Deep Learning for Remote Sensing Data

A technical tutorial on the state of the art

IMAGE LICENSED BY INGRAM PUBLISHING

Advances in Machine Learning for Remote Sensing and Geosciences

LIANGPEI ZHANG, LEFEI ZHANG, AND BO DU

Deep-learning (DL) algorithms, which learn the representative and discriminative features in a hierarchical manner from the data, have recently become a hotspot in the machine-learning area and have been introduced into the geoscience and remote sensing (RS) community for RS big data analysis. Considering the low-level features (e.g., spectral and texture) as the bottom level, the output feature representation from the top level of the network can be directly fed into a subsequent classifier for pixel-based classification. As a matter of fact, by carefully addressing the practical demands in RS applications and designing the input–output levels of the whole network, we have found that DL is actually everywhere in RS data analysis: from the traditional topics of image preprocessing, pixel-based classification, and target recognition, to the recent challenging tasks of high-level semantic feature extraction and RS scene understanding.

In this technical tutorial, a general framework of DL for RS data is provided, and the state-of-the-art DL methods in RS are regarded as special cases of input–output data combined with various deep networks and tuning tricks. Although extensive experimental results confirm the excellent performance of the DL-based algorithms in RS big data analysis, even more exciting prospects can be expected for DL in RS. Key bottlenecks and potential directions are also

Digital Object Identifier 10.1109/MGRS.2016.2540798
Date of publication: 13 June 2016

indicated in this article, guiding further research into DL for RS data.

ADVANTAGES OF REMOTE SENSING METHODS

RS techniques have opened a door to helping people widen their ability to understand the earth [1], [2]. In fact, RS techniques are becoming more and more important in data-collection tasks. Information technology companies depend on RS to update their location-based services [3], [4]. Google Earth employs high-resolution (HR) RS images to provide vivid pictures of the earth's surface. Governments have also utilized RS for a variety of public services, from weather reporting to traffic monitoring [5]–[7]. Nowadays, one cannot imagine a life without RS. Recent years have even witnessed a boom in RS satellites, providing for the first time an extremely large number of geographical images of nearly every corner of the earth's surface [8]. Data warehouses of RS images are increasing daily, including images with different spectral and spatial resolutions [9], [10].

How can we extract valuable information from the various kinds of RS data? How should we deal with the ever-increasing data types and volume? The traditional approaches exploit features from RS images with which information-extraction models can be constructed [11]. Handcrafted features have proved effective and can represent a variety of spectral, textural, and geometrical attributes of the images [12], [13]. However, since these features cannot easily consider the details of real data, it is impossible for them to achieve an optimal balance between discriminability and robustness. When facing the big data of RS images, the situation is even worse, since the imaging circumstances vary so greatly that images can change a lot in a short interval. Thanks to DL theory [14], which provides an alternative way to automatically learn fruitful features from the training set, unsupervised feature learning from very large raw-image data sets has become possible [15]. Actually, DL has proven to be a new and exciting tool that could be the next trend in the development of RS image processing.

RS images, despite the spectral and spatial resolution, are reflections of the land surface [16], with an important property being their ability to record multiple-scale information within an area. According to the type of information that is desired, pixel-based, object-based, or structure-based features can be extracted. However, an effective and universal approach has not yet been reported to optimally fuse these features, due to the subtle relationships between the data. In contrast, DL can represent and organize multiple levels of information to express complex relationships between data [17]. In fact, DL techniques can map different levels of abstractions from the images and combine them from low level to high level [18]. Consider scene recognition as an example, where, with the help of DL, the scenes can be represented as a unitary transformation by exploiting the variations in the local spatial arrangements and structural patterns captured by the low-level features, where no segmentation stage or individual object extraction stage is needed.

Despite its great potential, DL cannot be directly used in many RS tasks, with one obstacle being the large numbers of bands. Some RS images, especially hyperspectral ones, contain hundreds of bands that can cause a small patch to be a really large data cube, which corresponds to a large number of neurons in a pretrained network [19], [20]. In addition to the visual geometrical patterns within each band, the spectral curve vectors across bands are also important information. However, how to utilize this information still requires further research. Problems still exist in the high-spatial-resolution RS images, which have only green, red, and blue channels, the same as the benchmark data sets for DL. In practice, very few labeled samples are available, which may make a pretrained network difficult to construct. Furthermore, images acquired by different sensors present large differences. How to transfer the pretrained network to other images is still unknown.

In this article, we survey the recent developments in DL for the RS field and provide a technique tutorial on the design of DL-based methods for optical RS data. Although there are also several advanced techniques for DL for synthetic aperture radar images [21]–[26] and light detection and ranging (LiDAR) point clouds data [27], they share the similar basic DL ideas of the data analysis model.

THE GENERAL FRAMEWORK

Despite the complex hierarchical structures, all of the DL-based methods can be fused into a general framework. Figure 1 illustrates a general framework of DL for RS data analysis. The flowchart includes three main components, the prepared input data, the core deep networks, and the expected output data. In practice, the input–output data pairs are dependent on the particular application. For example, for RS image pan sharpening, they are the HR and low-resolution (LR) image patches from the panchromatic (PAN) images [28]; for pixel-based classification, they are the spectral–spatial features and their feature representations (unsupervised version) or label information (supervised version) [29]; while, for tasks of target recognition [30] and scene understanding [31], the inputs are the features extracted from the object proposals, as well as the raw pixel digital numbers from the HR images and RS image databases respectively, and the output data are always the same as in the application of pixel-based classification, as described previously.

When the input–output data pairs have been properly defined, the intrinsic and natural relationship between the input and output data is then constructed by a deep architecture composed of multiple levels of nonlinear operations, where each level is modeled by a shallow module such as an autoencoder (AE) or a sparse coding algorithm. It should be noted that, if a sufficient training sample set is available, such a deep network turns out to be a supervised

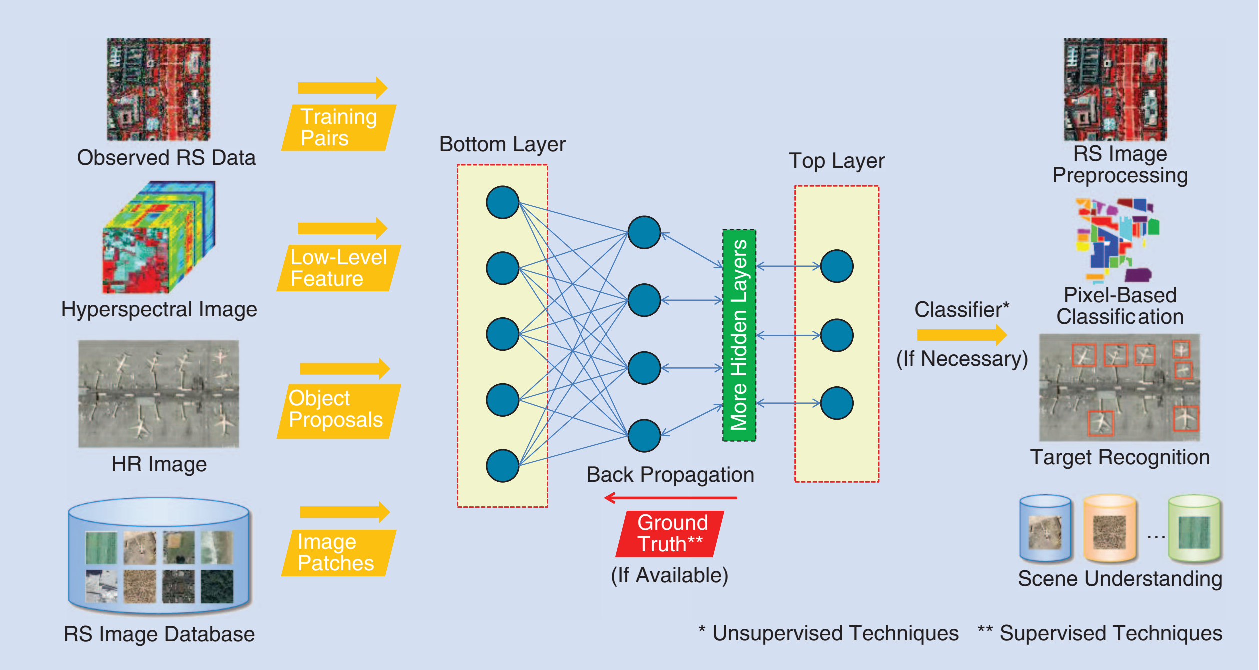

FIGURE 1. A general framework of DL for RS data analysis.

approach. It can be further fine-tuned by the use of the label information, and the top-layer output of the network is the label information rather than the abstract feature representation learned by an unsupervised deep network. When the core deep network has been well trained, it can be employed to predict the expected output data of a given test sample. Along with the general framework in Figure 1, we describe a few basic algorithms in the deep network-construction tutorial in the following section, and we then review the representative techniques in DL for RS data analysis from four perspectives: 1) RS image preprocessing, 2) pixel-based classification, 3) target recognition, and 4) scene understanding.

BASIC ALGORITHMS IN DEEP LEARNING

In recent years, the various DL architectures have thrived [32] and have been applied in fields such as audio recognition [33], natural language processing [34], and many classification tasks [35], [36], where they have usually outperformed the traditional methods. The motivation for such an idea is inspired by the fact that the mammal brain is organized in a deep architecture, with a given input percept represented at multiple levels of abstraction, for the primate visual system in particular [37]. Inspired by the architectural depth of the human brain, DL researchers have developed novel deep architectures as an alternative to shallow architectures. Deep belief networks (DBNs) [38] are a major breakthrough in DL research and train one layer at a time in an unsupervised manner by restricted Boltzmann machines (RBMs) [39]. A short while later, a number of AE-based algorithms were proposed that also train the intermediate levels of representation locally at each level (i.e., the AE and its variants, such as the sparse AE and the denoising AE [40], [41]). Unlike AEs, the sparse coding algorithms [42] generate sparse representations from the data themselves from a different perspective by learning an overcomplete dictionary via self-decomposition. In addition, as the most representative supervised DL model, convolutional neural networks (CNNs) [43] have outperformed most algorithms in visual recognition. The deep structure of CNNs allows the model to learn highly abstract feature detectors and to map the input features into representations that can clearly boost the performance of the subsequent classifiers. Furthermore, there are many optional techniques that can be used to train the DL architecture shown in Figure 1. In this review, we only provide a brief introduction to the following four typical models that have already been used in the RS community and can be embedded into the general framework to achieve the particular application. More detailed information regarding the DL algorithms in the machine-learning community can be found in [14] and [44].

CONVOLUTIONAL NEURAL NETWORKS

The CNN is a trainable multilayer architecture composed of multiple feature-extraction stages. Each stage consists of three layers: 1) a convolutional layer, 2) a nonlinearity layer, and 3) a pooling layer. The architecture of a CNN is designed to take advantage of the two-dimensional structure of the input image. A typical CNN is composed of one, two, or three such feature-extraction stages, followed by one or more traditional, fully connected layers and a final classifier layer. Each layer type is described in the following sections.

CONVOLUTIONAL LAYER

The input to the convolutional layer is a three-dimensional array with r two-dimensional feature maps of size $m \times n$. Each component is denoted as $x^i_{m,n}$, and each feature map is denoted as x^i. The output is also a three-dimensional array $m_1 \times n_1 \times k$, composed of k feature maps of size $m_1 \times n_1$. The convolutional layer has k trainable filters of size $l \times l \times q$, also called *the filter bank W*, which connects the input feature map to the output feature map. The convolutional layer computes the output feature map $z^s = \sum_{t=1}^{q} W_i^s * x^i + b_s$, where * is a two-dimensional discrete convolution operator and b is a trainable bias parameter.

NONLINEARITY LAYER

In the traditional CNN, this layer simply consists of a pointwise nonlinearity function applied to each component in a feature map. The nonlinearity layer computes the output feature map $a^s = f(z^s)$, as $f(\cdot)$ is commonly chosen to be a rectified linear unit (ReLU) $f(x) = \max(0, x)$.

POOLING LAYER

The pooling layer involves executing a max operation over the activations within a small spatial region G of each feature map: $p^s_G = \max\limits_{i \in G} a^s_i$. To be more precise, the pooling layer can be thought of as consisting of a grid of pooling units spaced s pixels apart, each summarizing a small spatial region of size $p * p$ centered at the location of the pooling unit. After the multiple feature-extraction stages, the entire network is trained with back propagation of a supervised loss function such as the classic least-squares output, and the target output y is represented as a 1–*of*–K vector, where K is the number of output and L is the number of layers:

$$J(\theta) = \sum_{i=1}^{N}\left(\frac{1}{2}\| h(x_i, \theta) - y \|^2\right) + \lambda \sum_{l}^{L} \text{sum}\left(\| \theta^{(l)} \|^2\right), \qquad (1)$$

where l indexes the layer number. Our goal is to minimize $J(\theta)$ as a function of θ. To train the CNN, we can apply stochastic gradient descent with back propagation to optimize the function.

CNNs have recently become a popular DL method and have achieved great success in large-scale visual recognition, which has become possible due to the large public image repositories, such as ImageNet [36]. In the RS community, there are also some recent works on CNN-based RS image pixel classification [45]–[47], target recognition [48], [49], and scene understanding [50].

AUTOENCODERS

An AE is a symmetrical neural network that is used to learn the features from a data set in an unsupervised manner by minimizing the reconstruction error between the input data at the encoding layer and its reconstruction at the decoding layer. During the encoding step, an input vector $x^i \in \mathbb{R}^N$ is processed by applying a linear mapping and a nonlinear activation function to the network:

$$\alpha^i = f(x) = g(W_1 x^i + b_1), \qquad (2)$$

where $W_1 \in \mathbb{R}^{K \times N}$ is a weight matrix with K features, $b_1 \in \mathbb{R}^K$ is the encoding bias, and $g(x)$ is the logistic sigmoid function $(1 + \exp(-x))^{-1}$. We decode a vector using a separate linear decoding matrix:

$$z^i = W_2{}^T \cdot \alpha^i + b_2, \qquad (3)$$

where $W_2 \in \mathbb{R}^{K \times N}$ is a weight matrix and $b_2 \in \mathbb{R}^N$ is the decoding bias. Feature extractors in the data set are learned by minimizing the cost function, and the first term in the reconstruction is the error term. The second term is a regularization term (also called a *weight decay term* in a neural network):

$$J(X, Z) = \frac{1}{2}\sum_{i=1}^{m} \| x^i - z^i \|^2 + \frac{\lambda}{2}\| W \|^2, \qquad (4)$$

where X and Z are the training and reconstructed data, respectively.

We recall that α denotes the activation of hidden units in the AE. Thus, when the network is provided with a specific input $x^i \in X^{N \times m}$, let $\hat{\rho} = \frac{1}{m}\sum_{i=1}^{m}[\alpha^i]$ be the average activation of α averaged over the training set. We want to approximately enforce the constraint $\hat{\rho} = \rho$, where ρ is the sparsity parameter, which is typically a small value close to zero. In other words, we want the average activation of each hidden neuron $\hat{\rho}$ to be close to zero. To satisfy this constraint, the hidden units activations must be mostly inactive and close to zero so that most of the neurons are inactive. To achieve this, the objective in the sparse AE learning is to minimize the reconstruction error with a sparsity constraint, i.e., a sparse AE:

$$J(X, Z) + \beta \sum_{j=1}^{K} KL(\rho \| \hat{\rho}), \qquad (5)$$

where β is the weight of the sparsity penalty, K is the number of features in the weight matrix, and $KL(\cdot)$ is the Kullback-Leibler divergence given by

$$KL(\rho \| \hat{\rho}) = \rho \log\frac{\rho}{\hat{\rho}} + (1 - \rho)\log\frac{1 - \rho}{1 - \hat{\rho}}. \qquad (6)$$

This penalty function has the property that $KL(\rho \| \hat{\rho}) = 0$ if $\hat{\rho} = \rho$. Otherwise, it increases monotonically as $\hat{\rho}$ diverges from ρ, which acts as the sparsity constraint. An AE can be directly employed as a feature extractor for RS data analysis [51], and it has been more frequently stacked into the AEs for DL from RS data [52]–[54].

RESTRICTED BOLTZMANN MACHINES

An RBM is commonly used as a layer-wise training model in the construction of a DBN. It is a two-layer network, presenting a particular type of Markov random field with visible units $v = \{0,1\}^D$ and hidden units $h = \{0,1\}^F$. A joint configuration of the units has an energy given by

$$E(v,h:\theta) = -\sum_{i=1}^{D} b_i v_i - \sum_{j=1}^{F} a_j h_j - \sum_{i=1}^{D}\sum_{j=1}^{F} w_i v_i h_j, \tag{7}$$

where $\theta = \{b_i, a_j, w_{ij}\}$ and 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 joint distribution over the units is defined by

$$P(v,h:\theta) = \frac{1}{Z(\theta)} e^{-E(v,h:\theta)} \tag{8}$$

$$Z(\theta) = \sum_v \sum_h E(v,h:\theta), \tag{9}$$

where $Z(\theta)$ is the normalizing constant. The network assigns a probability to every input vector via the energy function. The probability of the training vector can be raised by adjustment to lower the energy, as given in (7). The conditional distributions of hidden unit h and input vector v are given by the logistic function

$$p(h_j = 1|v) = g\left(\sum_{i=1}^{D} W_{ij} v_i + a_j\right), \tag{10}$$

$$p(v_j = 1|h) = g\left(\sum_{j=1}^{F} W_{ij} h_i + b_j\right), \tag{11}$$

$$g(x) = \frac{1}{1 + e^{(-x)}}. \tag{12}$$

Once the states of hidden units are chosen, the input data can be reconstructed by setting each v_i to 1 with the probability in (11). The hidden units' states are then updated to represent the features of the reconstruction. The learning of W is done through a method called *contrastive divergence (CD)*. The DBN has been applied to the RS image spatial–spectral classification and shows superior performance compared to the conventional feature dimensionality-reduction methods, such as principal component analysis (PCA), and classifiers, such as support vector machines (SVMs) [55], [29]. In recent years, it has also been successfully proposed for object recognition [56] and scene classification [57].

SPARSE CODING

Sparse coding is a type of unsupervised method for learning sets of overcomplete bases to represent data efficiently to find a set of basis vectors ϕ_i such that we can represent an input vector x as a linear combination of these basis vectors:

$$x = \sum_{i=1}^{k} a_i \phi_i. \tag{13}$$

While techniques such as PCA allow us to learn a complete set of basis vectors efficiently, we wish to learn an overcomplete set of basis vectors to represent the input vectors x. The advantage of having an overcomplete basis set is that our basis vectors are better able to capture structures and patterns inherent in the input data. However, with an overcomplete basis set, the coefficients a_i are no longer uniquely determined by the input vector x. Therefore, in sparse coding, we introduce the additional criterion of sparsity to resolve the degeneracy introduced by the overcompleteness.

We define the sparse coding cost function on a set of m input vectors as

$$\min_{a,\phi} \sum_{j=1}^{m} \left\| x^j - \sum_{i=1}^{k} a_i^j \phi_i \right\|_2^2 + \lambda \sum_{i=1}^{k} S(a_i^j), \tag{14}$$

where $S(\cdot)$ is a sparsity cost function that penalizes a_i for being far from zero. We can interpret the first term of the sparse coding objective as a reconstruction term that tries to force the algorithm to provide a good representation of x, and the second term can be defined as a sparsity penalty that forces our representation of x to be sparse.

A large number of sparse coding methods have been proposed. Notably, for RS scene classification, Cheriyadat [58] introduces a variant of sparse coding that combines local scale-invariant feature transform (SIFT)-based feature descriptors to generate a new sparse representation, while, in [59], the sparse coding is used to reduce the potential redundant information in the feature representation. In addition, as a computationally efficient unsupervised feature-learning technique, k-means clustering has also been played as a single-layer feature extractor for RS scene classification [60]–[62] and achieves state-of-the-art performance.

DEEP LEARNING FOR REMOTE SENSING DATA

The "Basic Algorithms in Deep Learning" section discussed some of the basic elements used in constructing a DL architecture as well as the general framework. In practice, the mathematical problems of the various RS data analysis techniques can be regarded as special cases of input–output data combined with a particular DL network based on the aforementioned algorithms. In this section, we provide a tutorial on DL for RS data from four perspectives: 1) image preprocessing, 2) pixel-based classification, 3) target recognition, and 4) scene understanding.

REMOTE SENSING IMAGE PREPROCESSING

In practice, the observed RS images are not always as satisfactory as we demand due to many factors, including the limitations of the sensors and the influence of the atmosphere. Therefore, there is a need for RS image preprocessing to enhance the image quality before the subsequent classification and recognition tasks. According to the related RS literature, most of the existing methods in RS image denoising, deblurring, superresolution, and pan sharpening are based on the standard image-processing techniques in the signal processing society, while there are very few machine-learning-based techniques. In fact, if we can effectively model the intrinsic correlation between the input (observed data) and output (ideal data) by a set of training samples, then the observed RS image could be enhanced by the same model. According to the basic techniques in the previous section, such an intrinsic correlation can be effectively explored by DL. In this tutorial, we consider two typical applications as the case studies, i.e., RS image restoration and pan sharpening, to show the state-of-the-art DL achievements in RS image preprocessing.

Followed by the general framework of DL-based RS data preprocessing that we introduced in the "General Framework" section, the input data of the framework are usually the whole original image or the local image patches. A specific deep network is then constructed, such as a deconvolution network [63] or a sparse denoising AE [28]. After that, the observed RS image is recovered by the learned DL model per spectral channel or per patch.

RESTORATION AND DENOISING

For RS image restoration and denoising, the original image is the input to a certain network that is trained with the clean image to obtain the restored and denoised image. For instance, Zhang et al. utilized the $L_{1/2}$-regularized deconvolution network for the restoration and denoising of RS images [63], which is an improved version of the L_1-regularized deconvolution network. The classical deconvolution network model is based on the convolutional decomposition of images under an L_1 regularization, which is a sparse constraint term. In the experiments undertaken in this study, adopting the $L_{1/2}$ regularization in the deep network gave sparser solutions than their L_1 counterpart and has achieved satisfactory results.

PAN SHARPENING

By introducing deep neural networks, Huang et al. proposed a new pan-sharpening method for RS image preprocessing [28] that used a stacked modified sparse denoising AE (S-MSDA) to train the relationship between HR and LR image patches. Similar to the structure of the sparse AE, S-MSDA is constructed by stacking a series of MSDAs. The MSDA is a modified version of the sparse denoising AE (SDA), which is obtained by combining sparsity and a denoising AE together. The SDA is trained to reconstruct a clean, repaired input from the corresponding corrupted version [64]. Meanwhile, the modified version (i.e., the MSDA) takes the HR image patches and the corresponding LR image patches as clean data and corrupted data, respectively, and represents the relationship between them. There is a key hypothesis that the HR and LR multispectral (MS) image patches have the same relationship as that between the HR and LR PAN image patches; thus, it is a learning-based method that requires a set of HR–LR image pairs for training. Since the HR PAN is already available, we have designed an approach to obtain its corresponding LR PAN. Therefore, we can use the fully trained DL network to reconstruct the HR MS image from the observed LR MS image. The experimental results demonstrated that the DL-based pan sharpening method outperforms the other traditional and state-of-the-art methods. The aforementioned methods are just two aspects of DL-based RS image preprocessing. In fact, we can use the general framework to generate more DL algorithms for RS image-quality improvement for different applications.

PIXEL-BASED CLASSIFICATION

Pixel-based classification is one of the most popular topics in the geoscience and RS community. Significant progress has been achieved in recent years, e.g., in the aspects of handcrafted feature description [65]–[68], discriminative feature learning [13], [69], [70], and powerful classifier designing [71], [72]. However, from the DL point of view, most of the existing methods can extract only shallow features of the original data (the classification step can also be treated as the top level of the network), which is not robust enough for the classification task. DL-based pixel classification for RS images involves constructing a DL architecture for the pixel-wise data representation and classification. By adopting DL techniques, it is possible to extract more robust and abstract feature representations and thus improve the classification accuracy.

The scheme of DL for RS image pixel-based classification consists of three main steps: 1) data input, 2) hierarchical DL model training, and 3) classification. A general flow chart of this scheme is shown in Figure 2. In the first steps, the input vector could be the spectral feature, the spatial feature, or the spectral–spatial feature, as we will discuss later. Then, for the hidden layers, a deep network structure is designed to learn the expected feature representation of the input data. In the related literature, both the supervised DL structures (e.g., the CNN [45]) and the unsupervised DL structures (e.g., the AEs [73]–[75], DBNs [29], [76], and other self-defined neurons in each layer [77]) are employed. The third step is the classification, which involves classification by utilizing the learned feature in the second step (the top layer of the DL network). In general, there are two main styles of classifiers: 1) the hard classifiers, such as SVMs, which directly output an integer number as the class label of each sample [76], and 2) the soft classifiers, such as logistic regression, which can simultaneously fine-tune the whole pretrained network and predict the class label in a probability distribution manner [29], [73], [74], [78].

SPECTRAL FEATURE CLASSIFICATION

The spectral information usually contains abundant discriminative information. A frequently used and direct approach for RS image classification is spectral feature-based classification, i.e., image classification with only the spectral feature. Most of the existing common approaches for RS image classification are shallow in their architecture, such as SVMs and k-nearest neighbor (KNN). Instead, DL adopts a deep architecture to deal with the complicated relationships between the original data and the specific class label.

For spectral feature classification, the spectral feature of the original image data is directly deployed as the input vector. The input pixel vector is trained in the network part to obtain the robust deep feature representation, which is used as the input for the subsequent classification step. The selected deep networks could be the deep CNN [45]

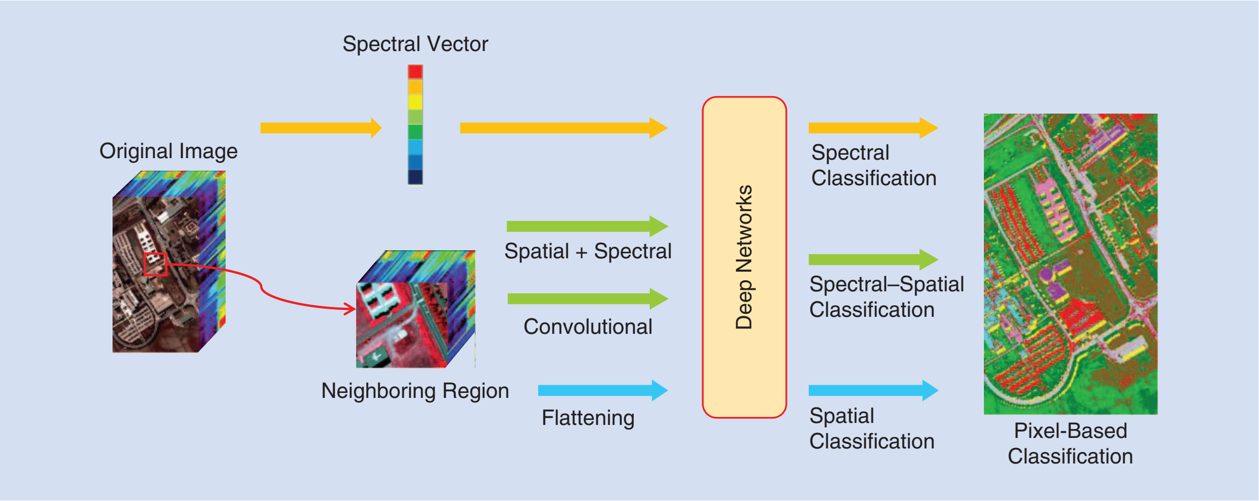

FIGURE 2. A general framework for the pixel classification of RS images using DL methods. Inputs of the DL networks can be divided into three categories: the spectral feature, the spatial feature, and the spectral–spatial feature.

and a stack of the AE [73], [75], [79], [80]. In particular, Lin et al. adopted an AE plus SVMs and a stacked AE plus logistic regression as the network structure and classification layer to perform the classification task with shallow and deep representation, respectively. It is worth noting that, due to the deeper network structure and the fine-tuning step, the deep spectral representation achieved a better performance than the shallow spectral representation [73].

CLASSIFICATION WITH SPATIAL INFORMATION

Land covers are known to be continuous in the spatial domain, and adjacent pixels in an RS image are likely to belong to the same class. As indicated in many spectral–spatial classification studies, the use of the spatial feature can significantly improve the classification accuracy [81]–[83]. However, traditional methods cannot extract robust deep feature representations due to their shallow properties. To address this problem, a number of DL-based feature-learning methods have been proposed to find a new way of extracting the deep spectral–spatial representation for classification [84].

For a certain pixel in the original RS image, it is natural to consider its neighboring pixels to extract the spatial feature representation. However, due to the hundreds of channels along the spectral dimension of a hyperspectral image, the region-stacked feature vector will result in too large an input dimension. As a result, it is necessary to reduce the spectral feature dimensionality before the spatial feature representation. PCA is commonly executed in the first step to map the data to an acceptable scale with a low information loss. Then, in the second step, the spatial information is collected by the use of a $w \times w$ (w is the size of window) neighboring region of every certain pixel in the original image [85]. After that, the spatial data is straightened into a one-dimensional vector to be fed into a DL network. Lin et al. [73] and Chen at al. [74] adopted the stacked AE as the deep network structure. When the abstract feature has been learned, the final classification step is carried out, which is similar to the spectral classification scheme.

When considering a joint spectral and spatial feature-extraction and classification scheme, there are two main strategies to achieve this goal under the framework summarized in Figure 2. Straightforwardly, differing from the spectral–spatial classification scheme, the spectral and initial spatial features are combined together into a vector as the input of the DL network in a joint framework, as presented in the works [29], [53]–[55], [73], and [74]. The preferred deep networks in these papers are SAEs and DBNs, respectively. Then, by the learned deep network, the joint spectral–spatial feature representation of each test sample is obtained for the subsequent classification task, which is the same as the spectral–spatial classification scheme described previously. The other approach is to address the spectral and spatial information of a certain pixel by a convolutional deep network, such as the CNNs [46], [47], [76], the convolutional AEs [78], and a particular defined deep network [78]. Moreover, there are a few hierarchical learning frameworks that take each step of operation (e.g., feature extraction, classification, and postprocessing) as a single layer of the deep network [86]–[90]. We also regard them as the spectral–spatial DL techniques in this tutorial article.

TARGET RECOGNITION

Target recognition in large HR RS images, such as ship, aircraft, and vehicle detection, is a challenging task due to the small size and large numbers of targets and the complex neighboring environments, which can cause the recognition algorithms to mistake irrelevant ground objects for target objects. However, objects in natural images are relatively large, and the environments in the local fields are not that complex compared to RS images, making the targets easier to recognize. This is one of the main differences between detecting RS targets and natural targets. Although many studies have been undertaken, we are still lacking an efficient location method and robust classifier for target recognition in complex environments. In the literature, Cai et al. [91] showed how difficult it is to segment aircraft from the background, and Chen et al. [30], [92] made great efforts in vehicle detection in HR RS images.

The performance of target recognition in such a complex context relies on the features extracted from the objects. DL methods are well suited for this task, as this type of algorithm can extract low-level features with a high frequency, such as edges, contours, and outlines of objects, whatever the shape, size, color, or rotation angle of the targets. This type of algorithm can also learn hierarchical representations from the input images or patches, such as the parts of the objects that are compounded by the lower-level features, making recognition of RS targets discriminative and robust. A number of these approaches achieved state-of-the-art performance in target recognition by use of a DL method [30], [48], [49], [52], [56], [93]–[96].

GENERAL DEEP-LEARNING FRAMEWORK OF REMOTE SENSING TARGET RECOGNITION

The DL methods used in target recognition can be divided into two main categories: unsupervised methods and supervised methods. The unsupervised methods learn features from the input data without knowing the correlated labels or other supervisory information, while the supervised methods use the input data as well as the supervisory information attached to the input to discriminatively learn the feature representations. However, both of these DL methods are utilized to learn features from the object images, and the learning processes can be unified into the same framework, as depicted in Figure 3.

The RS images are first preprocessed to subtract the mean and divide the variance, or to simply convert the images to gray images with only one channel. Other preprocessing techniques compute the gradient images [97] of the original image with a certain threshold [30]. The second term of this general pipeline is extracting the object proposals, which is a bounding box locating the probable targets. Following the process of selecting the proposals from the whole image, a simple feature extraction is conducted for each proposal or the whole image to extract the low-level descriptors that are invariant to shift, rotation, and scaling, to some extent, such as SIFT [98], Gabor [99], and the histogram of oriented gradients (HOG) [97]. Next, the middle-level feature representations can be generated by performing codebook learning on the learned descriptors. This step is not essential, but using these low- or middle-level features usually outperforms merely using the raw pixels when learning hierarchical feature representations by the following deep neural networks.

The deep neural networks such as the CNNs, sparse AEs, and DBNs are hierarchical models that can learn high-level feature representations in the deep layers automatically generated by the features learned in the shallow layers. Having learned the discriminative and robust representations of the proposals, a classifier such as an SVM is trained with training samples composed of the representations of some data and the corresponding supervisory information. When a new proposal is generated from a new image, this framework can automatically learn the high-level features from the raw image, and then classification is undertaken by the well-trained classifier to tell whether the proposal is the target or not.

SAMPLE SELECTION PROPOSALS

To choose the most accurate area that exactly contains the target, a number of proposals should be extracted from the input image. Each proposal is usually a bounding box covering an object that probably contains the target. The most satisfactory case is that the target is in the center of the bounding box, and the bounding box can just cover the edge of the object.

