Liu, L. Xiao, and Z. Wei, "Supervised spectral-spatial hyperspectral image classification with weighted markov random fields," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 3, pp. 1490–1503, March 2015.

[46] S. Sun, P. Zhong, H. Xiao, and R. Wang, "An mrf model-based active learning framework for the spectral-

spatial classification of hyperspectral imagery," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 6, pp. 1074–1088, Sept 2015.

[47] J. Li, M. Khodadadzadeh, A. Plaza, X. Jia, and J. M. Bioucas-Dias, "A discontinuity preserving relaxation scheme for spectral-spatial hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 2, pp. 625–639, Feb 2016.

[48] J. Zhao, Y. Zhong, H. Shu, and L. Zhang, "High-resolution image classification integrating spectral-spatial-location cues by conditional random fields," *IEEE Transactions on Image Processing*, vol. 25, no. 9, pp. 4033–4045, sep 2016. [Online]. Available: http://dx.doi.org/10.1109/TIP.2016.2577886

[49] J. Li, P. Marpu, A. Plaza, J. Bioucas-Dias, and J. A. Benediktsson, "Generalized composite kernel framework for hyperspectral image classification," *IEEE Trans. Geosci. and Remote Sens.*, 2013, in press.

[50] Y. Zhang and S. Prasad, "Locality preserving composite kernel feature extraction for multi-source geospatial image analysis," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 3, pp. 1385–1392, March 2015.

[51] J. Li, X. Huang, P. Gamba, J. M. Bioucas-Dias, L. Zhang, J. A. Benediktsson, and A. Plaza, "Multiple feature learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 3, pp. 1592–1606, March 2015.

[52] C. Zhao, X. Gao, Y. Wang, and J. Li, "Efficient multiple-feature learning-based hyperspectral image classification with limited training samples," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 7, pp. 4052–4062, July 2016.

[53] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.

[54] J. A. Benediktsson, P. H. Swain, and O. K. Ersoy, "Conjugate gradient neural networks in classification of very high dimensional remote sensing data," *Int. Jour. Remote Sens.*, vol. 14, no. 15, pp. 2883–2903, 1993.

[55] H. Yang, F. V. D. Meer, W. Bakker, and Z. J. Tan, "A back—propagation neural network for mineralogical mapping from AVIRIS data," *Int. Jour. Remote Sens.*, vol. 20, no. 1, pp. 97–110, 1999.

[56] J. A. Benediktsson, "Statistical methods and neural network approaches for classification of data from multiple sources," Ph.D. dissertation, PhD thesis, Purdue Univ., School of Elect. Eng. West Lafayette, IN, 1990.

[57] J. A. Richards, "Analysis of remotely sensed data: The formative decades and the future," *IEEE Trans. Geos. Remote Sens.*, vol. 43, no. 3, pp. 422–432, 2005.

[58] J. A. Benediktsson, P. H. Swain, and O. K. Ersoy, "Neural network approaches versus statistical methods in classification of multisource remote sensing data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 28, no. 4, pp. 540–552, 1990.

[59] E. Merényi, W. H. Farrand, J. V. Taranik, and T. B. Minor, "Classification of hyperspectral imagery with neural networks: comparison to conventional tools," *Eurasip Journal on Advances in Signal Processing*, vol. 2014, no. 1, pp. 1–19, 2014.

[60] F. D. Frate, F. Pacifici, G. Schiavon, and C. Solimini, "Use of neural networks for automatic classification from high-resolution images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 4, pp. 800–809, 2007.

[61] F. Ratle, G. Camps-Valls, and J. Wetson, "Semisupervised neural networks for efficient hyperspectral image classification," *IEEE Transactions on Geosciences and Remote Sens.*, vol. 48, no. 5, pp. 2271–2282, 2010.

[62] Y. Zhong and L. Zhang, "An adaptive artificial immune network for supervised classification of multi-/hyperspectral remote sensing imagery," *IEEE Transactions on Geosciences and Remote Sens.*, vol. 50, no. 3, pp. 894–909, 2012.

[63] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[64] M. Moller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, vol. 6, no. 4, pp. 525–533, 1993.

[65] M. T. Hagan and M. Menhaj, "Training feed-forward networks with the marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.

[66] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, pp. 489–501, 2006.

