Unsupervised Spatial–Spectral Feature Learning by 3D Convolutional Autoencoder for Hyperspectral Classification

Shaohui Mei, *Member, IEEE*, Jingyu Ji, Yunhao Geng, Zhi Zhang, Xu Li, *Member, IEEE*, and Qian Du, *Fellow, IEEE*

***Abstract*— Feature learning technologies using convolutional neural networks (CNNs) have shown superior performance over traditional hand-crafted feature extraction algorithms. However, a large number of labeled samples are generally required for CNN to learn effective features under classification task, which are hard to be obtained for hyperspectral remote sensing images. Therefore, in this paper, an unsupervised spatial–spectral feature learning strategy is proposed for hyperspectral images using 3-Dimensional (3D) convolutional autoencoder (3D-CAE). The proposed 3D-CAE consists of 3D or elementwise operations only, such as 3D convolution, 3D pooling, and 3D batch normalization, to maximally explore spatial–spectral structure information for feature extraction. A companion 3D convolutional decoder network is also designed to reconstruct the input patterns to the proposed 3D-CAE, by which all the parameters involved in the network can be trained without labeled training samples. As a result, effective features are learned in an unsupervised mode that label information of pixels is not required. Experimental results on several benchmark hyperspectral data sets have demonstrated that our proposed 3D-CAE is very effective in extracting spatial–spectral features and outperforms not only traditional unsupervised feature extraction algorithms but also many supervised feature extraction algorithms in classification application.**

***Index Terms*— Convolutional neural network (CNN), feature learning, hyperspectral, spatial–spectral.**

I. Introduction

HYPERSPECTRAL imaging technology, which collects electromagnetic spectral information in hundreds of contiguous narrow bands, can identify ground objects according to their unique spectral characteristics. Taking advantage of such rich spectral information, the hyperspectral image classification task, which classifies all the pixels into different categories, has been widely used in many applications, such as land-cover mapping, mineral exploration, water pollution detection, and so on [1], [2]. However, raw hyperspectral images always suffer from spectral variations caused by sensor noise and changes in illumination, environmental, atmospheric, and temporal conditions [3]. Such within-class variation degrades the performance of classification a lot [4]. Therefore, feature extraction is usually performed as a preprocessing step to enhance the separability between various classes in hyperspectral classification tasks.

During the past decades, many strategies have been proposed to extract effective features prior to classification tasks. According to whether labeled information is used or not, feature extraction can be classified into two categories: supervised and unsupervised methods. In the supervised feature extraction methods, samples with known class labels are required to enhance discriminability among different classes, in which the linear discriminant analysis (LDA) [5] and nonparametric weighted feature extraction (NWFE) [6] are two typical representatives. Many variants of these two methods have also been proposed in recent years, such as modified Fisher's LDA [7], regularized LDA [8], modified NWFE using spatial and spectral information [9], and kernel NWFE [10].

The unsupervised feature extraction algorithms automatically extract features from raw data without labeled information. One of the well-known unsupervised methods is the principal component analysis (PCA), which has been widely used for hyperspectral image processing [11]. A tensorial version of PCA has also been proposed to extract spectral-spatial features of hyperspectral images [12]. Many manifold learning-based methods have been applied to reduce the dimensionality of hyperspectral images [13], such as locally linear embedding [14], Laplacian eigenmap [15], and local tangent space alignment [16]. By considering spatial information around the data points, these local methods can preserve local spatial neighborhood and detect the manifold embedded in a high-dimensional feature space. Their linear approximations, such as neighborhood preserving embedding (NPE) [17], locality preserving projection (LPP) [18], and linear local tangent space alignment (LLTSA) [19], were also

Manuscript received December 12, 2018; revised February 24, 2019; accepted March 28, 2019. Date of publication April 22, 2019; date of current version August 27, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61671383 and Grant 61301235, in part by the Fundamental Research Funds for the Central Universities under Grant 3102018AX001, in part by the Natural Science Foundation of Shaanxi Province under Grant 2018JM6005, and in part by the China Postdoctoral Science Foundation under Grant 2014M550872. *(Corresponding author: Shaohui Mei.)*

S. Mei, J. Ji, Y. Geng, and X. Li are with the School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710129, China (e-mail: meish@nwpu.edu.cn).

Z. Zhang is with the State Key Laboratory of Remote Sensing Science, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing 100101, China.

Q. Du is with the Department of Electrical and Computer Engineering and the Geosystems Research Institute, Mississippi State University, Starkville, MS 39762 USA.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2019.2908756

applied to feature extraction of hyperspectral images [20]. In addition, graph-based discriminant analysis-based methods were also proposed, e.g., graph-based discriminant analysis with spectral similarity (GDA-SS) [21] and sparse and low-rank graph-based discriminant analysis (SLGDA) [22]. Laplacian-regularized collaborative graph-based discriminant analysis (LapCGDA) framework was also proposed in [23], in which a Laplacian graph of data manifold is incorporated into the CGDA [24].

The aforementioned feature extraction algorithms, namely, PCA, manifold learning, LDA, and so on, extract a fixed pattern of features using a small amount of adjustable parameters from the original data, mainly taking advantage of human ingenuity and prior knowledge [25], [26]. Different from these hand-crafted feature extraction algorithms [27], [28], feature learning methods can automatically learn effective features from the data itself. Owning to rapid developments in deep learning, feature learning by a deep neural network (DNN) has been developed to learn effective features using a huge amount of data in many applications, such as image classification [29]–[31], object detection [32], [33], and so on. As a typical deep learning technique for feature learning, convolutional neural network (CNN) often contains millions of parameters to be learned, e.g., VGG16 [34]. When these parameters are well optimized under a classification task, both feature quality and classification performance can be enhanced.

Recently, deep learning-based techniques have also been applied to hyperspectral image processing. Hu *et al.* [35] first used a CNN constructed by a spectral convolution operator for hyperspectral image classification (denoted as 1-dimensional (1D)-CNN in this paper). Makantasis *et al.* [36] integrated spatial–spectral inputs into CNN (randomized PCA-CNN) for classification. Li *et al.* [37] also proposed to use CNN to classify pixel pairs constructed from local neighborhood and assigned class labels by majority voting. Liu *et al.* [38] proposed a Siamese CNN (S-CNN) that adopted a margin ranking loss function to guarantee a low intraclass and high interclass variability. As for feature learning of hyperspectral images, Mei *et al.* [39] first proposed a sensor-specific spatial–spectral feature learning concept using CNN techniques, including the ability of feature extraction, transferring, and fine-tuning to different images acquired by the same sensor. Zhao and Du [40] fused the spectral feature extracted by a balanced local discriminant embedding algorithm and spatial feature learned in a CNN for hyperspectral classification. Although these deep learning-based methods have achieved satisfying performance in feature learning and classification, a large amount of labeled samples are required to train these DNNs under a supervised manner by classification task. Due to the difficulty in obtaining labeled training samples in hyperspectral images, it is difficult to increase the accuracy of these kinds of DNN-based supervised feature learning methods.

The unsupervised feature learning using DNNs has also gained much attention. For example, the highly efficient enforcing population and lifetime sparsity (EPLS) algorithm [41] is used to train DNNs in greedy layerwise fashion for unsupervised learning of sparse features of hyperspectral images [42]. Autoencoder (AE) is an artificial neural network used for learning a valid encoding of data in an unsupervised manner [26], [43]. It learns a representation of input samples by reconstructing their input patterns with a minimum reconstruction error. Deep AE (DAE) was first used in hyperspectral image classification and feature learning by Chen *et al.* [44], in which PCA was adopted for dimension reduction in spectral dimension and then flatten method was used to arrange "neighbor region" as a 1D vector to integrate spatial–spectral information. The stacked sparse AE (SSAE) first used AE for sparse spectral feature learning and multiscale spatial feature learning respectively and then fused these two kinds of feature for classification [45]. In these two algorithms, the spatial information may be flattened when using PCA for dimensionality reduction. An improved version of AE, namely, spatial-updated DAE (SDAE), was proposed by Ma *et al.* [46], in which sample similarity was considered by adding a regularization term in the energy function and features were updated by integrating contextual information. The 3D convolution has also been adopted in AE to explore the spatial context for feature extraction [47]. Although these AE-based techniques extract effective spatial–spectral features, spatial information is not sufficiently explored in the network. In this paper, by extending our preliminary work in [47], unsupervised feature learning by a 3D convolutional AE (3D-CAE) is proposed, in which only 3D or elementwise operations, such as 3D convolution, 3D pooling, 3D batch normalization, and parametric rectified linear unit (PReLU) [48], are used to maximally explore spatial structure information for spatial–spectral feature extraction. It should be noted that the proposed network is trained by developing a companion 3D convolutional decoder network to reconstruct the input to the proposed 3D-CAE, by which labeled samples is not required in the training process. As a result, spatial–spectral structure information can be encoded, and effective spatial–spectral features are learned in an unsupervised mode. Finally, extensive experiments on three benchmark hyperspectral data sets are conducted to demonstrate the effectiveness of the proposed 3D-CAE for unsupervised spatial–spectral feature learning.

In summary, the main contributions of this paper are twofold.

1) Unsupervised feature learning is conducted using 3D-CAE by which labeled samples are not required in the feature learning process. Instead, the input samples to the proposed 3D-CAE are used as ground truth to train the parameters involved in the 3D-CAE. Such unsupervised feature learning is especially useful for hyperspectral applications where training samples are rare and difficult to be obtained.
2) The structure information in hyperspectral images is preserved by constructing an AE using 3D and elementwise operations only, e.g., 3D convolution, 3D pooling, 3D batch normalization, and so on. Thus, the proposed 3D-CAE is very efficient in learning spatial–spectral features since all the flatten operations (e.g., fully connection layer) in traditional AE-based networks [44], [46], [47] are excluded.

