Self-Supervised Feature Learning With CRF Embedding for Hyperspectral Image Classification

Yuebin Wang, Jie Mei, Liqiang Zhang, Bing Zhang, Panpan Zhu, Yang Li, and Xingang Li

***Abstract*— The challenges in hyperspectral image (HSI) classification lie in the existence of noisy spectral information and lack of contextual information among pixels. Considering the three different levels in HSIs, i.e., subpixel, pixel, and superpixel, offer complementary information, we develop a novel HSI feature learning network (HSINet) to learn consistent features by self-supervision for HSI classification. HSINet contains a three-layer deep neural network and a multifeature convolutional neural network. It automatically extracts the features such as spatial, spectral, color, and boundary as well as context information. To boost the performance of self-supervised feature learning with the likelihood maximization, the conditional random field (CRF) framework is embedded into HSINet. The potential terms of unary, pairwise, and higher order in CRF are constructed by the corresponding subpixel, pixel, and superpixel. Furthermore, the feedback information derived from these terms are also fused into the different-level feature learning process, which makes the HSINet-CRF be a trainable end-to-end deep learning model with the back-propagation algorithm. Comprehensive evaluations are performed on three widely used HSI data sets and our method outperforms the state-of-the-art methods.**

***Index Terms*— Conditional random field (CRF), convolutional neural network (CNN), feature learning, hyperspectral image (HSI) classification, self-supervision.**

I. Introduction

Hyperspectral images (HSIs) are characterized in hundreds of continuous observation bands throughout the electromagnetic spectrum with high spectral resolution [1] in the field of remote sensing. These characteristics can help us to discriminate different materials of interest [2]. HSI classification has become one of the most important tasks like land cover/use classification. However, the noises and mixed spectral information in HSIs cause several theoretical and practical challenges for classification [3]. In recent years, many methods have been proposed to classify HSIs [4]–[7]. Based on the different levels of HSI feature extraction, these classification approaches usually can be divided into three types of approaches, i.e., subpixel-level, pixel-level, and superpixel-level methods [8]–[14]. Since pixels in HSIs usually contain mixed spectral information of different objects, subpixel-level classification methods have the ability to obtain features of pure spectral signals from subpixels. Although the pixel-level classification methods can combine the spectral and spatial features to enhance classification performance, they are susceptible to mixed pixels and noise, resulting in noisy classification results. Moreover, the above two types of methods fail to consider region boundary areas. For another, the spatial continuity has not been fully utilized in these approaches [3]. Superpixel can adaptively describe the local structure information with different sizes and shapes, and the object boundaries can be preserved well. Yet, the classification accuracy is deteriorated if undersegmentation cannot be fully avoided in superpixel-based approaches [15].

This enables us to take full advantage of the properties of three different levels in HSIs, i.e., spectral and spatial correlations, shape and contextual information as well as band-to-band variability. The difficulties in accurately describing the features of pixels from sublevel to superlevel are induced by the noises and mixed spectral information in HSIs and lack of contextual information among pixels, which are the challenges in HSI classification [4], [16], [17]. Aiming to obtain accurate classification results of HSIs, it usually needs a great deal of labeled pixels to train the parameters of the deep neural network. However, the amount of human resources needed to manually annotate such data sets represents a problem [18]. Inspired by recent works about self-supervised learning [19], [20], [52], the information about data can be adopted as a source of self-supervision for feature learning. Thus, the features learned in a certain level can be exploited for the HSI feature learning of other levels. The key benefit of this approach is that the annotations can be obtained for "free."

Manuscript received June 2, 2018; revised August 10, 2018; accepted October 4, 2018. Date of publication November 27, 2018; date of current version April 22, 2019. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC0213600, in part by the National Natural Science Foundation of China under Grant 41371324 and Grant 41801241, and in part by the China Postdoctoral Science Foundation under Grant 2016M600953 and Grant 2018T110065. *(Y. Wang and J. Mei contributed equally to this work.) (Corresponding author: Liqiang Zhang.)*

Y. Wang is with the School of Land Science and Technology, China University of Geosciences, Beijing 100083, China, and also with the Faculty of Geographical Science, State Key Laboratory of Remote Sensing Science, Beijing Normal University, Beijing 100875, China (e-mail: xxgcdxwyb@163.com).

J. Mei, L. Zhang, P. Zhu, Y. Li, and X. Li are with the Faculty of Geographical Science, State Key Laboratory of Remote Sensing Science, Beijing Normal University, Beijing 100875, China (e-mail: meijie@mail.bnu.edu.cn; zhanglq@bnu.edu.cn; zlyxbmsl@163.com; isliyang@mail.bnu.edu.cn; lixg95@126.com).

B. Zhang is with the Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing 100094, China (e-mail: zb@ceode.ac.cn).

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

Digital Object Identifier 10.1109/TGRS.2018.2875943

In this paper, HSINet is constructed to learn three complementary features of multilevels by the self-supervised way. To boost the self-supervised feature learning performance, the conditional random field (CRF) framework is embedded into HSINet. Our contributions are three folds.

1) We propose a deep network HSINet that effectively extracts and integrates complementary multilevel features from subpixel-level, pixel-level, and superpixel-level by self-supervision. To the best of our knowledge, this is the first use of the deep network model to learn and integrate the three-level complementary features for HSI classification.
2) We propose a HSINet-CRF model to alleviate the problem of the sampling complexity and gain robustness to noise in the input features. The feedback information derived from the terms of CRF is fed back to HSINet to further augment the performance of self-supervised feature learning. Moreover, our method significantly reduces the intensive labeling cost in previous works by self-supervision.
3) Our method is an efficient end-to-end trained system. It fully exploits the complementary characteristics of the subpixel, pixel, and superpixel to produce powerful feature representations that capture texture, shape, and contextual information of HSIs. It outperforms the state-of-the-art methods on three HSI data sets.

II. Related Work

In this section, we will introduce the contents about subpixel-based, pixel-based, and superpixel-based HSI classification methods.

A. Multilevel Features Fusion

Since the superpixel has the ability to provide contextual information, the superpixel-based classification approach can well avoid the “salt-and-pepper” problem. Just as Eches *et al.* [16] stated, it is difficult to obtain an accurate over-segmentation superpixel map for HSI classification only by one computational process. Once pixels within a superpixel belong to different classes, a wrong classification cannot be avoided. Under this condition, it is better to integrate the advantages of subpixel, pixel, and superpixel into the feature learning procedure and classification for HSIs. There exists some related works to fuse different levels of HSI features. In [14] and [16], multilevels of HSI features are learned independently, which would make the learned features be not consistent with each other.

B. Subpixel-Based HSI Classification

In the HSIs, mixed pixels are a mixture of more than one distinct substance. Aiming to obtain the pure spectral signals (also called as endmembers), some subpixel-based algorithms are developed to separate different category pixels. The subpixel mapping technique introduced by Atkinson [23] can obtain the subpixel location of each class in a pixel by dividing a pixel into subpixels. To improve the subpixel mapping accuracy, an adaptive subpixel mapping framework was proposed based on a multiagent system for remote sensing imagery [24]. In [25], a framework was developed for semisupervised HSI classification that naturally integrates the information provided by discriminative classification and spectral unmixing, where the complementarity information of classification and spectral unmixing can be better explored. Linear spectral unmixing is also a popular tool in remotely sensed hyperspectral data interpretation. Iordache *et al.* [26] have studied the linear spectral unmixing problem under the light of recent theoretical results published in those referred to areas and further indicates the potential of sparse regression techniques in the task of accurately characterizing the mixed pixels using the library spectra. With low-rank further embedding, sparse and low-rank matrices are simultaneously learned to exploit both spatial correlation and sparse representation of pixels lying in the homogeneous regions of HSI [27].

C. Pixel-Based HSI Classification

Pixel-based HSI classification methods are another important branch, which have appeared in many related works. In HSI classification, multiple features, e.g., spectral, texture, and shape features, can be employed to represent pixels from different perspectives. In [4], a patch alignment framework was introduced to linearly combine multiple features in the optimal way and obtain a unified low-dimensional representation of these multiple features for subsequent classification. A spectral–spatial feature learning method was proposed to obtain robust features of HSIs in [28]. It combines the spectral feature learning and spatial feature learning into a hierarchical structure. Gao *et al.* [29] constructed a bilayer graph-based learning framework for HSI classification, which can better explore the information of spectral and spatial of pixels. Some methods of HSI classification put emphasis on the classifiers. In [30], an HSI classification algorithm based on discriminative conditional random was developed. Ma *et al.* [31] combined local manifold learning and the k-nearest-neighbor classifier for HSI classification, where locally linear embedding, local tangent space alignment, and Laplacian eigenmaps are investigated with these classifiers. A semisupervised graph based was proposed in [32]. This method can well handle the special characteristics of the HSI, namely, high-input dimension of pixels, few labeled samples, and spatial variability of the spectral signature.

Based on the studies of subpixel-level and pixel-level HSI classification, it is observed that these classification approaches can well exploit discriminant spectral and spatial information of each pixel. Due to lack of contextual information among pixels, the subpixel-level and pixel-level classification methods are likely to generate noisy appearance in classification maps [16], [14].

D. Superpixel-Based HSI Classification

For better exploiting the contextual information among pixels, superpixel-based classification is adopted to establish the neighboring relationships among the pixels. Superpixels are generated with the graph-based algorithms like normalized cuts [33] and entropy rate superpixel segmentation [34] or the gradient-descent-based algorithms like SLIC [35] and SEEDS [36]. Superpixels can be used to smooth the features, labels and the similarities among neighboring pixels and avoid the "salt-and-pepper" problem generated in the subpixel-based and pixel-based classification methods. In [8], superpixels instead of pixels are applied to the graphical model to capture the contextual information and the spatial dependence among the superpixels. Fang *et al.* [9] considered a superpixel in an HSI as a small spatial region whose size and shape can be adaptively adjusted for different spatial structures. In their approach, pixels within each superpixel were jointly represented by a set of common atoms from a dictionary via a joint sparse regularization. A superpixel-level sparse representation classification framework with multitask learning was developed in [10]. The proposed algorithm exploited the class-level sparsity prior and the correlation of neighboring pixels for fusing multiple features. Based on the superpixels, a spectral–spatial adaptive sparse representation method was proposed for HSI compression by taking advantage of the spectral and spatial information of HSIs [37]. Sparse representation can transform spectral signatures of the pixels into sparse coefficients with very few nonzero entries.