[67] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in extreme learning machines: A review," *Neural Networks*, vol. 61, pp. 32–48, 2015.

[68] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Transactions on Neural Networks*, 2015.

[69] G.-B. Huang and C.-K. Siew, "Extreme learning machine: Rbf network case," in *Control, Automation, Robotics and Vision Conference, 2004. ICARCV 2004 8th*, vol. 2, 2004, pp. 1029–1036.

[70] G.-B. Huang, "An insight into extreme learning machines: Random neurons, random features and kernels," *Cognitive Computation*, vol. 6, pp. 376–390, 2014.

[71] Y. Zhou, J. Peng, and C. L. P. Chen, "Extreme learning machine with composite kernels for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sens.*, vol. 8, no. 6, pp. 2351–2360, 2015.

[72] A. B. amd A. Araujo and D. Menotti, "Combining multiple classification methods for hyperspectral data interpretation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sens.*, vol. 6, no. 3, pp. 1450–1459, 2013.

[73] J. Li, Q. Du, W. Li, and Y. Li, "Optimizing extreme learning machine for hyperspectral image classification," *Journal of Applied Remote Sensing*, vol. 9, no. 1, p. 097296, 2015.

[74] A. Samat, P. Du, S. Liu, and L. Cheng, "E2lms:

Ensemble extreme learning machines for hyperspectral image classfication," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sens.*, vol. 7, no. 4, pp. 1060–1069, 2014.

[75] V. N. Vapnik, *Statistical learning theory.* John Wiley & Sons, New York, NY, 1998.

[76] B. Pan, J. Lai, and L. Shen, "Ideal regularization for learning kernels from labels," *Neural Networks*, vol. 56, pp. 22–34, aug 2014. [Online]. Available: http://dx.doi.org/10.1016/j.neunet.2014.04.003

[77] B. Scholkopf and A. J. Smola, *Learning with Kernels.* MIT Press, Cambridge, MA, 2002.

[78] M. Fauvel, J. Chanussot, and J. A. Benediktsson, "Kernel principal component analysis for the classfication of hyperspectral remote-sensing data over urban areas," *EURASIP Jour. Adv. Signal Proc.*, pp. 1–14, 2009.

[79] L. Gmez-Chova, G. Camps-Valls, J. Muoz-Mar, and J. Calpe, "Semisupervised image classification with laplacian support vector machines," *IEEE Geos. Remote Sens. Let.*, vol. 5, no. 3, pp. 336–340, 2008.

[80] E. Blanzieri and F. Melgani, "Nearest neighbor classification of remote sensing images with the maximal margin principle," *IEEE Geos. Remote Sens. Let.*, vol. 46, no. 6, pp. 1804–1811, 2008.

[81] D. Tuia and G. Camps-vallas, "Semisupervised remote sensing image classification with cluster kernels," *IEEE Geos. Remote Sens. Let.*, vol. 6, no. 2, pp. 224–228, 2005.

[82] C. Castillo, I. Chollett, and E. klein, "Enhanced duckweed detection using bootstrapped svm classification on medium resolution rgb modis imagery," *Int. Jour. Remote Sens.*, vol. 29, no. 19, pp. 5595–5604, 2008.

[83] J. Li, P. Marpu, A. Plaza, J. Bioucas-Dias, and J. A. Benediktsson, "Generalized composite kernel framework for hyperspectral image classification," *IEEE Trans. Geos. Remote Sens.*, vol. 51, no. 9, pp. 4816–4829, 2013.

[84] J. Li, J. Bioucas-Dias, and A. Plaza, "Semi-supervised hyperspectral image segmentation using multinomial logistic regression with active learning," *IEEE Trans. Geos. Remote Sens.*, vol. 48, no. 11, pp. 4085–4098, 2010.

[85] ——, "Hyperspectral image segmentation using a new Bayesian approach with active learning," *IEEE Trans. Geos. Remote Sens.*, vol. 49, no. 10, pp. 3947–3960, 2011.

[86] B. Krishnapuram, L. Carin, M. Figueiredo, and A. Hartemink, "Sparse multinomial logistic regression: Fast algorithms and generalization bounds," *IEEE Trans. Pat. Analysis Mach. Intell.*, vol. 27, no. 6, pp. 957–968, 2005.

[87] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst, Man, Cyber.*, vol. 21, no. 3, pp. 294–300, 1991.