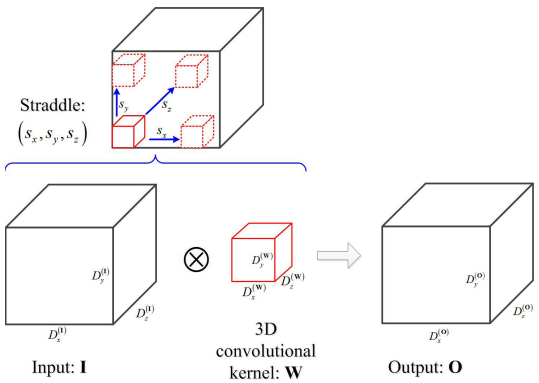


Fig. 1. Illustration of 3D convolution.

The remainder of this paper is organized as follows. Section II presents the proposed 3D-CAE for unsupervised spatial–spectral feature learning. Section III reports and discusses experimental results over three benchmark hyperspectral data sets. Finally, conclusions are drawn in Section IV.

II. Proposed 3D-CAE for Unsupervised Spatial–Spectral Feature Learning

A. 3D Operations for CAE

A hyperspectral image is represented by a 3D cube, which contains a 2-dimensional spatial context and 1D spectral information. In order to well explore both spatial context and spectral discrimination simultaneously, only 3D or elementwise operations is adopted in the proposed 3D-CAE for hyperspectral images, including 3D convolution, 3D deconvolution, 3D pooling, 3D batch normalization, and elementwise PReLU function.

1) 3D Convolution: As shown in Fig. 1, for an input $\mathbf{I} \in \mathscr{R}^3$ of size $D_x^{(\mathbf{I})} \times D_y^{(\mathbf{I})} \times D_z^{(\mathbf{I})}$, when a 3D convolution is applied using a kernel $\mathbf{W} \in \mathscr{R}^3$ of size $D_x^{(\mathbf{W})} \times D_y^{(\mathbf{W})} \times D_z^{(\mathbf{W})}$ ($D_x^{(\mathbf{W})} \leq D_x^{(\mathbf{I})}$, $D_y^{(\mathbf{W})} \leq D_y^{(\mathbf{I})}$, and $D_z^{(\mathbf{W})} \leq D_z^{(\mathbf{I})}$), its output is defined as

$$O^{x,y,z} = b + \sum_{p=0}^{D_x^{(\mathbf{W})}-1} \sum_{q=0}^{D_y^{(\mathbf{W})}-1} \sum_{r=0}^{D_z^{(\mathbf{W})}-1} W^{p,q,r} I^{x\cdot s_x+p, y\cdot s_y+q, z\cdot s_z+r}$$
$$x = 1, 2, \ldots, D_x^{\mathbf{O}},\ y = 1, 2, \ldots, D_y^{\mathbf{O}} \tag{1}$$

and

$$z = 1, 2, \ldots, D_z^{\mathbf{O}}$$

where $O^{x,y,z}$ denotes the (x, y, z)th element in the output $\mathbf{O} \in \mathscr{R}^3$, (s_x, s_y, s_z) represents the size of stride in three dimensions, b denotes the bias, $D_x^{\mathbf{O}}$, $D_y^{\mathbf{O}}$, and $D_z^{\mathbf{O}}$ represent the sizes of output $\mathbf{O}$ and are defined as

$$D_x^{\mathbf{O}} = \left\lfloor \frac{D_x^{\mathbf{I}} - D_x^{(\mathbf{W})}}{s_x} \right\rfloor + 1$$
$$D_y^{\mathbf{O}} = \left\lfloor \frac{D_y^{\mathbf{I}} - D_y^{(\mathbf{W})}}{s_y} \right\rfloor + 1$$
$$D_z^{\mathbf{O}} = \left\lfloor \frac{D_z^{\mathbf{I}} - D_z^{(\mathbf{W})}}{s_z} \right\rfloor + 1 \tag{2}$$

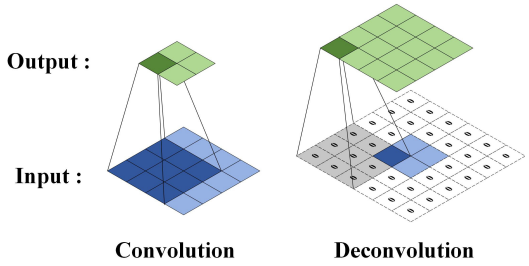


Fig. 2. Comparison of convolution and deconvolution.

where "$\lfloor \cdot \rfloor$" represents the round-to-zero process.

When such 3D convolution is applied to a hyperspectral image cube, spatial–spectral features can be extracted since 3D convolution is conducted in both spatial and spectral domains simultaneously. In general, dozens of 3D convolution kernels are stacked in just one layer to explore different kinds of spatial–spectral features in a local cube, producing dozens of feature cubes. When several such 3D convolution layers are connected sequentially, the 3D convolution should be conducted with an extra fixed dimension to handle these inputs from multiple feature cubes simultaneously. Therefore, in the proposed feature learning of hyperspectral images, 3D convolution kernel is defined as $\mathbf{W} \in \mathscr{R}^4$ of size $D_x \times D_y \times D_z \times D$, where the extra fourth dimension D represents the number of 3D feature cubes input to the convolutional layer. Suppose the input to the ith convolution layer is defined as $\mathbf{I}_i \in \mathscr{R}^4$ of size $D_x^{(\mathbf{I})_i} \times D_y^{(\mathbf{I})_i} \times D_z^{(\mathbf{I})_i} \times D_i$. Without loss of generality, if the original hyperspectral image cube is fed as input, $D = 1$. As a result, the 3D convolution in the ith convolution layer is represented as

$$O_{i,j}^{x,y,z} = b_{i,j} + \sum_{k=0}^{D_i-1} \sum_{p=0}^{D_x-1} \sum_{q=0}^{D_y-1} \sum_{r=0}^{D_z-1} W_{j,k}^{p,q,r} I_{i,k}^{x\cdot s_x+p, y\cdot s_y+q, z\cdot s_z+r} \tag{3}$$

where subscripts "i" and "j" index the convolutional layer and the convolutional kernels in a layer, respectively. Obviously, the structure of input is not flattened in such 3D convolution.

2) 3D Deconvolution: The deconvolution, also known as transposed convolution, can be viewed as the reverse of the convolutional layer. Being capable of mapping the input from a low-dimensional space to a high-dimensional space, it is often adopted in CNNs for image or voxel reconstruction in many applications, such as image semantic segmentation [49], style transfer [50], and image inpainting [51]. As shown in Fig. 2, deconvolution is realized by a padding process and convolution, in which the input is first padded with zero to generate a middle input that is of larger size than the output and then convolution is filtered on the middle input to generate output. Similarly, in the 3D deconvolution, the input is padded in all the three dimensions [i.e., x, y, and z in (3)] before the 3D convolution is applied.

3) 3D Batch Normalization: Assume that $\mathbf{X}_i (i = 1, 2, \ldots, M_i) \in \mathbb{R}^3$ is a minibatch of inputs, and the output of a 3D batch normalization, denoted as $\mathbf{Y}_i (i = 1, 2, \ldots, M_i) \in \mathbb{R}^3$, is of the same size with $\mathbf{X}_i$, in which M_i denotes the

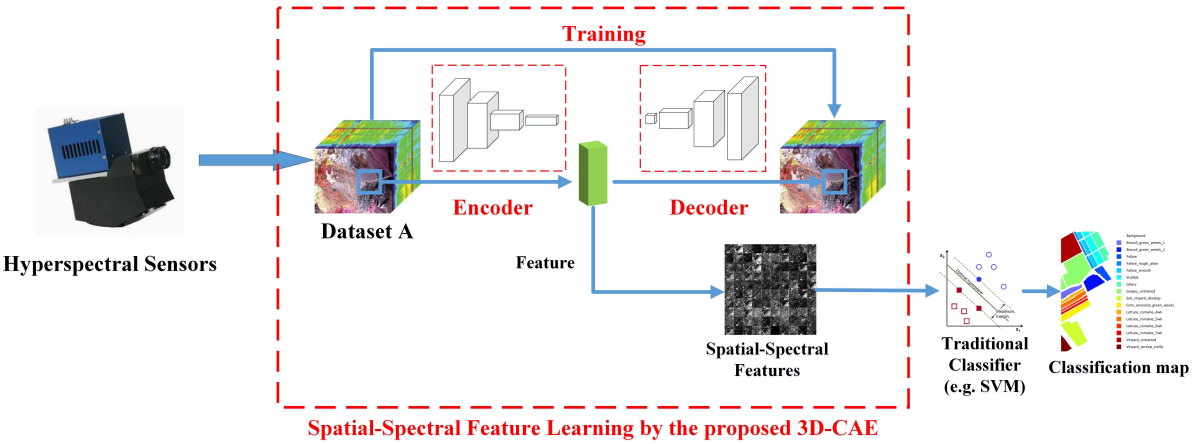


Fig. 3. Framework of the proposed 3D-CAE for unsupervised spatial–spectral feature learning of hyperspectral images.

number of feature maps in the minibatch. The 3D batch normalization is represented as

$$\boldsymbol{Y}_i = \frac{\boldsymbol{X}_i - \text{mean}^{(M_i)}[\boldsymbol{X}]}{\sqrt{\text{Var}^{(M_i)}[\boldsymbol{X}] + \epsilon}} * \boldsymbol{\gamma} + \boldsymbol{\beta}, \quad i = 1, 2, \ldots, M_i \tag{4}$$

where $\text{mean}^{(M_i)}[\boldsymbol{X}]$ and $\text{Var}^{(M_i)}[\boldsymbol{X}]$, respectively, represent the mean and standard deviation of $\mathbf{X}_i$ which are calculated in each of the three dimensions over a minibatch, γ and β are the learnable parameters, and ϵ was set to $1e-5$ as default. During training, this layer stores the mean and variance of all batches. The average of the mean and variance in the training process is used for normalization in the evaluation procedure. Such a strategy enables the network to be used for all kinds of samples without defining a specific value for a certain batch.