E. Semantic Segmentation

HSI classification also can be considered as the semantic segmentation of HSIs. Segmentation is essential for image analysis tasks. Semantic segmentation describes the process of associating each pixel of an image with a class label. The watershed segmentation algorithm was extended for HSIs [54]. The accuracy of the watershed algorithms was demonstrated by the further incorporation of the segmentation maps into a classifier. Long *et al.* [41] showed that convolutional networks by themselves, trained end-to-end, pixels-to-pixels, exceed most approaches in semantic segmentation. One can also apply high-capacity convolutional neural networks to bottom-up region proposals in order to localize and segment objects [53]. The framework of CNN for semantic segmentation also can be adopted for HSI semantic segmentation.

III. Proposed Approach

In this paper, we develop a novel HSI feature learning network (HSINet) to learn heterogeneous features by self-supervision (Fig. 1). HSINet contains a three-layer deep neural network (TDNN) and a multifeature convolutional neural network (MCNN). The TDNN learns subpixel-level features where noise is effectively removed from the original spectral information. The MCNN is utilized to obtain pixel-level features. Based on the features of subpixel level and pixel level, the HSI is over segmented into two kinds of superpixels. The extracted feature of each superpixel can be utilized to obtain the spatial constraints and contextual information among pixels. Then the self-supervised constraints are employed to make different-level features consistent.

Each level feature in HSINet can be considered as the supervisor for the feature learning of other levels. However, quality of the pure spectral and accurate relationships between pixels cannot be well ensured without the likelihood maximization. The CRF provides a solution to boost the performance of self-supervised feature learning. It is a flexible framework which can incorporate different kinds of features for visual recognition [21], [22]. Thus a novel approach named HSINet-CRF is developed to embed CRF framework into HSINet for further enhancing the performance of HSI feature learning. The potential terms of unary, pairwise, and higher order in CRF are constructed by the corresponding subpixel, pixel, and superpixel. Furthermore, the feedback information derived from these terms is also embedded into different-level feature learning process (Fig. 2). Our method alleviates the need of engineered features, and produces a powerful representation that captures texture, shape, and contextual information.

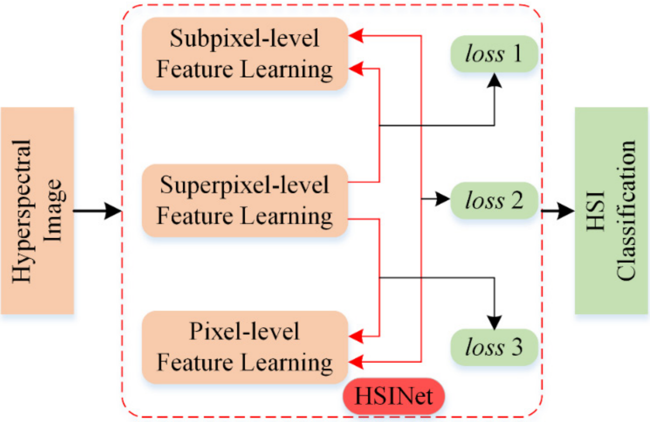

Hyperspectral Image
Subpixel-level Feature Learning
Superpixel-level Feature Learning
Pixel-level Feature Learning
loss 1
loss 2
loss 3
HSINet
HSI Classification

Fig. 1. Overview of HSINet. The red lines show the self-supervised constraints between different-level features. The three loss constraints are proposed to learn consistent features from three levels.

A. HSINet

To extract the complementary features from subpixel, pixel, and superpixel, we propose a TDNN and an MCNN, which utilize spatial–spectral information of neighboring pixels to explore contextual interactions of pixels simultaneously.

1) Subpixel-Level Feature Learning: The spectral information of pixels in an HSI is described by the vectors $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$, where N represents the number of image pixels, $\mathbf{x} \in \mathbb{R}^d$ and d denotes the dimension of the spectral vector (number of bands). Let y_i be the label variable assigned to the pixel i. The value of y_i is from a predefined set of labels $\mathcal{L} = \{l_1, l_2, \ldots, l_L\}$. $\mathbf{Y}$ is a vector formed by a set of variables $\{y_1, y_2, \ldots, y_\mathrm{N}\}$.

Since the HSIs usually contain mixed spectral information of different objects, it is necessary to remove the noise and get pure spectral signals of different land cover/use. It needs to reduce the dimension of the original spectral vector, and give the new representations for the HSI. Following the work [14], the dimension is set to $2 \times L$, where L is the number of classes in the HSI. We utilize the TDNN to extract the subpixel-level feature. The input of TDNN is the original spectral vector. The output of TDNN is the spectral vector with the dimension of $2 \times L$. For obtaining the new representations, the parameters of feature transformation are defined in each layer of TDNN.

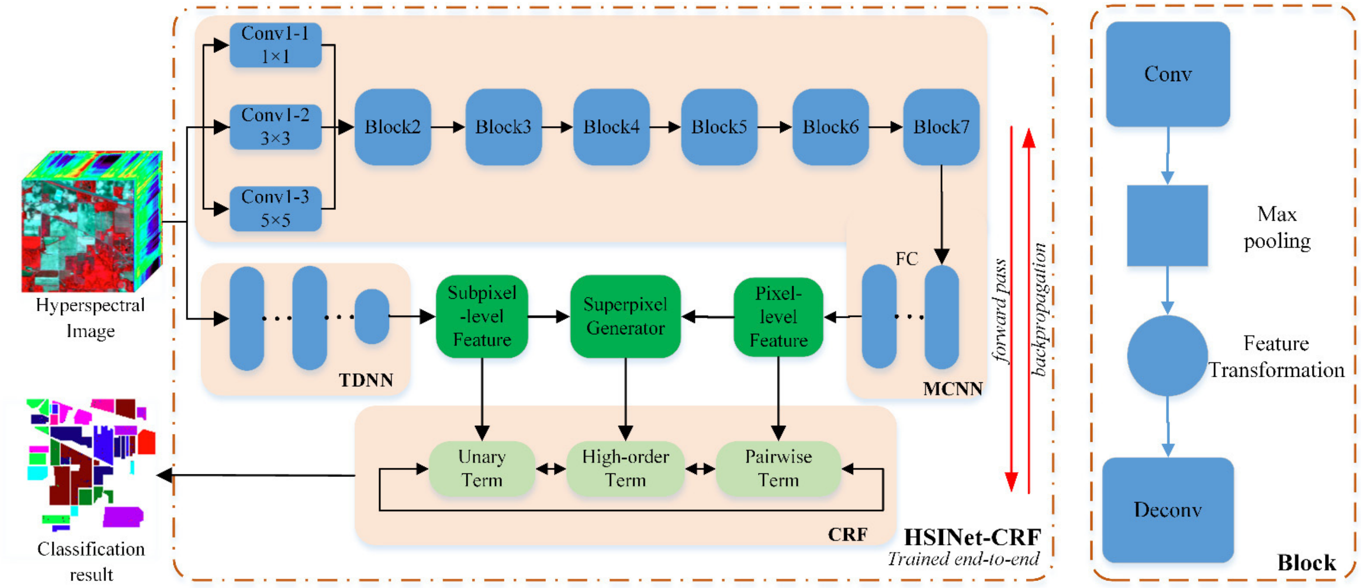


Fig. 2. Pipeline of the HSINet-CRF for HSI classification. HSINet including TDNN and MCNN learns three complementary features of multilevels by self-supervision. To boost the self-supervised feature learning performance, the CRF is embedded into HSINet.

By means of feature transformation with TDNN, the features of subpixel level can be achieved.

TDNN is a class of feed forward artificial neural network. Furthermore, TDNN includes three fully connected layers: an input and an output layer with one hidden layer. The first two layers have 600 neurons and the third layer has $2 \times L$ neurons. Fully connected layers connect every neuron in one layer to every neuron in another layer. All nodes are interconnected. Each node in one layer connects with a certain weight and a bias to every node in the following layer. The output $\mathbf{h}^m$ at the mth layer ($1 \leq m \leq 3$) is computed by the following nonlinear activation function, which is implemented by rectified linear unite (ReLU) to accelerate the training process

$$\mathbf{h}^m = \varphi(\mathbf{W}^m \mathbf{h}^{m-1} + \mathbf{b}^m) \tag{1}$$

where $\mathbf{W}^m$ is the weight matrix and $\mathbf{b}^m$ is the bias vector in this layer, $\varphi(\cdot)$ is the nonlinear activation function.

2) Pixel-Level Feature Learning: As shown in Fig. 2, the whole MCNN framework includes seven convolutional-based layers. The first layer is a "multiscale filter bank," which convolves the three input images with different sizes using 256 kernels of three different size convolutional filters. The second to the seventh layers are blocks containing convolution, max-pooling, feature transformation, and deconvolution. The convolution of the second layer filters the output of first layer with 256 kernels of size $5 \times 5 \times 768$. The third layer has 256 kernels of size $4 \times 4 \times 256$ connected to the $5 \times 5 \times 256$ outputs of the second layer. The fourth, fifth, sixth, and seventh layers have 256 kernels of size $3 \times 3 \times 256$. Then the output of the seventh layer is the input of two fully connected layers to learn pixel-level features. We use the dropout in the fully-connected layers to avoid overfitting. The output of the last fully connected layer is fed to a softmax layer, which is defined as the gradient-log-normalizer of the categorical probability distribution to produce a distribution over L class labels. The nonlinear activation function ReLU is applied to the output of every convolutional and fully connected layer.