[88] L. Breiman, J. H. Friedman, R. Olshen, and C. J. Stone, *Classification and Regression Tree*. Chapman and Hall/CRC, London, England, 1984.

[89] M. A. Friedl and C. E. Brodley, "Decision tree classification of land cover from remotely sensed data," *Remote Sens. Env.*, vol. 61, no. 3, pp. 399–409, 1997.

[90] M. Pal and P. Mather, "An assessment of the effectiveness of decision tree methods for land cover classification," *Remote Sens. Env.*, vol. 86, no. 4, pp. 554–565, 2003.

[91] G. J. Briem, J. A. Benediktsson, and J. R. Sveinsson, "Multiple classifiers applied to multisource remote sensing data," *IEEE Trans. Geos. Remote Sens.*, vol. 40, no. 10, pp. 2291–2299, 2003.

[92] P. O. Gislason, J. A. Benediktsson, and J. R. Sveinsson, "Random forests for land cover classification," *Pat. Recog. Lett.*, vol. 27, no. 4, pp. 294–300, 2006.

[93] L. Breiman, "Arcing classifier," *Ann. Statist.*, vol. 26, no. 3, pp. 801–849, 1998.

[94] ——, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 1, pp. 123–140, 1994.

[95] ——, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[96] J. Xia, P. Du, X. He, and J. Chanussot, "Hyperspectral remote sensing image classification based on rotation forest," *IEEE Geos. Remote Sens. Let.*, vol. 11, no. 1, pp. 239–243, 2014.

[97] B. Waske, J. A. Benediktsson, K. Arnason, and J. R. Sveinsson, "Mapping of hyperspectral AVIRIS data using machine-learning algorithms," *Canadian Jour. Remote Sens.*, vol. 35, pp. 106–116, 2009.

[98] Z. Zhi-Hua, *Ensemble Methods: Foundations and Algorithms*. Chapman and Hall/CRC, New York, NY, 2012.

[99] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse represenation," *IEEE Trans. Geosci. and Remote Sensing*, vol. 49, no. 10, pp. 3973–3985, Oct. 2011.

[100] Z. Lai, W. K. Wong, Y. Xu, C. Zhao, and M. Sun, "Sparse alignment for robust tensor learning," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 25, no. 10, pp. 1779–1792, oct 2014. [Online]. Available: http://dx.doi.org/10.1109/TNNLS.2013.2295717

[101] A. Castrodad, Z. Xing, J. Greer, E. Bosch, L. Carin, and G. Sapiro, "Learning discriminative sparse representations for modeling, source separation, and mapping of hyperspectral imagery," *IEEE Trans. Geosci. Remote Sensing*, vol. 49, no. 11, pp. 4263–4281, Dec. 2011.

[102] Y. Bengio, A. Courville, and P. Vincent, "Representation learning. a review and new perspectives," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, 2013.

[103] L. G. Chova, D. Tuia, G. Moser, and G. C. Valls, "Multimodal classification of remote sensing images: A review and future directions," *Proc. IEEE*, vol. 103, no. 9, pp. 1560–1584, 2015.

[104] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE Jour. Sel. Top. App. Earth Obs. Remote Sens.*, vol. 7, no. 6, pp. 2094 – 2107, 2014.

[105] C. Tao, H. Pan, Y. Li, and Z. Zou, "Unsupervised spectralspatial feature learning with stacked sparse autoen-

coder for hyperspectral imagery classification," *IEEE Geos. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2438 – 2442, 2015.

[106] Y. Chen, X. Zhao, and X. Jia, "Spectra-spatial classification of hyperspectral data based on deep belief network," *IEEE Jour. Sel. Top. App. Earth Obs. Remote Sens.*, vol. 8, no. 6, pp. 2381–2292, 2015.

[107] A. Romero, C. Gatta, and G. C. Valls, "Unsupervised deep feature extraction for remote sensing image classification," *IEEE Trans. Geos. Remote Sens.*, vol. 54, no. 3, pp. 1–14, 2016.

[108] J. A. Richards and X. Jia, *Remote Sensing Digital Image Analysis*, fourth, 2006 ed. Berlin: Springer-Verlag, 1986.

[109] P. C. Smits, S. G. Dellepiane, and R. A. Schowengerdt, "Quality assessment of image classification algorithms for land-cover mapping: a review and a proposal for a costbased approach," *Int. Jour. Remote Sens.*, vol. 20, pp. 1461–1486, 1999.