4) 3D Pooling: Max pooling layer can reduce the number of training parameters of a CNN [35]. Traditionally, 3D max pooling is usually used for spatial–temporal feature learning in video action recognition and detection task [52], [53]. Similarly, 3D max pooling can be used for spatial–spectral feature learning of hyperspectral images. In a DNN, dozens of 3D convolution kernels are applied to the same input in a layer to explore different features, and then pooling is used to summarize these features such that the dominating feature is retained in just one feature. Suppose pooling is applied to features extracted using T 3D convolution kernels $\mathbf{W}_t, t = 1, 2, \ldots, T$, the 3D max pooling is defined as

$$O^{x,y,z} = \max_t F_t^{x,y,z} \tag{5}$$

where $F_t^{x,y,z}$ represents the features extracted using 3D convolution kernels $\mathbf{W}_t$ and $O^{x,y,z}$ represents the feature at position (x, y, z) after 3D max pooling. It is observed that the structure information of input to the 3D convolutional layer does not flatten in such 3D max pooling.

B. Proposed 3D-CAE for Unsupervised Spatial–Spectral Feature Learning

When using the proposed 3D-CAE for unsupervised spatial–spectral feature learning, it is first trained with an "encoding–decoding" step in which a hyperspectral data cube is provided to the 3D-CAE for feature learning and then reconstructed using the learned features. After the network being trained to well recover the hyperspectral data cube using learned features, it can be used to extract spatial–spectral features. As shown in Fig. 3, the proposed 3D-CAE for unsupervised spatial–spectral feature learning is conducted by the following three steps: 1) constructing encoder for spatial–spectral feature learning; 2) constructing decoder to train the encoder under reconstruction task; and 3) extracting unsupervised features using the encoder of the proposed 3D-CAE from hyperspectral images.

CAE can automatically learn effective features under reconstruction task without labeled samples. Therefore, it has been applied for unsupervised feature extraction of hyperspectral images [44]. However, the flatten method in [44] to simply arrange different neighboring pixels as a vector loses the spatial structure information that has been demonstrated to be crucial in hyperspectral applications. Therefore, as shown in Fig. 4, a novel multilayer 3D-CAE is proposed for unsupervised feature learning of hyperspectral images, in which the neighboring cubes of pixels are directly fed into the encoder of the proposed 3D-CAE without any other handcraft operators and 3D convolution is then used to explore the spatial context for feature learning.

In this paper, for a pixel $\mathbf{p}_{x,y} \in \mathbb{R}^{c\times 1}$ located at (x, y) on the image plane, a square patch of size $s \times s$ centered at (x, y) is considered as its spatial context, where c is the number of channels (spectral bands). Therefore, in the proposed 3D-CAE, in order to fully explore spatial context of $\mathbf{p}_{x,y}$, its spatial neighborhood $\mathbf{I}_{(x,y)} \in \mathbb{R}^{s\times s\times c}$ is directly fed to the encoder without any other transformations.

As shown in Fig. 4, the encoder of the proposed 3D-CAE stacks 3D convolutional layers after the input layer to learn spatial–spectral features within a spatial neighborhood, in which 3D convolution is applied in spatial and spectral domain simultaneously. In the encoder, 3D batch normalization is adopted to normalize the features generated by each 3D convolutional layer such that feature weights are in the same range. Thus, a large learning rate can be used to speed up the training process [54]. After multiple 3D convolutional

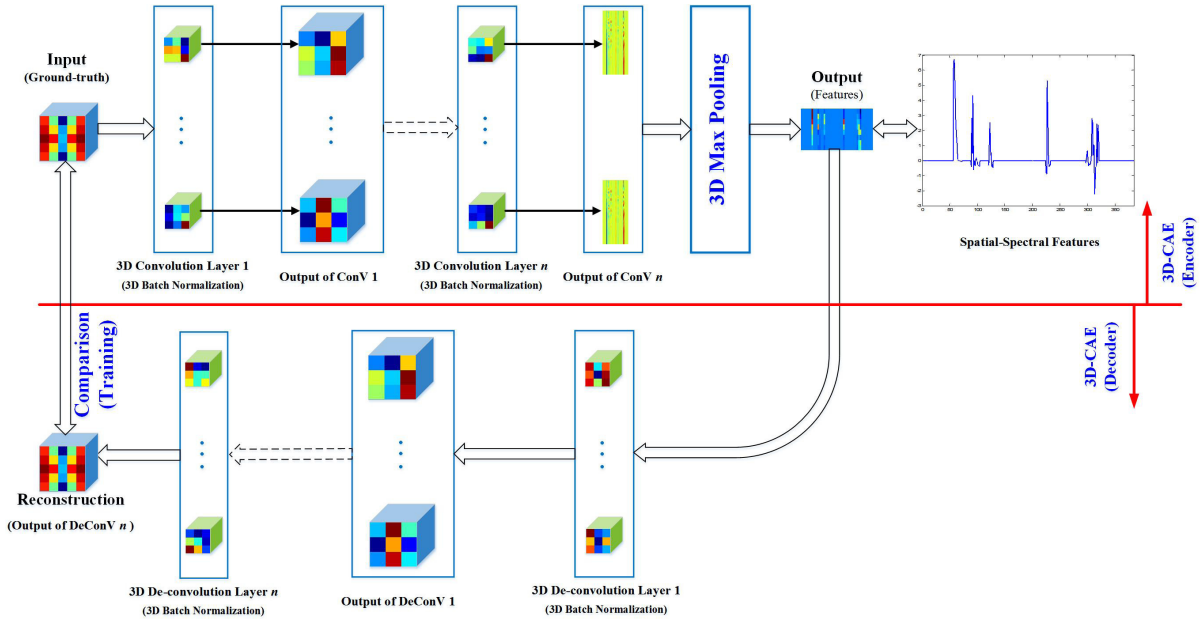


Fig. 4. Architecture of the proposed 3D-CAE for feature extraction of hyperspectral images.

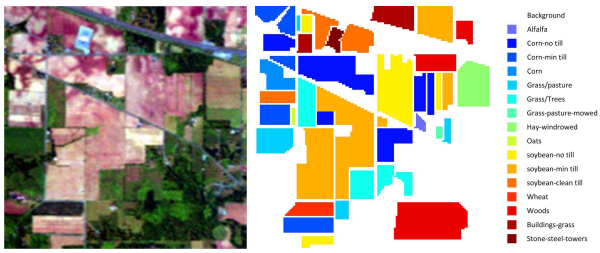
layers, 3D max pooling is finally utilized to gather local features explored by different 3D convolution filters. As a result, a global spatial–spectral feature is produced as the features extracted by the encoder of the proposed 3D-CAE.

In order to train the encoder of the proposed 3D-CAE for feature extraction, as shown in Fig. 4, a companion 3D convolutional decoder network is designed to reconstruct the input hyperspectral cubes from the features extracted by the encoder. By such a training strategy, a large amount of labeled samples are not required. In this paper, the 3D convolutional decoder network owns a mirrored structure with the encoder. However, 3D transposed convolutional layers are stacked to reconstruct hyperspectral cube from encoding features, instead of the 3D convolution layers in the encoder to extract features. The backpropagate (BP) method is used to trained the network with a loss function designed as

$$\text{loss} = \frac{1}{s \times s \times c} \sum_{x=0}^{s-1} \sum_{y=0}^{s-1} \sum_{z=0}^{c-1} (I^{x,y,z} - \hat{I}^{x,y,z})^2 + \frac{\lambda}{2} ||\mathbf{W}||_2^2 \quad (6)$$

where $I^{x,y,z}$ represents the value at position (x, y, z) of the input $\mathbf{I} \in \mathbb{R}^{s \times s \times c}$ to the encoder, $\hat{I}^{x,y,z}$ represents its reconstructed value by the 3D convolutional decoder network in the training process, $\mathbf{W}$ consists of the weights in all the layers, and λ is a hyperparameter set as 0.0005. The first term in (6) measures the reconstruction error, while the second regularization term forces the weights close to the origin. Such an attenuation term of weights can greatly reduce generalization errors over testing samples.

In both the encoder and its companion 3D convolutional decoder network of the proposed 3D-CAE, the PReLU activation function [48], which is an elementwise operation without flattening spatial structure, is adopted for all convolutional and deconvolutional layers since it can improve model fitting without extra computational cost and overfitting risk [48].



(a) (b)

Fig. 5. (a) Pseudocolor image of the Indian Pine data set. (b) Ground-truth classification map of the Indian Pine data set.

When the encoder is well trained by its companion 3D convolutional decoder network under a reconstruction task of hyperspectral images, it is then used independently to extract spatial–spectral features of pixels in the image for other applications, such as classification and object detection.

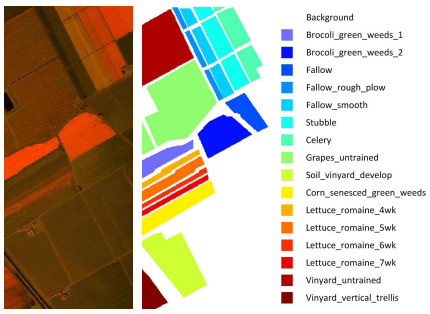
III. Experiments

In this section, extensive experiments are conducted to verify the performance of the proposed 3D-CAE for unsupervised spatial–spectral feature learning of hyperspectral images.

