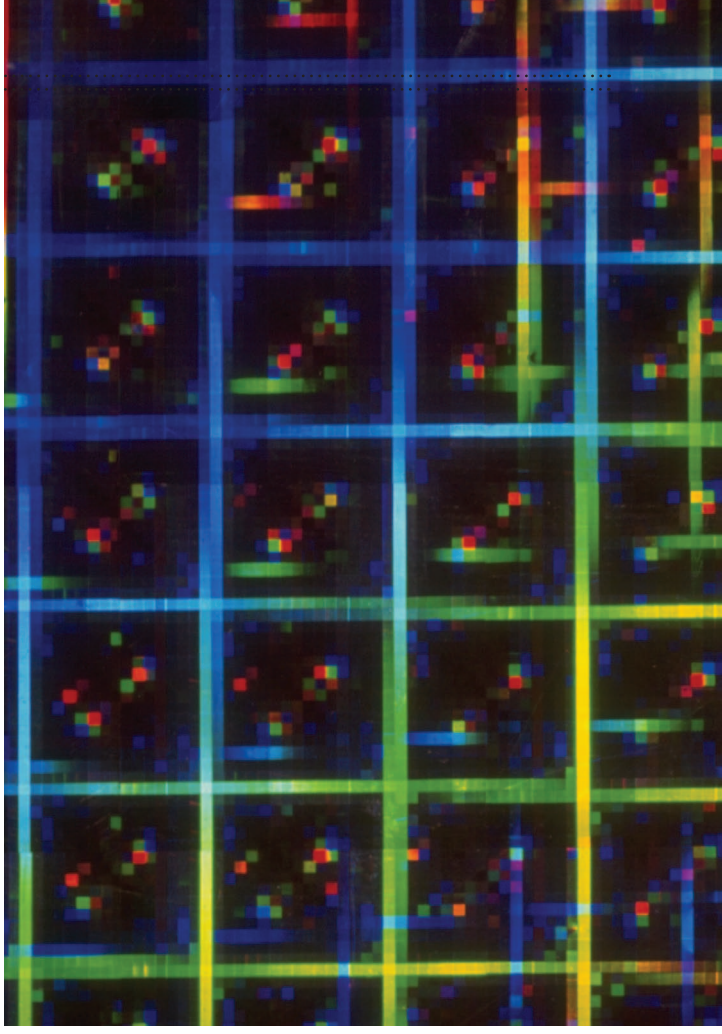
Advanced Spectral Classifiers for Hyperspectral Images

A review

PEDRAM GHAMISI, JAVIER PLAZA, YUSHI CHEN, JUN LI, AND ANTONIO PLAZA



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 for a number of reasons, such as the presence of redundant features, the imbalance among the limited number of available training samples, and 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 (SVMs), random forests (RFs), neural networks, deep approaches, and logistic regression-based techniques have attracted great interest in the hyperspectral community. This article reviews most of the existing spectral classification approaches in the literature. It also critically compares the most powerful hyperspectral classification approaches from different points of view, including their classification accuracy and computational complexity. The article goes on to provide several hints for readers about the logical choice of an appropriate classifier based on the application at hand.

Digital Object Identifier 10.1109/MGRS.2016.2616418
Date of publication: 16 March 2017

CLASSIFYING HYPERSPECTRAL DATA

Imaging spectroscopy (also known as *hyperspectral imaging*) is an important technique in remote sensing (RS). 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 earth's surface; while, in multispectral data, the number of spectral channels is 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. Several 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 in ecology, geology, hydrology, precision agriculture, and military applications.

Due to the detailed spectral information available from the hundreds of narrow bands collected by hyperspectral sensors, the accurate discrimination of different materials is possible. This fact makes hyperspectral data a valuable

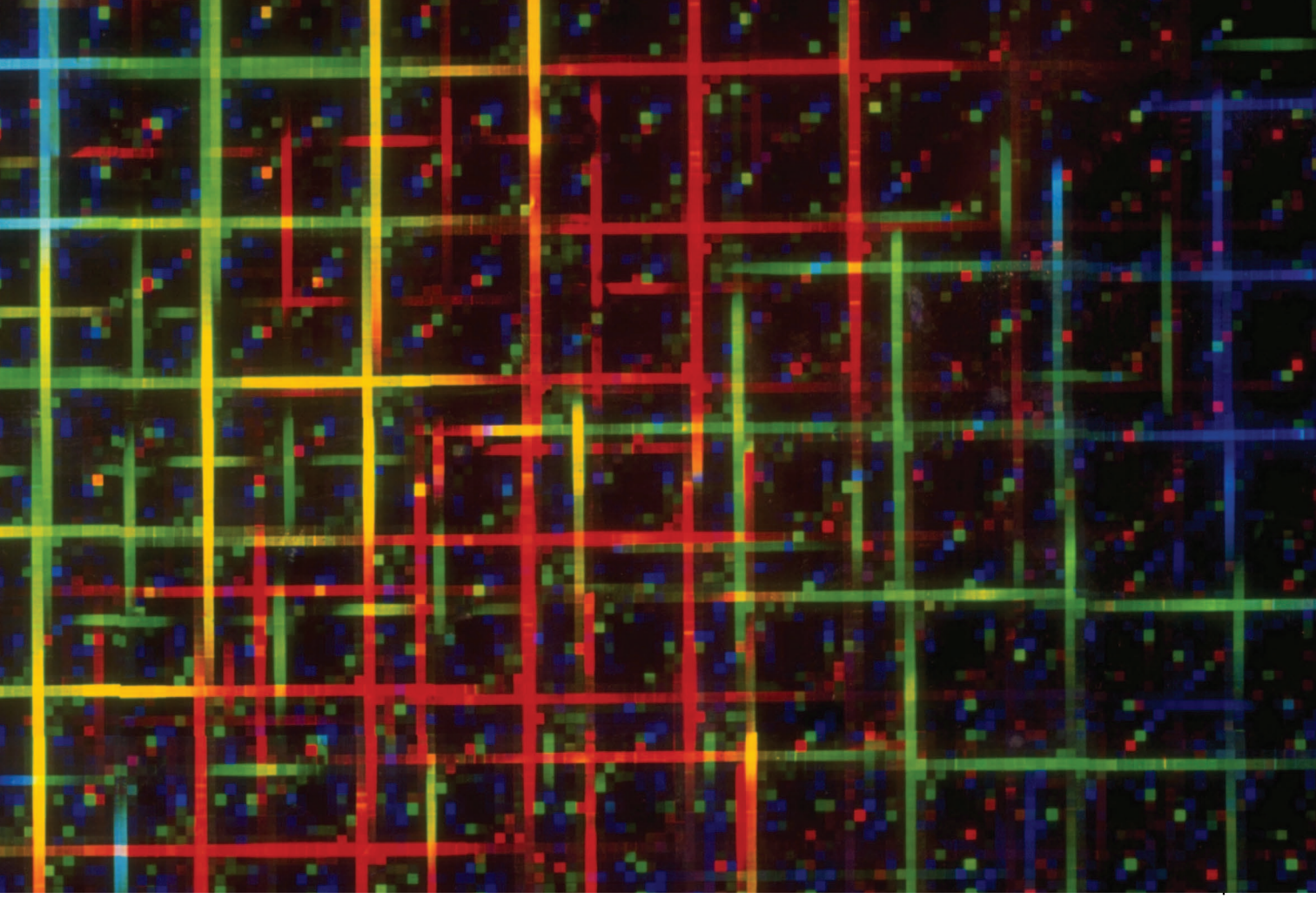


IMAGE ©COREL

source of information to be fed to advanced classifiers. The output of the classification step is known as the *classification map*.

Table 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 article, we tried to narrow down our description by excluding the items highlighted in green in Table 1, which have been extensively covered in other contributions. We reiterate that our main goal in this article is to provide a comparative assessment and best practice recommendations for the remaining contributions in Table 1.

With respect to the availability of training samples, classification approaches can be split into two categories, i.e., supervised and unsupervised classifiers. Supervised approaches classify input data for each class using a set of representative samples 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 approach classifies the data based only on an arbitrary number of initial cluster centers that may be either user specified or 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 cluster [4], [5].

There is a vast amount of literature on unsupervised classification approaches. Among these methods, Kmeans [6], Iterative Self-Organizing Data Analysis Technique (ISODATA) [7], and Fuzzy Cmeans [8] rank among the most popular. This set of approaches is known for being highly sensitive to the initial cluster configuration and may be trapped into suboptimal solutions [9]. To address this issue, researchers have tried to improve the resilience of the Kmeans (and its family) by optimizing it with bioinspired optimization techniques [3]. Since supervised approaches consider class-specific information provided by training samples, they lead to more precise classification maps than unsupervised approaches. In addition to unsupervised and

SINCE SUPERVISED APPROACHES CONSIDER CLASS-SPECIFIC INFORMATION PROVIDED BY TRAINING SAMPLES, THEY LEAD TO MORE PRECISE CLASSIFICATION MAPS THAN UNSUPERVISED APPROACHES.

TABLE 1. A TERMINOLOGY OF CLASSIFICATION APPROACHES BASED ON DIFFERENT CRITERIA. TO NARROW DOWN THIS ARTICLE'S RESEARCH LINE, WE INTENTIONALLY AVOID ELABORATING ON THE ITEMS HIGHLIGHTED IN GREEN.

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.
	Semisupervised classifiers	The training step in semisupervised approaches is based on both labeled and unlabeled training 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.
	Nonparametric classifiers	Nonparametric classifiers are not constrained by any assumptions on the distribution of input data.
Whether either a single classifier or an ensemble classifier is taken into account.	Single-classifier classifiers	In this approach, a single classifier is taken into account to allocate a class label for a given pixel.
	Ensemble (multiple) classifiers	In this approach, a set of classifiers (multiple classifiers) is taken into account to allocate a class label for a given pixel.
Whether or not the technique uses hard partitioning, in which each data point belongs to exactly one cluster.	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.
Whether spatial information is taken into account.	Spectral classifiers	This approach considers the hyperspectral image as a list of spectral measurements with no spatial organization.
	Spatial classifiers	This approach classifies the input data using spatially adjacent pixels, based on either a crisp or adaptive neighborhood system.
	Spectral–spatial classifiers	The 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 approach learns a model of the joint probability of the input and the labeled pixels and makes the prediction using Bayes rules.
	Discriminative classifiers	This approach 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 approach is able to predict, given a sample input, a probability distribution over a set of classes.
	Nonprobabilistic classifiers	This approach simply assigns the sample to the most likely class that the sample should belong to.
Which type of pixel information is used.	Subpixel classifiers	In this approach, the spectral value of each pixel is assumed to be a linear or nonlinear combination of endmembers (pure materials).
	Per-pixel	Input pixel vectors are fed to classifiers as inputs.
	Object- based and object-oriented classifiers	In this approach, 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 classifier is obtained using a combination of RS and geographic information system (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.

supervised approaches, semisupervised techniques have been introduced [10], [11]. With these, 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 semisupervised approaches can outperform that obtained by supervised classification. In this article, our focus is only on supervised classification approaches.

SUPERVISED CLASSIFICATION OF HYPERSPECTRAL DATA

A hyperspectral data set can be seen as a stack of many pixel vectors, here denoted by $\mathbf{x} = (x_1, \ldots, x_d)^T$, where d represents the number of bands or the length of the pixel vector. A common task when interpreting RS images is to differentiate between several land-cover classes. A classification algorithm is used to separate between different types

of patterns [5]. In RS, 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 γ classes $\Omega = \{\gamma_1, \gamma_2, \ldots, \gamma_K\}$, where K represents the number of classes, based on a set of training samples. The individual classes are discriminated based either on the similarity to a certain class or by decision boundaries that are constructed in the feature space [5].

PARAMETRIC VERSUS NONPARAMETRIC CLASSIFICATION

From another perspective, classification approaches can be split into parametric and nonparametric. For example, the widely used supervised maximum likelihood classifier (MLC) is often applied in the parametric context. In this 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 RFs) can be applied, even if the class-conditional densities are not known or cannot be reliably estimated [1]. Therefore, for hyperspectral data with a limited number of available training samples, such techniques may lead to more accurate classification results.

CHALLENGES FOR THE CLASSIFICATION OF HYPERSPECTRAL DATA

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

THE CURSE OF DIMENSIONALITY

In [14]–[16], researchers have reported some distinguishing geometrical, statistical, and asymptotical properties of high-dimensional data through some experimental examples, e.g., 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 $\mathbb{R}$ are 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 so-called *curse of dimensionality*.

CONVENTIONAL TECHNIQUES DEVELOPED FOR MULTISPECTRAL DATA MAY NOT BE SUITABLE FOR THE CLASSIFICATION OF HYPERSPECTRAL DATA.

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

UNCERTAINTIES

Uncertainties generated at different stages of the data acquisition and classification procedure can dramatically influence the classification accuracies and the quality of the final classification map [19]–[22]. There are many reasons for such uncertainties, including atmospheric conditions at the 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].

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 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.

THE USE OF NEURAL NETWORKS IN COMPLEX CLASSIFICATION SCENARIOS IS A CONSEQUENCE OF THEIR SUCCESSFUL APPLICATION IN THE FIELD OF PATTERN RECOGNITION.

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 uncertainty and can influence classification results dramatically. Fine spatial resolution can provide detailed information about the 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 in the classification system is a vibrant research topic in the hyperspectral community, and it has been investigated in many works such as [1] and [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 shadows, which lead to high spectral variations within the same land-cover class. These disadvantages may reduce classification accuracy if classifiers cannot effectively handle such effects [30].

LITERATURE REVIEW

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

PROBABILISTIC APPROACHES

A common subclass of classifiers is based on probabilistic approaches. This group of classifiers uses statistical terminologies to find the best class for a given pixel. In contrast with other algorithms that 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 the processing of hyperspectral images of median or big volume and an extremely large number of classes, using a high number of training samples [36], [37]. More recently, a subspace-based version of this classifier, called MLR*sub* [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 MLR*sub* for hyperspectral models presents two important advantages. On one hand, with the advantages of good algorithm generalization and fast computation, MLR has been widely 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.

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 the classification of hyperspectral images [54], [55]. The advantage of such approaches over probabilistic methods result mainly from the fact that neural networks do not need prior knowledge about the statistical distribution of the classes. Their attractiveness increased because of 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 that consider both supervised and unsupervised nonparametric approaches [58]–[62]. The feedforward neural network (FN)-based classifiers are 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. The BP presents two main problems, i.e., 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 fewer 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 FNs (SLFN) [66], [67]. Then, the concept has been extended to multihidden-layer networks [68], radial basis function (RBF) networks [69], and kernel learning [70]. The main characteristic of the ELM is that the hidden layer (feature mapping) is randomly fixed and need not be iteratively tuned. ELM-based networks are remarkably efficient in terms of accuracy and computational complexity and have been successfully applied as nonlinear classifiers for hyperspectral data, providing results comparable with state-of-the-art methodologies [71]–[74].

KERNEL METHODS, INCLUDING SVMs

SVMs are another example of a supervised classification approach. They have been widely used for the classification of hyperspectral data because of their ability to handle high-dimensional data with a limited number of training samples [1], [75], [76]. SVMs were originally introduced to classify linear classification problems. To generalize the SVM for nonlinear classification problems, the so-called *kernel trick* was introduced [77]. The sensitivity to the choice of the kernel and regularization parameters is the most important disadvantage of a kernel SVM. For the former, the Gaussian RBF is widely used in RS [77]. The latter is classically addressed using cross validation techniques that employ training data [78]. Gómez et al. proposed an approach by combining both labeled and unlabeled pixels using clustering and the mean map kernel to increase the classification accuracy and reliability of SVMs [79]. In [80], a local k-nearest neighbor adaptation was taken into account to formulate localized SVM variants. 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 the 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 that an incorrectly classified training sample in a given learning step is removed from the training pool, reassigned a correct label, and reintroduced into the training set in the subsequent training cycles.

In addition to the SVM, a composite kernel framework for the classification of hyperspectral images was recently investigated. In [83], a linearly weighted composite kernel framework with SVMs was used for the classification of hyperspectral data. However, classification using composite kernels and SVMs demands a convex combination of kernels and a time-consuming optimization process. To overcome these limitations, a generalized composite kernel framework for spectral–spatial classification was developed in [83]. The MLR [33], [37], [84] was also investigated as an alternative to the SVM classifier for the construction of composite kernels, and a set of generalized composite kernels that can be linearly combined without any constraint of convexity was proposed.

DECISION TREES

Decision trees represent another subclass of nonparametric approaches that can be used for both classification and regression. Safavian and Landgrebe [85] 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 that generally use the entire feature space at once and make a single membership decision per class [86]. The relative structural simplicity of decision trees and the relatively short training time required (compared to methods that can be computationally demanding) are the main advantages of such classifiers [1], [87], [88]. 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 degraded 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 [89], [90].

SVMs HAVE BEEN WIDELY USED FOR THE CLASSIFICATION OF HYPERSPECTRAL DATA BECAUSE OF THEIR ABILITY TO HANDLE HIGH-DIMENSIONAL DATA WITH A LIMITED NUMBER OF TRAINING SAMPLES.

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 to increase classification accuracies [1]. To develop an efficient multiple classifier, one needs to determine an effective combination of classifiers such that each is able to benefit the others while avoiding the weaknesses of each [89]. Two highly used multiple classifiers are boosting and bagging [89], [91], [92], which were elaborated in detail in [1].

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 the number of available training samples. In [96], rotation forest is proposed based on the idea of RFs to simultaneously encourage both member diversities and individual accuracy within a classifier ensemble. For a detailed description of this approach, see [1], [90], [93], [95], and [96].

SPARSE REPRESENTATION CLASSIFIERS

Another important development has been the use of sparse representation classifiers (SRCs) with dictionary-based generative models [97], [98]. In this case, an input signal is represented by a sparse linear combination of samples (atoms) from a dictionary [97], where the training data are generally used as the dictionary. The main advantage of SRCs is that they avoid the heavy training procedure that 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 [99].

DEEP LEARNING

Deep learning is a kind of neural network with multilayers, typically deeper than three layers, that tries to hierarchically learn the features of input data. Deep learning is a fast-growing topic that has shown usefulness in many research areas, including computer vision and natural language processing [100]. In the context of RS, some deep models have been proposed for hyperspectral data feature extraction and classification [101]. The stacked autoencoder (SAE) and the autoencoder (AE) with sparse constrain were proposed for hyperspectral data classification [102], [103]. Later, another deep model, i.e., the deep belief network (DBN), was proposed for the classification of hyperspectral data [104]. Very recently, an unsupervised convolutional neural network (CNN) was proposed for RS image analysis, which uses greedy layer-wise unsupervised learning to formulate a deep CNN model [105].

DEEP LEARNING IS A FAST-GROWING TOPIC THAT HAS SHOWN USEFULNESS IN MANY RESEARCH AREAS, INCLUDING COMPUTER VISION AND NATURAL LANGUAGE PROCESSING.

CLASSIFICATION ACCURACY ASSESSMENT

Accuracy assessment is a crucial step in evaluating 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 the reference data is due to 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 RS [1], [106]–[111].

THIS ARTICLE'S CONTRIBUTIONS

The main aim of this article is to critically compare representative spectral-based classifiers (such as those outlined in the "Literature Review" section) from different perspectives. Without any doubt, classification plays an important role in 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 more specific, we consider only spectral and per-pixel-based classifiers in this article. In other words, spatial classifiers, fuzzy approaches, subpixel classifiers, object-based approaches, and per-field RS-GIS approaches are considered to be out of scope.

Compared to previous review papers, such as [112] published in 2009 that provides a general review of the advances in techniques for hyperspectral image processing to that date, this article deals specifically with spectral classifiers and includes the most recent and advanced spectral classification approaches in the hyperspectral community (with many new developments since the publication of the previous paper). In addition, we believe that a few specific classifiers have gained great interest in the hyperspectral community because of their ability to handle high-dimensional data with a limited number of training samples. Among those approaches, neural networks, RFs, MLR, SVMs, and deep CNN-based classifiers are the most widely used at present. As a result, we first elaborate on these approaches and then further compare them based on different scenarios, such as the capability of the methods in terms of having different numbers of training samples, spatial resolution, stability, complexity, and the 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. To make the equations easier to follow, Table 2 details all of the notations used in this article.

Figure 1 shows the classification approaches investigated in this article along with their publication year and the number of citations obtained 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 RS community. We used [58] for neural networks, [90] for RFs, [33] for MLR, [113] for SVMs, [114] for ELM, and [115] for kernel ELM (KELM). Since the CNN was published only very recently, it is not shown in Figure 1.

NEURAL NETWORKS

Artificial neural networks (ANNs) have been traditionally used in multihyperspectral data classification. FNs, in particular, have been extensively applied because of their ability to approximate complex nonlinear mappings directly from the input samples using a single hidden layer [116]. Traditional learning techniques are based on the original BP algorithm [63]. The most popular group is the gradient

TABLE 2. A 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	v	Stack variable	k	Kernel
$\|\cdot\|$	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	$p(y_i \| x_i)$	Probability of pixel i
α	Lagrange multiplier	β	Output weight	$\mathbf{v}$	Visible units	h	Hidden units

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 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 a large number of weights. Lastly, Newton-based learning algorithms generally provide better and faster 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 introduces an extra computational burden for the calculation of the Hessian matrix.

Recently, the ELM algorithm has been proposed to train SLFNs [66], [67] and has emerged as an efficient algorithm that provides accurate results in much less time. Traditional gradient-based learning algorithms assume that all of 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 [117] and [118], it was first shown that an SLFN (with N hidden nodes) with randomly chosen input weights and hidden-layer biases can learn exactly 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 proven that the input weights and hidden-layer biases of an SLFN can be randomly assigned if the activation function of the hidden layer is infinitely differentiable, which allows for the analytical determination of the rest of the parameters (the weights between the hidden and output layers) since the SLFN is a linear system. This fact leads to a significant decrease of the computational complexity of the algorithm, making it much faster than its predecessors, and turning ELM into the main alternative for the analysis of large amounts of data.

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

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

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

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

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

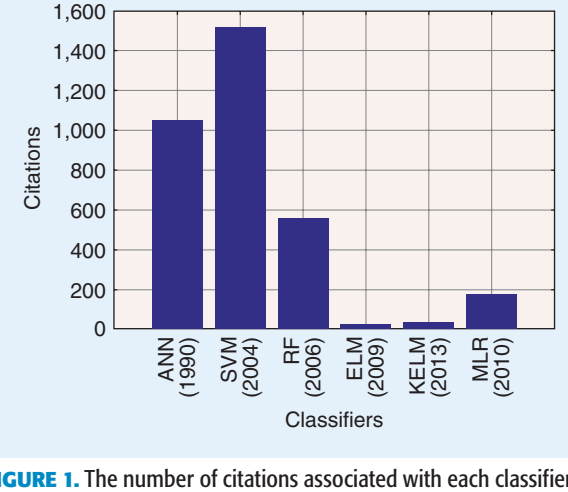


FIGURE 1. The number of citations associated with each classifier.

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

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

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

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

- First and foremost, all gradient-based learning algorithms are very time consuming in most applications. This became an important problem when classifying hyperspectral data.
- The size of the learning rate parameter strongly affects the performance of the network. Values that are too small generate very slow convergence processes, while scores in η that are too large make the learning algorithm diverge and become unstable.
- The error surface generally presents local minima. Gradient-based learning algorithms can get stuck at local minima. This can be an important issue if local minima are far above global minima.
- 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 [119].

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 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 finding 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.

RECENTLY, THE ELM ALGORITHM HAS BEEN PROPOSED TO TRAIN SLFNs AND HAS EMERGED AS AN EFFICIENT ALGORITHM THAT PROVIDES ACCURATE RESULTS IN MUCH LESS TIME.

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

Let $\mathbf{h}(\mathbf{x}) = [f(\mathbf{w}_1 \cdot \mathbf{x} + b_1), \ldots, f(\mathbf{w}_L \cdot \mathbf{x} + b_L)]$, if we express (6) from the optimization theory point of view

$$\min_\beta \frac{1}{2} \| \beta \|_2^2 + C\frac{1}{2}\sum_{i=1}^{n} \xi_i^2, \quad (7)$$

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

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

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

This expression can be generalized to a 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:

- There is no need to iteratively tune the input weights $\mathbf{w}_i$ and the hidden-layer biases b_i using slow gradient-based learning algorithms.
- 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.
- ELM's learning speed 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].
- The use of ELM avoids inherent problems with gradient-descent methods such as getting stuck in a local minima or overfitting the model [66].

SUPPORT VECTOR MACHINES

SVMs [113] have often been used for the classification of hyperspectral data because of their ability 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 the 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 to statistical approaches, SVMs minimize classification error on unseen data without any prior assumptions made on the probability distribution of the data [120].

The SVM tries to maximize the margins between the hyperplane and the closest training samples [75]. In other words, to train the classifier, only samples that are close to the class boundary are needed to locate the hyperplane vector. This is why the training samples closest to the hyperplane are called *support vectors*. More importantly, since

only the closest training samples are influential in placing the hyperplane in the feature space, the SVM can classify the input data efficiently even if only a limited number of training samples is available [2], [113], [121], [122]. In addition, SVMs can efficiently handle the classification of noisy patterns and multimodal feature spaces.

With regard to a binary classification problem in a d-dimensional feature space, $\mathbb{R}^d, \mathbf{x}_i \in \mathbb{R}^d, i = 1, \ldots, n$ 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 $w\mathbf{x} + b = \pm 1$ 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 nonseparable 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 previously 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], [95]. When the Mercer's conditions are met, the final hyperplane can be defined by

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

where α_i denotes the Lagrange multipliers. For a detailed derivation of (13), we refer readers to [123]. 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 was introduced [124]. In [125], a genetic algorithm-based approach was used to regulate the hyperplane parameters of an SVM while it found efficient features to be fed to the classifier.

In terms of kernels, the Gaussian RBF kernel may be the most widely used in RS [77], [95]. 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], [126].

SVMs were originally developed for binary classification problems. In general, one needs to deal with multiple classes in RS [1]. To address this, several multiclass strategies have been introduced in the literature. Among those approaches, two main strategies are best known and are based on the separation of the multiclass problem into several binary classification problems [127]. These are the one-against-one strategy and the one-against-rest strategy [95]. The following are some important points:

- The capability of the SVM in handling a limited number of training samples, self-adaptability, a swift training stage, and ease of use are considered as the main advantages of this classifier. In addition, SVMs are resilient to becoming trapped in local minima, since the convexity of the cost function enables the classifier to consistently identify the optimal solution [120]. More precisely, SVMs deal with quadratic problems and, as a result, they guarantee 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 is the case for many approaches such as neural networks. Last but not least, SVMs are nonparametric and do not assume a known statistical distribution of the data to be classified. This is considered an important advantage because the data acquired from remotely sensed imagery usually have unknown distributions [120].
- One drawback of the SVM lies in setting 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 tradeoff between maximizing the margin and minimizing the training error, is highly important.

ALL GRADIENT-BASED LEARNING ALGORITHMS ARE VERY TIME CONSUMING IN MOST APPLICATIONS.

For further reading, a detailed introduction of SVMs is given by Burges [123], Cristianini and Shawe-Taylor [130], and Scholkopf and Smola [77].

MULTINOMIAL LOGISTIC REGRESSION

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

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

where $\boldsymbol{\omega} = [\boldsymbol{\omega}^{(1)^T}, \ldots, \boldsymbol{\omega}^{(K-1)^T}]^T$ are the logistic regressors. Again, y_i is the class label of pixel $\mathbf{x}_i \in \mathbb{R}^d$, d is the number of bands, and K is the number of classes. Since the density in (14) does not depend on translations of the regressors $\boldsymbol{\omega}^{(k)}$, we take $\boldsymbol{\omega}^{(K)} = 0$. The term $\Phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \ldots, \phi_l(\mathbf{x})]^T$ is the fixed functions of the input, often termed *features*. The open structure of $\Phi(\mathbf{x})$ leads to flexible selection of the input features, i.e, they can be linear, kernel, or nonlinear functions. To control the algorithm complexity and its generalization capacity, the regressor $\boldsymbol{\omega}$ is modeled as a random vector with Laplacian density [129]

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

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 posteriori estimate of $\boldsymbol{\omega}$

$$\hat{\boldsymbol{\omega}} = \underset{\omega}{\operatorname{argmax}} \ \ell(\boldsymbol{\omega}) + \log p(\boldsymbol{\omega}), \tag{16}$$

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

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

where n is the number of training samples. Although convex, (16) is difficult to compute because the term of $\ell(\boldsymbol{\omega})$ is nonquadratic and the term $\log p(\boldsymbol{\omega})$ is nonsmooth. Following [32], $\ell(\boldsymbol{\omega})$ can be estimated by a quadratic function. However, the problem is still difficult, as $\log p(\boldsymbol{\omega})$ is nonsmooth. Optimization problem (16) can be solved by the SMLR in [129] and by the fast SMLR 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. To address this issue, the LORSAL algorithm is proposed in [36] and [37] to deal with high-dimensional features and leads to good success in hyperspectral classification. For more information about the LORSAL algorithm, see [33] and [37].

MLR CLASSIFIERS ARE ABLE TO DIRECTLY LEARN THE POSTERIOR CLASS DISTRIBUTIONS AND DEAL WITH THE HIGH DIMENSIONALITY OF HYPERSPECTRAL DATA IN A VERY EFFECTIVE WAY.

Finally, the advantages of MLR are as follows:

- MLR classifiers are able to directly learn 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.
- The sparsity-inducing prior on the regressors leads to sparse estimates, which allows us to control the algorithm complexity and their generalization capacity.
- The open structure of the MLR results in a good flexibility for the input functions, which can be linear, kernel-based, or nonlinear.

RANDOM FORESTS

RFs were proposed in [93] 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, is trained and their individual results are then combined through a voting process [130], [131]. In other words, the classification label is allocated to the input vector $(\mathbf{x})$ through $y_{rf}^{B} = \textit{majority vote}\{y_b(\mathbf{x})\}_1^B$, where $y_b(\mathbf{x})$ is the class prediction of the bth tree and B shows the total number of trees. RFs can be considered to be a particular case of decision trees. However, since RFs are composed of many classifiers, this implies special characteristics that make them completely different from traditional classification trees; therefore, they should be understood as a new type of classifier [132].

The training algorithm for RFs applies the general technique of bootstrap aggregating, or bagging, to tree learners [92]. 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) [132]. 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 [133]. By training many trees on a single training set (or even the same tree many times if the training algorithm is deterministic), strongly correlated trees are produced. Bootstrap sampling decorrelates the trees by showing them different training sets. RF uses trees as base classifiers, $\{h(\mathbf{x}, \theta_k), \ k = 1, \ldots, \}$, 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 others may not be used, greater classifier stability is achieved. This makes the classifier more robust when a slight variation in input data occurs, and consequently higher classification accuracy can be obtained [132], [134].

As mentioned in several studies, such as [88], [89], [132], and [135], 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 to generate the prediction model, i.e., the number of trees and the number of prediction variables. The number of trees is a free parameter that 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 [93], [131], [136]. For detailed information regarding RFs and their different implementations, see [1], [130], and [131]. The number of prediction variables is referred to as 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 to approximately the square root of the number of input features [130], [131], [137], [138].

By using RFs, the out-of-bag error, the variable importance, and the proximity analysis can be driven. To find detailed information about the RF and its derived parameters, see [1], [86], [93], [130], [131], and [136]. The following are some important points about RFs:

- RFs are quite flexible, and they can handle different scenarios, such as large numbers of attributes, very limited numbers of training samples, and small or large data sets. In addition, they are easy and quick to evaluate.
- RFs do not assume any underlying probability distribution for input data, can provide a good classification result in terms of accuracies, and can handle many variables and a large amount of missing data. Another advantage of the 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.
- The generated forest can be saved and used for other data sets.
- 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.
- 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 been 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 [139].

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 intraclass variability make it extremely difficult to effectively extract the features. Moreover, hyperspectral data quickly increase in volume, velocity, and variety, so they are difficult to analyze in complicated real situations. On the other hand, it is believed that deep models can progressively lead to more invariant and abstract features at higher layers [100]. Therefore, deep models have the potential to be a promising tool. Deep learning involves a number of models, including the SAE [140], DBN [141], and deep CNN [142].

SINCE RFs ARE COMPOSED OF MANY CLASSIFIERS, THIS IMPLIES SPECIAL CHARACTERISTICS THAT MAKE THEM COMPLETELY DIFFERENT FROM TRADITIONAL CLASSIFICATION TREES.

THE SAE

The AE is the basic part of the SAE [140]. As shown in Figure 2, an AE contains one visible layer of d inputs, one hidden layer of L units, and one reconstruction layer of d units. During the 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 the *encoder*. Then, $\mathbf{z}$ is mapped to $\mathbf{r} \in \mathbb{R}^d$ by a decoder, which is called the *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 of the hidden and output units, and $f(.)$ denotes the activation function.

Stacking the input and hidden layers of AEs together layer by layer constructs an SAE. Figure 3 shows a typical instance of an SAE connected with a subsequent logistic regression classifier. The SAE can be used as a spectral classifier.

DBNs

The restricted Boltzmann machine (RBM) is a layer-wise training model in the construction of a DBN [143]. As shown in Figure 4, it is a two-layer network with visible units

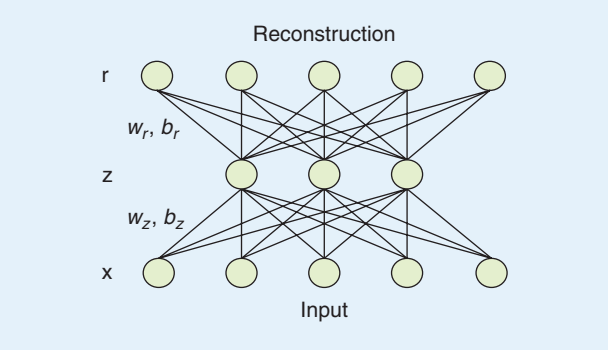


FIGURE 2. A single hidden-layer AE. The model learns a hidden feature z from input x by reconstructing it on r.

$\mathbf{v} = \{0,1\}^d$ and hidden units $\mathbf{h} = \{0,1\}^L$. A joint configuration of the units has an energy given by

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

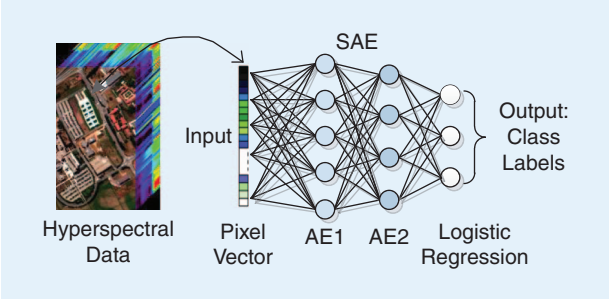


FIGURE 3. A typical instance of an SAE connected with a subsequent logistic regression classifier.

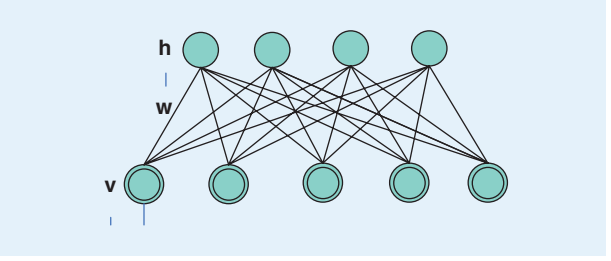


FIGURE 4. A graphical illustration of an RBM. The top layer (h) represents the hidden units and the bottom layer (v) represents the visible units. w: input weight.

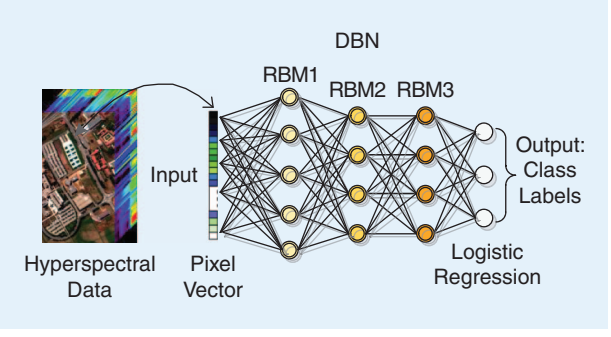


FIGURE 5. A spectral classifier based on a DBN. The classification scheme shown here has four layers: one input layer, two RBMs, and a logistic regression layer.

where $\theta = \{b_i, a_j, w_{ij}\}$, in which w_{ij} is the weight between visible unit i and hidden unit j, and b_i and a_j are bias terms of the visible and hidden unit, respectively. The learning of w_{ij} is done by a method called *constructive divergence* [141].

Due to the complexity of input hyperspectral data, RBM is not the best way to capture the features. After the training of RBM, the learned 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 Figure 5, a DBN is employed for feature learning and adds a logistic regression layer above the DBN to constitute a DBN-logistic regression framework.

THE DEEP CNN

The CNN is a special type of deep learning model that 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:

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

where $\mathbf{x}_i^{l-1}$ is the ith feature map of $(l-1)$th layer, $\mathbf{x}_j^l$ is the jth feature map of current (i)th layer, and M is the number of input feature maps. 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. It should be noted that here we explain one-dimensional (1-D) CNN, as this article deals with spectral classifiers. To find detailed information about two-dimensional (2-D) and three-dimensional (3-D) CNN for the classification of hyperspectral data, see [145].

The 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 [144]. By stacking several convolution layers with nonlinear operation and several pooling layers, a deep CNN can be formulated. A deep CNN can hierarchically extract the features of inputs, which tend to be invariant and robust [100].

The architecture of a deep CNN for spectral classification is shown in Figure 6. The input of the system is a pixel vector of hyperspectral data, and the output is the

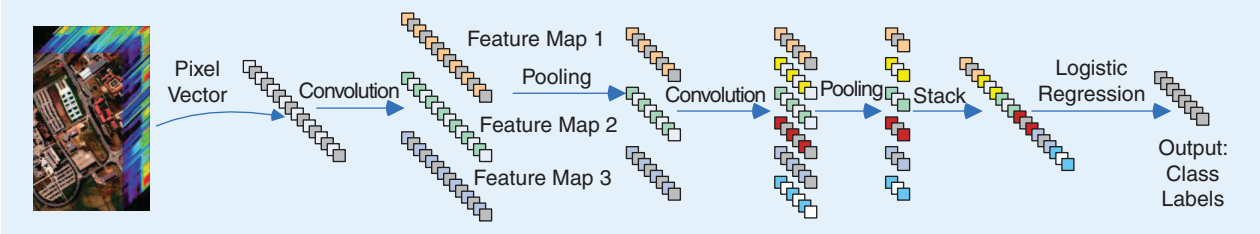


FIGURE 6. A spectral classifier based on a deep CNN.

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 that captures the spectral information.

DISCUSSION OF DEEP LEARNING APPROACHES

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

- Recently, some deep models have been used in hyperspectral data feature extraction and classification. Deep learning opens a new window for future research, showcasing the deep learning-based methods' huge potential [145].
- 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 the machine learning community, though we may be able to use grid searches to find a proper deep model.
- 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 this issue, it is necessary to use powerful regularization methods.
- Deep learning methods can be combined with other methods, such as sparse coding and ensemble learning, which is another research area in hyperspectral data classification.

EXPERIMENTAL RESULTS

This section describes our experimental results, including the different hyperspectral data sets used in experiments, the setup for the different algorithms to be compared, and the obtained results with a detailed discussion about the use of the different classifiers tested in different applications. The sets of training and test samples used in this article are available on request by e-mailing the authors.

DATA DESCRIPTION

PAVIA UNIVERSITY

This hyperspectral data set has been repeatedly used. It was captured over the city of Pavia, Italy, by the Reflective Optics Spectrographic Imaging System (ROSIS-03) airborne instrument. The flight over the city of Pavia 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 were processed. The data have been corrected atmospherically but not geometrically. The spatial resolution is 1.3 m per pixel. The data set, with dimensions of 640×340 pixels, 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 (see Table 3). Figure 7 presents a false color image of ROSIS-03 Pavia University data and their 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.

TABLE 3. PAVIA UNIVERSITY: THE NUMBER OF TRAINING AND TEST SAMPLES.

CLASS		NUMBER OF SAMPLES
NUMBER	NAME	TOTAL
1	Asphalt	6,304
2	Meadow	18,146
3	Gravel	1,815
4	Tree	2,912
5	Metal sheets	1,113
6	Bare soil	4,572
7	Bitumen	981
8	Brick	3,364
9	Shadow	795
Total		40,002

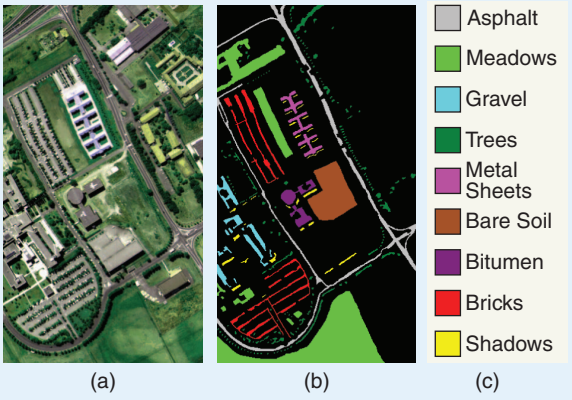


FIGURE 7. Some ROSIS-03 Pavia University hyperspectral data: (a) the three-band false color composite, (b) the reference data, and (c) the color code.

DEEP LEARNING OPENS A NEW WINDOW FOR FUTURE RESEARCH, SHOWCASING THE DEEP LEARNING-BASED METHODS' HUGE POTENTIAL.

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.

TABLE 4. INDIAN PINES: THE NUMBER OF TRAINING AND TEST SAMPLES.

CLASS		NUMBER OF SAMPLES
NUMBER	NAME	TOTAL
1	Corn-no till	1,434
2	Corn-minimum till	834
3	Corn	238
4	Grass/pasture	497
5	Grass/trees	747
6	Hay-windrowed	489
7	Soybean-no till	968
8	Soybean-minimum till	2,468
9	Soybean-clean	614
10	Wheat	212
11	Woods	1,294
12	Building/grass/tree-drives	380
13	Stone-steel towers	95
14	Alfalfa	54
15	Grass/pasture-mowed	26
16	Oats	20
Total		10,366

TABLE 5. HOUSTON: THE NUMBER OF TRAINING AND TEST SAMPLES.

CLASS		NUMBER OF SAMPLES	
NUMBER	NAME	TRAINING	TEST
1	Grass-healthy	198	1,053
2	Grass-stressed	190	1,064
3	Grass-synthetic	192	505
4	Tree	188	1,056
5	Soil	186	1,056
6	Water	182	143
7	Residential	196	1,072
8	Commercial	191	1,053
9	Road	193	1,059
10	Highway	191	1,036
11	Railway	181	1,054
12	Parking lot 1	192	1,041
13	Parking lot 2	184	285
14	Tennis court	181	247
15	Running track	187	473
Total		2,832	12,197

Its spatial dimensions are 145×145 pixels, and its spatial resolution is 20 m per pixel. This data set originally included 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 contain 16 classes of interest that represent mostly different types of crops and are detailed in Table 4. Figure 8 shows a three-band false color image and its corresponding reference samples.

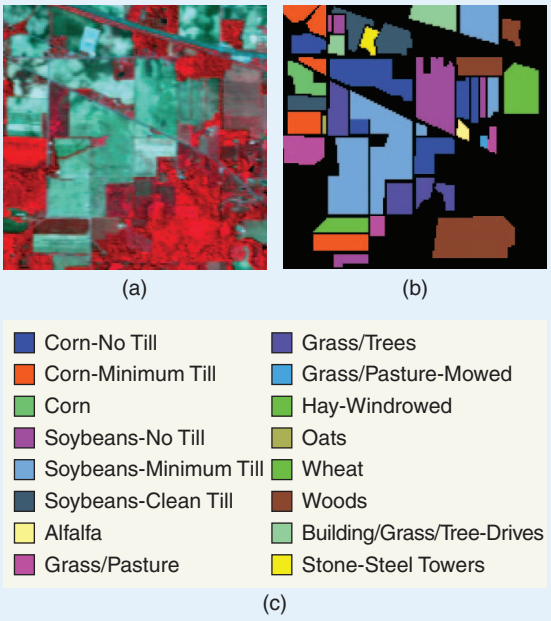


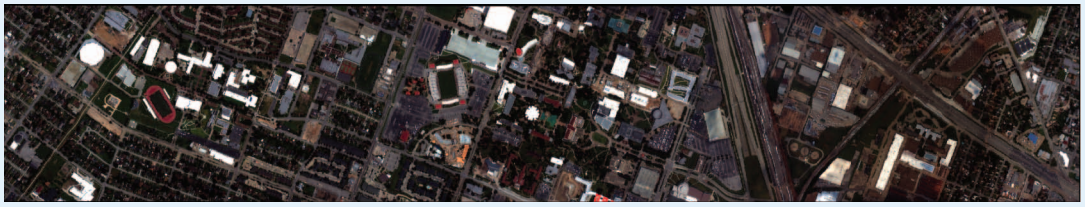
FIGURE 8. Some AVIRIS Indian Pines hyperspectral data: (a) the three-band false color composite, (b) the reference data, and (c) the color code.

HOUSTON DATA

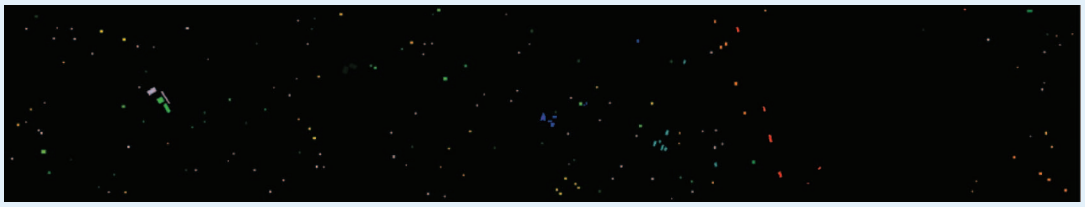
This data set was captured by the Compact Airborne Spectrographic Imager (CASI) over the University of Houston campus and the neighboring urban area in June 2012. With a size of 349×1905 pixels and a spatial resolution of 2.5 m, this data set is composed of 144 spectral bands ranging from 0.38 to 1.05 m. These data consist of 15 classes, including healthy grass, stressed grass, synthetic grass, trees, soil, water, residential, commercial, road, highway, railway, parking lot 1, parking lot 2, tennis court, and running track. Parking lot 1 includes parking garages at the ground level and also in elevated areas, while parking lot 2 corresponds to parked vehicles. Table 5 demonstrates the different classes with the corresponding number of training and test samples. Figure 9 shows a three-band false color image and its corresponding already-separated training and test samples.

ALGORITHM SETUP

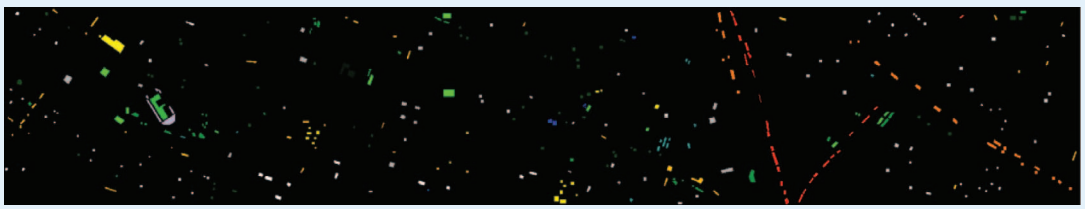
In this article, two different scenarios were defined to evaluate different approaches. In the first scenario, different percentages of the available reference data were chosen as training samples. In this scenario, only Indian Pines and Pavia University were considered. For Indian Pines, 1, 5, 10, 15, 20, and 25% of the whole sample were randomly selected as training samples, except for classes alfalfa,



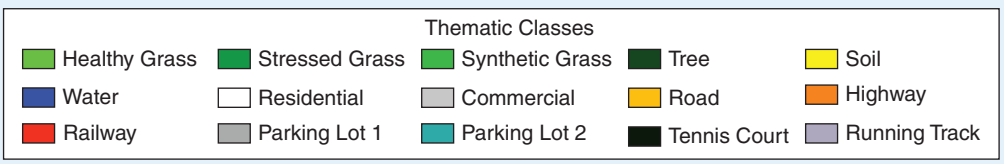
(a)



(b)



(c)



(d)

FIGURE 9. Some CASI Houston hyperspectral data: (a) a color composite representation of the data, using bands 70, 50, and 20 as R, G, and B, respectively; (b) training samples; (c) test samples; and (d) a legend of the different classes.

grass/pasture-mowed, and oats. These classes contain only a small number of samples in the reference data. Therefore, only 15 samples in 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% of the whole samples were randomly selected as training samples and the rest as test samples. The experiments were repeated ten times, and the mean and the standard deviation of the obtained overall accuracy (OA) are reported.

For the second scenario, the Houston data were taken into account. The training and test samples of these data were separated (Table 5). The results were evaluated using OA, average accuracy (AA), kappa coefficient (Kappa), and class-specific accuracies.

The following classifiers were investigated and compared in the two different scenarios discussed previously:

- SVM
- RF
- BP (also known as *multilayer perceptron*)
- ELM
- KELM
- 1-D CNN
- MLR.

For the MLR classifier, which is executed by the LORSAL algorithm [36,] [37], we use a Gaussian 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 [146]. For the parameters involved in the algorithm, we use the default settings provided in the online demo (http://www.lx.it.pt/~jun/demo_LORSAL_AL.rar), 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 (the parameter that controls the amount of penalty during the SVM optimization) and γ (the spread of the RBF kernel) have been traced in the range of $C = 10^{-2}, 10^{-1}, \ldots, 10^{4}$ and $\gamma = 10^{-3}, 10^{-2}, \ldots, 10^{4}$ using fivefold 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

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	

FIGURE 10. The architectures of the 1-D CNN on three data sets.

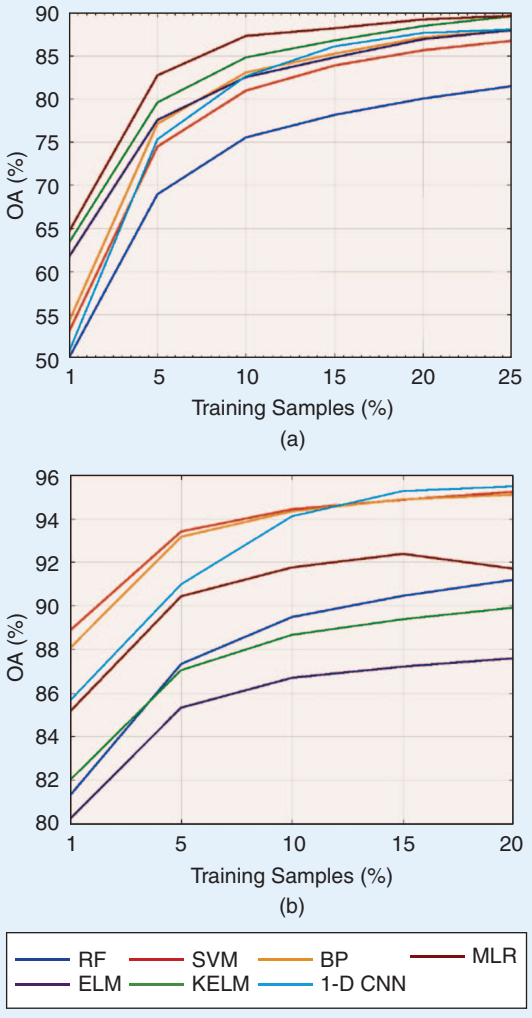


FIGURE 11. Scenario 1: OA. The OA of different approaches (i.e., the average value over ten runs) using different percentages of training samples from (a) Indian Pines and (b) Pavia University obtained by different classification approaches.

parameters were used for all experiments, stating that the RF is insensitive to the parameter initialization.

Regarding the BP-based neural network classifier, the network has only one hidden layer, and the number of hidden nodes has been empirically set within the range $((n+K)\times 2)/3 \pm 10$. 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 classes. 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 adjusted the early stopping parameters to achieve reasonable performance goals.

In the case of the ELM, the network also has one single hidden layer. The number of nodes L and the regularization parameter C [147] were traced in the ranges of $L = 400, 600, 800, \ldots, 2000$ and $C = 10^{-3}, 10^{-2}, \ldots, 10^{4}$ using fivefold cross validation.

For the KELM, the RBF kernel is considered. Again, the regularization parameter C and the kernel parameter γ were searched in the ranges $C = 10^{-3}, 10^{-1}, \ldots, 10^{4}$ and $\gamma = 2^{-3}, 2^{-2}, \ldots, 2^{4}$ also using fivefold cross validation. For the 1-D CNN, the important parameters are the kernel size, the number of layers, the number of feature maps, the number of neurons in the hidden layer, and the learning rate. Figure 10 shows the architectures of the deep 1-D 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 to the 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 epochs are 700 for the Indian Pines data set. For the 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.

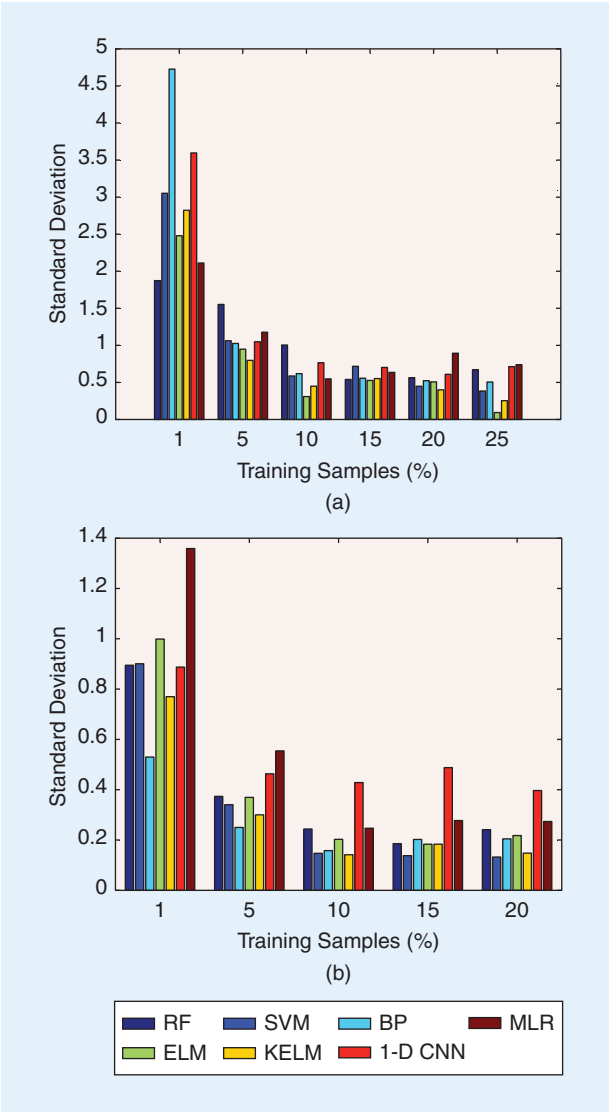


FIGURE 12. Scenario 1: stability. The standard deviation value over ten runs using different percentages of training samples from (a) Indian Pines and (b) Pavia University obtained by different classification approaches.

Figure 11 shows the OA of different approaches (i.e., the average value over ten runs) on different percentages of training samples on Indian Pines and Pavia University. To evaluate the stability of different classifiers on the change of training samples, the standard deviation value over ten runs for each percentage is estimated and shown in Figure 12.

For the Houston hyperspectral data, since the training and test sets were already separated, we performed the classifiers on the standard set of training and test samples. The classification accuracies (i.e., OA, AA, Kappa, and class specific accuracies) are reported in Table 6. The classification maps of this data set are shown in Figure 13.

RESULTS AND DISCUSSION

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

TABLE 6. SCENARIO 2: THE 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

- *SVM versus RF:* Although both classifiers have the same number of hyperparameters to tune (i.e., the 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 the classification accuracy of RFs that can be obtained. RFs are trained faster than a 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 versus BP:* The SVM classifier presents a series of advantages over the BP classifier. The SVM exhibits less computational complexity, even when the kernel trick is used, and usually provides better results when a small number of training samples is available. However, if the 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 BP 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 is 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 versus ELM:* From an optimization point of view, the ELM presents the same optimization cost function as the least squares SVM [148] but much less computational complexity. In general terms, ELM training is tens or hundreds of times faster than a traditional SVM.

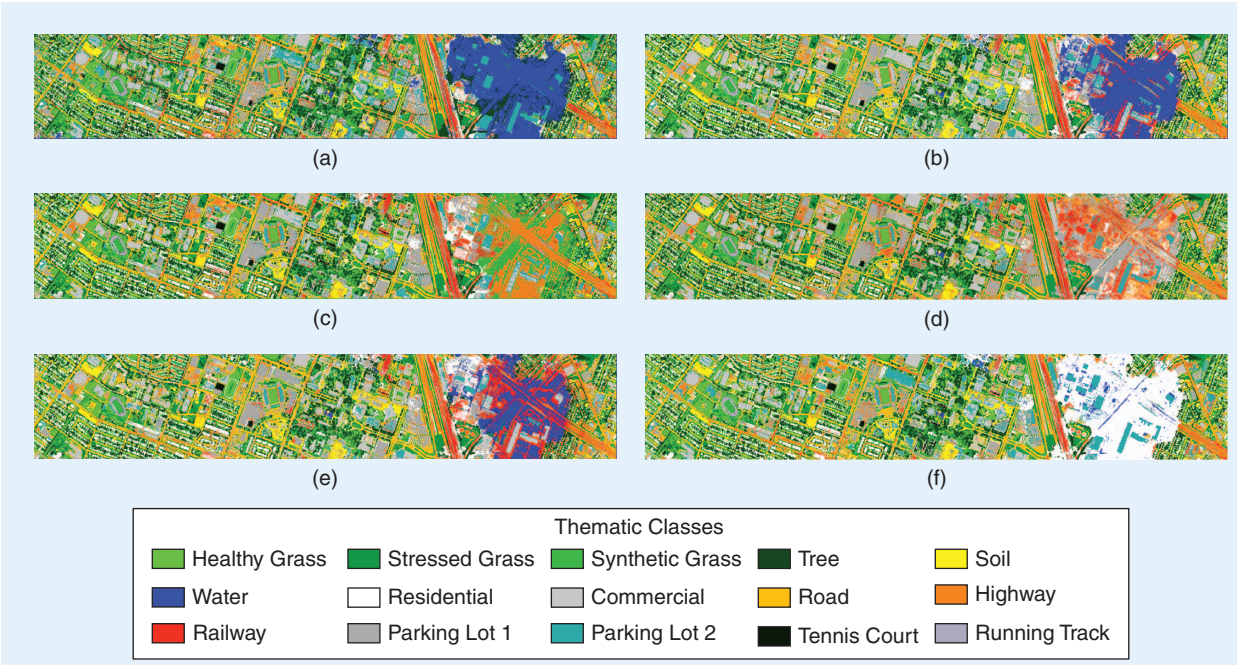


FIGURE 13. Scenario 2: classification maps for Houston data using (a) RF, (b) SVM, (c) BP, (d) KELM, (e) MLR, and (f) 1-D CNN.

Regarding the classification accuracy, it can be seen that the ELM achieves comparable results.

- *SVM versus KELM:* The computational complexity of the SVM is much bigger than the KELM. It can be seen that the KELM slightly outperforms the SVM in terms of classification accuracy. Experimental validation shows that the kernel used in the KELM and SVM is more efficient than the activation function used in ELM.
- *BP versus ELM versus KELM:* In light of the results, it can be seen how the three versions of the SLFN provide competitive results in terms of accuracy. However, it should be noticed that both the ELM and KELM are on the order of hundreds or even thousands of times faster than the BP. Actually, the ELM and KELM have a practical complexity of $O(L^3 + L^2 n + (K+d)Ln)$ and $O(2n^3 + (K+d)n^2)$, respectively [149].
- *SVM versus 1-D CNN:* The main advantage of 2-D and 3-D CNNs is that they use local connections to handle spatial dependencies. In this work, however, the 1-D CNN is taken to have a fair comparison with other spectral approaches. In general, the SVM can obtain higher classification accuracies and work faster than the 1-D CNN, so the use of SVMs over the 1-D CNN is recommended. In terms of central processing unit (CPU) processing time, deep-learning methods are time consuming in the training step. Compared to the SVM, the training time of the 1-D deep CNN is about two or three times longer than the RBF-SVM. On the other hand, the advantage of the deep CNN is that it is extremely fast on the testing stage.
- *MLR (executed via LORSAL) versus other methods:* Some of the MLR advantages are as follows: 1) It converges very fast and is relatively insensitive to parameter settings. In our experiments, we used the same settings for all data sets and received very competitive results in comparison with those obtained by other methods. 2) MLR has a very low computational cost, with a practical complexity of $O(d^2(K-1))$.

DIFFERENT SOLUTIONS DEPEND ON THE COMPLEXITY OF THE ANALYSIS SCENARIO AND ON THE CONSIDERED APPLICATION DOMAIN.

For illustrative purposes, Figure 11 provides a comparison of the different classifiers tested in this work with the Indian Pines and Pavia University scenes (in terms of OA). As shown by Figure 11, 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 Figure 12, which demonstrates how much a classifier is stable with respect to some changes in the available training sets. Furthermore, Table 6 gives detailed information about the 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 by the obtained classification results. This is an important observation, as it is felt that the hyperspectral community has reached a point at which many classifiers are able to provide very high classification

accuracies. However, the competitive differences between existing classifiers are 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.

With the aforementioned observations in mind, we can interpret the results provided in Table 7 in more detail. In this table, one bullet denotes to the worst performance while four bullets denotes the best. 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. The SVM and MLR also show a fair balance between accuracy, automation (obtained with respect to the number of parameters needed to be adjusted), speed (evaluated based on the demanded CPU processing time of different classifiers), and stability, which can be advantageous for applications where a tradeoff between these elements is needed. In contrast, the 1-D CNN does not display enough advantages, either in terms of classification accuracy and stability or speed and automation.

TABLE 7. THE PERFORMANCE EVALUATION OF DIFFERENT SPECTRAL CLASSIFIERS.

TECHNIQUES	ACCURACY	AUTOMATION	SIMPLICITY AND SPEED	STABILITY
RF	•	••••	••••	••
SVM	••••	•••	•••	•••
BP	••••	••	••	••
ELM	••	••	•••	•••
KELM	••••	••	•••	•••
1-D CNN	••	•	•	••
MLR	••••	••••	••••	••

One bullet indicates the worst performance while four bullets indicates the best.

CONCLUSIONS

In this article, 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 various algorithms. The compared techniques include popular approaches such as SVMs, RFs, 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). Instead, different solutions depend on the complexity of the analysis scenario (e.g., the availability of training samples, processing requirements, tuning parameters, and speed of the algorithm) and on the considered application domain. Combined, the insights provided in this article may facilitate the selection of a specific classifier by an end user depending on his/her expectations and/or exploitation goals.

ACKNOWLEDGMENTS

The authors would like to thank the National Center for Airborne Laser Mapping for providing the Houston data set. The ROSIS Pavia University and Indian Pines data and the corresponding reference information were kindly provided by Prof. P. Gamba, University of Pavia, Italy, and Prof. D. Landgrebe, Purdue University, West Lafayette, Indiana, respectively. This research was supported by the Chinese 1000 people program B under project 41090427 and by the Guangdong Provincial Science Foundation under project 42030397. This work was also partly supported by the Alexander von Humboldt Fellowship for postdoctoral researchers.

AUTHOR INFORMATION

Pedram Ghamisi (p.ghamisi@gmail.com) received his B.Sc. degree in civil (survey) engineering from the Tehran South Campus, Azad University, Iran. He received his M.E. degree with first-class honors in remote sensing at K.N. Toosi University of Technology in 2012, and he received his Ph.D. degree in electrical and computer engineering from the University of Iceland, Reykjavik, in 2015. He then worked as a postdoctoral research fellow at the University of Iceland. In 2015, he 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. He has also been working as a researcher at the German Aerospace Center, Remote Sensing Technology Institute, Germany, on deep learning since October 2015. His research interests include machine learning, deep learning, and hyperspectral image analysis. He is a Member of the IEEE.

Javier Plaza (jplaza@unex.es) received his B.S. degree in 2002, his M.Sc. degree in 2004, and his Ph.D. degree in 2008, all in computer engineering. In 2008, he was the recipient of the Outstanding Ph.D. Dissertation Award at the University of Extremadura, Spain, where he is an associate professor in the Department of Technology of Computers and Communications. He has authored or coauthored more than 120 scientific publications. He is currently serving as associate editor of *IEEE Geoscience and Remote Sensing Letters*. He has served as a reviewer for more than 180 papers submitted to more than 30 different journals, and he has served as a proposal evaluator for the Spanish Ministry of Science and Innovation since 2008. He has also served as a proposal evaluator for the Czech Science Foundation

and the Chilean National Science and Technology Commission. He is a Senior Member of the IEEE.

Yushi Chen (chenyushi@hit.edu.cn) received his Ph.D. degree from Harbin Institute of Technology, China, in 2008, where he is currently an associate professor in the School of Electrical and Information Engineering. His research interests include remote sensing data processing and machine learning. He is a Member of the IEEE.

Jun Li (lijun48@mail.sysu.edu.cn) received her B.S. degree in geographic information systems from Hunan Normal University, Changsha, China, in 2004; her M.E. degree in remote sensing from Peking University, Beijing, China, in 2007; and her Ph.D. degree in electrical engineering from the Instituto de Telecomunicaes, Instituto Superior Técnico (IST), Universidade Técnica de Lisboa, Lisbon, Portugal, in 2011. From 2007 to 2011, she was a Marie Curie Research Fellow with the Departamento de Engenharia Electrotécnica e de Computadores and the Instituto de Telecomunicaes, IST, Universidade Técnica de Lisboa, in the framework of the European Doctorate for Signal Processing. Since 2011, she has been a postdoctoral researcher with the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Escuela Politécnica, University of Extremadura, Cáceres, 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. She is a Senior Member of the IEEE.

Antonio Plaza (aplaza@unex.es) received his M.Sc. degree in 1999 and his Ph.D. degree in 2002 from the University of Extremadura, Spain, both in computer engineering. He is head of the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, at the University of Extremadura, and his main research interests lie in hyperspectral data processing and parallel computing of remote sensing data. He has authored more than 500 publications, including 182 Journal Citation Report papers (132 in IEEE journals), 20 book chapters, and over 250 peer-reviewed conference proceeding papers. In 2015, he received the Best Column Award of *IEEE Signal Processing Magazine*. He served as the director of education activities for the IEEE Geoscience and Remote Sensing Society (GRSS) in 2011–2012, and he is currently serving as president of the Spanish Chapter of the IEEE GRSS. He has reviewed more than 500 manuscripts for over 50 journals. He currently serves as the editor-in-chief of *IEEE Transactions on Geoscience and Remote Sensing*. He is a Fellow of the IEEE.

REFERENCES

[1] J. A. Benediktsson and P. Ghamisi, *Spectral-Spatial Classification of Hyperspectral Remote Sensing Images*. Norwood, MA: Artech House, 2015.

[2] P. Ghamisi and J. A. Benediktsson, "Feature selection based on hybridization of genetic algorithm and particle swarm optimization," *IEEE Geosci. 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," *IEEE J. Sel. Topics Appl. Earth Observ. in Remote Sens.*, 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. Pattern. Anal. Mach. Intell.*, 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. Berlin, Germany: Springer-Verlag, 2014.

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

[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," *Comput. Geosci.*, 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," *Intell. Cont. Aut.*, vol. 1, pp. 3604–3607, June 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. Geosci. 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. Geosci. 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. Geosci. Remote Sens.*, vol. 40, no. 3, pp. 593–598, 2002.

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

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

[15] E. J. Wegman, "Hyperdimensional data analysis using parallel coordinates," *J. Amer. Stat. 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. Syst., Man, Cybern. A, Syst., Humans*, vol. 28, no. 1, pp. 39–54, 1998.

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

[18] Y. Qian, F. Yao, and S. Jia. (2009). Band selection for hyperspectral imagery using affinity propagation. *IET Comput. Vis. 3(4)*, p. 213. [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," *Photogrammetric 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. Hoboken, NJ: Wiley, 2002.

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

[22] X. Wang, "Learning from big data with uncertainty editorial," *J. Intell. and Fuzzy Syst.*, 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. Geosci. 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," *Proc. IEEE*, vol. 101, no. 3, pp. 652–675, 2013.

[27] C. Xu, H. Liu, W. Cao, and J. Feng. (2012, Jan.). Multispectral image edge detection via Clifford gradient. *Sci. China Inform. Sci. 55(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. (2013, Apr.). Edge-preserving texture suppression filter based on joint filtering schemes. *IEEE Trans. Multimedia*. *15(3)*, pp. 535–548. [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," *Inform. Sci.*, 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. J. Remote Sens.*, vol. 8, no. 1, pp. 15–29, 1987.

[31] Y. Zhong, Q. Zhu, and L. Zhang. (2015, Nov.). Scene classification based on the multifeature fusion probabilistic topic model for high spatial resolution remote sensing imagery. *IEEE Trans. Geosci. Remote Sens. 53(11)*, pp. 6207–6222. [Online]. Available: http://dx.doi.org/10.1109/TGRS.2015.2435801

[32] D. Böhning, "Multinomial logistic regression algorithm," *Ann. Inst. Statist. Math.*, vol. 44, no. 1, 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. Remote Sens.*, vol. 48, no. 11, 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 Geosci. Remote Sens. Lett.*, vol. 5, no. 2, pp. 280–284, Apr. 2008.

[35] J. S. Borges, J. M. Bioucas-Dias, and A. R. S. Marcal, "Bayesian hyperspectral image segmentation with discriminative class learning," *IEEE Trans. Geosci. Remote Sens.*, 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, Portugal, 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. Remote Sens.*, vol. 49, no. 19, pp. 3947–3960, 2011.

[38] J. Li, J. Bioucas-Dias, and A. Plaza, "Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields," *IEEE Trans. Geosci. 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 Trans. Image Process.*, 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 Trans. Geosci. Remote Sens.*, vol. 51, no. 4, pp. 2276–2291, Apr. 2013.

[41] P. Zhong and R. Wang, "Jointly learning the hybrid CRF and MLR model for simultaneous denoising and classification of hyperspectral imagery," *IEEE Trans. Neural Netw. Learn. Syst.*, 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 Geosci. Remote Sens. Lett.*, 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 Trans. Geosci. Remote Sens.*, 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 Trans. Geosci. Remote Sens.*, 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 Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1490–1503, Mar. 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 J. Sel. Topics Signal Process.*, 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 J. Sel. Topics Appl. Earth Observ. in Remote Sens.*, vol. 9, no. 2, pp. 625–639, Feb. 2016.

[48] J. Zhao, Y. Zhong, H. Shu, and L. Zhang. (2016, Sept.). High-resolution image classification integrating spectral-spatial-location cues by conditional random fields. *IEEE Trans. Image Process. 25(9)*, pp. 4033–4045. [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. Remote Sens.*, vol. 51, no. 9, pp. 4816–4829, Feb. 2013.

[50] Y. Zhang and S. Prasad, "Locality preserving composite kernel feature extraction for multi-source geospatial image analysis," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 3, pp. 1385–1392, Mar. 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 Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1592–1606, Mar. 2015.

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

[53] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: 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. J. Remote Sens.*, vol. 14, no. 15, pp. 2883–2903, 1993.

[55] H. Yang, "A backpropagation neural network for mineralogical mapping from AVIRIS data," *Int. J. 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, School of Elect. Eng., Purdue Univ., West Lafayette, IN, 1990.

[57] J. A. Richards, "Analysis of remotely sensed data: The formative decades and the future," *IEEE Trans. Geosci. 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 Trans. Geosci. Remote Sens.*, 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 J. 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 Trans. Geosci. Remote Sens.*, 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 Trans. Geosci. 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 Trans. Geosci. 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 Netw.*, vol. 6, no. 4, pp. 525–533, 1993.

[65] M. T. Hagan and M. Menhaj, "Training feed-forward networks with the Marquardt algorithm," *IEEE Trans. Neural Netw.*, 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, no. 1–3, pp. 489–501, Dec. 2006.

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

[68] J. Tang, C. Deng, and G. B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 809–21, Apr. 2016.

[69] G. B. Huang and C. K. Siew, "Extreme learning machine: RBF network case," in *Proc. 8th Control, Automation, Robotics and Vision Conf., (ICARCV 2004)*, 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, no. 3, pp. 376–390, Sept. 2014.

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

[72] A. B. Santos, A. Araujo, and D. Menotti, "Combining multiple classification methods for hyperspectral data interpretation," *IEEE J. Sel. Topics Appl. Earth Observ. 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," *J. Appl. Remote Sens.*, vol. 9, no. 1, pp. 097296, 2015.

[74] A. Samat, P. Du, S. Liu, and L. Cheng, "E2LMs: Ensemble extreme learning machines for hyperspectral image classfication," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 4, pp. 1060–1069, 2014.

[75] V. N. Vapnik, *Statistical Learning Theory*. New York, NY: Wiley, 1998.

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

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

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

[79] L. Gómez-Chova, G. Camps-Valls, J. Muoz-Mar, and J. Calpe, "Semisupervised image classification with Laplacian support vector machines," *IEEE Geosci. Remote Sens. Lett.*, 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 Geosci. Remote Sens. Lett.*, vol. 46, no. 6, pp. 1804–1811, 2008.

[81] D. Tuia and G. Camps-Valls, "Semisupervised remote sensing image classification with cluster kernels," *IEEE Geosci. Remote Sens. Lett.*, 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. J. 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. Geosci. Remote Sens.*, vol. 51, no. 9, pp. 4816–4829, 2013.

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

[85] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 21, no. 3, pp. 660–674, 1991.

[86] L. Breiman, J. H. Friedman, R. Olshen, and C. J. Stone, *Classification and Regression Tree*. London, U.K.: Chapman & Hall, 1984.

[87] 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.

[88] 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.

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

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

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

[92] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 1, pp. 123–140, 1994.

[93] L. Breiman, "Random forests," *Mach. Learn*, vol. 45, no. 1, pp. 5–32, 2001.

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

[95] B. Waske, J. A. Benediktsson, K. Arnason, and J. R. Sveinsson, "Mapping of hyperspectral AVIRIS data using machine-learning algorithms," *Canadian J. Remote Sens.*, vol. 35, suppl. 1, pp. 106–116, 2009.

[96] Z. Zhi-Hua, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL: CRC, 2012.

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

[98] Z. Lai, W. K. Wong, Y. Xu, C. Zhao, and M. Sun. (2014, Oct.). Sparse alignment for robust tensor learning. *IEEE Trans. Neural Netw. Learn. Syst.* 25*(10)*, pp. 1779–1792. [Online]. Available: http://dx.doi.org/10.1109/TNNLS.2013.2295717

[99] 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 Sens.*, vol. 49, no. 11, pp. 4263–4281, Dec. 2011.

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

[101] 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.

[102] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, 2014.

[103] C. Tao, H. Pan, Y. Li, and Z. Zou, "Unsupervised spectral-spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2438–2442, 2015.

[104] Y. Chen, X. Zhao, and X. Jia, "Spectral-spatial classification of hyperspectral data based on deep belief network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2381–2292, 2015.

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

[106] J. A. Richards and X. Jia, *Remote Sensing Digital Image Analysis*, 4th ed. Berlin, Germany: Springer-Verlag, 2006.

[107] 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 cost-based approach," *Int. J. Remote Sens.*, vol. 20, no. 8, pp. 1461–1486, 1999.

[108] W. D. Hudson and C. V. Ramm, "Correct formulation of the kappa coefficient of agreement," *Photogrammetric Eng. Remote Sens.*, vol. 53, pp. 21–422, Aug. 1987.

[109] R. G. Congalton, "A review of assessing the accuracy of classification of remotely sensed data," *Remote Sens. Env.*, vol. 37, no. 1, pp. 35–46, July 1991.

[110] L. L. F. Janssen and F. J. M. Vanderwel, "Accuracy assessment of satellite derived land-cover data: A review," *Photogrammetric Eng. Remote Sens.*, vol. 60, no. 4, pp. 419–426, Apr. 1994.

[111] G. M. Foody, "Status of land cover classification accuracy assessment," *Remote Sens. Env.*, vol. 80, no. 1, pp. 185–201, Apr. 2002.

[112] 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, suppl. 1, pp. 110–122, Sept. 2009.

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

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

[115] 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.

[116] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Netw.*, vol. 4, pp. 251–257, 1991.

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

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

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

[120] G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing: A review," *ISPRS J. Photogrammetry Remote Sens.*, vol. 66, no. 3, pp. 247–259, 2011.

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

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

[123] 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.

[124] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Mach. Learn.*, vol. 46, no. 1, pp. 131–159, 2002.

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

[126] S. S. Keerthi and C. Lin, "Asymptotic behaviors of support vector machines with Gaussian kernel," *Neur. Comp*, vol. 15, no. 7, pp. 1667–1689, 2003.

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

[128] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

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

[130] 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. Boca Raton, FL.: CRC, 2007, pp. 327–344.

[131] B. Waske, 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. New York, NY: CRC, 2012, pp. 363–374.

[132] 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 J. Photogrammetry Remote Sens.*, vol. 67, pp. 93–104, Jan. 2012.

[133] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. New York, NY: Springer-Verlag, 2008.

[134] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Reading, MA: Addison-Wesley, 2009.

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

[136] 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.

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

[138] S. R. Joelsson, J. A. Benediktsson, and J. R. Sveinsson, "Random forest classifiers for hyperspectral data," in *Proc. IEEE Int. Geoscience Remote Sensing Symp., (IGARSS 05)*, 2005, pp. 25–29.

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

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

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

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

[143] 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. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

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

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

[146] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 6, pp. 1351–1362, June 2005.

[147] G. B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, 2012.

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

[149] A. Iosifidis, A. Tefas, and I. Pitas, "On the kernel extreme learning machine classifier," *Pattern Recog. Lett.*, vol. 54, pp. 11–17, Mar. 2015.

GRS