[110] W. D. Hudson and C. W. RAMM, "Correct formulation of the kappa coefficient of agreement," *Photo. Eng. Remote Sens.*, vol. 53, pp. 21–422, 1987.

[111] R. G. Congalton, "A review of assessing the accuracy of classification of remotely sensed data," *Remote Sens. Env.*, vol. 37, pp. 35–46, 1991.

[112] L. F. J. Janssen and F. J. M. V. D. Wel, "Accuracy assessment of satellite derived land-cover data: a review," *Photo. Eng. Remote Sens.*, vol. 60, pp. 419–426, 1994.

[113] G. M. Foody, "Status of land cover classification accuracy assessment," *Remote Sens. Env.*, vol. 80, pp. 185–201, 2002.

[114] A. Plaza, J. A. Benediktsson, J. W. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri, M. Marconcini, J. C. Tilton, and G. Trianni, "Recent advances in techniques for hyperspectral image processing," *Remote Sens. Env.*, vol. 113, no. Supplement 1, pp. 110–122, Sep. 2009.

[115] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geos. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, 2004.

[116] M. Pal, "Extreme-learning-machine-based land cover classification," *Int. Jour. Remote Sens.*, vol. 30, no. 14, pp. 3835–3841, 2009.

[117] M. Pal, A. E. Maxwell, and T. A. Warner, "Kernel-based extreme learning machine for remote-sensing image classification," *Remote Sens. Lett.*, vol. 4, no. 9, pp. 853–862, 2013.

[118] K. Hornik, "Approximation capabilities of multilayered feedforward networks," *Neural Networks*, vol. 4, pp. 251–257, 1991.

[119] S. Tamura and M. Tateishi, "Capabilities of a four-layered feedforward neural network: four layers versus three," *IEEE Transactions on Neural Networks*, vol. 8, no. 2, pp. 251–255, 1997.

[120] G.-B. Huang, "Learning capability and storage capacity of two hidden-layer feedforward networks," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 274–281, 2003.

[121] L. Prechelt, "Automatic early stopping using cross validation: quantifying the criteria," *Neural Networks.*, vol. 11, no. 4, pp. 761–767, 1998.

[122] G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing: A review," *ISPRS*, vol. 66, pp. 247–259, 2011.

[123] P. Ghamisi, M. S. Couceiro, and J. A. Benediktsson, "A novel feature selection approach based on FODPSO and SVM," *IEEE Trans. Geos. Remote Sens.*, vol. 53, no. 5, pp. 2935–2947, 2015.

[124] M. Pal and P. Mather, "Some issues in the classification of dais hyperspectral data," *Inter. Jour. Remote Sens.*, vol. 27, no. 14, pp. 2895–2916, 2006.

[125] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[126] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Mach. Learn.*, vol. 46, pp. 131–159, 2002.

[127] Y. Bazi and F. Melgani, "Toward an optimal SVM classification system for hyperspectral remote sensing images," *IEEE Trans. Geos. Remote Sens.*, vol. 44, no. 11, pp. 3374–3385, 2006.

[128] S. S. Keerthi and C. Lin, "Asymptotic behaviors of support vector machines with Gaussian kernel," *Neur. Comp.*, vol. 15, no. 7, pp. 1667–1689, 1998.

[129] G. Foody and A. Mathur, "A relative evaluation of multiclass image classification by support vector machines," *IEEE Trans. Geos. Remote Sens.*, vol. 42, no. 6, pp. 1335–1343, 2002.

[130] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[131] B. Krishnapuram, L. Carin, M. Figueiredo, and A. Hartemink, "Sparse multinomial logistic regression: Fast algorithms and generalization bounds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 957–968, 2005.

[132] S. R. Joelsson, J. A. Benediktsson, and J. R. Sveinsson, "Random forest classification of remote sensing data," in *Signal and Image Processing for Remote Sensing*, C. H. Chen, Ed. CRC Press, Boca Raton, FL., 2007, pp. 327–344.

[133] B. Waske, J. A. Benediktsson, and J. R. Sveinsson, "Random forest classifcation of remote sensing data," in *Signal and Image Processing for Remote Sensing*, C. H. Chen, Ed. CRC Press, New York, NY, 2012, pp. 363–374.