A. Experimental Results Over Data Sets Acquired by AVIRIS Sensor

In this experiment, two benchmark data sets acquired by the AVIRIS sensor, i.e., the Indian Pine data set and the Salinas Valley data set, are adopted for evaluation.[1] As shown in Fig. 5(a), the Indian Pine data set contains 145×145 pixels with a ground resolution of 17 m. According to the ground-truth classification map of the Indian Pine data set shown

[1] Available online from http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes



(a) (b)

Fig. 6. (a) Pseudocolor image of the Salinas Valley data set. (b) Ground-truth classification map of the Salinas Valley data set.

TABLE I

CLASS LABELS AND TRAIN–TEST DISTRIBUTION OF SAMPLES FOR THE INDIAN PINES DATA SET

#	Class	Training	Testing
1	Alfalfa	5	41
2	Corn-no till	143	1285
3	Corn-min till	83	747
4	Corn	24	213
5	Grass/pasture	48	435
6	Grass/Trees	73	657
7	Grass-pasture-mowed	3	25
8	Hay-windrowed	48	430
9	Oats	2	18
10	soybean-no till	97	875
11	soybean-min till	246	2209
12	soybean-clean till	59	534
13	Wheat	21	184
14	Woods	127	1140
15	Buildings-grass	39	347
16	Stone-steel-towers	9	84
	Total	1027	9222

TABLE II

CLASS LABELS AND TRAIN–TEST DISTRIBUTION OF SAMPLES FOR THE SALINAS DATA SET

#	Class	Training	Testing
1	Broccoli green weeds 1	100	1909
2	Broccoli green weeds 2	186	3540
3	Fallow	99	1877
4	Fallow rough plow	70	1324
5	Fallow smooth	134	2544
6	Stubble	198	3761
7	Celery	179	3400
8	Grapes untrained	564	10707
9	Soil vineyard develop	310	5893
10	Corn senesced green weeds	164	3114
11	Lettuce romaine, 4 wk	53	1015
12	Lettuce romaine, 5 wk	96	1831
13	Lettuce romaine, 6 wk	46	870
14	Lettuce romaine, 7 wk	54	1016
15	Vineyard untrained	363	6903
16	Vineyard vertical trellis	90	1717
	Total	2706	51421

in Fig. 5(b), 16 different land-cover classes of agriculture are mainly contained in this area as listed in Table I. The Salinas Valley data set, which is shown in Fig. 6, contains 512×217 pixels with a ground resolution of 3.7 m. As shown in Table II, 16 classes such as vegetables, bare soils, and vineyard fields are adopted.

The proposed 3D-CAE is designed, trained, and tested using keras framework.[2] The parameter settings of the proposed 3D-CAE are listed in Table III. The network is trained using "adagrad" on a Geforce GTX 1080 GPU for 200 epochs, with a learning rate of 0.01 and minibatch of 32. In order to learn the spatial–spectral feature of a pixel, pixels in its 5×5 neighborhood are fed to the network, in which the border pixel is padded by mirror. In addition, the training set and validation set are divided by a ratio of 1:9.

After the network is well trained under the reconstruction task, it is then used to extract spatial–spectral features of pixels by flattening the output of "Pool2" layer as feature vectors. In order to evaluate the effectiveness of these learned features by the proposed 3D-CAE, the traditional support vector machine (SVM) classifier with radial basis function kernel, which is implemented using the LIBSVM package [55], is used for classification. In this experiment, as shown in Tables I and II, 10% and 5% samples of each class are used for training and others are used for the validation in the Indian Pine and Salinas Valley data sets, respectively. In addition, a tenfold cross-validation strategy is used for evaluation.

In this experiment, both unsupervised feature reduction methods and supervised feature reduction methods are adopted for comparison. For unsupervised feature reduction, PCA, NPE [17], LPP [58], DAE [44], tensorial PCA (TPCA) [12], SSAE [45], and EPLS [42] are adopted, whereas four supervised feature reduction methods, including LDA [7], local Fisher's discriminant analysis (LFDA) [56], sparse graph-based discriminant analysis (SGDA) [57], and SLGDA [22] are adopted for comparison. Note that the results of LFDA, SGDA, and SLGDA are selected from [22]. The PCA, NPE, LPP, and LDA are implemented according to the published code online,[3] in which the reduced dimension of features in PCA is set as 40, and the size of neighborhood in NPE and LPP is set as 5×5. The TPCA is implemented as explained in [12], in which 10% of pixels are randomly selected to construct tensor of correlation. In the DAE [44], the dimensionality of spectral dimension is reduced to 4 by PCA and the size of "neighbor region" is set as 5×5. In multiscale spatial feature learning of SSAE [45], the spatial scale is set as 3, 5, and 7. In EPLS [42], the spatial neighbor is set as 5×5. In addition, two typical CNNs that learned effective features in a supervised manner, i.e., 1D-CNN in [35] and S-CNN [38], are also adopted for comparison. In these two CNN based algorithms, the number of samples adopted for training in the feature extraction step is identical to that used to train the subsequent classifier. The quantitative results over these two data sets, evaluated by average accuracy (AA) and overall accuracy (OA), are listed in Tables IV and V, respectively. Their corresponding visual results are shown in Figs. 7 and 8, respectively. The observation is as follows.

1) The features learned in the proposed 3D-CAE nearly outperform all the other considered features over these two data sets, including both supervised and unsupervised feature extraction algorithms. The proposed

[2] https://keras.io/

[3] http://www.cad.zju.edu.cn/home/dengcai/Data/DimensionReduction.html

TABLE III

PARAMETER SETTINGS OF THE PROPOSED 3D-CAE WHEN APPLIED TO DATA SETS ACQUIRED BY THE AVIRIS SENSOR

	input size	kernel size	strides	output size
Conv1	$224 \times 5 \times 5 \times 1$	$24 \times 3 \times 3 \times 24$	$1 \times 1 \times 1 \times 1$	$201 \times 3 \times 3 \times 24$
Bn1	$201 \times 3 \times 3 \times 24$	—	—	$201 \times 3 \times 3 \times 24$
Conv2	$201 \times 3 \times 3 \times 24$	$24 \times 3 \times 3 \times 48$	$1 \times 1 \times 1 \times 1$	$178 \times 1 \times 1 \times 48$
Bn2	$178 \times 1 \times 1 \times 48$	—	—	$178 \times 1 \times 1 \times 48$
Pool2	$178 \times 1 \times 1 \times 48$	$18 \times 1 \times 1$	$18 \times 1 \times 1$	$9 \times 1 \times 1 \times 48$
Deconv3	$9 \times 1 \times 1 \times 48$	$9 \times 3 \times 3 \times 24$	$22 \times 1 \times 1 \times 1$	$198 \times 3 \times 3 \times 24$
Bn3	$198 \times 3 \times 3 \times 24$	—	—	$198 \times 3 \times 3 \times 24$
Deconv4	$198 \times 3 \times 3 \times 24$	$27 \times 3 \times 3 \times 24$	$1 \times 1 \times 1 \times 1$	$224 \times 5 \times 5 \times 1$
Bn4	$224 \times 5 \times 5 \times 1$	—	—	$224 \times 5 \times 5 \times 1$

TABLE IV

CLASSIFICATION ACCURACY OF DIFFERENT FEATURE EXTRACTION ALGORITHMS OVER THE INDIAN PINES DATA SET

Class	Supervised Feature Extraction						Unsupervised Feature Extraction							
	LDA [7]	LFDA [56]	SGDA [57]	SLGDA [22]	1D-CNN [35]	S-CNN [38]	PCA	NPE [17]	LPP [58]	DAE [44]	TPCA [12]	SSAE [45]	EPLS [42]	3DCAE
1	58.54	29.63	42.59	39.62	43.33	83.33	39.02	68.29	75.61	35.52	60.97	56.25	58.72	**90.48**
2	69.88	75.59	80.89	85.56	73.13	81.41	72.30	65.21	64.59	61.13	87.00	69.58	59.91	**92.49**
3	65.86	75.42	65.71	74.82	65.52	74.02	72.02	63.59	69.75	53.28	**94.51**	75.36	71.34	90.37
4	73.71	58.12	64.10	49.32	51.31	71.49	55.87	40.85	33.80	63.53	79.34	64.58	74.31	**86.90**
5	90.32	95.17	94.57	95.35	87.70	90.11	93.09	82.95	85.25	63.26	93.08	88.81	**97.95**	94.25
6	92.09	96.12	**98.39**	95.59	95.10	94.06	94.67	93.91	89.35	88.31	96.34	87	96.44	97.07
7	**96.00**	11.53	50.00	36.92	56.92	84.61	80.00	64.00	76.00	30.98	76.00	90	54.02	91.26
8	98.14	93.87	**99.80**	99.75	96.64	98.37	98.37	97.67	90.70	95.65	99.76	89.72	88.99	97.79
9	11.11	0.00	0.00	5.00	28.89	33.33	88.89	33.33	27.78	48.89	**100.00**	**100**	58.89	75.91
10	73.80	80.89	59.30	69.11	75.12	86.05	74.49	65.90	74.14	75.15	79.51	77.19	73.10	**87.34**
11	55.41	83.02	84.44	89.91	83.49	82.98	69.58	63.24	61.70	78.78	85.42	77.58	70.78	**90.24**
12	76.92	86.32	77.04	86.78	67.55	73.40	65.29	54.97	54.60	49.03	84.24	72.00	57.51	**95.76**
13	91.30	79.25	99.09	**99.51**	96.86	87.02	98.37	82.61	85.87	89.97	98.91	87.80	99.25	97.49
14	93.32	88.87	92.50	96.45	96.51	94.38	91.39	89.89	87.79	91.65	**98.06**	93.48	95.07	96.03
15	67.72	60.00	68.68	61.79	39.08	75.57	48.99	40.63	39.48	54.61	87.31	72.36	**91.26**	90.48
16	90.36	53.68	85.26	84.16	89.40	79.76	87.95	83.13	90.36	85.92	96.38	97.22	91.27	**98.82**
AA(%)	76.89	66.72	72.65	73.10	71.66	84.44	76.89	68.14	69.17	66.60	89.31	81.18	77.43	**92.04**
OA(%)	76.88	81.79	80.05	85.19	79.66	80.72	76.88	70.49	70.47	73.16	88.55	79.78	77.18	**92.35**