Since the available HSI training data is highly limited, we use two modules of residual learning [38] to avoid overfitting under the network having enough layers. This strategy is shown to be able to significantly improve training efficiency of the networks and is easier to optimize the weights with the residual mapping than with the unreferenced mapping. The following network settings play important role in the structure of MCNN:

a) Multiscale filter bank: The first layer of the MCNN is a multiscale filter bank, which can explore spectral correlations and local spatial structure. The multiscale filter bank consists of three different size convolutional filters $1 \times 1 \times d$, $3 \times 3 \times d$ and $5 \times 5 \times d$. The filters $1 \times 1 \times d$ are utilized to handle spectral correlations, while the filters $3 \times 3 \times d$ and $5 \times 5 \times d$ are used to exploit the local spatial correlations among neighboring pixels. The sizes of the input images corresponding to three different size convolutional filters are $1 \times 1 \times d$, $3 \times 3 \times d$ and $5 \times 5 \times d$, and the sizes of the feature maps from the three convolutional filters are 1×1, 3×3 and 5×5. To combine the three feature maps into a joint feature map, a space of two-pixel width filled with zeros is padded around the feature maps so that the size of three feature maps become 5×5. Then we can combine the three convolutional feature maps from the first layer to form a joint spatial–spectral feature map used as the input of the subsequent layers. It is noted that the residual learning and the multiscale filter bank are effective for increasing the width and depth of the network [38], [39], which can help to effectively learn the network with a small number of training data.

b) Block in the network: We exploit the advantage of the blocks from the second to the seventh layers to process the influence of pixel clustering results for feature learning. The blocks are constructed based on the superpixels that provide the information of clustering results, which is useful for feature learning of self-supervision. As shown in Fig. 2, each block contains convolution, max-pooling, feature transformation, and deconvolution. The convolution of each block takes the 5×5 output of the previous layer as the input. We obtain the 1×1 feature map after convolution and max-pooling, which is taken as a 2-D feature matrix of the feature transform with the information of clustering results. Then the feature map after transformation is deconvoluted to 5×5 for fully exploiting spatial information of pixels.

Feature Transformation: Before feature transformation, HSI is divided into a set of nonoverlapping homogenous regions via spectral clustering (SC) method [40], and each region corresponds to a superpixel. Since the pixels in each superpixel have spectral and spatial similarity, we calculate the average spectral data to characterize the common spatial–spectral information. Then the transformation matrix $\mathbf{H}$ is obtained using the average spectral data of each superpixel

$$\mathbf{H}_{i,i} = \frac{\exp\left(-\|\mathbf{u}_a - \mathbf{u}_s\|_2^2/\sigma_b\right)}{\sum_{t=1}^{K} \exp\left(-\|\mathbf{u}_a - \mathbf{u}_t\|_2^2/\sigma_b\right)} \tag{2}$$

where $\mathbf{u}_s$ is the average spectral value of the superpixel containing pixel i, $\mathbf{u}_a$ is the average spectral value of all superpixels, $\mathbf{u}_t$ is the average spectral value of each superpixel, σ_b is an alternative parameter, and K is the number of superpixels. The following equation is then defined to transform the feature map after convolution and max-pooling in each block:

$$\mathbf{z} = \mathbf{H}\{\mathbf{W}_c * \mathbf{x} + \mathbf{b}_c\} \tag{3}$$

where x and z represent the input and output vectors of the used layers, $\mathbf{W}_c$ and $\mathbf{b}_c$ are the weight and bias, $\{\cdot\}$ denotes the output after convolution and max-pooling, $\mathbf{H}$ is the transformation matrix obtained using superpixel information, and $*$ denotes a convolution.

Note that each superpixel in HSI corresponds to a set of spatially connected and spectrally similar pixels that can better exploiting the spectral–spatial structure information, so the feature transformation in each block can effectively improve the performance of HSI classification. Since the feature maps learned by MCNN are changeable during the iteration, it is not appropriate to use the average spectral value of superpixels segmented from the original whole image throughout the iteration process in the feature transformation. We develop the adaptive superpixels for the block, which segment the superpixels using the pixel-level feature learned from previous iteration. Then the transformation matrix is changing during iterations so that the feature learning process can make better use of clustering information.

Deconvolution: We adopt the deconvolution to make feature map size after transformation upsample to 5×5, which is taken as the input to the subsequent convolutional layers. However, unlike the fully convolutional network [41], which uses the deconvolution to recover the pixel level prediction information from the feature maps extracted by the convolutional layers. We add the deconvolution filter into every block to fully exploit the spatial correlation. In our experiments, we find the upsampling in each block is effective for learning spatial information of pixels.

3) Superpixel-Level Feature Learning: After getting the subpixel-level feature by TDNN and the pixel-level feature by MCNN, we can learn superpixel-level feature. We obtain two kinds of superpixels based on the subpixel-level feature and pixel-level feature using the SC method, and thus two kinds of superpixel-level features can be obtained by calculating the average spectral data of pixels in each superpixel.

4) Self-Supervised Feature Learning Constraints: Trough the multilevel feature learning, we obtain different feature representations of the HSI. Aiming to increase the accuracy of feature learning without label introduced, we define three constraints to learn heterogeneous features from three levels by self-supervision and make different-level features of the HSI be consistent. Fixed one level feature, the feature of this level can be considered as the supervisor for the feature learning of other levels. The constraints of self-supervised feature learning include three parts: subpixel-pixel constraint, pixel-superpixel constraint, and superpixel-superpixel constraint. To fulfil the feature learning through the subpixel-pixel constraint, the manifold learning is introduced with the assumption: if two pixels i and j have the similar subpixel-level feature, they are expected to have same labels. To deal with the feature learning under the constraints of subpixel-pixel and superpixel–superpixel, the L-2 norm function is adopted to compute the corresponding learned features of different level.

a) Subpixel-pixel constraint: If two pixels i and j have the similar subpixel-level feature, they are expected to have same labels. We utilize this property to regularize the pixel-level feature of each pixel by the similar relationship in subpixel level. An adaptive graph $\mathbf{G}_E$ is introduced to describe the relationships among the pixels. In $\mathbf{G}_E$, each vertex corresponds to one pixel, and the nearest neighbors are selected according to the weight matrix $\mathbf{U}$. $\mathbf{U}$ is defined using the following function:

$$\mathbf{U}_{ij} = \begin{cases} \exp\left(-\left\|\mathbf{f}_i^{\text{sub}} - \mathbf{f}_j^{\text{sub}}\right\|_2^2\right) & i \in \mathcal{N}_{k_1}(j) \ or \ j \in \mathcal{N}_{k_1}(i) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $\mathcal{N}_{k_1}(i)$ denotes the k_1-nearest neighbors of pixel i. $\mathbf{f}_i^{\text{sub}}$ is the subpixel-level feature vector for pixel i. The subpixel-pixel constraint is obtained as follows using the manifold smoothness with pixel-level feature:

$$J_1 = \frac{1}{2}\sum_{i,j=1}^{n} \mathbf{U}_{ij}\left\|\mathbf{f}_i^{\text{pix}} - \mathbf{f}_j^{\text{pix}}\right\|_2^2 = tr((\mathbf{f}^{\text{pix}})^T \mathbf{L}_E \mathbf{f}^{\text{pix}}) \tag{5}$$

where $\mathbf{f}_i^{\text{pix}}$ is the pixel-level feature vector for pixel i, $\mathbf{f}^{\text{pix}}$ is the pixel-level feature matrix, $\mathbf{D}_E$ is a diagonal matrix where the (i, i)th element equals to the sum of the ith row of U, then $\mathbf{L}_E = \mathbf{D}_E - \mathbf{U}$. This regularization is used to simultaneously optimize the relationships among pixels and classification scores.

b) Pixel-superpixel constraint: Pixels in one superpixel are expected to have similar data representation and labeling information. We define the following constraint to avoid the "salt-and-pepper" problem and further refine the HSI classification results with pixel-level feature:

$$J_2 = \left\|\mathbf{f}_i^{\text{pix}} - \mathbf{u}_s^{\text{pix}}\right\|_2^2 \tag{6}$$

where $\mathbf{u}_s^{\text{pix}}$ is the average feature of the superpixel containing pixel i, which is calculated by pixel-level feature.

c) Superpixel-superpixel constraint: We can obtain two kinds of superpixels using subpixel-level and pixel-level features. However, the two kinds of superpixels are usually inconsistent, which can lead incorrect classification results. To solve this problem, we utilize the superpixel-superpixel constraint to make subpixel-level and pixle-level features have consistent over-segmentation results, further for consistent feature learning of different levels. Thus, the following objective function is defined to obtain consistent classification results:

$$J_3 = \sum_{i=1}^{n} \left\|\mathbf{u}_t^{\text{sub}} - \mathbf{u}_t^{\text{pix}}\right\|_2^2 \tag{7}$$

where $\mathbf{u}_t^{\text{sub}}$ and $\mathbf{u}_t^{\text{pix}}$ are the superpixel-level features obtained by subpixel-level feature and pixel-level feature, respectively. J_3 is calculated by two kinds of superpixels to show their difference.

B. HSINet-CRF Model

Aiming to boost the self-supervised feature learning by the probabilistic model, we propose an HSINet-CRF model to embed the CRF framework into HSINet.

Subpixels are usually utilized to describe information about pure spectral signals like land cover/use. We define the unary term of the CRF through the subpixel-level feature. The relationships between a pixel and neighboring pixels can be determined by the spectral-spatial graph. The pairwise term of the CRF can fully exploit the advantage of the pixel-level feature. The higher order term of CRF describes the pixel clustering information, so the superpixel-level feature is introduced to define the higher order term. The detail of CRF terms is referred as the following.

Given a graph G = $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{y_1, y_2, \ldots, y_N\}$. A CRF (**X**, **Y**) can be characterized by a Gibbs distribution of the form $P(\mathbf{Y}|\mathbf{X}) = (1/Z(\mathbf{X}))\exp(-E(\mathbf{y}|\mathbf{X}))$, where E(**y**|**X**) is the Gibbs energy of a labeling $\mathbf{y} \in \mathcal{L}^N$ and Z(**X**) is the partition function [42]. For notational convenience, we omit the conditioning on **X** in the rest of this paper.

In the fully connected pairwise CRF model [43], the unary potential of CRF is given by

$$E_1(\mathbf{y}) = \sum_i \psi_u(y_i) \tag{8}$$

where $i \in [1, N]$, the unary potential $\psi_u(y_i)$ measures the cost of assigning label y_i to the pixel i, which is computed independently for each pixel from the softmax function using the subpixel-level feature in our CRF model. The pairwise potentials are modeled as follows:

$$E_2(\mathbf{y}) = \sum_{i<j} \psi_p(y_i, y_j) = \sum_{i<j} \mu(y_i, y_j) \sum_{n=1}^{P} w^{(n)} k_G^{(n)}\left(\mathbf{f}_i^{\text{pix}}, \mathbf{f}_j^{\text{pix}}\right) \tag{9}$$

where $j \in [1, N]$, $\mu(\cdot)$ is a label compatibility function, $w^{(\text{n})}$ are linear combination weights, each $k_G^{(\text{n})}$ is a Gaussian kernel. The pixel-level feature vectors $\mathbf{f}_i^{\text{pix}}$ and $\mathbf{f}_j^{\text{pix}}$ for pixels i and j are obtained from MCNN. $\psi_p(y_i, y_j)$ measures the cost of the pixels i, j taking labels y_i, y_j simultaneously, which are obtained from the pixel-level feature in MCNN and predict pixel labels considering consistency and smoothness of the labels assignment.