[134] V. F. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J. P. Rigol-Sanchez, "An assessment of the effectiveness of a random forest classifier for land-cover classification," *ISPRS*, vol. 67, pp. 93–104, 2012.

[135] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. Springer, 2008.

[136] ——, "The elements of statistical learning: Data mining, inference, and prediction," *Massachusetts: Addison-*

Wesley, 2009.

[137] J. C. Chan and D. Paelinckx, "Evaluation of random forest and adaboost treebased ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery," *Remote Sens. Env.*, vol. 112, no. 6, pp. 2999–3011, 2008.

[138] D. R. Cutler, T. C. Edwards, K. H. Beard, A. Cutler, and K. T. Hess, "Random forests for classification in ecology," *Ecology*, vol. 88, no. 11, pp. 2783 – 2792, 2007.

[139] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Trans. Geos. Remote Sens.*, vol. 43, no. 3, pp. 492 – 501, 2005.

[140] S. R. Joelsson, J. A. Benediktsson, and J. R. Sveinsson, "Random forest classifiers for hyperspectral data," pp. 25–29, 2005.

[141] J. L. Cushnie, "Analysis of a random forests model," *Jour. Mach. Learn. Res.*, vol. 13, pp. 1063–1095, 2012.

[142] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol, "Stacked denoising autoencoders," *Jour. Machine Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, 2010.

[143] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Comp.*, vol. 18, no. 7, pp. 1106–1114, 2012.

[144] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Neural Information Processing Systems 25*, Lake Tahoe, Nevada, USA, 2012, pp. 1527–1554.

[145] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geos. Remote Sens.*, 2016.

[146] Y. L. nad L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[147] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geos. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, 2016.

[148] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Trans. Geosci. and Remote Sens.*, vol. 43, pp. 1351–1362, 2005.

[149] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions Systems, Man and Cybernetics - Part B: Cybernetics*, vol. 42, no. 2, pp. 513–529, 2012.

[150] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[151] A. Iosifidis, A. Tefas, and I. Pitas, "On the kernel extreme learning machine classifier," *Pattern Recognition Letters*, vol. 54, pp. 11–17, 2015.

Pedram Ghamisi (SM'12-M'15) graduated with a B.Sc. in civil (survey) engineering from the Tehran South Campus of Azad University. Then, he obtained an M.E. degree with first class honors in remote sensing at K. N. Toosi University of Technology in 2012. He received a Ph.D. in electrical and computer engineering at the University of Iceland, Reykjavik, Iceland in 2015. Then, he worked as a postdoctoral research fellow at the University of Iceland. In 2015, Dr. Ghamisi won the prestigious Alexander von Humboldt Fellowship and started his work as a postdoctoral research fellow at Technische Universität München (TUM), Signal Processing in Earth Observation, Munich, Germany and GIScience and 3-D spatial data processing at the Institute of Geography, Heidelberg University, Heidelberg, Germany from October, 2015. He has also been working as a researcher at German Aerospace Center (DLR), Remote Sensing Technology Institute (IMF), Germany on deep learning since October, 2015. In the academic year 2010-2011, he received the Best Researcher Award for M.Sc. students in K. N. Toosi University of Technology. At the 2013 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Melbourne, July 2013, Dr. Ghamisi was awarded the IEEE Mikio Takagi Prize for winning the Student Paper Competition at the conference among almost 70 people. He was selected as a talented international researcher by Iran's National Elites Foundation in 2016. His research interests include machine learning, deep learning, hyperspectral image analysis. For more information, please see http://pedram-ghamisi.com/

Yushi Chen (M'11) received the Ph.D. degree from Harbin Institute of Technology, Harbin, China, in 2008. Currently, he is an Associate Professor in the School of Electrical and Information Engineering, Harbin Institute of Technology, China. His research interests include remote sensing data processing and machine learning.