TABLE V

CLASSIFICATION ACCURACY OF DIFFERENT FEATURE EXTRACTION ALGORITHMS OVER THE SALINAS VALLEY DATA SET

Class	Supervised Feature Extraction						Unsupervised Feature Extraction							
	LDA [7]	LFDA [56]	SGDA [57]	SLGDA [22]	1D-CNN [35]	S-CNN [38]	PCA	NPE [17]	LPP [58]	DAE [44]	TPCA [12]	SSAE [45]	EPLS [42]	3DCAE
1	99.16	99.20	99.65	98.14	97.98	99.55	97.48	99.53	99.69	96.51	99.88	**100.00**	99.99	**100.00**
2	**99.94**	99.19	99.33	99.44	99.25	99.43	99.52	99.66	99.86	98.35	99.49	99.52	99.92	99.29
3	99.79	99.75	99.30	99.29	94.43	98.81	99.41	**99.89**	99.79	95.08	99.04	94.24	98.75	97.13
4	99.77	99.71	99.21	99.57	99.42	97.45	99.77	99.09	98.79	98.57	**99.84**	99.17	98.52	97.91
5	98.98	98.47	99.07	98.06	96.60	97.96	98.70	**99.10**	98.66	97.19	98.96	98.82	98.33	98.26
6	99.89	99.09	99.57	99.32	99.51	99.83	99.65	99.23	99.65	99.50	99.80	**100**	99.92	99.98
7	**99.97**	99.66	99.27	99.33	99.27	99.59	99.94	99.85	99.44	98.73	99.84	99.94	97.69	99.64
8	81.84	86.89	89.78	89.48	86.79	**94.40**	83.90	84.83	82.67	83.83	84.11	80.73	78.86	91.58
9	99.90	97.34	**100.00**	99.65	99.08	98.85	99.97	99.90	99.22	97.67	99.60	99.47	99.54	99.28
10	96.31	95.85	**97.99**	97.94	93.71	97.35	96.89	97.24	97.27	92.55	95.76	92.12	95.98	96.65
11	**99.61**	97.94	99.53	99.06	94.55	97.71	96.84	98.82	99.11	90.89	96.14	96.62	98.60	97.74
12	99.67	99.64	**100.00**	**100.00**	99.59	98.73	99.95	**100.00**	99.51	99.16	99.07	97.75	99.44	98.84
13	99.20	97.38	98.47	97.82	97.50	96.72	99.54	96.78	99.66	96.87	**100.00**	95.81	98.85	99.26
14	96.56	93.18	96.07	90.47	94.08	95.22	97.24	93.70	92.03	95.54	95.74	96.65	**98.56**	97.49
15	73.60	63.04	65.40	69.51	66.52	**95.61**	76.68	74.51	81.21	74.78	79.54	79.73	83.13	87.85
16	98.48	99.00	99.34	99.00	97.48	**99.44**	97.90	98.60	98.19	87.50	98.40	99.12	99.50	98.34
AA(%)	96.42	95.33	96.37	96.01	94.73	97.39	96.46	96.30	96.55	93.92	93.24	95.61	96.55	**97.45**
OA(%)	92.18	91.22	91.82	93.31	91.30	**97.92**	92.87	92.81	93.17	91.04	96.57	92.11	92.35	95.81

3D-CAE offers a similar performance as the S-CNN and TPCA on the Salinas Valley data set in terms of OA.

2) In the Indian Pine data set where all the classes are a little difficult to be discriminated, the superiority of the features learned in the proposed 3D-CAE is more obvious. Of the 16 classes, the proposed 3D-CAE yields the best performance in 7 classes while approaching the best for the other 9 classes.

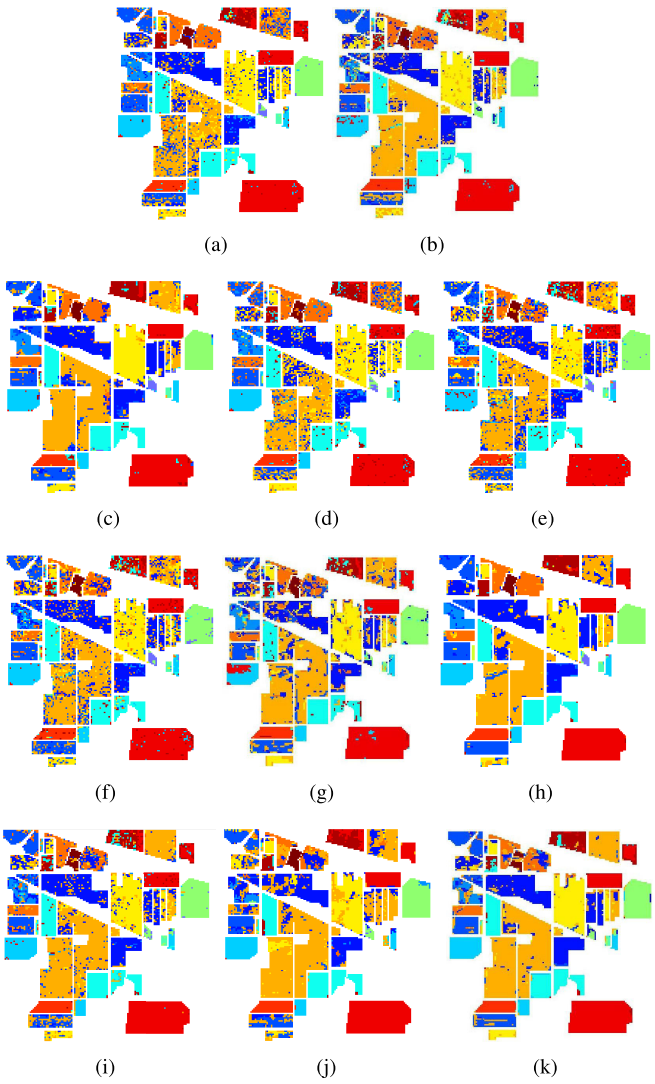


Fig. 7. Classification maps generated by different feature extraction algorithms over the Indian Pine data set. (a) LDA [7]. (b) 1D-CNN [35]. (c) S-CNN [38]. (d) PCA. (e) NPE [17]. (f) LPP [58]. (g) DAE [44]. (h) TPCA [12]. (i) SSAE [45]. (j) EPLS [42]. (k) Proposed 3D-CAE.

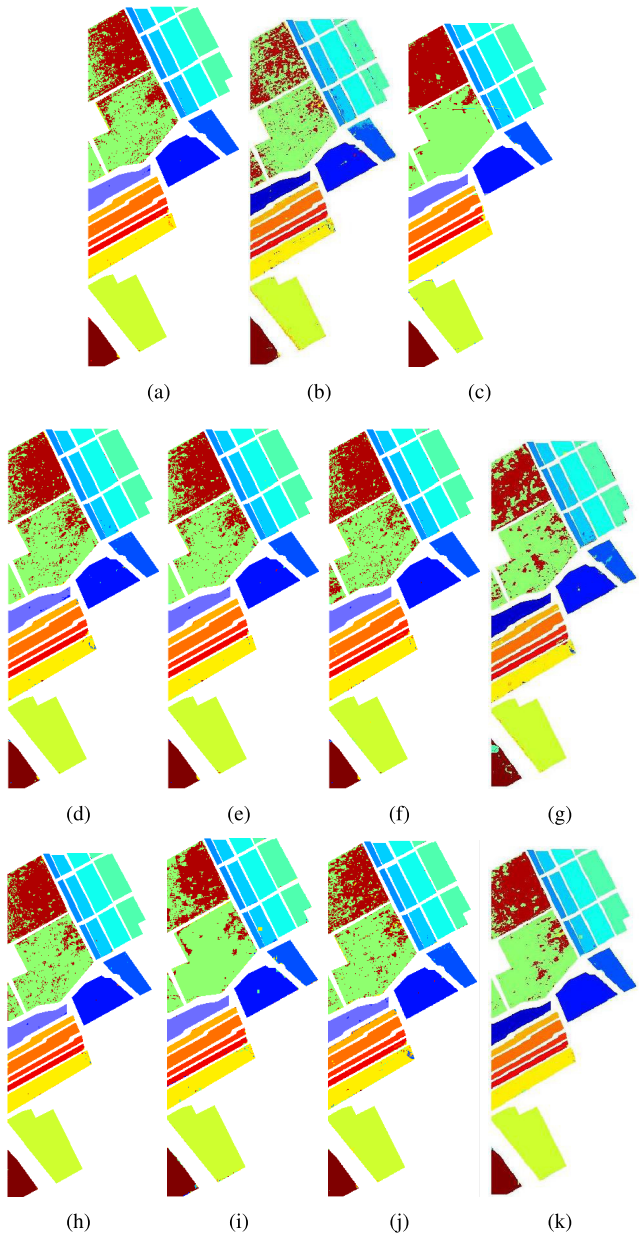
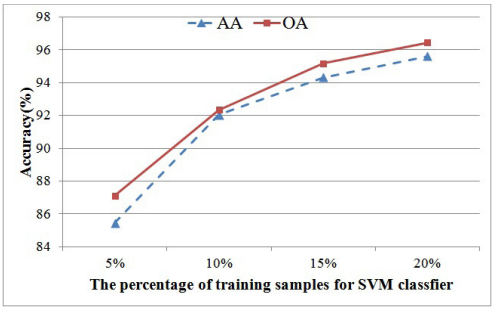


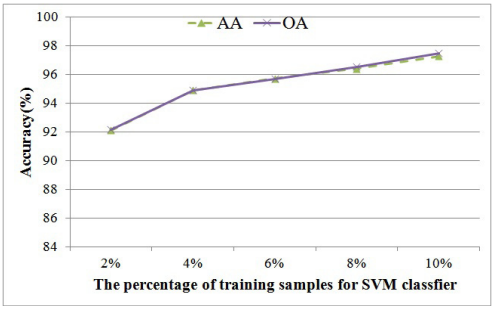
Fig. 8. Classification maps generated by different feature extraction algorithms over Salinas Valley data set. (a) LDA [7]. (b) 1D-CNN [35]. (c) S-CNN [38]. (d) PCA. (e) NPE [17]. (f) LPP [58]. (g) DAE [44]. (h) TPCA [12]. (i) SSAE [45]. (j) EPLS [42]. (k) Proposed 3D-CAE.