Considering that pixels in a superpixel are likely to belong to the same class, we develop superpixel-based higher order potentials defined as follows:

$$\begin{aligned} E_3(\mathbf{y}) &= \sum_{i=1}^{n} w_i^{\text{sub}} k^{\text{sub}}\left(\mathbf{f}_i^{\text{sub}}\right) + \sum_{i=1}^{n} w_i^{\text{pix}} k^{\text{pix}}\left(\mathbf{f}_i^{\text{pix}}\right) \\ &= \sum_{i=1}^{n} w_i^{\text{sub}} \exp\left(\frac{\left\|\mathbf{f}_i^{\text{sub}} - \mathbf{u}_t^{\text{sub}}\right\|_2^2}{\sigma_s^{\text{sub}}}\right) \\ &\quad + \sum_{i=1}^{n} w_i^{\text{pix}} \exp\left(\frac{\left\|\mathbf{f}_i^{\text{pix}} - \mathbf{u}_t^{\text{pix}}\right\|_2^2}{\sigma_s^{\text{pix}}}\right) \end{aligned} \tag{10}$$

where w_i^{sub} and w_i^{pix} are learnable weights. $\mathbf{f}_i^{\text{sub}}$ and $\mathbf{f}_i^{\text{pix}}$ are the subpixel-level and pixel-level feature vectors learned by the TDNN and MCNN, respectively. $\mathbf{u}_t^{\text{sub}}$ and $\mathbf{u}_t^{\text{pix}}$ are the average feature of pixels within the superpixels where $\mathbf{f}_i^{\text{sub}}$ and $\mathbf{f}_i^{\text{pix}}$ locate, σ_s^{sub} and σ_s^{pix} are alternative parameters. It is noted that superpixel-based potential is effective for avoiding the "salt-and-pepper," which is inconsistent with the correct labels of its neighbors.

Since the three potentials are defined, the whole energy function is formed as

$$E(\mathbf{y}) = E_1(\mathbf{y}) + E_2(\mathbf{y}) + E_3(\mathbf{y}). \tag{11}$$

The most probable label y for each pixel in the HSI can be obtained by minimizing the energy E(**y**). Since L-BFGS [43] requires computing the gradient of the partition function Z, which is intractable to estimate exactly. Then an approximate inference such as mean field approximation is helpful.

In the traditional CRF model, the unary, pairwise and higher order potentials are usually independent with each other, which is not appropriate for gaining robustness to noise in the HSI. Therefore, we embed the CRF framework into HSINet to form HSINet-CRF, which can boost the self-supervised feature learning by probabilistic model. Simultaneously, the three constraints in HSINet can make the three potentials in CRF collaborate with each other, and then the finally objective function of the whole network HSINet-CRF is formed as

$$J = \frac{1}{Z}\exp(-E(\mathbf{y})) + \lambda_1 J_1 + \lambda_2 J_2 + \lambda_3 J_3 \tag{12}$$

where Z is the partition function, λ_1, λ_2, and λ_3 are tradeoff factors.

C. Implementation

Given such a nonlinear optimization in (12), solving the variables in HSINet and CRF simultaneously is intractable by directly applying gradient descent method due to the highly nonlinear nature of J, which makes the gradient and the Hessian difficult to compute. In this paper, we adopt a customized iterative procedure to optimize the variables in the HSINet-CRF. The optimization procedure mainly refers to two parts: the loss from three terms of CRF, and the loss from feature learning.

For the loss from the three terms of CRF, the mean field approximation inference is adopted to approximate the exact distribution $P(\mathbf{Y})$ by a simpler distribution $Q(\mathbf{Y})$, which can be expressed as the product of independent marginal distributions, $Q(\mathbf{Y}) = \Pi_i Q_i(y_i)$. For unary and pairwise potentials, the following iterative update equation is derived from [43]:

$$Q_i(y_i = l) = \frac{1}{Z_i}\exp\Bigg\{-\psi_u(y_i) - \sum_{l' \in L}\mu(l, l') \times \sum_{n=1}^{P} w^{(n)} \sum_{j \neq i} k_G^{(n)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(l')\Bigg\}. \tag{13}$$

For the superpixel-based higher order potentials, the update equation is defined as follows:

$$Q_i(y_i = l) = \frac{1}{Z_i}\exp\Bigg\{-\sum_{l' \in \mathcal{L}}\sum_{i=1}^{n} w_i^{\text{sub}} k^{\text{sub}}\big(\mathbf{f}_i^{\text{sub}}\big) Q_j(l') - \sum_{l' \in L}\sum_{i=1}^{n} w_i^{\text{pix}} k^{\text{pix}}\big(\mathbf{f}_i^{\text{pix}}\big) Q_j(l')\Bigg\}. \tag{14}$$

This operation is differentiable with respect to weights w_i^{sub} and w_i^{pix}, we can optimize them using back propagation. It is also differentiable with respect to the $Q(\mathbf{Y})$, allowing us to optimize previous layers in the network.

For the loss from feature learning, it can be treated as the general error of the objective function which is fed back to the deep network using the chain rule [55], [56]. Since the parameters of unary, pairwise, and higher order potentials in CRF are differentiable with respect to $Q_i(y_i)$ distribution inputs at each iteration, it makes the CRF another layer of a neural network. Therefore, HSINet-CRF fully integrates the HSINet and CRF, making it possible to train the whole deep network end-to-end with the usual back-propagation algorithm. Then the two components HSINet and CRF can learn how to cooperate with each other to obtain the optimal results during the training of the HSINet-CRF.

The architecture of the HSINet is implemented using TensorFlow deep learning library. In the deconvolution process, the bilinear interpolation is designed to upsample the feature maps. During training, five iterations of mean field inference are performed to avoid gradient vanishing, and the parameters of the whole network are optimized end-to-end utilizing the back-propagation algorithm. For the tradeoff factors in HSINet-CRF, we set $\lambda_1 = 0.01$, $\lambda_2 = 0.01$, and $\lambda_3 = 0.001$ in the experiment. A certain number of pixels from the HSI are random sampled for training and the remaining data is used to evaluate the performance of the proposed network. For each pixel, we crop the 1×1 and its surrounding 3×3, 5×5 neighboring pixels for learning convolutional layers. We augment the number of training samples four times by mirroring the training samples across the horizontal, vertical and diagonal axes.

IV. Experiments

In this section, we evaluate the performance of the proposed method for HSIs classification. We first briefly describe the used HSI data sets. Afterward, we evaluate the self-supervised classification results of HSINet. Then the performance of the proposed method HSINet-CRF is validated for HSI classification, and meanwhile we validate the effectiveness of the submodules in HSINet-CRF.

A. Experimental Data Sets

Three HSI data sets are used to evaluate the performance of the proposed method.

The first data set is the Indian Pines data set, which was gathered by AVIRIS sensor over the Indian Pines test site of North-Western Indiana in 1992. It consists of 145 × 145 pixels and 224 spectral reflectance bands in the wavelength range 0.4–2.5 μm with a spatial resolution of 20 m. The bands covering the region of water absorption (104–108, 150–163, and 220) are removed and hence 200 out of the 224 bands are preserved. The data set contains 10 classes and 9620 labeled pixels. The detailed information is listed in Table I.

The second data set is the Salinas data set, which was collected by the 224-band AVIRIS sensor over Salinas Valley, CA, USA. The image size is 512 × 217 pixels and is characterized by high spatial resolution (3.7 m pixels). As with Indian Pines data set, 20 water absorption bands (108–112, 154–167, and 224) out of 224 bands are discarded, thus 204 bands are used in our experiment. The Salinas data set contains 16 classes and 54 129 labeled pixels as shown in Table I.

The third data set is the University of Pavia data set (PaviaU), which was acquired by the ROSIS-03 sensor over an urban area, northern Italy. The spatial size is 610 × 340 and the spatial resolution is 1.3 m. 12 noisy bands are removed and 103 out of the 115 bands are used in our experiment. There are nine classes in PaviaU and 42 776 labeled pixels. The details are shown in Table I.

Three metrics of overall accuracy (OA), average accuracy (AA), and Kappa coefficient are used to evaluate the classification results.

B. Self-Supervised Classification Evaluation

In order to demonstrate the quality of the three level features learned by our HSINet in a self-supervision format, we compare the self-supervised classification results between HSINet and the following state-of-the-art methods: KNN, LLE [44], NNLRS [45], LRR [46], LapLRR [47], and SSAE [48]. For HSI classification, we use the local and global consistency (LGC) [49] to compare the effectiveness of different methods. In LGC, the labeled data and unlabeled data need to be specified for HSI classification. Thus, we randomly select 15 samples of the training data as the labeled data for the Indian Pines data set and select 20 samples of the training data as the labeled data for the Salinas and University of Pavia data sets, the remaining data are unlabeled data.