Javier Plaza (M09-SM15) received the Computer Engineer degree in 2002, the M.Sc. degree in 2004, and the Ph.D. degree in 2008, all in Computer Engineering. He was the recipient of the Outstanding Ph.D. Dissertation Award at the University of Extremadura in 2008 where he is an Associate Professor with the Department of Technology of Computers and Communications. Dr. Plaza has authored or co-authored more than 120 scientific publications, including 32 JCR journal papers, 1 Guest Edited special issue, 10 book chapters and more than 80 peer-reviewed international conference proceedings. He is currently serving as Associate Editor of the IEEE Geoscience and Remote Sensing Letters. He has served as a Reviewer fore more than 180 papers submitted to more than 30 different journals. Dr. Plaza has served as a proposal evaluator for the Spanish Ministry of Science and Innovation since 2008. He has also served as proposal evaluator for the Czech Science Foundation and the Chilean National Science and Technology Commission. He is a recipient of the IEEE Signal Processing Magazine Best Column Paper Award in 2015. He received the Top Cited Article award of Elsevier's Journal of Parallel and Distributed Computing (in 2005-2010). He is also a recipient of the 2008 Best Paper award at the IEEE Symposium on Signal Processing and Information Technology. Additional information: http://www.umbc.edu/rssipl/people/jplaza

Jun Li (M13-SM16) Jun Li received the B.S. degree in geographic information systems from Hunan Normal University, Changsha, China, in 2004, the M.E. degree in remote sensing from Peking University, Beijing, China, in 2007, and the Ph.D. degree in electrical engineering from the Instituto de Telecomunicaes, Instituto Superior Tcnico (IST), Universidade Tcnica de Lisboa, Lisbon, Portugal, in 2011.

From 2007 to 2011, she was a Marie Curie Research Fellow with the Departamento de Engenharia Electrotcnica e de Computadores and the Instituto de Telecomunicaes, IST, Universidade Tcnica de Lisboa, in the framework of the European Doctorate for Signal Processing (SIGNAL). She has also been actively involved in the Hyperspectral Imaging Network, a Marie Curie Research Training Network involving 15 partners in 12 countries and intended to foster research, training, and cooperation on hyperspectral imaging at the European level. Since 2011, she has been a Postdoctoral Researcher with the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Escuela Politcnica, University of Extremadura, Cceres, Spain. Currently, she is a Professor with Sun Yat-Sen University, Guangzhou, China. Her research interests include hyperspectral image classification and segmentation, spectral unmixing, signal processing, and remote sensing.

Dr. Li is an Associate Editor for the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING. She has been a reviewer of several journals, including the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTESENSING, THE IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, Pattern Recognition, Optical Engineering, Journal of Applied Remote Sensing,and Inverse Problems and Imaging.

Antonio Plaza (M05-SM07-F15) is the Head of the Hyperspectral Computing Laboratory at the Department of Technology of Computers and Communications, University of Extremadura, where he received the M.Sc. degree in 1999 and the PhD degree in 2002, both in Computer Engineering. His main research interests comprise hyperspectral data processing and parallel computing of remote sensing data. He has authored more than 500 publications, including 182 JCR journal papers (132 in IEEE journals), 20 book chapters, and over 250 peer-reviewed conference proceeding papers. He has guest edited 9 special issues on hyperspectral remote sensing for different journals. Dr. Plaza is a Fellow of IEEE for contributions to hyperspectral data processing and parallel computing of Earth observation data. He is a recipient of the recognition of Best Reviewers of the IEEE Geoscience and Remote Sensing Letters (in 2009) and a recipient of the recognition of Best Reviewers of the IEEE Transactions on Geoscience and Remote Sensing (in 2010), for which he served as Associate Editor in 2007-2012. He is also an Associate Editor for IEEE Access, and was a member of the Editorial Board of the IEEE Geoscience and Remote Sensing Newsletter (2011-2012) and the IEEE Geoscience and Remote Sensing Magazine (2013). He was also a member of the steering committee of the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (JSTARS). He is a recipient of the Best Column Award of the IEEE Signal Processing Magazine in 2015, the 2013 Best Paper Award of the JSTARS journal, and the most highly cited paper (2005-2010) in the Journal of Parallel and Distributed Computing. He received best paper awards at the IEEE International Conference on Space Technology and the IEEE Symposium on Signal Processing and Information Technology. He served as the Director of Education Activities for the IEEE Geoscience and Remote Sensing Society (GRSS) in 2011-2012, and is currently serving as President of the Spanish Chapter of IEEE GRSS. He has reviewed more than 500 manuscripts for over 50 different journals. He is currently serving as the Editor-in-Chief of the IEEE Transactions on Geoscience and Remote Sensing journal. Additional information: http://www.umbc.edu/rssipl/people/aplaza