3) Although both SSAE and DAE use the same idea of AE as in the proposed 3D-CAE, the spatial context in these two algorithms has been flatten by using PCA for dimensionality reduction before feature extraction. On the contrary, the proposed 3D-CAE significantly improves the performance of the existing AE-based algorithms using 3D or elementwise operations only to retain structure information.
4) The proposed 3D-CAE obviously outperform EPLS over these two data sets, indicating that the end-to-end fashion is more effective than the greedy layerwise fashion to train DNNs.
5) Although the proposed 3D-CAE is trained under reconstruction task without label information, it is even more effective than the CNN using label information. In 1D-CNN [35], only spectral information is used as input and spatial information is absent in the learned features. This also demonstrates that the spatial context is of crucial importance in hyperspectral applications. Without spatial information, the features learned in a supervised CNN are even worse than the spatial–spectral features learned in an unsupervised manner. When spatial information is considered in S-CNN, the performance of supervised learning has been greatly improved.

In order to further evaluate the effectiveness of the spatial–spectral feature learned in the proposed 3D-CAE, the training samples for the SVM classifier are also varied from 5% to 20% for the Indian Pine data set and from 2% to 10% for the Salinas data set, respectively. Fig. 9 shows the classification results with different percentage of training samples over the Indian Pine data set and the Salinas data set, respectively.

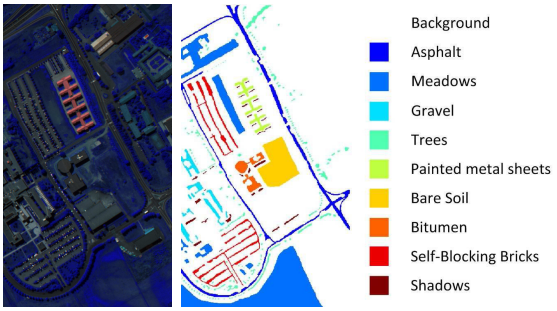


(a)



(b)

Fig. 9. Experimental results of SVM with different percentage of training samples on the feature extracted by the proposed 3D-CAE over (a) Indian Pine data set and (b) Salinas Valley data set.



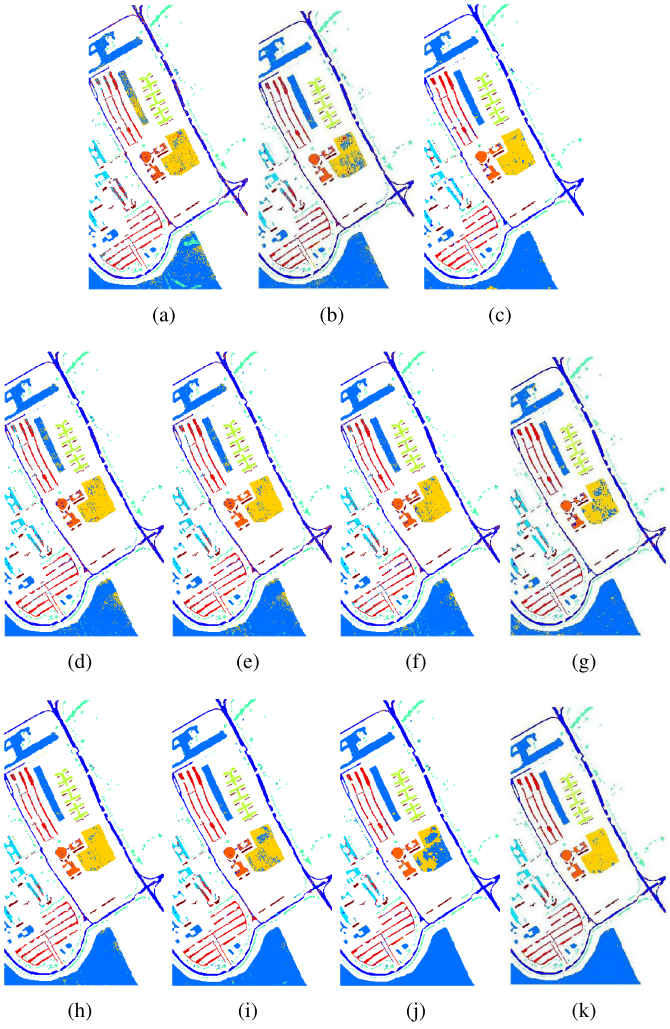
(a) (b)

Fig. 10. (a) Pseudocolor image of the Pavia University data set. (b) Ground-truth classification map of the Pavia University data set.

It is observed that the performance of classification increases steadily when more training samples are used to train the SVM classifier, also demonstrating that the spatial–spectral features learned in the proposed 3D-CAE are very effective for the classification task.

B. Experimental Results Over Data Set Acquired by ROSIS Sensor

The scenes acquired by the ROSIS sensor during a flight campaign over Pavia, Northern Italy, namely, Pavia University, are also selected for evaluation. As shown in Fig. 10(a), Pavia University contains 610×340 pixels. The geometric resolution is 1.3 m. The number of spectral bands is 103. According to the ground-truth maps shown in Fig. 10(b), nine classes are adopted for quantitative evaluation. The structure of the proposed 3D-CAE in this experiment is listed in Table VI. All



(a) (b) (c)

(d) (e) (f) (g)

(h) (i) (j) (k)

Fig. 11. Classification maps generated by different algorithms for Pavia University data set. (a) LDA [7]. (b) 1D-CNN [35]. (c) S-CNN [38]. (d) PCA. (e) NPE [17]. (f) LPP [58]. (g) DAE [44]. (h) TPCA [12]. (i) SSAE [45]. (j) EPLS [42]. (k) Proposed 3D-CAE.

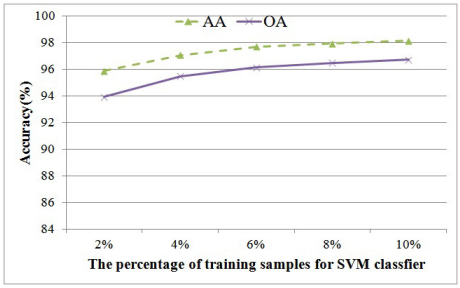


Fig. 12. Experimental results of SVM with different percentage of training samples on the feature extracted by the proposed 3D-CAE over the Pavia University data set.

the other settings in this experiment are the same as that in the previous experiment. The qualitative experimental results are listed in Table VII and their corresponding visual results are shown in Fig. 11. It is observed that the proposed 3D-CAE outperforms all other considered algorithms in terms of OA and approaching the best in terms of AA. The number of

TABLE VI
PARAMETER SETTINGS OF THE PROPOSED 3D-CAE WHEN APPLIED TO DATA BY THE PAVIA UNIVERSITY DATA SET

	input size	kernel size	strides	output size
Conv1	$103 \times 5 \times 5 \times 1$	$11 \times 3 \times 3 \times 24$	$1 \times 1 \times 1 \times 1$	$93 \times 3 \times 3 \times 24$
Bn1	$93 \times 3 \times 3 \times 24$	—	—	$93 \times 3 \times 3 \times 24$
Conv2	$93 \times 3 \times 3 \times 24$	$11 \times 3 \times 3 \times 48$	$1 \times 1 \times 1 \times 1$	$83 \times 1 \times 1 \times 48$
Bn2	$83 \times 1 \times 1 \times 48$	—	—	$83 \times 1 \times 1 \times 48$
Pool2	$83 \times 1 \times 1 \times 48$	$9 \times 1 \times 1$	$9 \times 1 \times 1$	$9 \times 1 \times 1 \times 48$
Deconv3	$9 \times 1 \times 1 \times 48$	$9 \times 3 \times 3 \times 24$	$10 \times 1 \times 1 \times 1$	$90 \times 3 \times 3 \times 24$
Bn3	$90 \times 3 \times 3 \times 24$	—	—	$90 \times 3 \times 3 \times 24$
Deconv4	$90 \times 3 \times 3 \times 24$	$14 \times 3 \times 3 \times 24$	$1 \times 1 \times 1 \times 1$	$103 \times 5 \times 5 \times 1$
Bn4	$103 \times 5 \times 5 \times 1$	—	—	$103 \times 5 \times 5 \times 1$