The classification results are listed in Tables II–IV, in comparison with the other methods, the HSINet achieves the best

TABLE I

LAND COVER CLASSES WITH SAMPLES NUMBER FOR THE INDIAN PINES, SALINAS, AND PAVIA UNIVERSITY DATA SETS

Indian Pines			Salinas			University of Pavia		
Class	Land Cover Type	Number	Class	Land Cover Type	Number	Class	Land Cover Type	Number
1	Alfalfa	46	1	Weeds-1	2009	1	Asphalt	6631
2	Corn-notill	1428	2	Weeds-2	3726	2	Meadows	18649
3	Corn-mintill	830	3	Fallow	1976	3	Gravel	2099
4	Corn	237	4	Fallow-plow	1394	4	Trees	3064
5	Grass-pasture	483	5	Fallow-smooth	2678	5	Metal sheets	1345
6	Grass-trees	730	6	Stubble	3959	6	Bare Soil	5029
7	Grass-pasture-mowed	28	7	Celery	3579	7	Bitumen	1330
8	Hay-windrowed	478	8	Grapes	11271	8	Bricks	3682
9	Oats	20	9	Soil	6203	9	Shadows	947
10	Soybean-notill	972	10	Corn	3278		Total	42776
11	Soybean-mintill	2455	11	Lettuce-4wk	1068			
12	Soybean-clean	593	12	Lettuce-5wk	1927			
13	Wheat	205	13	Lettuce-6wk	916			
14	Woods	1265	14	Lettuce-7wk	1070			
15	Bldg-grass-trees	386	15	Vinyard-untrained	7268			
16	Stone-Steel-Towers	93	16	Vinyard-trellis	1807			
	Total	10249		Total	54129			

TABLE II

HSI CLASSIFICATION RESULTS (%) BY SELECTING 15 LABELED SAMPLES FOR EACH CLASS ON THE INDIAN PINES DATA SET

Class	KNN	LLE	NNLRS	LRR	LapLRR	SSAE	HSINet
Alfalfa	92.59	96.43	100	81.48	100	100	100
Corn-notill	35.49	43.40	39.52	38.58	58.51	58.64	66.89
Corn-mintill	38.59	31.01	36.41	33.05	30.34	49.88	62.44
Corn	62.84	59.82	92.24	47.25	98.28	100	100
Grass-pasture	62.61	59.35	55.76	67.89	62.34	79.06	83.23
Grass-trees	64.84	70.51	69.89	67.93	67.40	90.76	98.03
Grass-pasture-mowed	77.78	80.00	100	77.78	100	100	100
Hay-windrowed	73.86	65.00	99.15	68.63	73.73	100	99.69
Oats	100	100	100	100	100	100	100
Soybean-notill	52.98	33.44	56.11	45.75	58.39	82.22	77.46
Soybean-mintill	42.20	54.25	33.71	49.67	55.84	62.79	78.38
Soybean-clean	25.61	31.22	51.02	24.74	32.31	69.55	74.96
Wheat	83.87	74.87	96.00	74.19	100	98.95	99.47
Woods	95.91	96.79	80.95	94.22	74.44	85.92	96.47
Bldg-grass-trees	16.25	10.05	30.53	21.53	18.26	54.32	69.11
Stone-Steel-Towers	87.84	93.33	100	94.59	100	100	100
OA	53.47 ± 0.56	55.44 ± 0.26	55.79 ± 0.29	53.93 ± 0.37	60.62 ± 0.25	75.16 ± 0.33	**83. 01 ± 0.22**
AA	63.33 ± 0.18	62.47 ± 0.23	71.33 ± 0.22	61.71 ± 0.42	70.62 ± 0.19	83.26 ± 0.31	87. 88 **± 0.17**
Kappa	0.476 ± 0.004	0.493 ± 0.002	0.517 ± 0.003	0.477 ± 0.005	0.568 ± 0.001	0.724 ± 0.003	**0.8**19 **± 0.002**

classification accuracies. It indicates that the HSINet can learn heterogeneous features from three levels by self-supervision and make HSI different-level features be consistent.

C. Classification Results of HSINet-CRF

To validate the performance of the HSINet-CRF with a relative small number of training samples, we randomly select 200 samples from each class as the training samples. Since pixel number of some classes in the Indian data set is insufficient, we select 10 classes in the Indian to train and test. The rests are used for testing the proposed network. We report the HSI classification performance of different methods over 20 random splits on the testing data set (see Tables V–VII).

The following four approaches are utilized to compare with the HSINet-CRF in terms of HSI classification accuracy.

TABLE III
HSI CLASSIFICATION RESULTS (%) BY SELECTING 20 LABELED SAMPLES FOR EACH CLASS ON THE SALINAS DATA SET

Class	KNN	LLE	NNLRS	LRR	LapLRR	SSAE	HSINet
Weeds-1	97.79	99.43	99.33	99.45	98.34	100	100
Weeds-2	99.15	93.11	99.72	94.90	96.32	68.60	100
Fallow	82.58	67.02	70.21	80.34	84.83	89.36	100
Fallow-plow	100	99.19	99.23	99.16	99.06	93.80	100
Fallow-smooth	96.77	74.42	75.19	95.56	98.39	76.36	100
Stubble	99.73	98.45	98.19	98.96	98.67	97.93	100
Celery	97.93	89.37	92.53	99.65	99.41	98.85	100
Grapes	54.83	59.71	61.06	79.49	63.14	89.08	67.57
Soil	96.17	99.35	99.51	99.71	97.50	99.84	99.10
Corn	80.84	89.31	88.05	90.91	82.79	87.74	97.30
Lettuce-4wk	98.85	95.88	99.63	98.85	98.85	95.88	100
Lettuce-5wk	100	96.17	96.17	82.08	98.95	74.32	100
Lettuce-6wk	97.22	87.80	89.02	94.44	95.83	97.56	100
Lettuce-7wk	96.55	72.16	72.16	83.91	93.10	68.04	99.10
Vinyard-untrained	71.00	93.86	93.86	56.58	69.31	73.36	85.59
Vinyard-trellis	92.55	99.68	99.16	95.65	96.89	100	100
OA	83.14 ± 0.33	85.36 ± 0.31	86.42 ± 0.22	86.56 ± 0.29	85.04 ± 0.21	87.61 ± 0.32	**96.82 ± 0.19**
AA	91.37 ± 0.26	88.43 ± 0.33	89.56 ± 0.17	90.60 ± 0.23	91.96 ± 0.19	88.17 ± 0.27	**96.79 ± 0.17**
Kappa	0.812 ± 0.01	0.842 ± 0.003	0.851 ± 0.002	0.857 ± 0.003	0.833 ± 0.003	0.862 ± 0.003	**0.966 ± 0.002**

The best results are highlighted in bold.

TABLE IV
HSI CLASSIFICATION RESULTS (%) BY SELECTING 20 LABELED SAMPLES FOR EACH CLASS ON THE PAVIA UNIVERSITY DATA SET

Class	KNN	LLE	NNLRS	LRR	LapLRR	SSAE	HSINet
Asphalt	18.20	12.75	35.83	4.04	18.82	37.83	90.20
Meadows	82.87	74.25	65.44	49.49	82.76	68.57	95.07
Gravel	50.53	35.79	81.00	81.05	77.89	91.50	61.90
Trees	92.66	96.50	89.86	93.36	90.91	77.36	84.31
Metal sheets	100	100	94.40	99.13	99.13	96.00	99.26
Bare Soil	23.40	25.26	57.20	12.22	26.71	89.86	58.45
Bitumen	69.03	61.95	99.32	34.51	50.44	94.31	80.45
Bricks	79.31	84.77	62.57	16.67	57.76	74.86	77.17
Shadows	93.33	98.67	47.06	88.00	98.67	63.53	95.79
OA	64.89 ± 0.13	60.32 ± 0.19	63.66 ± 0.31	41.39 ± 0.16	64.20 ± 0.27	70.06 ± 0.21	**85.76 ± 0.18**
AA	67.70 ± 0.18	65.55 ± 0.21	70.30 ± 0.25	53.16 ± 0.22	67.01 ± 0.22	77.09 ± 0.17	**82.51 ± 0.15**
Kappa	0.533 ± 0.002	0.480 ± 0.002	0.551 ± 0.003	0.303 ± 0.002	0.539 ± 0.002	0.627 ± 0.002	**0.808 ± 0.002**

The best results are highlighted in bold.

1) A multilayer fully connected neural network (MDNN) is compared to demonstrate the performance difference between DNN and our proposed network.
2) A shallower CNN consists of two convolutional layers and two fully connected layers (SCNN) [50] is used to compare the performance between CNN and our proposed network.
3) A recent contextual deep CNN (CDCNN) with nine convolutional layers [7] and a deep belief network (D-DBN) [51] are utilized to compare with our network.

From the classification results listed in Tables V–VII and classification maps shown in Figs. 3–5, we have the following observations.

1) The classification accuracies obtained by the HSINet-CRF are higher than those obtained by other methods on the three HSI data sets. It indicates that the HSINet-CRF can effectively exploit the spatio–spectral information of the HSI.
2) As shown in Figs. 3–5, the HSINet-CRF has more compact HSI classification maps on the three HSI data sets, which validates the superpixel-level feature can provide useful spatial information for the HSI classification.

D. Discussion

We go deeper into the efficacy of HSINet-CRF by five experiments: end-to-end trained system, collaboration among CRF terms, block, superpixel, and multiscale filter bank.

1) Effectiveness of the End-to-End Trained System: To validate the effectiveness of the end-to-end trained system, we compare the HSINet-CRF with NN-CRF, where the

TABLE V
HSI CLASSIFICATION RESULTS (%) BY SELECTING 200 SAMPLES FOR EACH CLASS ON THE INDIAN PINES DATA SET

Class	MDNN	SCNN	D-DBN	CDCNN	HSINet-conv	NN-CRF	HSINet-CRF-in	HSINet-SP	HSINet-CRF
Corn-notill	70.88	75.99	86.07	90.10	94.41	99.30	96.26	94.59	95.45
Corn-mintill	91.98	93.21	88.71	97.10	100	92.77	97.99	96.40	99.40
Grass-pasture	88.34	93.09	98.26	100	98.97	100	100	100	100
Grass-trees	99.44	99.44	99.86	93.80	100	100	100	100	100
Hay-windrowed	100	100	100	100	100	100	100	100	100
Soybean-notill	91.39	93.80	85.05	95.90	93.81	98.45	96.58	98.20	97.94
Soybean-mintill	85.87	87.72	89.97	87.10	94.09	96.74	97.28	90.99	99.19
Soybean-clean	96.68	97.73	91.83	96.40	100	91.60	98.31	100	99.16
Woods	87.63	93.98	96.01	99.40	100	100	99.74	94.59	99.21
Bldg-grass-trees	89.34	90.71	80.38	93.50	81.82	83.12	87.93	94.59	100
OA	87.68 ±0.09	90.16 ±0.12	90.97 ±0.12	93.61 ± 0.56	96.26 ± 0.16	97.09 ± 0.11	97.61 ± 0.10	96.83 ± 0.13	**98.70 ± 0.08**
AA	90.15 ±0.85	92.57 ±0.05	91.61 ±0.11	95.33 ± 0.48	96.31 ± 0.19	96.20 ± 0.09	97.41 ± 0.15	96.94 ± 0.11	**99.04 ± 0.06**
Kappa	0.856 ±0.002	0.888 ±0.001	0.895 ± 0.001	0.914 ± 0.005	0.983 ± 0.002	0.966 ± 0.001	0.972 ± 0.002	0.966 ± 0.002	**0.985 ± 0.001**

The best results are highlighted in bold.

TABLE VI
HSI CLASSIFICATION RESULTS (%) BY SELECTING 200 SAMPLES FOR EACH CLASS ON THE SALINAS DATA SET

Class	MDNN	SCNN	D-DBN	CDCNN	HSINet-conv	NN-CRF	HSINet-CRF-in	HSINet-SP	HSINet-CRF
Weeds-1	100	100	100	100	100	100	100	100	100
Weeds-2	100	97.80	98.92	100	100	100	100	100	100
Fallow	99.47	100	85.86	100	100	95.00	100	97.30	96.67
Fallow-plow	94.57	93.80	100	99.30	100	100	100	100	100
Fallow-smooth	86.05	67.44	98.51	98.50	98.94	100	98.32	98.20	100
Stubble	91.97	99.74	100	100	100	100	100	100	100
Celery	99.71	83.91	100	99.80	99.20	100	100	100	100
Grapes	87.65	88.09	85.46	83.40	94.67	96.00	86.12	84.68	98.82
Soil	100	100	99.35	99.60	100	100	100	99.10	100
Corn	94.03	95.28	98.17	94.60	99.13	100	99.12	93.69	100
Lettuce-4wk	95.88	90.72	90.57	99.30	100	100	100	100	100
Lettuce-5wk	68.85	87.98	100	100	100	100	100	100	100
Lettuce-6wk	73.17	87.80	100	100	100	100	100	100	100
Lettuce-7wk	88.66	78.35	100	100	100	100	100	99.10	100
Vinyard-untrained	85.63	95.40	75.21	100	89.76	93.79	94.36	78.38	98.17
Vinyard-trellis	100	100	100	98.00	100	100	100	100	100
OA	91.82 ±0.06	92.42 ±0.05	92.61 ±0.16	95.07 ±0.23	97.36 ±0.18	98.15 ±0.19	98.56 ±0.13	97.02 ±0.17	**99.38 ±0.05**
AA	91.60 ±0.12	91.64 ±0.78	95.75 ±0.12	98.28 ±0.21	98.86±0.16	99.05 ±0.13	98.63 ±0.15	96.90 ±0.13	**99.60 ±0.03**
Kappa	0.909 ±0.001	0.915 ±0.001	0.918 ±0.003	0.954 ±0.002	0.971±0.003	0.979 ±0.002	0.985 ±0.002	0.970 ±0.002	**0.993 ±0.002**

The best results are highlighted in bold.

TDNN and MCNN are trained first, and then the CRF is applied on top of the neural networks output. As shown in Tables V–VII and Figs. 3–5, the end-to-end HSINet-CRF significantly outperforms the offline application of CRF as a postprocessing method. This observation supports the fact that the two components HSINet and CRF can learn how to cooperate with each other to obtain the optimal results during the training of the HSINet-CRF.

2) Effectiveness of Collaboration Among CRF Terms: To verify the effectiveness of the collaboration among CRF terms, we compare the HSINet-CRF with HSINet-CRF-in, which means the CRF terms are independent and do not incorporate

TABLE VII

HSI CLASSIFICATION RESULTS (%) BY SELECTING 200 SAMPLES FOR EACH CLASS ON THE PAVIA UNIVERSITY DATA SET

Class	MDNN	SCNN	D-DBN	CDCNN	HSINet-conv	NN-CRF	HSINet-CRF-in	HSINet-SP	HSINet-CRF
Asphalt	78.53	87.34	89.58	94.60	100	100	99.56	99.05	100
Meadows	87.23	94.63	93.93	96.00	97.85	96.79	92.06	97.16	99.82
Gravel	89.33	86.47	88.41	95.5000	100	100	95.51	95.73	100
Trees	97.50	96.29	95.64	95.90	98.91	95.65	100	97.63	100
Metal sheets	100	99.65	99.56	100	100	100	100	100	100
Bare Soil	88.11	93.23	93.87	94.10	88.74	97.33	98.49	87.68	94.70
Bitumen	85.48	93.19	93.19	97.50	100	100	100	100	100
Bricks	85.01	86.42	91.07	88.80	98.18	100	98.93	94.31	100
Shadows	100	100	100	99.50	100	100	100	100	100
OA	90.06 ±0.51	92.56 ±0.18	93.11 ±0.06	95.97 ±0.46	96.57 ±0.19	97.97±0.22	98.22 ±0.13	96.88 ±0.15	**99.30±0.11**
AA	90.13 ±0.53	93.02 ±0.15	93.92 ±0.07	95.77 ±0.39	98.19±0.21	98.86±0.18	98.28 ±0.15	96.84 ±0.13	**99.39±0.12**
Kappa	0.888 ±0.006	0.901 ±0.002	0.908 ±0.001	0.936 ±0.005	0.967±0.003	0.973±0.002	0.980 ±0.003	0.964 ± 0.002	**0.991±0.002**

The best results are highlighted in bold.

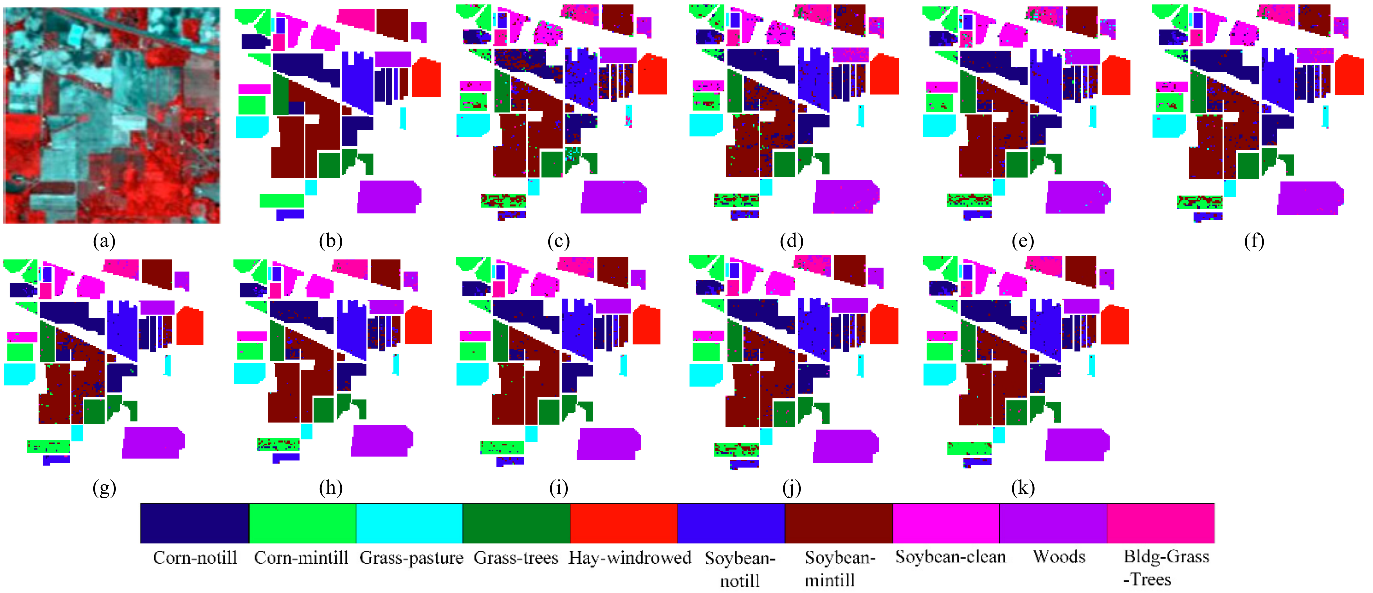


Fig. 3. Classification maps of the Indian Pine data set. (a) False-color image. (b) Ground truth. (c)–(k) Classification map obtained by MDNN, SCNN, D-DBN, CDCNN, HSINet-conv, NN-CRF, HSINet-CRF-in, HSINet-SP, and HSINet-CRF.

the loss of feature learning. From the results listed in Tables V–VII and Figs. 3–5, we observe that there is an obvious advantage of the HSINet-CRF with the subpixel-pixel loss, pixel-superpixel loss, and superpixel-superpixel loss over the network without these regularizations. In terms of the most classes of the three data sets, HSINet-CRF obtains higher classification accuracy than HSINet-CRF-in, such as Corn-mintill, Soybean-notill and Soybean-mintill in the Indian Pines data set. It attributes to the fact that the three loss terms help make the three CRF terms collaborate with each other.

3) Effectiveness of the Block: To verify the effectiveness of the block, we also compare the HSINet-CRF with the network in which each convolutional layer is regular. The classification results are lisited in Tables V–VII and Figs. 3–5, HSINet-conv means the second to the seventh layers only have convolutions. Since our method integrates the information of the pixel clustering results into feature learning, performance of the HSINet-CRF significantly outperforms the network with each convolutional layer being regular. The classification results in almost classes of the three data sets acquired by HSINet-CRF are better, and there are less "salt-and-pepper" in the classification maps obtained by HSINet-CRF than HSINet-conv.

4) Effectiveness of the Superpixel: To verify the effectiveness of the superpixel, we compare the HSINet-CRF with the HSINet-SP, which only learns subpixel-level feature and pixel-level feature, and does not consider the superpixel-level feature. As shown in Tables V–VII and Figs. 3–5,

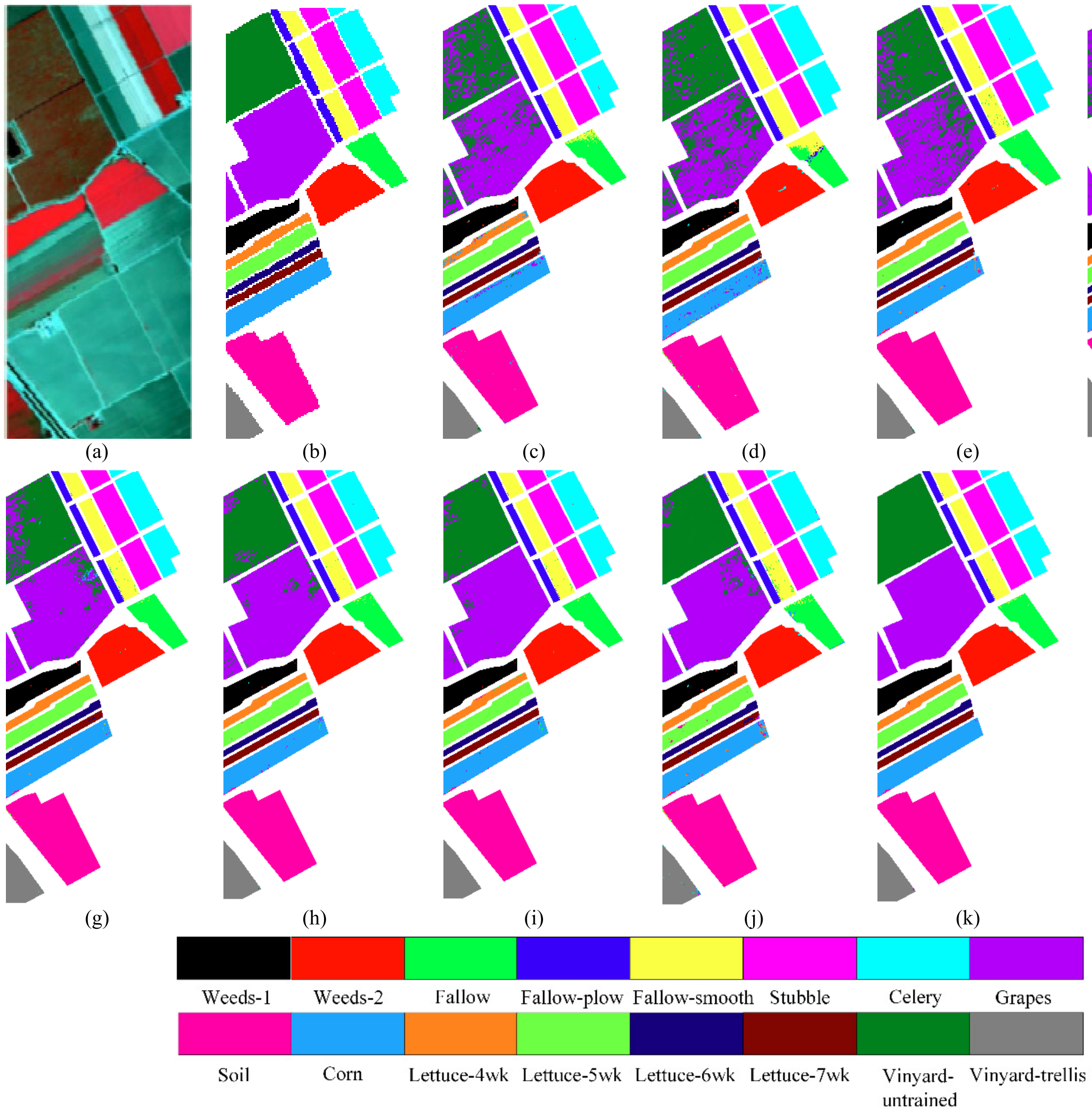


Fig. 4. Classification maps of the Salinas data set. (a) False-color image. (b) Ground truth. (c)–(k) Classification map obtained by MDNN, SCNN, D-DBN, CDCNN, HSINet-conv, NN-CRF, HSINet-CRF-in, HSINet-SP, and HSINet-CRF.

TABLE VIII

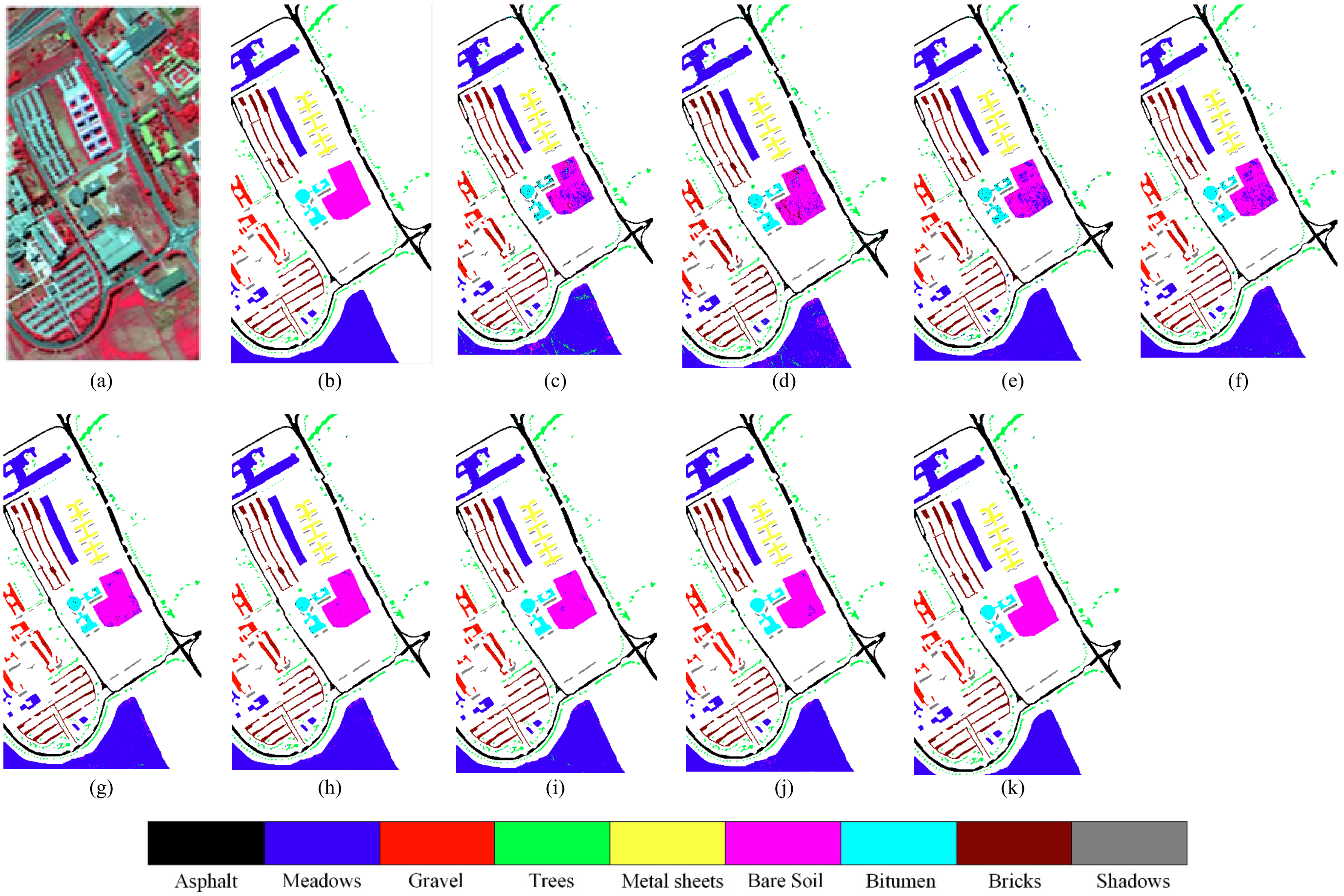
PERFORMANCE COMPARISON OF THE MULTISCALE FILTER BANKS WITH DIFFERENT CONFIGURATIONS OF THE PROPOSED NETWORK. $\sim 7 \times 7$ MEANS THE MULTISCALE FILTER BANK CONSISTING OF 1×1, 3×3, 5×5 AND 7×7 CONVOLUTIONAL FILTERS

Dateset	Indian Pines	Salinas	PaviaU
1×1	81.69 ±0.30	86.91 ±0.58	82.34 ±0.45
$\sim 3 \times 3$	89.13 ±0.39	92.13 ±0.17	93.52 ±0.19
$\sim 5 \times 5$	**97.61 ±0.08**	**98.56 ±0.03**	**98.22 ±0.13**
$\sim 7 \times 7$	97.52 ±0.06	98.39 ±0.11	98.15 ±0.09

The best results are highlighted in bold.

the performance of the HSINet-CRF significantly outperforms the network without considering the superpixel-level feature. The classification results support the fact that the superpixel can provide more contextual information than the subpixel and pixel, and can well avoid the "salt-and-pepper" problem generated in the subpixel-based or pixel-based classification. As shown in Figs. 3–5, HSINet-CRF obviously performs better than HSINet-SP on preserving the boundary. The boundaries of the classification results obtained by HSINet-CRF are more smooth.

5) Effectiveness of the Multiscale Filter Bank: To validate the effectiveness of the multiscale filter bank used to jointly exploit spatial–spectral information, we compare the multiscale filter bank with different configurations consisting of: 1×1, $\sim 3 \times 3$, $\sim 5 \times 5$, and $\sim 7 \times 7$. The $\sim 7 \times 7$ means the multiscale filter bank consisting of 1×1, 3×3, 5×5, and 7×7 convolutional filters, others are similar. As shown in Table VIII, our multiscale filter bank significantly outperforms the network with only 1×1 convolutional filter on the three data sets. The reason is that the network with only 1×1 convolutional filter fails to use the data augmentation due to the nonexistence of spatial filtering and cannot exploit the spatial–spectral information. We also compare the HSINet-CRF with the multiscale filter bank configuration $\sim 7 \times 7$.



(a) (b) (c) (d) (e) (f)

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

Fig. 5. Classification maps of the Pavia University data set. (a) False-color image. (b) Ground truth. (c)–(k) Classification map obtained by MDNN, SCNN, D-DBN, CDCNN, HSINet-conv, NN-CRF, HSINet-CRF-in, HSINet-SP, and HSINet-CRF.

Since the ∼7 × 7 contains more noise, the selected ∼5 × 5 obtains much higher classification accuracies.

V. Conclusion

In this paper, we have developed the HSINet-CRF to learn complementary and consistent features for HSI classification. HSINet is first proposed to learn three complementary features of different levels by self-supervision, which contains TDNN and MCNN. To boost the self-supervised feature learning by probabilistic model, the CRF framework is embedded into HSINet. In the experiment, we evaluate the performance of the proposed method for HSIs classification, and also validate the effectiveness of the submodules in HSINet-CRF. The experiment results prove that our method performs better than other related approaches. The future work will involve exploiting this framework on other labeling and regression tasks such as 3-D point clouds parsing, and image denoising.

References

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

[2] P. Ghamisi, J. A. Benediktsson, and M. O. Ulfarsson, "Spectral–spatial classification of hyperspectral images based on hidden Markov random fields," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2565–2574, May 2014.

[3] C. Shi and C.-M. Pun, "Superpixel-based 3D deep neural networks for hyperspectral image classification," *Pattern Recognit.*, vol. 74, pp. 600–616, Feb. 2017.

[4] 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, Mar. 2012.

[5] X. Huang and L. Zhang, "An adaptive mean-shift analysis approach for object extraction and classification from urban hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 12, pp. 4173–4185, Dec. 2008.

[6] L. Zhang, Q. Zhang, B. Du, X. Huang, Y. Y. Tang, and D. Tao, "Simultaneous spectral-spatial feature selection and extraction for hyperspectral images," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 16–28, Jan. 2018.

[7] H. Lee and H. Kwon, "Going deeper with contextual CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4843–4855, Oct. 2017.

[8] G. Zhang, X. Jia, and J. Hu, "Superpixel-based graphical model for remote sensing image mapping," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 11, pp. 5861–5871, Nov. 2015.

[9] L. Y. Fang, S. T. Li, X. D. Kang, and J. A. Benediktsson, "Spectral-spatial classification of hyperspectral images with a superpixel-based discriminative sparse model," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 8, pp. 4186–4201, Aug. 2015.

[10] J. Li, H. Zhang, and L. Zhang, "Efficient superpixel-level multitask joint sparse representation for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 10, pp. 5338–5351, Oct. 2015.

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

[12] 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, Mar. 2013.

[13] 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.

[14] T. Lu, S. Li, L. Fang, X. Jia, and J. A. Benediktsson, "From subpixel to superpixel: A novel fusion framework for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4398–4411, Aug. 2017.

[15] S. Li, T. Lu, L. Fang, X. Jia, and J. A. Benediktsson, "Probabilistic fusion of pixel-level and superpixel-level hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7416–7430, Dec. 2016.

[16] O. Eches, N. Dobigeon, C. Mailhes, and J.-Y. Tourneret, "Bayesian estimation of linear mixtures using the normal compositional model. Application to hyperspectral imagery," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1403–1413, Jun. 2010.

[17] 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.

[18] L. Gomez, Y. Patel, M. Rusiol, D. Karatzas, and C. V. Jawahar, "Self-supervised learning of visual features through embedding images into text topic spaces," in *Proc. CVPR*, 2017, pp. 2017–2026.

[19] A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba, "Ambient sound provides supervision for visual learning," in *Proc. ECCV*, 2016, pp. 801–816.

[20] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proc. ICCV*, 2015, pp. 1422–1430.

[21] L. Bertelli, T. Yu, D. Vu, and B. Gokturk, "Kernelized structural SVM learning for supervised object segmentation," in *Proc. CVPR*, Jun. 2011, pp. 2153–2160.

[22] J. Yang and M.-H. Yang, "Top-down visual saliency via joint CRF and dictionary learning," in *Proc. CVPR*, Jun. 2012, pp. 2296–2303.

[23] P. M. Atkinson, "Mapping sub-pixel boundaries from remotely sensed images," in *Innovations in GIS*, vol. 4. London, U.K.: Taylor & Francis, 1997, pp. 166–180.

[24] X. Xu, Y. Zhong, and L. Zhang, "Adaptive subpixel mapping based on a multiagent system for remote-sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 2, pp. 787–804, Feb. 2014.

[25] J. Li, I. Dópido, P. Gamba, and A. Plaza, "Complementarity of discriminative classifiers and spectral unmixing techniques for the interpretation of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2899–2912, May 2015.

[26] M.-D. Iordache, J. Bioucas-Dias, and A. Plaza, "Sparse unmixing of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 6, pp. 2014–2039, Jun. 2011.

[27] P. V. Giampouras, K. E. Themelis, A. A. Rontogiannis, and K. D. Koutroumbas, "Simultaneously sparse and low-rank abundance matrix estimation for hyperspectral image unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4775–4789, Aug. 2016.

[28] Y. Zhou and Y. Wei, "Learning hierarchical spectral–spatial features for hyperspectral image classification," *IEEE Trans. Cybern.*, vol. 46, no. 7, pp. 1667–1678, Jul. 2016.

[29] Y. Gao, R. Ji, P. Cui, Q. Dai, and G. Hua, "Hyperspectral image classification through bilayer graph-based learning," *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 2769–2778, Jul. 2014.

[30] P. Zhong and R. Wang, "Learning conditional random fields for classification of hyperspectral images," *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 1890–1907, Jul. 2014.

[31] 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.

[32] G. Camps-Valls, T. V. B. Marsheva, and D. Zhou, "Semi-supervised graph-based hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 10, pp. 3044–3054, Oct. 2007.

[33] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[34] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2011, pp. 2097–2104.

[35] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[36] M. Van den Bergh, X. Boix, G. Roig, and L. Van Gool, "SEEDS: Superpixels extracted via energy-driven sampling," *Int. J. Comput. Vis.*, vol. 111, no. 3, pp. 298–314, 2015.

[37] W. Fu, S. Li, L. Fang, and J. A. Benediktsson, "Adaptive spectral–spatial compression of hyperspectral image with sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 671–682, Feb. 2017.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Jun. 2016, pp. 770–778.

[39] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. CVPR*, Jun. 2015, pp. 1–9.

[40] W.-Y. Chen, Y. Song, H. Bai, C.-J. Lin, and E. Y. Chang, "Parallel spectral clustering in distributed systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 568–586, Mar. 2011.

[41] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. CVPR*, Jun. 2015, pp. 3431–3440.

[42] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. ICML*, 2001, pp. 282–289.

[43] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *Proc. NIPS*, 2011, pp. 109–117.

[44] J. Wang, F. Wang, C. Zhang, H. C. Shen, and L. Quan, "Linear neighborhood propagation and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 9, pp. 1600–1615, Sep. 2009.

[45] L. Zhuang *et al.*, "Constructing a nonnegative low-rank and sparse graph with data-adaptive features," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3717–3728, Nov. 2015.

[46] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proc. ICML*, 2010, pp. 663–670.

[47] W. Yang, Y. Gao, Y. Shi, and L. Cao, "MRM-lasso: A sparse multiview feature selection method via low-rank analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 11, pp. 2801–2815, Nov. 2015.

[48] 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.

[49] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. NIPS*, 2003, pp. 321–328.

[50] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 43, no. 6, pp. 1351–1362, 2015.

[51] P. Zhong, Z. Gong, S. Li, and C. B. Schönlieb, "Learning to diversify deep belief networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3516–3530, Jun. 2017.

[52] B. Du, X. Tang, Z. Wang, L. Zhang, and D. Tao, "Robust graph-based semisupervised learning for noisy labeled data via maximum correntropy criterion," *IEEE Trans. Cybern.*, vol. 48, no. 12, pp. 1–14, Dec. 2018, doi: 10.1109/TCY.B.2018.2804326.

[53] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. CVPR*, Jun. 2014, pp. 580–587.

[54] Y. Tarabalka, J. Chanussot, and J. A. Benediktsson, "Segmentation and classification of hyperspectral images using watershed transformation," *Pattern Recognit.*, vol. 43, no. 7, pp. 2367–2379, 2010.

[55] B. Du, Z. Wang, L. Zhang, L. Zhang, and D. Tao, "Robust and discriminative labeling for multi-label active learning based on maximum correntropy criterion," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1694–1707, Apr. 2017.

[56] B. Du, S. Wang, C. Xu, N. Wang, L. Zhang, and D. Tao, "Multi-task learning for blind source separation," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4219–4231, Sep. 2018.

Yuebin Wang received the Ph.D. degree from the School of Geography, Beijing Normal University, Beijing, China, in 2016.

He was a Post-Doctoral Researcher with the School of Mathematical Sciences, Beijing Normal University. He is currently an Assitant Professor with the School of Land Science and Technology, China University of Geosciences, Beijing. His research interests include remote sensing imagery processing and 3-D urban modeling.

Jie Mei is currently pursuing the master's degree with the Faculty of Geographical Science, Beijing Normal University, Beijing, China.

His research interests include remote sensing image processing and image-based and LiDAR-based segmentation and reconstruction.

Liqiang Zhang received the Ph.D. degree in geoinformatics from the Institute of Remote Sensing Applications, Chinese Academy of Sciences, Beijing, China, in 2004.

He is currently a Professor with the Faculty of Geographical Science, Beijing Normal University, Beijing. His research interests include remote sensing image processing, 3-D urban reconstruction, and spatial object recognition.

Bing Zhang is currently a Full Professor and the Deputy Director of the Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences (CAS), Beijing, China, where he has been leading lots of key scientific projects in the area of hyperspectral remote sensing for more than 20 years. He has authored over 300 publications, including more than 190 journal papers. He has edited six books/contributed book chapters on hyperspectral image processing and subsequent applications. He has developed five software systems in the image processing and applications. His research interests include the development of mathematical and physical models and image processing software for the analysis of hyperspectral remote sensing data in many different areas.

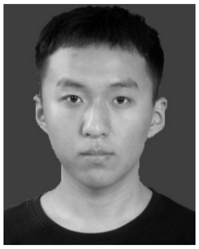
Dr. Zhang was a recipient of the National Science Foundation for Distinguished Young Scholars of China in 2013, and the 2016 Outstanding Science and Technology Achievement Prize of the Chinese Academy of Sciences for his special achievements in hyperspectral remote sensing. He is currently serving as an Associate Editor for the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING. He is also the Guest Editor for the series of special issues of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, the IEEE GEOSCIENCE AND REMOTE SENSING SOCIETY LETTERS, the *Proceedings of the IEEE*, and*Pattern Recognition Letters*. He has been served as a Technical Committee Member of the IEEE Workshop on Hyperspectral Image and Signal Processing since 2011 and the President of the Hyperspectral Remote Sensing Committee of China National Committee of International Society for Digital Earth since 2012.

Panpan Zhu is currently pursuing the Ph.D. degree with the Faculty of Geographical Science, Beijing Normal University, Beijing, China.

Her research interests include remote sensing image processing and image-based classification and retrieval.

Yang Li is currently pursuing the master's degree with the Faculty of Geographical Science, Beijing Normal University, Beijing, China.

His research interests include deep learning and remote sensing image processing.



Xingang Li is currently pursuing the master's degree with the Faculty of Geographical Science, Beijing Normal University, Beijing, China.

His research interests include remote sensing imagery processing and machine learning.