There are different ways of selecting the proposals. The baseline technique is the sliding window method [100], which slides the bounding box over the whole image with a small stride to generate a number of proposals. The sliding window technique is accurate and will not miss any possible proposals that may exactly contain the target, yet it is slow and burdens the subsequent feature-learning

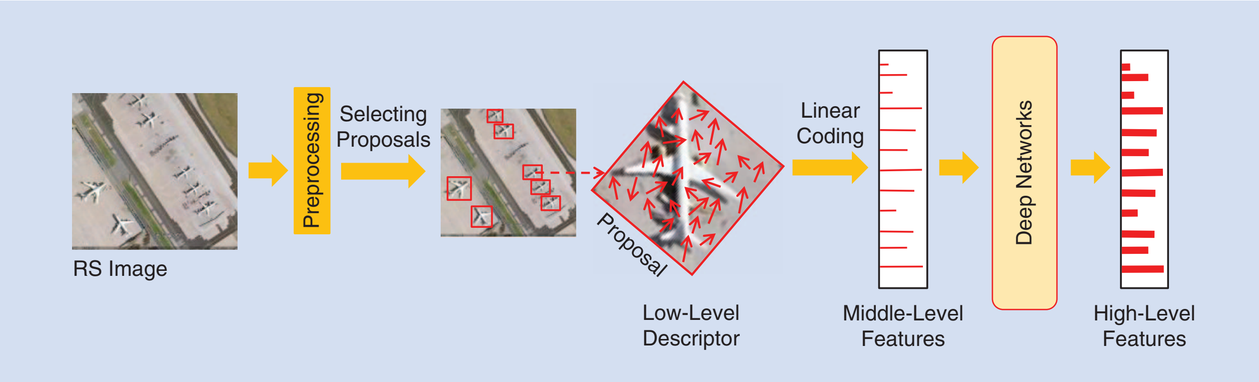

FIGURE 3. A general framework of target recognition using DL methods. The high-level features learned by the deep networks are sent to the classifiers to be classified (or directly classified by the deep networks for a supervised network).

algorithms and classifiers, especially when there are quite a lot of objects in an image (e.g., in RS images). Other methods have been proposed to solve this problem, e.g., Chen et al. [30] proposed an object-location technique that can discover the coarse locations of the targets, and hence can greatly reduce the number of proposals. The search efficiency of this method is more than 20 times the baseline sliding window method. Tang et al. [101] proposed a coarse ship location technique that can extract the candidate locations of the targets with little decrease in accuracy.

LOW- TO MIDDLE-LEVEL FEATURE LEARNING

Low-level features, which have the ability to handle variations in terms of intensity, rotation, scale, and affine projection, are utilized to characterize the local region of the key points in each image or patch. Han et al. [94] utilized SIFT descriptors to represent the image patches, which made the subsequent training process of the DBMs easier. Dalal and Triggs [97] proposed a method for object detection using the HOG.

The low-level descriptors can boost the feature-learning performance of the deep networks. However, they catch only limited local spatial geometric characteristics, which can lead to poor classification or detection performance when they are directly used to describe the structural contents of image patches. To tackle this problem, some work has been done to learn codebooks that are used to encode the local descriptors and generate middle-level feature representations and to alleviate the unrecoverable loss of discriminative information. For instance, Han et al. [94] applied the locality-constrained linear coding model [102] to encode the SIFT descriptors into the image patch representation.

TRAINING THE DEEP-LEARNING NETWORKS

Although the middle-level features are extracted based on the low-level local descriptors to obtain the structural information and preserve the local relevance of elements in the local region, they cannot provide enough strong description and generalization abilities for object detection when confronted with objects and backgrounds with a large variance. To better understand the complexity of the environments in an image, better descriptors should be utilized. The DL methods can handle complex ground objects with large variance, as the features learned by the deep neural networks can be highly abstract, which makes them invariant to relatively large deformations, including different shapes, sizes, and rotations, and discriminative to some objects that belong to different categories but resemble each other in some other aspect, such as white targets on a white background. Generally speaking, the DL methods used in target recognition in RS images can be divided into two categories: 1) the supervised DL algorithms and 2) the unsupervised DL algorithms.

SUPERVISED METHODS

There are two typical supervised DL methods for target recognition: the CNN and the multilayer perceptron (MLP) [103]. The CNNs are hierarchical models that transform the input image or image patch into layers of feature maps, which are high-level discriminative features representing the original input data. For the MLP model, the input image or patch should be reshaped into a vector. Then, after the transformation of each fully connected layer, the final feature representation can be generated. The final features are then sent to the classification layer to generate the label of the input image. Both types of supervised networks transform the input image into a two-dimensional vector for a one-class object detection. This vector indicates the predicted label (whether the input candidate is the target or not, or the probability of the proposal being the target). In the training stage, to learn the weights or kernels, the supervised networks are trained with the training samples composed of positive samples that contain the target and negative samples that do not contain the target. In the testing stage, the proposals extracted from a new RS image are processed by the models and attached with a probability y. The candidates then considered to contain the target are selected by a given empirical threshold or other criteria.

Although the CNN has shown robustness to distortion, it only extracts features of the same scale and, therefore, cannot tolerate a large-scale variance of objects. When it comes to RS images that have a large variance in the backgrounds and objects, training a CNN that extracts multiscale feature representations is necessary for a better detection accuracy. Chen et al. [30] proposed a hybrid deep neural network (HDNN) by dividing the maps of the final convolutional layer and the max-pooling layer of the deep neural network into multiple blocks of variable receptive field sizes or max-pooling field sizes to enable the HDNN to extract variable-scale features for detecting the RS objects. The input of the HDNN with L convolutional layers is a gray image. The image is filtered by the filters in the first convolutional layers f^1 to get the feature maps C^1, which are then subsampled by the first max-pooling layer to select the representative features as well as reduce the number of parameters to be processed. After transferring the L layers' activations or feature maps, the final convolutional feature maps of the Lth layer C^L are generated. In the architecture of the conventional CNNs, the final layer is followed by some fully connected layers and finally the classification layer. However, this kind of feature-processing method does not make full use of the features and the filters. The receptive field size of each convolutional layer is fixed, and thus it cannot extract multiscale features. However, there are still rich features in the final convolutional layer that can be learned and transformed into more discriminative representations. One way to better utilize the rich features is to increase the depth of the convolutional layers, which may, however, introduce a huge amount of computational burden when training the model. Another way is to use a multiscale receptive field size that can train filters with different sizes and generate multiscale feature maps.

In the HDNN, the last layer's feature maps are divided into T blocks $\{B^1, B^2, \ldots, B^T\}$ with filter sizes of $\{s^1 \times s^1, s^2 \times s^2, \ldots, s^T \times s^T\}$, respectively. The ith block covers n_i feature maps of the final convolutional layer. Then the activation propagation between the last two convolutional layers can be formulated as

$$B^t = \sigma(C^{L-1} * f^t + b^t), \tag{15}$$

where B^t denotes the tth block of the last feature maps, f^t denotes the filters of the corresponding block, and σ denotes the activation function.

Having learned the multiscale feature representations to form the final convolutional layer, an MLP network is used to classify the features. The output of the HDNN is a two-node layer, which indicates the probability of whether the input image patch contains the target. Some of the vehicle-detection results are referred to in [30], from which it can be concluded that, although there are a number of vehicles in the scene, the modified CNN model can successfully recognize the precise location of most of the targets, indicating that the HDNN has learned fairly discriminative feature representations to recognize the objects.

UNSUPERVISED METHODS

Although the supervised DL methods like the CNN and its modified models can achieve acceptable performances in target recognition tasks, there are limitations to such methods since their performance relies on large amounts of labeled data, while, in RS image data sets, high-quality images with labels are limited. It is therefore necessary to recognize the targets with a few labeled image patches while learning the features with the unlabeled images.

Unsupervised feature-learning methods are models that can learn feature representations from the patches with no supervision. Typical unsupervised feature-learning methods are RBMs, sparse coding, AEs, k-means clustering, and the Gaussian Mixture Model [104]. All of these shallow feature-learning models can be stacked to form deep unsupervised models, some of which have been successfully applied to recognizing RS scenes and targets. For instance, the DBN generated by stacking RBMs has shown its superiority over conventional models in the task of recognizing aircraft in RS scenes [105].

The DBN is a deep probabilistic generative model that can learn the joint distribution of the input data and its ground truth. The general framework of the DBN model is illustrated in Figure 4. The weights of each layer are updated through layer-wise training using the CD algorithm, i.e., training each layer separately. The joint distribution between the observed vector x and the L hidden layers is $P(x, h^1, h^2, \ldots, h^l) = (\prod_{k=0}^{l-2} P(h^k | h^{k+1}))\, P(h^{l-1}, h^l)$, where $P(h^k | h^{k+1})$ is a conditional distribution for the visible units conditioned on the hidden units of the RBM at level k, and $P(h^{l-1}, h^l)$ is the visible–hidden joint distribution in the top-level RBM. Some aircraft detection results from large airport scenes can be seen in [105], from which we can see that most aircrafts with different shapes and rotation angles have been detected.

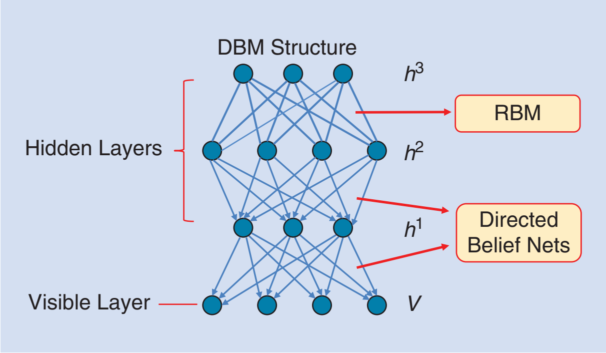

FIGURE 4. The simple structure of the standard DBN.

SCENE UNDERSTANDING

Satellite imaging sensors can now acquire images with a spatial resolution of up to 0.41 m. These images, which are usually called *very high-resolution (VHR) images*, have abundant spatial and structural patterns. However, due to the huge volume of the image data, it is difficult to directly access the VHR data containing the scenes of interest. Due to the complex composition and large number of land-cover types, efficient representation and understanding of the scenes from VHR data have become a challenging problem, which has drawn great interest in the RS field.

Recently, a lot of work in RS scene understanding has been proposed that focuses on learning hierarchical internal feature representations from image data sets [50], [106]. Good internal feature representations are hierarchical. In an image, pixels are assembled into edgelets, edgelets into motifs, motifs into parts, and parts into objects. Finally, objects are assembled into scenes [107], [108]. This suggests that recognizing and analyzing scenes from VHR images should have multiple trainable feature-extraction stages stacked on top of each other, and we should learn the hierarchical internal feature representations from the image.

UNSUPERVISED HIERARCHICAL FEATURE-LEARNING-BASED METHODS

As indicated in the "General Framework" section, there is some work that focuses on unsupervised feature-learning techniques for RS images scene classification, such as sparse coding [58], k-means clustering [60], [109], and topic model [110], [111]. These shallow models could be considered to stack into deep versions in a hierarchical manner [31], [106]. Here, we summarize an overall architecture of

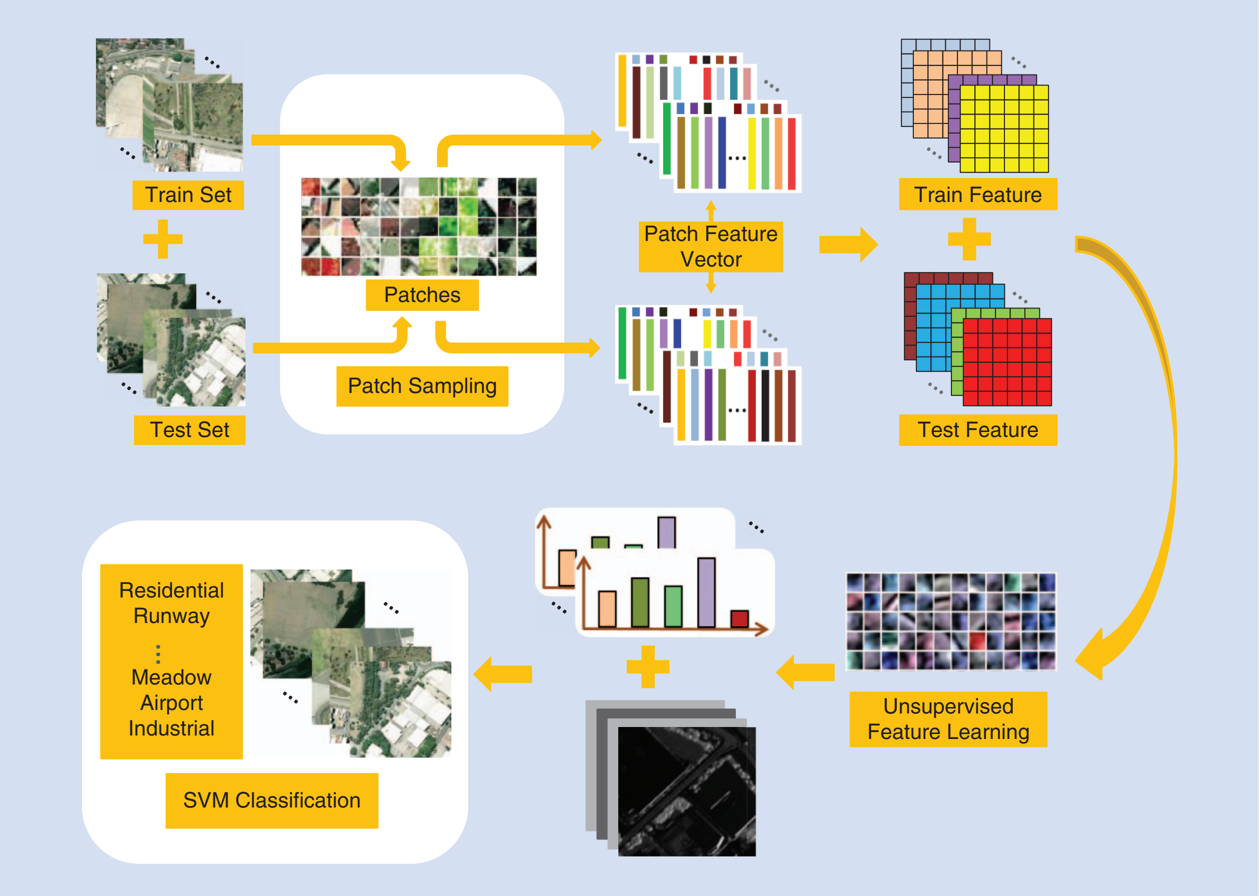

FIGURE 5. The overall architecture of the unsupervised feature-learning framework.

the unsupervised feature-learning framework for RS scene classification. As depicted in Figure 5, the framework consists of four parts: 1) patch extraction, 2) feature extraction, 3) feature representation, and 4) classification.

PATCH EXTRACTION

In this step, the patches are extracted from the image by using random sampling or another method. Each patch has a dimension of $w \times w$ and has three bands (R, G, and B), with w referred to as the receptive field size. Each $w \times w$ patch can be represented as a vector in $\mathbb{R}^N$ of the pixel intensity values, with $N = w \times w \times 3$. A data set of m sampled patches can thus be constructed. Then various features can be extracted from the image patch to construct the training and test data, such as raw pixel or other low-level features (e.g., color histogram, local binary pattern, and SIFT). The patch feature composed of the training feature and test feature is then fed into an unsupervised feature-learning method that is used for the unsupervised learning of the feature extractor W.

FEATURE EXTRACTION

After the unsupervised feature learning, the features can be extracted from the training and test images using the learned feature extractor W, as illustrated in Figure 6. Given a w-×-w image patch, we can now extract a representative $\alpha^i \in \mathbb{R}^K$ for that patch by using the learned feature extractor $f: \mathbb{R}^N \to \mathbb{R}^K$. We then define a new representation of the entire image using the feature extractor function $f: \mathbb{R}^N \to \mathbb{R}^K$ with each image. Specifically, given an image of n-×-n pixels (with three channels: R, G, and B), we can define an $(n - w + 1)$-×-$(n - w + 1)$ representation (with K channels) by computing the representative $\alpha^i \in \mathbb{R}^K$ for each w-×-w subpatch of the input image. More formally, we denote $\alpha_{(ij)}$ as the K-dimensional feature extracted from location i, j of the input image. For computational efficiency, we can also convolute our n-×-n image with a step size (or stride) greater than 1 across the image.

FEATURE REPRESENTATION

After the feature extraction, the new feature representation for an image scene will usually have a very high dimensionality. For computational efficiency and storage volume, it is standard practice to use max-pooling or another strategy to reduce the dimensionality of the image representation [112], [36]. For a stride of $s = 1$, the feature mapping produces an $(n - w + 1)$-×-$(n - w + 1)$-×-K representation. We can reduce this by finding the maximum over local regions of $\alpha_{(ij)}$, as done previously. This procedure is commonly used in computer vision, with many variations, as well as in deep feature learning.

CLASSIFICATION

Finally, the extracted feature is combined with the SVM or another classifier to predict the scene label. However, most methods for unsupervised feature learning produce filters that operate either on intensity or color information. Vladimir [113] proposed a quaternion PCA and k-means combined approach for unsupervised feature learning that makes joint encoding of the intensity and color information possible. In addition, Cheriyadat [58] introduced a variant of sparse coding that combines local SIFT-based feature descriptors to generate a new sparse representation, producing an excellent classification accuracy. The sparse AE-based method also produces excellent performance. In [31], Zhang et al. proposed a saliency-guided sparse AE method to learn a set of feature extractors that are robust and efficient, proposing a saliency-guided sampling strategy to extract a representative set of patches from a VHR image so that the salient parts of the image that contain the representative information in the VHR image can be explored, which differs from the traditional random sampling strategy. They also explored the new dropout technique in the feature-learning procedure to reduce data overfitting [114]. The extracted feature generated from the learned feature extractors can characterize a complex scene very well and can produce an excellent classification accuracy.

SUPERVISED HIERARCHICAL FEATURE-LEARNING-BASED METHODS

Before 2006, it was believed that training deep supervised neural networks was too difficult to perform (and indeed did not work). The first breakthrough in training happened in Geoff Hinton's lab with an unsupervised pretraining by RBMs, as discussed in the previous subsection. However, more recently, it was discovered that one could train deep supervised networks by proper initialization, just large enough for the gradients to flow well and the activations to convey useful information. These good results with the pure supervised training of deep networks seem to be especially apparent when large quantities of labeled data are available.

In the early years after 2010, based on the latent Dirichlet allocation (LDA) model [115], various supervised hierarchical feature-learning methods have been proposed in the RS community [116]–[120]. LDA is a generative probabilistic graphical model for independent collections of discrete data and is a three-level hierarchical model, in which the documents inside a corpus are represented as random mixtures over a set of latent variables called *topics*. Each topic is in turn characterized by a distribution over words. The LDA model captures all of the important information contained in a corpus by considering only the statistics of the words. The contextual relationships are neglected due to the Bayesian assumption. For this reason, LDA is categorized as a *bag of words* model. Its main characteristic is based on the words' exchangeability. The LDA-based supervised hierarchical feature-learning approaches have been shown to generate excellent hierarchical feature representations for RS scene classification.

In fact, the LDA-based models are still not deep enough compared to the other techniques in the DL family. More recently, a few pure DL methods have been proposed for RS image scene understanding based on CNNs [121]. Zhang et al. proposed in detail a gradient-boosting random convolutional network framework for RS scene classification that can effectively combine many deep neural networks [50]. Marmanis et al. considered a pretrained CNN by the ImageNet challenge and exploited it to extract an initial set of representations for earth observation classification [122]. Hu et al. investigated how to transfer features from the existing successfully pretrained CNNs for RS scene classification [123]. Luus et al. suggested a multiscale input strategy for multiview DL with the aid of convolutional layers to shift the burden of feature determination from hand-engineering to a deep CNN [124]. These advanced supervised DL methods all outperform the state-of-the-art methods with the various RS scene classification data sets.

EXPERIMENTS AND ANALYSIS

In this section, we present some experimental results on the DL algorithms for RS data scene understanding that we

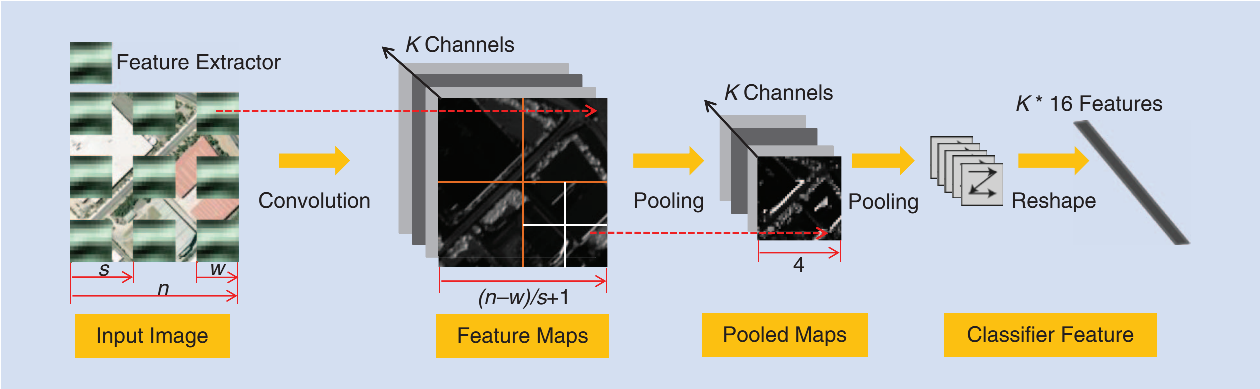

FIGURE 6. The feature extraction using a *w*-×-*w* feature extractor and a stride of *s*. We first extract the *w*-×-*w* patches, each separated by *s* pixels, then map them to the *K*-dimensional feature vectors to form a new image representation. These vectors are then pooled over 16 quadrants of the image to form a feature vector for classification.

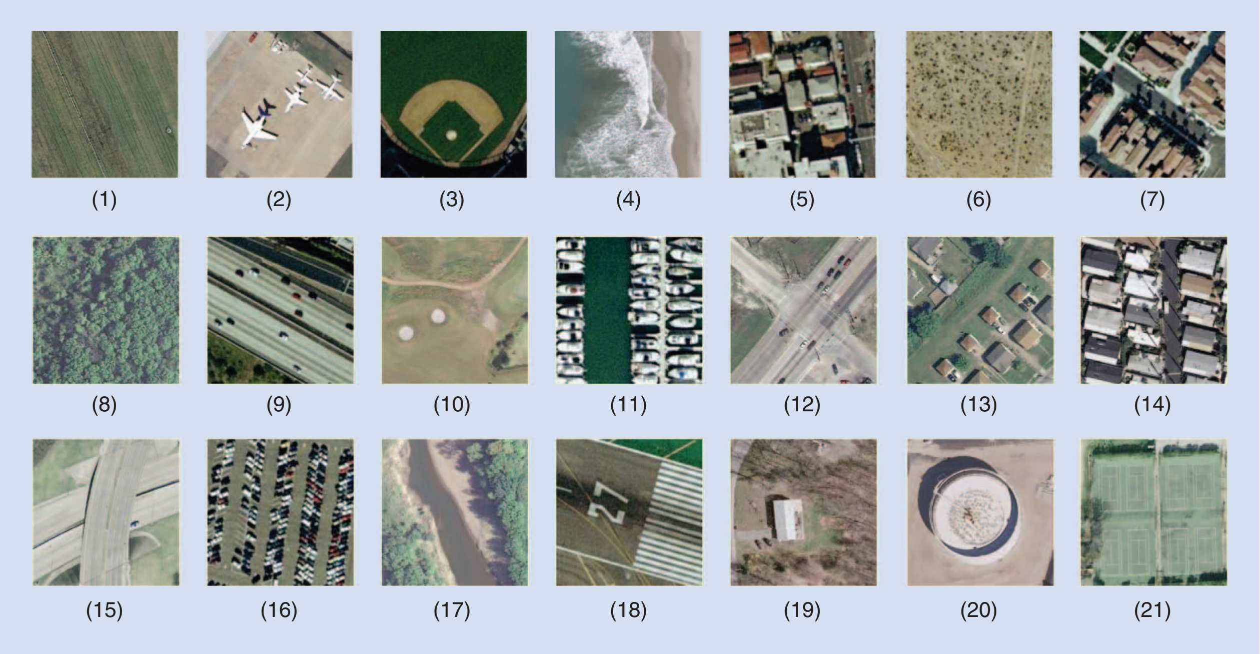

FIGURE 7. Example images associated with the 21 land-use categories in the UC Merced data set: 1) agricultural, 2) airplane, 3) baseball diamond, 4) beach, 5) buildings, 6) chaparral, 7) dense residential, 8) forest, 9) freeway, 10) golf course, 11) harbor, 12) intersection, 13) medium residential, 14) mobile-home park, 15) overpass, 16) parking lot, 17) river, 18) runway, 19) sparse residential, 20) storage tanks, and 21) tennis court.

TABLE 1. A COMPARISON OF THE PREVIOUSLY REPORTED ACCURACIES AND THE UC MERCED DATA SET.

METHOD	SPMK	SSC	SSAE	RCNet
Accuracy	74%	81.67%	82.72%	**94.53%**

believe can bring the most significant improvement compared to existing methods. Due to page limitations, experimental results relating to RS image preprocessing, pixel-based classification, and target recognition can be found within the papers in the reference list.

The first data set chosen for investigation is the well-known University of California (UC) Merced data set [125]. Figure 7 shows a few example images representing various RS scenes that are included in this data set, which contains 21 challenging scene categories with 100 image samples per class. Following the experimental setup in [58], we randomly selected 80% of the samples from each class for training and set the remaining images for testing. For this data set, we compared our proposed supervised DL method, i.e., the random convolutional network (RCNet) [50], with the spatial pyramid matching kernel (SPMK) method [112], the SIFT + sparse coding (SSC) approach described in [58], and our unsupervised feature-learning method, i.e., the saliency-guided sparse AE (SSAE) method that was previously proposed in [31]. For the RCNet algorithm, we trained the RCNet function using stochastic gradient descent with a batch size of 64, a momentum of 0.9, a weight decay of 0.0005, and a learning rate of 0.01. In addition, we trained each RCNet for roughly 500 cycles with the whole training set. All of these experiments were run on a personal computer (PC) with a single Intel core i7 central processing unit, an NVIDIA Titan graphics processing unit, and 6-GB memory. The operating system was Windows 7, and the implementation environment was under MATLAB 2014a with a CUDA kernel. We compared the reported classification performances with the challenging UC Merced data set, and, among the four strategies we compared, the supervised DL method RCNet produced the best performance, as shown in Table 1.

The other image data set was constructed from a large satellite image that was acquired from Google Earth of Sydney, Australia. The spatial resolution of the image was approximately 1.0 m. The large image to be annotated was of 7,849 × 9,073 pixels, as shown in Figure 8. There were eight classes of training images: residential, airplane, meadow, rivers, ocean, industrial, bare soil, and runway. Figure 8 shows some examples of such images. This data set consisted of not only the eight defined classes, but also some other classes that had not been learned such as the bridges and the main roads. We manually labeled part of the image to obtain a subregion image data set, in which each subregion was of the size of 128 × 128, whereby each subimage was supposed to contain a certain scene. The training set for each class contained 25 samples of the labeled images for

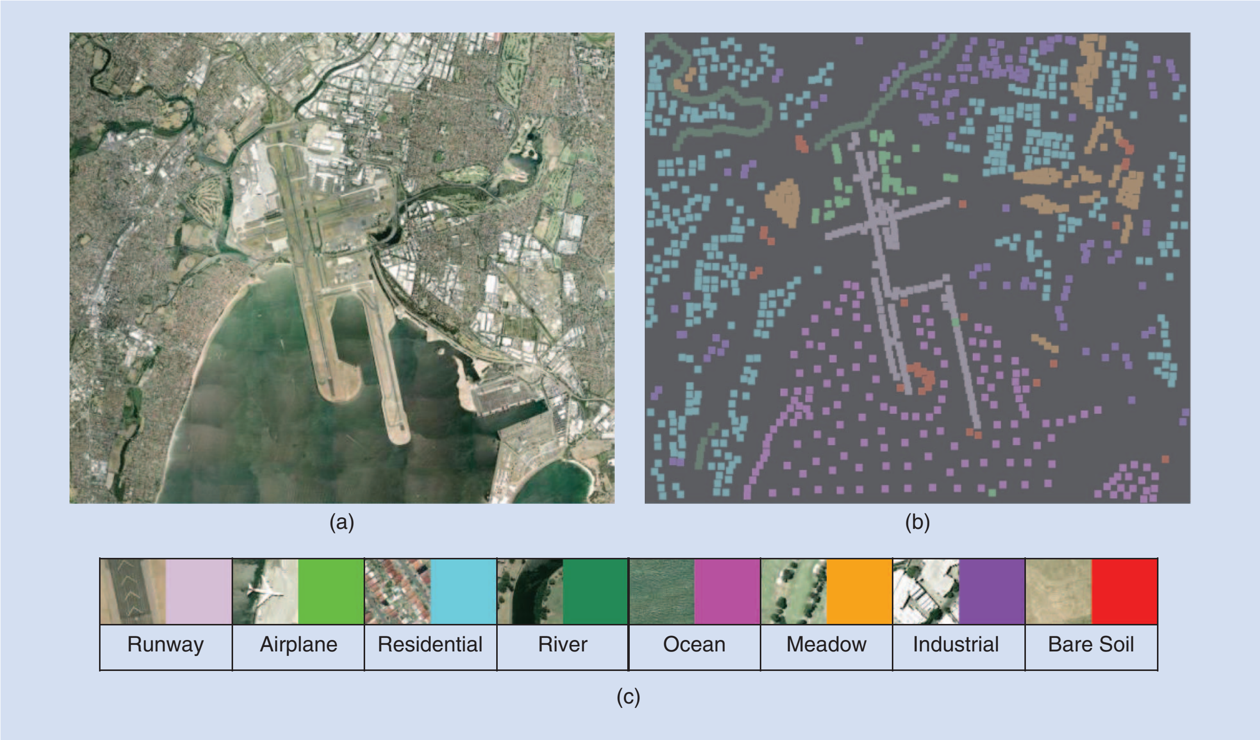

FIGURE 8. (a) The whole image for image annotation. (b) The image ground truth. (c) Example images associated with the eight land-use categories from the image: 1) runway, 2) airplane, 3) residential, 4) river, 5) ocean, 6) meadow, 7) industrial, and 8) bare soil.

each class, while the remaining images were used for testing, as shown in Table 2.

For the Sydney data set, we trained the RCNet function using stochastic gradient descent with a batch size of 32, a momentum of 0.9, a weight decay of 0.0005, and a learning rate of 0.01. We trained each RCNet for roughly 800 cycles with the whole training set. The PC environment was the same as previously mentioned. We also compared the final classification accuracies for RCNet and the traditional methods. Table 3 shows the average overall accuracies for the four methods. The results confirm that using the supervised DL method is an efficient way to increase the RS scene classification accuracy.

TABLE 2. THE TRAINING AND TEST SAMPLES FOR THE SYDNEY DATA SET.

	CLASS	SAMPLES	
NO.	NAME	TRAINING	TEST
1.	Runway	25	97
2.	Airplane	25	16
3.	Residential	25	381
4.	River	25	59
5.	Ocean	25	133
6.	Meadow	25	102
7.	Industrial	25	101
8.	Bare soil	25	9
Total		200	898

TABLE 3. THE OVERALL ACCURACIES FOR THE DIFFERENT METHODS WITH THE SYDNEY DATA SET.

METHOD	SPMK	SSC	SSAE	RCNet
Accuracy	89.67%	91.33%	92.20%	**98.78%**

CONCLUSIONS AND FUTURE WORK

In this technical tutorial, we have systematically reviewed the state-of-the-art DL techniques in RS data analysis. The DL techniques were originally rooted in machine-learning fields for classification and recognition tasks, and they have only recently appeared in the geoscience and RS community. From the four perspectives of image preprocessing, pixel-based classification, target recognition, and scene understanding, we have found that DL techniques have had significant successes in the areas of target recognition and scene understanding, i.e., areas that have been widely accepted as challenges in recent decades in the RS community because such applications require us to abstract the high-level semantic information from the bottom-level features (usually the raw pixel representation), while the traditional RS methods of feature describing feature extraction classification are shallow models, with which it is extremely difficult or impossible to uncover the high-level representation.

On the other hand, the achievements of DL techniques in image preprocessing and pixel-based classification (especially considering the cost of the large training set) have not been as dramatic, which is partly because the image-quality improvement is more likely to relate to the image-degradation model (as with the traditional approaches), and the task of predicting the label of pixels in the RS image is relative shallow for most conditions, even when only addressing the spectral feature. Despite this, we strongly believe that DL techniques are crucial and important in RS data analysis, particularly for the age of RS big data.

However, the research in DL is still young and many questions remain unanswered [44]. The following are some potentially interesting topics in RS data analysis.

1) The number of training samples: Although DL methods can learn highly abstract feature representations from raw RS images, the detection and recognition performance relies on large numbers of training samples. However, there is usually a lack of high-quality training images because the collection of labeled HR images is difficult. Under these circumstances, how to retain the representation learning performance of the DL methods with fewer adequate training samples remains a big challenge.
2) The complexity of RS images: Unlike natural scene images, HR RS images include various types of objects with different sizes, colors, rotations, and locations in a single scene, while distinct scenes belonging to different categories may resemble each other in many respects. The complexity of RS images contributes a lot to the difficulty of learning robust and discriminative representations from scenes and objects with DL.
3) Transfer between data sets: An interesting direction is the transfer of the feature detectors learned by deep networks from one data set to another, since there is often a lack of training images in some fields of RS. Especially when facing the large variations of RS data sets, the problem may be even worse, which needs further and systematic research.
4) Depth of the DL model: The deeper the deep networks are, the better the performance of the models. For supervised networks such as CNNs, deeper layers can learn more complex distributions, but they may result in many more parameters to learn, and hence can lead to the problem of overfitting, especially when the training samples are inadequate. The computation time is also a vital factor that should be considered. Exploring the proper depth of a DL model for a given data set is still an open topic to be researched.

ACKNOWLEDGMENTS

The authors would like to thank the guest editors and the anonymous reviewers for their careful reading and helpful remarks that have contributed to improving the quality of this article. This article is supported by the National Natural Science Foundation of China under grants 41431175, 61401317, and 61471274.

AUTHOR INFORMATION

Liangpei Zhang (zlp62@whu.edu.cn) received a B.S. degree in physics from Hunan Normal University, Changsha, China, in 1982; an M.S. degree in optics from the Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, in 1988; and a Ph.D. degree in photogrammetry and remote sensing from Wuhan University, China, in 1998. He is currently the head of the Remote Sensing Division, State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University. He is also a Chang-Jiang Scholar chair professor appointed by the Ministry of Education of China and a principal scientist for the China State Key Basic Research Project (2011–2016), appointed by China's Ministry of National Science and Technology to lead the country's remote sensing program. He has more than 450 research papers and five books to his credit and holds 15 patents. He is currently serving as an associate editor of *IEEE Transactions on Geoscience and Remote Sensing*. He is a Senior Member of the IEEE.

Lefei Zhang received B.S. and Ph.D. degrees from Wuhan University, China, in 2008 and 2013, respectively. From August 2013 to July 2015, he was with the School of Computer at Wuhan University as a postdoctoral researcher, and he was a visiting scholar with the State Key Laboratory of Computer Aided Design and Computer Graphics at Zhejiang University in 2015. He is currently a lecturer at the School of Computer, Wuhan University, and a Hong Kong Scholar with the Department of Computing, Hong Kong Polytechnic University. His research interests include pattern recognition, image processing, and remote sensing. He is a Member of the IEEE.

Bo Du received B.S. and Ph.D. degrees in photogrammetry and remote sensing from the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, China, in 2005 and 2010, respectively. He is currently a professor at the School of Computer, Wuhan University. He has more than 40 research papers published in journals, such as *IEEE Transactions on Geoscience and Remote Sensing* (*TGRS*), *IEEE Transactions on Image Processing* (*TIP*), *IEEE Journal of Selected Topics in Earth Observations and Applied Remote Sensing* (*JSTARS*), and *IEEE Geoscience and Remote Sensing Letters* (*GRSL*). He was session chair for the Fourth IEEE Geoscience and Remote Sensing Society Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing. He also serves as a reviewer of 20 Science Citation Index magazines, including *IEEE TGRS*, *TIP*, *JSTARS*, and *GRSL*. He is a Senior Member of the IEEE.

REFERENCES

[1] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. M. Nasrabadi, and J. Chanussot, “Hyperspectral remote sensing data analysis and future challenges,” *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, 2013.

[2] J. A. Benediktsson, J. Chanussot, and W. M. Moon, “Very high-resolution remote sensing: Challenges and opportunities,” *Proc. IEEE*, vol. 100, no. 6, pp. 1907–1910, 2012.

[3] U. C. Benz, P. Hofmann, G. Willhauck, I. Lingenfelder, and M. Heynen, “Multi-resolution, object-oriented fuzzy analysis of remote sensing data for gis-ready information,” *ISPRS J. Photogramm. Remote Sens.*, vol. 58, no. 3–4, pp. 239–258, 2004.

[4] T. Blaschke, “Object based image analysis for remote sensing,” *ISPRS J. Photogramm. Remote Sens.*, vol. 65, no. 1, pp. 2–16, 2010.

[5] M. Bevis, S. Businger, T. A. Herring, C. Rocken, R. A. Anthes, and R. H. Ware, “GPS meteorology: Remote sensing of atmospheric water vapor using the Global Positioning System,” *J. Geophys. Res. [Atmos.]*, vol. 97, no. D14, pp. 15,787–15,801, Oct. 1992.

[6] M. H. Ward, J. R. Nuckols, S. J. Weigel, S. K. Maxwell, K. P. Cantor, and R. S. Miller, “Identifying populations potentially exposed to agricultural pesticides using remote sensing and a geographic information system,” *Environ. Health Perspect.*, vol. 108, no. 1, pp. 5–12, 2000.

[7] S. P. Hoogendoorn, H. J. V. Zuylen, M. Schreuder, B. Gorte, and G. Vosselman, “Microscopic traffic data collection by remote sensing,” *Transport. Res. Rec.: J. Transport. Res. Board*, vol. 1855, no. 1, pp. 121–128, 2003.

[8] M. A. Lefsky, W. B. Cohen, G. G. Parker, and D. J. Harding, “LiDAR remote sensing for ecosystem studies,” *Biosci.*, vol. 52, no. 1, pp. 19–30, Jan. 2002.

[9] B. Aiazzi, L. Alparone, S. Baronti, and A. Garzelli, “Context-driven fusion of high spatial and spectral resolution images based on oversampled multiresolution analysis,” *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 10, pp. 2300–2312, 2002.

[10] J. Liu, J. Li, W. Li, and J. Wu, “Rethinking big data: A review on the data quality and usage issues,” *ISPRS J. Photogramm. Remote Sens.*, vol. 115, pp. 134–142, May 2016.

[11] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. A. Benediktsson, “Advances in hyperspectral image classification: Earth monitoring with statistical learning methods,” *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 45–54, 2014.

[12] J. A. Benediktsson, M. Pesaresi, and K. Arnason, “Classification and feature extraction for remote sensing images from urban areas based on morphological transformations,” *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 9, pp. 1940–1949, 2003.

[13] X. Jia, B. C. Kuo, and M. M. Crawford, “Feature mining for hyperspectral image classification,” *Proc. IEEE*, vol. 101, no. 3, pp. 676–697, 2013.

[14] 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.

[15] G. E. Hinton, “Learning multiple layers of representation,” *Trends Cogn. Sci.*, vol. 11, no. 10, pp. 428–434, 2007.

[16] J. R. Jensen, and K. Lulla, “Introductory digital image processing: A remote sensing perspective,” *Geocarto Int.*, vol. 2, no. 1, pp. 65, 1987.

[17] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, “Deep, big, simple neural nets for handwritten digit recognition,” *Neural Comput.*, vol. 22, no. 12, pp. 3207–3220, Dec. 2010.

[18] M. Riesenhuber and T. Poggio, “Hierarchical models of object recognition in cortex,” *Nat. Neurosci.*, vol. 2, no. 11, pp. 1019–1025, 1999.

[19] C.-I. Chang, Q. Du, T.-L. Sun, and M. L. G. Althouse, “A joint band prioritization and band-decorrelation approach to band selection for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 6, pp. 2631–2641, 1999.

[20] A. Plaza, P. Martinez, J. Plaza, and R. Perez, “Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations,” *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 466–479, 2005.

[21] S. De and A. Bhattacharya, “Urban classification using PolSAR data and deep learning,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Milan, Italy, 2015, pp. 353–356.

[22] Q. Lv, Y. Dou, X. Niu, J. Xu, and B. Li, “Classification of land cover based on deep belief networks using polarimetric RADARSAT-2 data,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Québec, Canada, 2014, pp. 5132–5136.

[23] H. Xie, S. Wang, K. Liu, S. Lin, and B. Hou, “Multilayer feature learning for polarimetric synthetic radar data classification,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Québec, Canada, 2014, pp. 5132–5136.

[24] H. Wang, S. Chen, F. Xu, and Y.-Q. Jin, “Application of deep-learning algorithms to MSTAR data,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Milan, Italy, 2015, pp. 3743–3745.

[25] J. Geng, J. Fan, H. Wang, X. Ma, B. Li, and F. Chen, “High-resolution SAR image classification via deep convolutional AE,” *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 11, pp. 2351–2355, 2015.

[26] C. Bentes, D. Velotto, and S. Lehner, “Target classification in oceanographic SAR images with deep neural networks: Architecture and initial results,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Milan, Italy, 2015, pp. 3703–3706.

[27] Y. Yu, J. Li, H. Guan, F. Jia, and C. Wang, “Learning hierarchical features for automated extraction of road markings from 3-D mobile LiDAR point clouds,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 2, pp. 709–726, 2015.

[28] W. Huang, L. Xiao, Z. Wei, H. Liu, and S. Tang, “A new pan sharpening method with deep neural networks,” *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 5, pp. 1037–1041, 2015.

[29] 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. 2381–2392, 2015.

[30] X. Chen, S. Xiang, C. L. Liu, and C. H. Pan, “Vehicle detection in satellite images by hybrid deep convolutional neural networks,” *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1797–1801, 2014.

[31] F. Zhang, B. Du, and L. Zhang, “Saliency-guided unsupervised feature learning for scene classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 2175–2184, 2015.

[32] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[33] A.-R. Mohamed, T. N. Sainath, G. Dahl, B. Ramabhadran, G. E. Hinton, and M. A. Picheny, "Deep belief networks using discriminative features for phone recognition," in *Proc. IEEE Int. Conf. Acoust. Speech, and Signal Processing*, Prague, Czech Republic, 2011, pp. 5060–5063.

[34] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. Int. Conf. Mach. Learning*, Helsinki, Finland, 2008, pp. 160–167.

[35] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. Advances in Neural Inform. Processing Syst.*, Vancouver, Canada, 2006, p. 153.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Advances in Neural Inform. Processing Syst.*, Lake Tahoe, NV, 2012, pp. 1106–1114.

[37] T. Serre, G. Kreiman, M. Kouh, C. Cadieu, U. Knoblich, and T. Poggio, "A quantitative theory of immediate visual recognition," *Prog. Brain Res.*, vol. 165, pp. 33–56, 2007.

[38] G. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[39] Y. Freund and D. Haussler, "Unsupervised learning of distributions on binary vectors using two layer networks," University of California, Santa Cruz, Tech. Rep. UCSC-CRL-94-25, 1994.

[40] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," University of California, San Diego, Institute for Cognitive Science Report 8506, 1985.

[41] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzago, "Extracting and composing robust features with denoising AE," in *Proc. Int. Conf. Mach. Learning*, Helsinki, Finland, 2008, pp. 1096–1103.

[42] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Proc. Advances in Neural Inform. Processing Syst.*, Vancouver, Canada, 2006, pp. 801–808.

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

[44] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[45] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, Article ID 258619, pp. 1–12, 2015.

[46] J. Yue, W. Zhao, S. Mao, and H. Liu, "Spectral-spatial classification of hyperspectral images using deep convolutional neural networks," *Remote Sens. Lett.*, vol. 6, no. 6, pp. 468–477, 2015.

[47] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Milan, Italy, 2015, pp. 4959–4962.

[48] M. Vakalopoulou, K. Karantzalos, N. Komodakis, and N. Paragios, "Building detection in very high resolution multispectral data with deep learning features," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Milan, Italy, 2015, pp. 1873–1876.

[49] L. Zhang, Z. Shi, and J. Wu, "A hierarchical oil tank detector with deep surrounding features for high-resolution optical satellite imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 10, pp. 4895–4909, 2015.

[50] F. Zhang, B. Du, and L. Zhang, "Scene classification via a gradient boosting random convolutional network framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1793–1802, 2016.

[51] W. Zhou, Z. Shao, C. Diao, and Q. Cheng, "High-resolution remote sensing imagery retrieval using sparse features by autoencoder," *Remote Sens. Lett.*, vol. 6, no. 10, pp. 775–783, 2015.

[52] J. Tang, C. Deng, G. B. Huang, and B. Zhao, "Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1174–1185, 2015.

[53] J. Li, L. Bruzzone, and S. Liu, "Deep feature representation for hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Milan, Italy, 2015, pp. 4951–4954.

[54] X. Ma, J. Geng, and H. Wang, "Hyperspectral image classification via contextual deep learning," *EURASIP J. Image Video Process.*, vol. 2015, no. 20, pp. 1–12, 2015.

[55] T. Li, J. Zhang, and Y. Zhang, "Classification of hyperspectral image based on deep belief networks," in *Proc. IEEE Int. Conf. Image Processing*, Paris, France, 2014, pp. 5132–5136.

[56] W. Diao, X. Sun, F. Dou, M. Yan, H. Wang, and K. Fu, "Object recognition in remote sensing images using sparse deep belief networks," *Remote Sens. Lett.*, vol. 6, no. 10, pp. 745–754, 2015.

[57] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep learning based feature selection for remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 11, pp. 2321–2325, 2015.

[58] A. M. Cheriyadat, "Unsupervised feature learning for aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 439–451, 2014.

[59] W. Yang, X. Yin, and G. S. Xia, "Learning high-level features for satellite image classification with limited labeled samples," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 8, pp. 4472–4482, 2015.

[60] F. Hu, G. S. Xia, Z. Wang, X. Huang, L. Zhang, and H. Sun, "Unsupervised feature learning via spectral clustering of multidimensional patches for remotely sensed scene classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 5, pp. 2015–2030, 2015.

[61] J. Hu, G.-S. Xia, F. Hu, H. Sun, and L. Zhang, "A comparative study of sampling analysis in scene classification of high-resolution remote sensing imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Milan, Italy, 2015, pp. 2389–2392.

[62] M. Fu, Y. Yuan, and X. Lu, "Unsupervised feature learning for scene classification of high resolution remote sensing image," in *Proc. IEEE China Summit and Int. Conf. Signal and Inform. Processing*, Chengdu, China, 2015, pp. 206–210.

[63] J. Zhang, P. Zhong, Y. Chen, and S. Li, "L(1/2)-regularized deconvolution network for the representation and restoration of optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2617–2627, 2014.

[64] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Proc. Advances in Neural Inform. Processing Syst.*, Lake Tahoe, NV, 2012, pp. 350–358.

[65] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 480–491, 2005.

[66] L. Zhang, X. Huang, B. Huang, and P. Li, "A pixel shape index coupled with spectral information for classification of high spatial resolution remotely sensed imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 10, pp. 2950–2961, 2006.

[67] X. Huang and L. Zhang, "Morphological building/shadow index for building extraction from high-resolution imagery over urban areas," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 1, pp. 161–172, 2012.

[68] X. Huang, Q. Lu, and L. Zhang, "A multi-index learning approach for classification of high-resolution remotely sensed images over urban areas," *ISPRS J. Photogramm. Remote Sens.*, vol. 90, pp. 36–48, Apr. 2014.

[69] L. Zhang, L. Zhang, D. Tao, and X. Huang, "On combining multiple features for hyperspectral remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 879–893, 2012.

[70] L. Zhang, Q. Zhang, L. Zhang, D. Tao, X. Huang, and B. Du, "Ensemble manifold regularized sparse low-rank approximation for multiview feature embedding," *Pattern Recognit.*, vol. 48, no. 10, pp. 3102–3112, 2015.

[71] 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.

[72] W. Li, E. W. Tramel, S. Prasad, and J. E. Fowler, "Nearest regularized subspace for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 477–489, 2014.

[73] Z. Lin, Y. Chen, X. Zhao, and G. Wang, "Spectral-spatial classification of hyperspectral image using AE," in *Proc. IEEE Int. Conf. Inform., Commun. and Signal Processing,* Tainan, Taiwan, 2013, pp. 1–5.

[74] 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.

[75] C. Xing, L. Ma, and X. Yang, "Stacked denoise AE based feature extraction and classification for hyperspectral images," *J. Sensors,* vol. 2015, Article ID 3632943, pp. 1–10, 2015.

[76] A. Lagrange, B. L. Saux, A. Beaupere, A. Boulch, A. Chan-Hon-Tong, S. Herbin, H. Randrianarivo, and M. Ferecatu, "Benchmarking classification of earth-observation data: From learning explicit features to convolutional networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.,* Milan, Italy, 2015, pp. 4173–4176.

[77] Y. Zhou and Y. Wei, "Learning hierarchical spectral-spatial features for hyperspectral image classification," *IEEE Trans. Cybern.*, 2016. doi: 10.1109/TCYB.2015.2453359.

[78] W. Zhao, Z. Guo, J. Yue, X. Zhang, and L. Luo, "On combining multiscale deep learning features for the classification of hyperspectral remote sensing imagery," *Int. J. Remote Sens.*, vol. 36, no. 13, pp. 3368–3379, 2015.

[79] J. Zabalza, J. Ren, J. Zheng, H. Zhao, C. Qing, Z. Yang, P. Du, and S. Marshall, "Novel segmented stacked AE for effective dimensionality reduction and feature extraction in hyperspectral imaging," *Neurocomputing*, vol. 185, pp. 1–10, 2016.

[80] K. Karalasa, G. Tsagkatakis, M. Zervakisa, and P. Tsakalides, "Deep learning for multi-label land cover classification," in *Proc. SPIE*, 9643, Image and Signal Processing for Remote Sensing XXI, Toulouse, France, 2015.

[81] 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.

[82] L. Zhang, L. Zhang, D. Tao, and X. Huang, "Tensor discriminative locality alignment for hyperspectral image spectral-spatial feature extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 242–256, 2013.

[83] X. Kang, S. Li, and J. A. Benediktsson, "Spectral-spatial hyperspectral image classification with edge-preserving filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2666–2677, 2014.

[84] W. Zhao and S. Du, "Learning multiscale and deep representations for classifying remotely sensed imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 113, pp. 155–165, 2016.

[85] F. Zhang, B. Du, L. Zhang, and L. Zhang, "Hierarchical feature learning with dropout k-means for hyperspectral image classification," *Neurocomputing*, vol. 187, pp. 75–82, 2016.

[86] Y. Liu, G. Cao, Q. Sun, and M. Siegel, "Hyperspectral classification via deep networks and superpixel segmentation," *Int. J. Remote Sens.*, vol. 36, no. 13, pp. 3459–3482, 2015.

[87] H. Alhichri, Y. Bazi, N. Alajlan, and N. Ammour, "A hierarchical learning paradigm for semi-supervised classification of remote sensing images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.,* Milan, Italy, 2015, pp. 4388–4391.

[88] E. Othman, Y. Bazi, H. AlHichri, and N. Alajlan, "A deep learning approach for unsupervised domain adaptation in multitemporal remote sensing images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.,* Milan, Italy, 2015, pp. 2401–2404.

[89] J. Wang, Q. Qin, Z. Li, X. Ye, J. Wang, X. Yang, and X. Qin, "Deep hierarchical representation and segmentation of high resolution remote sensing images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.,* Milan, Italy, 2015, pp. 4320–4323.

[90] D. Tuia, R. Flamary, and N. Courty, "Multiclass feature learning for hyperspectral image classification: Sparse and hierarchical solutions," *ISPRS J. Photogramm. Remote Sens.*, vol. 105, pp. 272–285, 2015.

[91] K. Cai, W. Shao, X. Yin, and G. Liu, "Co-segmentation of aircrafts from high-resolution satellite images," in *Proc. IEEE Int. Conf. Signal Processing,* Beijing, 2012, pp. 993–996.

[92] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, "Vehicle detection in satellite images by parallel deep convolutional neural networks," in *Proc. Asian Conf. Pattern Recognit.,* Naha, Japan, 2013, pp. 181–185.

[93] Y. Yu, H. Guan, and Z. Ji, "Rotation-invariant object detection in high-resolution satellite imagery using superpixel-based deep Hough forests," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 11, pp. 2183–2187, 2015.

[94] J. Han, D. Zhang, G. Cheng, L. Guo, and J. Ren, "Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 6, pp. 3325–3337, 2015.

[95] J. Wang, J. Song, M. Chen, and Z. Yang, "Road network extraction: A neural-dynamic framework based on deep learning and

a finite state machine," *Int. J. Remote Sens.*, vol. 36, no. 12, pp. 3144–3169, 2015.

[96] Y. Yu, H. Guan, D. Zai, and Z. Ji, "Rotation-and-scale-invariant airplane detection in high-resolution satellite images based on deep-Hough-forests," *ISPRS J. Photogramm. Remote Sens.*, vol. 112, pp. 50–64, Feb. 2016.

[97] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Int. Conf. Comput. Vision and Pattern Recognit.*, San Diego, CA, 2005, pp. 886–893.

[98] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[99] J. P. Jones and L. A. Palmer, "An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex," *J. Neurophysiol.*, vol. 58, no. 6, pp. 1233–1258, 1987.

[100] J. Hosang, R. Benenson, P. Dollar, and B. Schiele, "What makes for effective detection proposals?" vol. 1, arXiv:1502.05082, Aug. 2015.

[101] J. Tang, C. Deng, G. B. Huang, and B. Zhao, "Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine" *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1174–1185, 2014.

[102] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Int. Conf. Comput. Vision and Pattern Recognit.*, San Francisco, 2010, pp. 3360–3367.

[103] M. Gardner and S. Dorling, "Artificial neural networks (the multilayer perceptron): A review of applications in the atmospheric sciences," *Atmos. Environ.*, vol. 32, no. 14-15, pp. 2627–2636, Aug. 1998.

[104] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Proc. Int. Conf. Pattern Recognit.*, Cambridge, U.K., 2004, pp. 28–31.

[105] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, "Aircraft detection by deep belief nets," in *Proc. Asian Conf. Pattern Recognit.*, Naha, Japan, 2013, pp. 54–58.

[106] A. Romero, C. Gatta, and G. Camps-Valls, "Unsupervised deep feature extraction for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1349–1362, 2016.

[107] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *Proc. IEEE Int. Conf. Comput. Vision*, Kyoto, Japan, 2009, pp. 2146–2153.

[108] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proc. IEEE Int. Symp. Circuits and Syst.*, Paris, France, 2010, pp. 253–256.

[109] F. Hu, G.-S. Xia, Z. Wang, L. Zhang, and H. Sun, "Unsupervised feature coding on local patch manifold for satellite image scene classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Québec, Canada, 2014, pp. 1273–1276.

[110] B. Zhao, Y. Zhong, G. S. Xia, and L. Zhang, "Dirichlet-derived multiple topic scene classification model for high spatial resolution remote sensing imagery" *IEEE Trans. Geosci. Remote Sens*, vol. 54, no. 4, pp. 2108–2123, 2016.

[111] Y. Zhong, Q. Zhu, and L. Zhang, "Scene classification based on the multifeature fusion probabilistic topic model for high spatial resolution remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 11, pp. 6207–6222, 2015.

[112] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Int. Conf. Comput. Vision and Pattern Recognit.*, New York, 2006, pp. 2169–2178.

[113] V. Risojevic and Z. Babic, "Unsupervised learning of quaternion features for image classification," in *Proc. Int. Conf. Telecommun. in Modern Satellite, Cable, and Broadcast. Services*, Nis, Serbia, 2013, pp. 345–348.

[114] G. E. Hinton, N. Srivastava, A. Krizhevky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing coadaptation of feature detectors," in *Arxiv*, arXiv:1207.0580, 2012.

[115] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Jan. 2003.

[116] M. Lienou, H. Maitre, and M. Datcu, "Semantic annotation of satellite images using latent dirichlet allocation," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 1, pp. 28–32, 2010.

[117] C. Vaduva, I. Gavat, and M. Datcu, "Deep learning in very high resolution remote sensing image information mining communication concept," in *Proc. European Signal Processing Conf.*, Bucharest, Romania, 2012, pp. 2506–2510.

[118] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *Proc. Int. Conf. Learning Representations*, Scottsdale, AZ, 2013, pp. 1–16.

[119] B. Zhao, Y. Zhong, and L. Zhang, "Scene classification via latent dirichlet allocation using a hybrid generative/discriminative strategy for high spatial resolution remote sensing imagery," *Remote Sens. Lett.*, vol. 4, no. 12, pp. 1204–1213, 2013.

[120] C. Vaduva, I. Gavat, and M. Datcu, "Latent dirichlet allocation for spatial analysis of satellite images," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 5, pp. 2770–2786, 2013.

[121] O. A. B. Penatti, K. Nogueira, and J. A. dos Santos, "Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?" in *Proc. IEEE Int. Conf. Comput. Vision and Pattern Recognit. Workshop on Earth Vision*, Boston, 2015, pp. 44–51.

[122] D. Marmanis, M. Datcu, T. Esch, and U. Stilla, "Deep learning earth observation classification using ImageNet pretrained networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 1, pp. 105–109, 2016.

[123] F. Hu, G. S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sens.*, vol. 7, no. 11, pp. 14,680–14,707, 2015.

[124] F. P. S. Luus, B. P. Salmon, F. van den Bergh, and B. T. J. Maharaj, "Multiview deep learning for land-use classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2448–2452, 2015.

[125] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. ACM Int. Conf. Adv. Geograph. Inf. Syst.*, San Jose, CA, 2010, pp. 270–279.

GRS