TABLE VII
CLASSIFICATION ACCURACY OF DIFFERENT FEATURE EXTRACTION ALGORITHMS OVER THE PAVIA UNIVERSITY DATA SET

Class	Supervised Feature Extraction						Unsupervised Feature Extraction							
	LDA [7]	LFDA [56]	SGDA [57]	SLGDA [22]	1D-CNN [35]	S-CNN [38]	PCA	NPE [17]	LPP [58]	DAE [44]	TPCA [12]	SSAE [45]	EPLS [42]	3DCAE
1	78.41	93.45	91.37	94.66	90.93	95.40	90.89	85.33	90.60	93.69	**96.17**	95.72	95.95	95.21
2	83.69	97.36	97.23	97.83	96.94	97.31	93.27	93.42	94.91	96.41	**97.95**	94.13	95.91	96.06
3	73.02	71.41	66.08	77.27	69.43	81.21	82.60	80.94	79.49	71.88	86.50	87.47	**94.33**	91.32
4	93.68	91.00	91.19	93.18	90.32	95.83	92.41	94.43	95.40	96.70	94.84	96.91	**99.28**	98.28
5	**100.00**	97.99	99.33	98.51	99.44	99.91	98.98	99.06	98.83	99.37	**100.00**	99.76	99.92	95.55
6	88.51	87.55	80.31	90.08	73.69	95.29	92.00	92.84	89.89	78.83	94.76	**95.76**	93.57	95.30
7	85.75	80.23	75.26	85.34	83.42	87.05	85.83	93.43	88.04	76.83	91.89	91.18	**98.17**	95.14
8	74.49	86.98	84.11	90.49	83.65	87.35	82.96	84.99	79.58	88.76	89.04	82.47	91.23	**91.38**
9	99.11	99.26	95.14	99.37	98.23	95.66	**100.00**	**100.00**	**100.00**	96.72	98.94	100	99.78	99.96
AA(%)	86.29	89.47	86.67	91.86	87.34	94.75	90.99	91.60	90.75	88.80	95.64	93.71	**96.33**	95.36
OA(%)	83.75	92.77	90.58	94.15	89.99	92.78	91.37	91.15	91.63	91.57	94.45	93.51	95.13	**95.39**

training samples for the SVM classifier also varies from 2% to 10%, and the experimental results are shown in Fig. 12. It is also confirmed that the proposed 3D-CAE is very effective to learn spatial–spectral features under an unsupervised manner.

C. Parameter Sensitivity Analysis

In this section, the performance of the proposed 3D-CAE with various parameters is discussed.

1) Experiment With Different Size of Input: First, the performance of the proposed 3D-CAE with different sizes of spatial context as input is discussed. In the Indian Pine data set, the spatial context input to the 3D-CAE varies from 3×3 to 19×19. For the spatial context of $224 \times n \times n \times 1$, the size of the convolution kernel is set to $24 \times (n-1)/2 \times (n-1)/2 \times 24$, and thus, a larger convolution kernel is considered for border spatial context, such that the spatial dimension of the feature output from the last layer of the encoder is kept as 1×1. All the other parameters are set as that in the previous experiment.

The experimental results of the proposed 3D-CAE with different spatial context as input are shown in Fig. 13. It is observed that the performance of classification first slightly decreases and then increases steadily when the spatial context fed into the 3D-CAE increases. The OA is more than 94% when a spatial context of 17×17 is considered, indicating that the unsupervised spatial–spectral features learned in the proposed 3D-CAE are very effective for classification.

2) Number of Convolutional Layers: In this section, the effect of the number of convolutional layers is analyzed by varying it from 1 to 5 in the proposed 3D-CAE, where

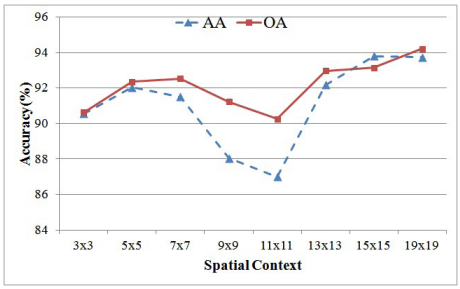


Fig. 13. Experimental results of the proposed 3D-CAE with different spatial contexts as input over the Indian Pine data set.

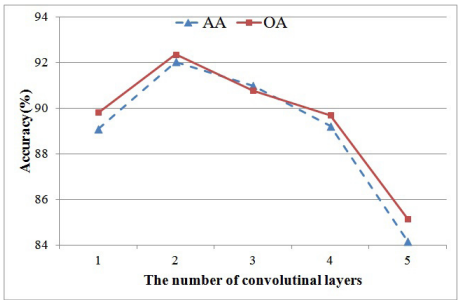


Fig. 14. Experimental results of the proposed 3D-CAE with different number of convolutional layers over the Indian Pine data set.

the kernel size in all the convolutional layers is fixed as $24 \times 3 \times 3 \times 24$. Fig. 14 shows the classification results on the Indian Pine data set. It is observed that the performance of classification first increases and then decreases a lot when the number of convolutional layers in the proposed 3D-CAE

TABLE VIII
COMPUTATIONAL ANALYSIS OF DIFFERENT FEATURE EXTRACTION ALGORITHMS

Indian Pines dataset (20245 pixels)							
	1D-CNN [35]	S-CNN [38]	DAE [44]	TPCA [12]	SSAE [45]	EPLS [42]	3D-CAE
Training(s)	20.6	119	74.4	44	240.1	111.9	1156
Feature Extraction (s)	17.4	3.2	15.9	150	2.76	47.3	5.22
Feature extraction per pixel ($\times 10^{-3}$ ms)	819.2	152.2	751.6	7134.4	131.5	2251	248.3
Salinas Valley dataset (111104 pixels)							
	1D-CNN [35]	S-CNN [38]	DAE [44]	TPCA [12]	SSAE [45]	EPLS [42]	3D-CAE
Training(s)	43.1	134	12.3	241	513	103.5	1159
Feature Extraction (s)	83.8	62	98.5	816	16.3	192.2	26.4
Feature extraction per pixel ($\times 10^{-3}$ ms)	754.0	558.0	859.7	7344.5	146.78	1729.6	237.3
Pavia University dataset (207400 pixels)							
	1D-CNN [35]	S-CNN [38]	DAE [44]	TPCA [12]	SSAE [45]	EPLS [42]	3D-CAE
Training(s)	32.1	332	93	44	491	127.13	1168
Feature Extraction (s)	150.07	106	812	150	31.9	141.5	32.04
Feature extraction per pixel ($\times 10^{-3}$ ms)	723.6	511.1	731.6	3915.1	154.0	682.3	154.5

increases. The proposed 3D-CAE achieves its best performance when the number of convolutional layers is 2, indicating that the proposed 3D-CAE can achieve superior performance of unsupervised feature learning under a simple structure with only two convolutional layers.

D. Computational Complexity

In this paper, all the experiments are carried out using a PC equipped with Intel i76850K CPU and a single GPU of GeForce GTX 1080. The computational time of the proposed 3D-CAE algorithm on these data sets, including both training and inference, is summarized in Table VIII. In general, neural network-based methods have the characteristics of extremely long training time and very short execution time. This is because the BP algorithm for training requires to iterate thousands of epochs for convergence, while only one forward operator is conducted for testing. As demonstrated in Table VIII, the proposed 3D-CAE takes several hours for training in these three data sets. However, when it is used for feature extraction, about 240 ms is consumed for 1000 samples for data sets acquired by the AVIRIS sensor, i.e., Indian Pine and Salinas data sets, which is about 0.2 ms for each sample. For the ROSIS sensor with less bands, much less time is taken for feature extraction over the Pavia University data set.

The computation time of 1D-CNN in [35], S-CNN [38], DAE [44], TPCA [12], SSAE [45], and EPLS [42] algorithms is also summarized in Table VIII. It is observed that the proposed 3D-CAE spends much longer time than other algorithms for training. However, such a long-time training can fully explore spatial–spectral information of pixels, which enables better feature representation ability though no label information is used. Although DAE and SSAE can also make use of all the samples in the image for training, the dimension reduction by PCA saves a lot of training time. The greedy layerwise training strategy in EPLS also saves a lot of training time. As for 1D-CNN and S-CNN, the training time is not very long since a small amount of labeled pixels are used for feature learning, e.g., 10% of pixels in the Indian Pines data set and 5% of pixels in the Salinas Valley data set. In TPCA, the computation time to construct tensor of correlation is recorded as training time, which is of the same level as that of 1D-CNN and DAE. In terms of the time for feature extraction, the proposed 3D-CAE costs nearly the least amount of time for feature extraction than all the other considered algorithms over all the data sets.

IV. CONCLUSION

In this paper, a 3D-CAE is constructed for unsupervised spatial–spectral feature extraction of hyperspectral images, in which only 3D and elementwise operations are used to avoid flattening the structure information of pixels. The proposed network is trained under a reconstruction task with a decoding network such that the labeled samples are not necessary. As a result, effective spatial–spectral features can be automatically learned in an unsupervised manner. Experiments over three benchmark data sets demonstrate that the spatial–spectral features learned in the proposed 3D-CAE are very effective for the classification task.

REFERENCES

[1] W. Li, Q. Du, and B. Zhang, "Combined sparse and collaborative representation for hyperspectral target detection," *Pattern Recognit.*, vol. 48, no. 12, pp. 3904–3916, 2015.
[2] L. He, J. Li, C. Liu, and S. Li, "Recent advances on spectral-spatial hyperspectral image classification: An overview and new guidelines," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1579–1597, Mar. 2017.
[3] A. Zare and K. Ho, "Endmember variability in hyperspectral analysis: Addressing spectral variability during spectral unmixing," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 95–104, Jan. 2014.
[4] S. Mei, Q. Bi, J. Ji, J. Hou, and Q. Du, "Spectral variation alleviation by low-rank matrix approximation for hyperspectral image analysis," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 6, pp. 796–800, Jun. 2016.
[5] C.-I. Chang and H. Ren, "An experiment-based quantitative and comparative analysis of target detection and image classification algorithms for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 2, pp. 1044–1063, Mar. 2000.
[6] B.-C. Kuo and D. A. Landgrebe, "Nonparametric weighted feature extraction for classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 5, pp. 1096–1105, May 2004.
[7] Q. Du, "Modified Fisher's linear discriminant analysis for hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 4, no. 4, pp. 503–507, Oct. 2007.
[8] T. V. Bandos, L. Bruzzone, and G. Camps-Valls, "Classification of hyperspectral images with regularized linear discriminant analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 3, pp. 862–873, Mar. 2009.

[9] B.-C. Kuo, C.-C. Hung, C.-W. Chang, and H.-P. Wang, "A modified nonparametric weight feature extraction using spatial and spectral information," in *Proc. IEEE Int. Symp. Geosci. Remote Sens.*, Jul./Aug. 2006, pp. 172–175.

[10] B. C. Kuo, C. H. Li, and J. M. Yang, "Kernel nonparametric weighted feature extraction for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 4, pp. 1139–1155, Apr. 2009.

[11] 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, Mar. 2005.

[12] Y. Ren, L. Liao, S. J. Maybank, Y. Zhang, and X. Liu, "Hyperspectral image spectral-spatial feature extraction via tensor principal component analysis," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 9, pp. 1431–1435, Sep. 2017.

[13] L. Ma, M. M. Crawford, and J. Tian, "Local manifold learning-based k-nearest-neighbor for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4099–4109, Nov. 2010.

[14] G. Chen and S.-E. Qian, "Dimensionality reduction of hyperspectral imagery using improved locally linear embedding," *J. Appl. Remote Sens.*, vol. 1, no. 1, p. 013509, 2007.

[15] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. Int. Conf. Neural Inf. Process. Syst., Natural Synth.*, 2001, pp. 585–591.

[16] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM J. Sci. Comput.*, vol. 26, no. 1, pp. 313–338, 2005.

[17] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Oct. 2005, pp. 1208–1213.

[18] X. He and P. Niyogi, "Locality preserving projections," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, vol. 16, no. 1, pp. 186–197.

[19] T. Zhang, J. Yang, D. Zhao, and X. Ge, "Linear local tangent space alignment and application to face recognition," *Neurocomputing*, vol. 70, nos. 7–9, pp. 1547–1553, 2007.

[20] H.-Y. Huang and B.-C. Kuo, "Double nearest proportion feature extraction for hyperspectral-image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4034–4046, Nov. 2010.

[21] F. Feng, W. Li, Q. Du, and B. Zhang, "Dimensionality reduction of hyperspectral image with graph-based discriminant analysis considering spectral similarity," *Remote Sens.*, vol. 9, no. 4, p. 323, 2017.

[22] W. Li, J. Liu, and Q. Du, "Sparse and low-rank graph for discriminant analysis of hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 7, pp. 4094–4105, Jul. 2016.

[23] W. Li and Q. Du, "Laplacian regularized collaborative graph for discriminant analysis of hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7066–7076, Dec. 2016.

[24] N. H. Ly, Q. Du, and J. E. Fowler, "Collaborative graph-based discriminant analysis for hyperspectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2688–2696, Jun. 2014.

[25] 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, Aug. 2013.

[26] L. Deng and D. Yu, "Deep learning: Methods and applications," *Found. Trends Signal Process.*, vol. 7, nos. 3–4, pp. 197–387, Jun. 2014.

[27] V. Singhal, H. K. Aggarwal, S. Tariyal, and A. Majumdar, "Discriminative robust deep dictionary learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5274–5283, Sep. 2017.

[28] X. Yang, Y. Ye, X. Li, R. Y. K. Lau, X. Zhang, and X. Huang, "Hyperspectral image classification with deep learning models," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5408–5423, Sep. 2018.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[30] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[32] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[33] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2015.

[34] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[35] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 501, Jan. 2015, Art. no. 258619.

[36] 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.*, Jul. 2015, pp. 4959–4962.

[37] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 844–853, Feb. 2017.

[38] B. Liu, X. Yu, P. Zhang, A. Yu, Q. Fu, and X. Wei, "Supervised deep feature extraction for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 1909–1921, Apr. 2018.

[39] S. Mei, J. Ji, J. Hou, X. Li, and Q. Du, "Learning sensor-specific spatial-spectral features of hyperspectral images via convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4520–4533, Aug. 2017.

[40] W. Zhao and S. Du, "Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Aug. 2016.

[41] R. Adriana, R. Petia, and G. Carlo, "Meta-parameter free unsupervised sparse feature learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 8, pp. 1716–1722, Aug. 2015.

[42] 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, Mar. 2016.

[43] Y. Bengio, I. Goodfellow and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[44] 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, Jun. 2014.

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

[46] X. Ma, H. Wang, and J. Geng, "Spectral–spatial classification of hyperspectral image based on deep auto-encoder," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 9, pp. 4073–4085, Sep. 2016.

[47] J. Ji, S. Mei, J. Hou, X. Li, and Q. Du, "Learning sensor-specific features for hyperspectral images via 3-dimensional convolutional autoencoder," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2017, pp. 1820–1823.

[48] K. He, X. Zhang, S. Ren, and J. Sun. (2015). "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification." [Online]. Available: https://arxiv.org/abs/1502.01852

[49] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[50] L. A. Gatys, A. S. Ecker, and M. Bethge. (2015). "A neural algorithm of artistic style." [Online]. Available: https://arxiv.org/abs/1508.06576

[51] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros. (2016). "Context encoders: Feature learning by inpainting." [Online]. Available: https://arxiv.org/abs/1604.07379

[52] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[53] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4489–4497.

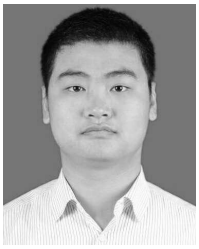
[54] S. Ioffe and C. Szegedy. (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift." [Online]. Available: https://arxiv.org/abs/1502.03167

[55] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011, Art. no. 27.

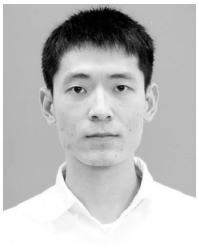
[56] W. Li, S. Prasad, J. E. Fowler, and L. M. Bruce, "Locality-preserving dimensionality reduction and classification for hyperspectral image analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 4, pp. 1185–1198, Apr. 2012.
[57] N. H. Ly, Q. Du, and J. E. Fowler, "Sparse graph-based discriminant analysis for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 7, pp. 3872–3884, Jul. 2014.
[58] X. He and P. Niyogi, "Locality preserving projections (LPP)," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, vol. 16, no. 1, pp. 186–197.

Shaohui Mei (S'10–M'12) received the B.S. degree in electronics and information engineering and the Ph.D. degree in signal and information processing from Northwestern Polytechnical University, Xi'an, China, in 2005 and 2011, respectively.

From 2007 to 2008, he was a Visiting Student with the University of Sydney, Sydney, NSW, Australia. He is currently an Associate Professor with the School of Electronics and Information, Northwestern Polytechnical University. His research interests include hyperspectral remote sensing image processing and applications, intelligent signal and information acquisition and processing, neural networks, pattern recognition, and machine learning.



Jingyu Ji received the B.E. degree in electronics and information engineering from Northwestern Polytechnical University, Xi'an, China, in 2016, where he is currently pursuing the master's degree in signal and information processing, with a focus on deep learning and hyperspectral remote sensing image processing.



Yunhao Geng received the B.E. degree in electronics and information engineering from Northwestern Polytechnical University, Xi'an, China, in 2018, where he is currently pursuing the master's degree in signal and information processing.

His research interests include image classification and superresolution.

Zhi Zhang received the B.Sc. and M.Sc. degrees in applied mathematics from Air Force Engineering University, Xi'an, China, in 2003 and 2006, respectively, and the Ph.D. degree in electrical engineering from the National University of Defense Technology, Changsha, China, in 2010.

He currently holds a post-doctoral position at the State Key Laboratory of Remote Sensing Science, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, China. His research interests include hyperspectral remote sensing image processing and applications, pattern recognition, and information fusion.

Xu Li (S'13–M'16) received the B.Sc. degree in electronics and information technology, the M.Sc. degree in signal and information processing, and the Ph.D. degree in circuits and systems from Northwestern Polytechnical University, Xi'an, China, in 2001, 2004, and 2010, respectively.

In 2004, he joined the School of Electronics and Information, Northwestern Polytechnical University. From 2007 to 2008, he was a Visiting Ph.D. Student with Télécom ParisTech, Paris, France. From 2012 to 2014, he was an Enterprise Post-Doctoral Fellow with the Jiangsu R&D Center for Internet of Things, Wuxi, China. From 2015 to 2016, he was a Marie Curie Research Fellow with the School of Computer Science, University of Lincoln, Lincoln, U.K. He is currently an Associate Professor with the School of Electronics and Information, Northwestern Polytechnical University. His research interests include image fusion, pansharpening, and image enhancement.

Qian Du (S'98–M'00–SM'05–F'18) received the Ph.D. degree in electrical engineering from the University of Maryland–Baltimore County, Baltimore, MD, USA, in 2000.

She is currently a Bobby Shackouls Professor with the Department of Electrical and Computer Engineering, Mississippi State University, Starkville, MS, USA. Her research interests include hyperspectral remote sensing image analysis and applications, pattern classification, data compression, and neural networks.

Dr. Du is a fellow of the SPIE-International Society for Optics and Photonics. She was a recipient of the 2010 Best Reviewer Award from the IEEE Geoscience and Remote Sensing Society. She served as the Co-Chair for the Data Fusion Technical Committee of the IEEE Geoscience and Remote Sensing Society from 2009 to 2013 and the Chair for the Remote Sensing and Mapping Technical Committee of the International Association for Pattern Recognition from 2010 to 2014. She is the General Chair for the 4th IEEE GRSS Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing in Shanghai, China, in 2012. She served as an Associate Editor for the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, the *Journal of Applied Remote Sensing*, and IEEE SIGNAL PROCESSING LETTERS. Since 2016, she has been the Editor-in-Chief of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING.