Unsupervised Spatial-Spectral CNN-Based Feature Learning for Hyperspectral Image Classification

Shuyu Zhang, Meng Xu, *Member, IEEE*, Jun Zhou, *Senior Member, IEEE*, and Sen Jia, *Senior Member, IEEE*

***Abstract*— The rapid development of remote sensing sensors makes the acquisition, analysis, and application of hyperspectral images (HSIs) more and more extensive. However, the limited sample sets, high-dimensional features, highly correlated bands, and mixing spectral information make the classification of HSIs a great challenge. In this article, an unsupervised multiscale and diverse feature learning (UMsDFL) method is proposed for HSI classification, which deeply considers the spatial–spectral features via convolutional neural networks (CNNs). Specifically, after employing the simple noniterative clustering (SNIC) algorithm with the heuristic calculation of superpixel size, the HSIs are segmented into superpixels for feature learning. The unsupervised network is designed with the convolutional encoder and decoder, the additional clustering branch, and the multilayer feature fusion to enhance the distinguishability of feature learning and the reusability of feature maps. Then, the spatial relationships and object attributes in large- and small-scale contexts are learned collaboratively through the unsupervised network to utilize the complementary multiscale characteristics. Moreover, the diverse features of hyperspectral information and nonsubsampled contourlet transform (NSCT) textures are learned simultaneously via the unsupervised network to alleviate the insufficiency of geometric representation. Finally, the random forest (RF) is adopted as the comprehensive classifier for land cover mapping based on the UMsDFL, and superpixel regularization is adopted to optimize the classification results. A series of experiments are performed on three real-world HSI datasets to demonstrate the effectiveness of our UMsDFL approach. The experimental results show that the proposed UMsDFL can achieve the overall accuracy of 79.23%, 96.49%, and 77.26% for Houston, Pavia, and Dioni datasets, respectively, when there are only five samples per class for training.**

***Index Terms*— Convolutional neural network (CNN), feature fusion, hyperspectral image (HSI), superpixel segmentation, unsupervised feature learning.**

Manuscript received July 28, 2021; revised November 22, 2021 and January 15, 2022; accepted February 16, 2022. Date of publication February 22, 2022; date of current version March 31, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 41971300 and Grant 61901278, in part by the Key Project of the Department of Education of Guangdong Province under Grant 2020ZDZX3045, and in part by the Natural Science Foundation of Guangdong Province under Grant 2021A1515011413. *(Corresponding author: Sen Jia.)*

Shuyu Zhang, Meng Xu, and Sen Jia are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China, and also with the Key Laboratory for Geo-Environmental Monitoring of Coastal Zone of the Ministry of Natural Resources, Shenzhen University, Shenzhen 518060, China (e-mail: shuyu-zhang@szu.edu.cn; m.xu@szu.edu.cn; senjia@szu.edu.cn).

Jun Zhou is with the School of Information and Communication Technology, Griffith University, Nathan, QLD 4111, Australia (e-mail: jun.zhou@griffith.edu.au).

Digital Object Identifier 10.1109/TGRS.2022.3153673

I. Introduction

With the rapid development of remote sensing observation technology, the acquisition, analysis, and application of hyperspectral images (HSIs) have become more and more extensive. Many characteristics of ground objects hidden in the narrow spectral ranges of HSIs are gradually being discovered. In contrast to the multispectral images, the HSIs can obtain abundant information on hundreds of continuous spectral bands to enhance the ability of feature extraction and object recognition [1]. Hyperspectral remote sensing plays an important role in the fields of land use and land cover (LULC) classification, target detection, agricultural monitoring, mineral mapping, environmental management, and national defense [2]. However, the limited sample sets, high-dimensional features, highly correlated bands, and mixing spectral information make the HSI classification a great challenge.

Up until now, diverse kinds of methods for HSI feature extraction and land cover classification have been developed. Early pixel-based methods use spectral information and simple features for recognition with classifiers such as K-nearest neighbor (KNN) [3], extreme learning machine (ELM) [4], and support vector machine (SVM) [5]. Later, methods based on graph embedding [6], sparse representation [7], and low-rank representation [8] are proposed to enhance the hyperspectral discriminative abilities. However, there exist the phenomena of similar objects with different spectra and different objects with similar spectra, and thus, it is difficult to distinguish confusing objects only utilizing the spectral information. Hence, spatial–spectral methods are developed to combine the spectral values and spatial structure to improve the performance. In terms of spatial processing units, methods of graph construction [9], morphological segmentation [10], and superpixel segmentation [11] are employed to divide the images into meaningful patches for the subsequent analysis. In terms of spatial contexts, methods of the Markov random field (MRF) [12], the morphological operation, and the texture extraction are widely adopted to integrate the adjacent pixel information. Specifically, morphological methods measure the shape features in HSIs through definition and iterative calculation of structural elements, such as the morphological profiles (MPs) [13], attribute profiles (APs) [14], and invariant APs [15]. Textural methods of Gabor [16], wavelet [17], and contourlet [18] transform measure the spatial relationships and changing patterns among adjacent pixels through contextual

statistical functions. However, the aforementioned methods using handcrafted features rely on HSI content and domain knowledge, and lack adaptive parameter learning and flexible deep feature extraction.

To address this defect, deep learning (DL) methods have been extensively studied and employed for the HSI classification, such as convolutional neural networks (CNNs) [19]–[22], recurrent neural networks (RNNs) [23]–[25], stacked autoencoders (SAEs) [26]–[28], graph convolutional networks (GCNs) [29]–[31], and generative adversarial networks (GANs) [32]–[35]. DL methods can automatically learn the deep features in HSIs from concrete to abstract through the network layer by layer, thereby enhancing the semantic expression. Among them, CNN methods are the most widely used, due to their intrinsic feature learning ability of convolution and the aggregation of feature maps. Yu *et al.* [36] improved the parameter optimization in CNN and applied it to HSI classification, alleviating the problems of highly correlated bands and insufficient training samples. Furthermore, Mei *et al.* [37] proposed an unsupervised spatial–spectral feature learning method using 3-D convolutional autoencoder and achieved better classification accuracy than other supervised algorithms. In order to promote the sample selection, Hu *et al.* [38] introduced the active learning strategy to construct a valuable sample set for CNN training and boost the feature extraction. For combining the advantages of different DL models, Yue *et al.* [39] merged the spectral and spatial features obtained via SAE and CNN, respectively, and employed spatial pyramid pooling to accept inputs on inconsistent scales. Similarly, Hao *et al.* [40] adopted stacked denoising autoencoder and CNN to encode the features separately and fused the features of multiple branches through adaptive class-specific weights. Another extension of CNN methods for HSI classification is to design different effective network structures to emphasize the information extraction and raise the recognition precision, such as residual CNN [41], attention-based CNN [42], and densely connected CNN [43].

However, the CNN-based methods for HSI classification are usually trained in a supervised manner, and large-scale samples are required to optimize a great number of parameters [44], [45]. The acquisition of manually labeled samples costs a lot of time and expertise, resulting in the problem of inadequate network training with a small sample set. In addition, the unsupervised methods of autoencoders learn features from unlabeled samples through image encoding and reconstruction, whereas the target of maximizing categorical discrimination is not explicit, reducing the expression ability [46], [47]. Furthermore, the insufficient utilization of multilayer feature maps from low to high levels in CNN losses part of the HSI characteristics and limits the learning performance. From a contextual perspective, spectral attributes and spatial relationships in the neighborhood are important for ground object recognition, and multiscale contexts reflect the structural information at different levels [48]–[50]. Small- and large-scale contexts mainly focus on the complementary internal features and external relationships, respectively, but there exist challenges for multiscale feature learning in an unsupervised way. On the other hand, texture information is a conductive feature description for HSI classification to reflect the spatial distribution and changing patterns of ground objects [18], [51]. CNN methods using spectral bands extract high-level semantic features of spectral attributes rather than geometric textures, which is inadequate for comprehensive and discriminative feature representation. It is significant to integrate diverse feature learning through the deep network for better performance, but how to collaboratively accomplish the texture learning with other characteristics in the unsupervised framework remains to be studied.

Consequently, in this article, we propose an unsupervised multiscale and diverse feature learning (UMsDFL) method for HSI classification, which deeply considers the spatial–spectral features via CNN, as shown in Fig. 1. In our UMsDFL, the HSI is first segmented into superpixels using the simple noniterative clustering (SNIC) algorithm [52], which is an improvement of the simple linear iterative clustering (SLIC) algorithm [53]. The SNIC has the advantages of low computational complexity and good segmentation results, and it is modified to be compatible with the hyperspectral input. On the other hand, the principal component analysis (PCA) and nonsubsampled contourlet transform (NSCT) are adopted to HSI sequentially for dimensionality reduction and geometric texture extraction, respectively. By employing the SNIC segmentation boundaries to HSI and NSCT images, HSI superpixels and NSCT superpixels are obtained for diverse deep feature learning. Based on the segmented superpixels, an effective unsupervised spatial–spectral CNN network is built for feature learning, which is synthetically designed with the convolutional encoder and decoder, the additional clustering branch, and multilayer feature map combination. Two branches of the decoder and K-means clustering are structured for image reconstruction and feature discrimination, respectively, followed by the backward error propagation. The multilayer feature map combination is accomplished through global average pooling (GAP) and adaptive weighted concatenation to enhance the feature reusability and integrity. For multiscale feature learning, we propose to learn the object attributes and spatial relationships of superpixels within small- and large-scale contexts via the unsupervised network. Multiscale feature learning is performed on HSI superpixels and NSCT superpixels in parallel, and it is beneficial to represent the complementary information of objects and patterns in multiple contexts. For diverse feature learning, we design to learn spectral attributes of HSI superpixels and geometric structure of NSCT superpixels via the unsupervised network. The deep features of contourlet textures emphasize object contours at various levels and in different directions for geometric representation, which makes up for CNN's shortcomings. Finally, the random forest (RF) algorithm is employed as the comprehensive classifier after UMsDFL, considering that it can obtain stable performance with unbalanced and limited training samples. Furthermore, superpixel regularization is adopted to optimize the pixel classification results to improve the continuity of boundaries corresponding to the superpixel segmentation and superpixel-based feature learning.

The main contributions of this article are given as follows.

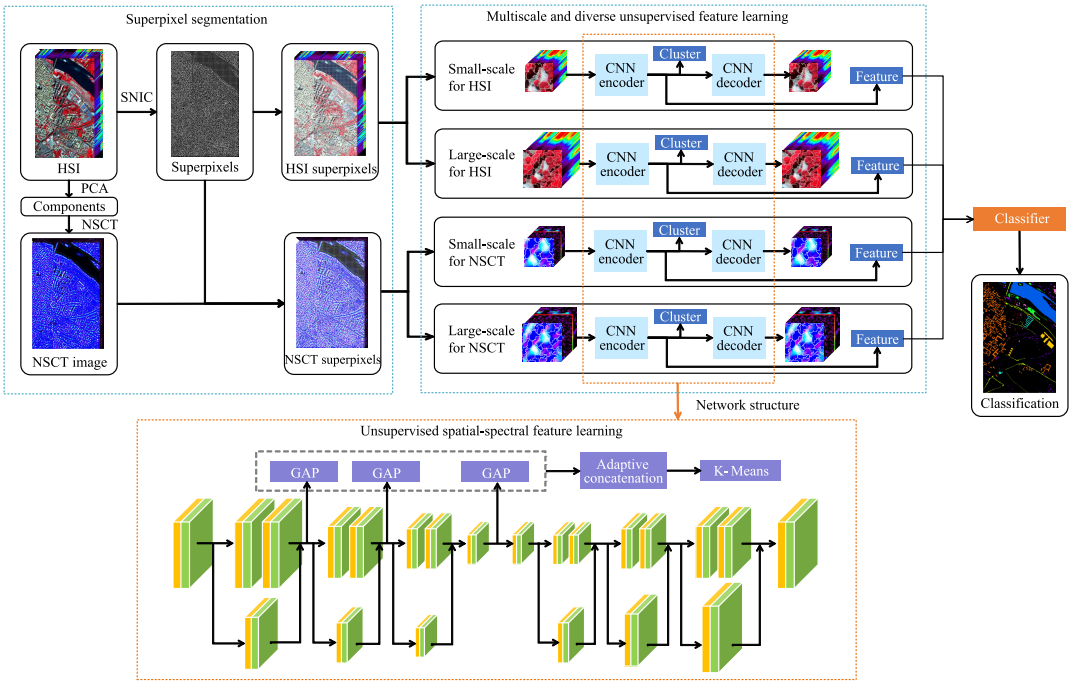

Fig. 1. Flowchart of the proposed UMsDFL method for HSI classification.

1) We propose a novel framework of effective unsupervised spatial–spectral CNN for HSI feature learning and land cover classification. With two branches of decoder and clustering, the network is trained and optimized iteratively under the error feedback of image reconstruction and pseudolabel classification to learn features from unlabeled samples and improve the feature discrimination among categories. Moreover, the information of multilayer feature maps is combined for distinguishable clustering through GAP and adaptive weighted concatenation to reuse the low-, middle-, and high-level features and enhance the intrinsic cohesion.
2) To utilize the complementary multiscale characteristics, we design to learn the object attributes and spatial relationships in small- and large-scale contexts via the unsupervised spatial–spectral CNN. The small-scale features mainly describe hyperspectral properties and the local structure of objects, and the large-scale features represent spatial relationships and distribution patterns between objects. The unsupervised multiscale deep feature learning effectively extracts the abundant contextual information of unlabeled samples in various sizes and from different perspectives.
3) To alleviate the insufficiency of geometric representation, we propose to learn the deep features of hyperspectral information and contourlet textures through the unsupervised spatial–spectral CNN. The spectral features focus on the object characteristics of optical absorption and reflection, and the textural features focus on the geometric patterns of spatial structure and distribution. The unsupervised diverse feature learning is beneficial to raise the comprehensiveness of HSI feature expression and mine the connotation of unlabeled samples.

The rest of this article is organized as follows. The proposed UMsDFL method is introduced in Section II. The experimental setup and results are illustrated in Sections III and IV, respectively. The conclusion is presented in Section V.

II. Methodology

As shown in Fig. 1, our UMsDFL method has five main steps. First, the modified SNIC algorithm is used to segment the HSI into superpixels with relatively uniform size and regular shape, which are adopted as the basic units for subsequent feature learning. Second, the effective unsupervised spatial–spectral CNN with clustering branch and multilayer combination is built for superpixel feature learning to enhance the feature distinguishability. Third, the multiscale deep features of object attributes and spatial relationships in small- and large-scale contexts are learned through the unsupervised CNN. Fourth, the diverse deep features of hyperspectral information and contourlet textures are extracted, respectively, in the unsupervised manner. Finally, based on the multiscale and diverse unsupervised feature learning, limited labeled samples are utilized to train the RF classifier and obtain the classification maps. Superpixel regularization is adopted to optimize the classification results and achieve better performance.

A. Superpixel Segmentation

The SNIC [48] algorithm is an improvement of the SLIC [49] superpixel segmentation, retaining the desirable properties of simple implementation, efficient computation,

and control over the superpixel compactness and number. In addition, the SNIC is noniterative and faster using less memory and enforces connectivity from the start, which is suitable for HSI processing. In order to be compatible with hyperspectral input, the affinity of a pixel to the centroid is modified and measured using the distance in high-dimensional space of spectral and spatial coordinates. With the spatial position X and hyperspectral vector S, the distance $D_(i, j)$ between pixel i and j is calculated by

$$D_{i,j} = \sqrt{\frac{\|X_i - X_j\|^2}{\omega_X} + \frac{\|S_i - S_j\|^2}{\omega_S}} \tag{1}$$

where $\|\cdot\|^2$ means the Euclidean distance calculation of spectral and spatial coordinates. X_i and S_i are the hyperspectral vector and the spatial position of the ith superpixel, respectively. ω_X and ω_S are the weights of spatial and spectral distances, respectively. For an image of N pixels segmented into K superpixels, assuming the square shape of a superpixel, the value of ω_X is set to be $(N/K)^{1/2}$. ω_S is the compactness factor to adjust the shape compactness and boundary adherence, which is user-defined.

First, from the initial centroids, the SNIC algorithm uses a priority queue to choose the next pixel to add to a cluster. The priority queue is filled with connected candidate pixels and pops up the candidate with the smallest distance. Then, an online updating of the corresponding centroid is performed according to each new pixel added to the superpixel. The online updating is executed effectively through a single iteration due to the local similarity in images. After executing the modified SNIC algorithm, the HSI is segmented into superpixels with relatively uniform size and regular shape, which are adopted as the basic units for subsequent unsupervised feature learning and classification result regularization.

Although the grid partition can also generate image patches for unsupervised CNN training, the grids do not adapt to the shape and distribution of ground objects, which influences the feature extraction and reduces the learning efficiency. Employing the SNIC superpixels to generate spectral and textural training patches is conductive to optimize the characteristic identification and enhance the unsupervised CNN convergence. On the other hand, there exists a salt and pepper effect in the pixel classification results, which is discrete and discontinuous in the complicated regions. It is necessary to adopt the superpixel regularization to improve the classification boundaries since SNIC superpixel segmentation has good adherence to the ground objects. Therefore, the advantages of SNIC superpixel segmentation in HSI processing are generating adaptive training patches and optimizing classification maps.

B. Unsupervised Spatial–Spectral Feature Learning

The labeled samples are generally limited for HSI classification, and meanwhile, a great number of unlabeled samples contain abundant information and potential image features. It is difficult for the supervised CNN methods to learn rich characteristics and correct relationship training with small sample sets due to insufficient feature expression and model optimization. In order to utilize the unlabeled samples, an effective unsupervised spatial–spectral CNN is synthetically built with the convolutional encoder and decoder, the additional clustering branch, and the multilayer feature map combination, as shown in Fig. 1. The encoder compresses the input HSI into a latent-space representation, and the decoder reconstructs the input HSI from the latent-space feature, making the input and output as close as possible in an unsupervised manner. For feature compression, the CNN encoder is designed with depthwise separable convolution [54], nonlinear activation, and downsampling layers with residual connections [55]. In detail, the depthwise separable convolution is integrated by individual depthwise and pointwise convolution to extract the feature maps faster with fewer parameters. For image reconstruction, the decoder is designed to be symmetrical with the encoder to implement feature decompression. The encoder and the decoder are represented as

$$\begin{aligned} F^k &= \mathrm{ReLU}\left(C_{\mathrm{spb}} \otimes F^{k-1} + b_{\mathrm{spb}}\right) \\ &= \mathrm{ReLU}\left(C_{\mathrm{pw}} \otimes \left(C_{\mathrm{dw}} \otimes F^{k-1} + b_{\mathrm{dw}}\right) + b_{\mathrm{pw}}\right) \end{aligned} \tag{2}$$

$$F_{\mathrm{en}}^{k+1} = \mathrm{Maxpool}\left(F_{\mathrm{en}}^{k}\right) \tag{3}$$

$$F_{\mathrm{de}}^{k+1} = \mathrm{Upsample}\left(F_{\mathrm{de}}^{k}\right) \tag{4}$$

where F^{k-1} and F^k denote the feature maps at last and next layers, respectively. The symbol $\otimes$ means the convolutional operation between kernel and feature map. C_{spb} is the separable convolution kernel integrated by C_{dw} and C_{pw}, namely, the depthwise kernel with channel-by-channel calculation and the pointwise kernel with point-by-point calculation, respectively, and b_{spb}, b_{dw}, and b_{pw} are the corresponding bias parameters. $\mathrm{ReLU}(x)$ is the nonlinear activation function employed in this study, which equals to x when x is positive and keeps 0 when x is negative. F_{en} and F_{de} denote the feature maps of the encoder and the decoder, respectively. Maxpool$(\cdot)$ and Upsample$(\cdot)$ are the maxpooling and upsampling functions to compress and amplify feature maps in the encoder and the decoder, respectively.

Considering to balance the learning ability and model simplicity, the encoder is composed of five blocks with 1, 2, 2, 2, and 1 convolutional layers, respectively. Following the second, third, and fourth blocks, there are maxpooling operations and residual connection, compressing the feature maps and raising the learning efficiency. Then, GAP is utilized to encode the feature maps of the fifth block to obtain a feature vector as the latent-space representation. The structure of the decoder is symmetrical with the encoder for unsupervised learning via image reconstruction and error feedback, and the number of convolution kernels at the last layer is set to the dimension of input HSI.

To further differentiate the features learned through unsupervised CNN, the K-means branch is designed to add to the network for latent-space feature clustering in addition to the decoder branch. The K-means algorithm is adopted due to the stable clustering performance, simple implementation, and effective execution during network training. In each epoch of training, the K-means branch inputs with the GAP encoded features and outputs the clustered assignments, by iteratively

calculating distance and adjusting centroids. Then, the clustered assignments are regarded as the pseudolabels of unlabeled samples for network prediction, error calculation, and label updating. The error propagates backward to adjust the gradients and parameters through network training. Besides, the number of clusters is set according to the land cover types in HSI, keeping the feature extraction consistent with the subsequent classification. With the structure of decoder and K-means output branches, the network is trained iteratively and interactively with the error feedback of image reconstruction and pseudolabel classification. Given a training set $X = \{x_1, x_2, \ldots, x_N\}$ of N superpixels, $F_\alpha(x_n)$ denotes the encoded feature of x_n with network parameter α, and z_n denotes the pseudolabel associated with x_n. The K-means branch produces a centroid matrix M in size of $d \times k$, where d and k are the dimension of x_n and the number of clusters, respectively. M_{z_n} is a centroid vector corresponding to the pseudolabel z_n in M. Then, the object functions of K-means clustering and pseudolabel classification are expressed as

$$H_{\text{kms}} = \min_{M} \frac{1}{N} \sum_{n=1}^{N} \min_{z_n} \| F_\alpha(x_n) - M_{z_n} \|^2 \tag{5}$$

$$H_{\text{psd}} = \min_{\alpha,\beta} \frac{1}{N} \sum_{n=1}^{N} L\big(S_\beta(F_\alpha(x_n)), z_n\big) \tag{6}$$

where $\| \cdot \|^2$ means the Euclidean distance calculation of encoded features and centroids within the same clustering. H_{kms} and H_{psd} are the clustering and classification object functions, respectively. S_β means the classifier with parameter β for the encoded features, and $L(\cdot)$ is the loss function using categorical cross-entropy. The unlabeled superpixels are trained in the minibatch using the Adam strategy for the adjustment of the learning rate.

In the unsupervised network, the encoder extracts HSI features layer by layer from concrete to abstract, and the high-level features contain more discriminative semantics than low-level features. However, for some objects that are recognized with significant concrete features, such as color and edges, the low-level features are more representative than high-level ones. Moreover, the feature maps at different levels are complementary to each other and describe objects from multiple perspectives, which are conductive to raise the clustering effect. Therefore, we combine the feature maps of the second, third, and fifth blocks to enhance the feature grouping and differentiating, which contain low-, middle-, and high-level information of the encoder, respectively. Considering that there exist expression redundancy and clustering difficulty in high-dimensional features, GAP is employed to compress the feature maps effectively, and then, multilayer GAP features are concatenated and input into the K-means branch. Furthermore, the importance and contribution of multilayer features are unequal for various samples due to the diversity of object attributes and spatial patterns. Hence, the adaptive concatenation is designed with adjustable and trainable weights to highlight the more significant features for each sample. The multilayer feature combination and adaptive concatenation are represented as

$$\begin{aligned} F_{\text{comb}} &= \theta_1 G_{b_2} + \theta_2 G_{b_3} + \theta_3 G_{b_5} \\ &= \theta_1 \text{GAP}\big(F_{b_2}\big) + \theta_2 \text{GAP}\big(F_{b_3}\big) + \theta_3 \text{GAP}\big(F_{b_5}\big) \end{aligned} \tag{7}$$

where GAP$(\cdot)$ means the operator of GAP and F_{comb} is the combined multilayer feature with adaptive weighting. F_{b_i} and G_{b_i} denote the output feature maps and GAP features of the ith block, respectively. θ_1, θ_2, and θ_3 are the adjustable weights for adaptive concatenation, which can be trained through the clustering branch. The K-means branch clusters samples using the combined multilayer vectors and propagates error backward to optimize the encoder parameters. After network training, the encoded features of the fifth block are extracted as the unsupervised deep features for subsequent classification using small sample sets.

C. Multiscale Unsupervised Feature Learning

Conventionally, CNN extracts image deep features layer by layer with fixed-size receptive fields and sliding convolution kernels. However, the ground objects in HSI show great heterogeneity in size, shape, and spatial relationships with obvious multiscale characteristics. In CNN, the fixed-size receptive field limits the observation context and is not useful to capture the scale-related information, thus reducing the performance. Therefore, it is necessary to integrate the multiple contextual information in the spatial domain and perform multiscale unsupervised deep feature learning. As shown in Fig. 1, small- and large-scale branches are designed to extract the unsupervised features of object attributes and spatial patterns in various contexts, and no more scale branches are adopted due to the feature redundancy and model complexity with more contexts. For the small-scale branch, the unsupervised spatial–spectral CNN mainly learns the inner hyperspectral and structural features of objects. For the large-scale branch, the network mainly extracts the external surrounding and associated relationships of objects. In this study, the size of contexts in two branches is set to 32×32 and 64×64, respectively, which is tested practically to be suitable for multiscale feature extraction. The 32×32 and 64×64 contexts are beneficial to extract the local attributes and distribution patterns of ground objects, respectively, which is complementary for spatial feature representation. The multiscale unsupervised deep feature learning and fusion can be represented as

$$F_{32}^i = \text{Encode}\big(I_{32}^i\big) \tag{8}$$

$$F_{64}^i = \text{Encode}\big(I_{64}^i\big) \tag{9}$$

$$F_{\text{ms}}^i = \big[F_{32}^i, F_{64}^i\big] \tag{10}$$

where I_{32}^i and I_{64}^i denote the patches of ith superpixel within 32×32 and 64×64 contexts, respectively, which are extracted according to the centroid of superpixels. Encode$(\cdot)$ means the feature transformation of the encoder, changing an HSI patch into a vector. The unsupervised learned features F_{32}^i and F_{64}^i are concatenated to obtain the multiscale feature F_{ms}^i for subsequent classification.

More precisely, for the superpixels inside class boundaries, the neighborhood information is relatively uniform and

consistent, and a large-scale context is required for feature learning to achieve better recognition. For the superpixels across class boundaries, the neighborhood is more complex and inconsistent, and a small-scale context is needed to avoid introducing disturbing noise. Therefore, multiscale feature learning obtains deep characteristics of superpixels in different contexts through unsupervised spatial–spectral CNN, utilizing the abundant information of unlabeled samples and improving the performance with a small sample set.

D. Diverse Unsupervised Feature Learning

In order to improve the comprehensiveness and diversity of unsupervised learning, deep features of hyperspectral information and geometric textures are designed to learn and fuse. Considering that direct stacking of multiple input costs less execution time but is not beneficial for targeted information extraction and parameter training, diverse features are learned separately from two branches for better discrimination and classification. For objects with typical hyperspectral characteristics, the spectral branch contributes more, and for objects with significant geometric patterns, the textural branch is more important. Contourlet texture [56] is a multidirection and multiscale transformation, which combines the Laplacian pyramid [57] and the directional filter bank (DFB) from the framing pyramid [58]. Subsequently, Cunha *et al.* [59] proposed an overcomplete contourlet method, called NSCT, which has the advantages of fast implementation, shift invariance, and multiscale and multidirection expansion. The NSCT is suitable for HSI processing to extract and highlight the geometric structure of ground objects, followed by unsupervised deep textural feature learning. It is more effective and appropriate than wavelet and Gabor transformations since the wavelet and Gabor perform insufficiently for multidirection and multiscale conditions, respectively. The NSCT can use fewer coefficients to capture more edge contours in the HSI, which is significant for textural description and object identification. Specifically, the core of NSCT is a nonseparable two-channel nonsubsampled filter bank (NSFB), and the NSCT can be divided into two shift-invariant parts, namely, the multiscale nonsubsampled pyramid (NSP) and the nonsubsampled directional filter bank (NSDFB). In NSP, the multiscale characteristics are gained using two-channel and 2-D NSFBs, and the nonsubsampled Laplacian decomposition can be expressed as

$$\begin{cases} x_H^{(1)} = x * \mathrm{PF}_H^{(D)} \in R^{n\times m} \\ x_L^{(1)} = x * \mathrm{PF}_L^{(D)} \in R^{n\times m} \end{cases} \tag{11}$$

where $x \in R^{n\times m}$ denotes the decomposed input signal. $\mathrm{PF}_H^{(D)}$ and $x_H^{(1)}$ represent the high-pass filter and the high-frequency part of decomposition at level 1, respectively, and $\mathrm{PF}_L^{(D)}$ and $x_L^{(1)}$ represent the low-pass filter and the low-frequency part, respectively.

In NSDFB, the DFB is constructed by combining sampled two-channel filter bank and resampling operation, and the result is a tree-structured filter bank. The NSDFB is built by eliminating the downsampler and the upsampler in DFB, and the filter is upsampled through turning off the downsampler/upsampler in each two-channel filter bank. The DFB is calculated by

$$\begin{cases} x_{H,1}^{(1)} = x_H^{(1)} * \mathrm{DF}_1 \in R^{n\times m} \\ x_{H,2}^{(1)} = x_H^{(1)} * \mathrm{DF}_2 \in R^{n\times m} \\ \vdots \\ x_{H,K}^{(1)} = x_H^{(1)} * \mathrm{DF}_K \in R^{n\times m} \end{cases} \tag{12}$$

where $\mathrm{DF}_k(k = 1, 2, \ldots, K)$ represents the DFB and K is an exponent of 2 in general. The high-frequency part $x_H^{(1)}$ is decomposed into multiple directional subbands ($x_{H,1}^{(1)}$, $x_{H,2}^{(1)}, \ldots, x_{H,K}^{(1)}$). The input signal is decomposed in subsequent stages via decomposing low-frequency components obtained by the previous again. NSCT decomposition consists of the multiscale decomposition of NSP and the tree decomposition of NSDFB. The NSDFB is a two-channel filter bank, and 2^K directional subbands are gained through the decomposition of K-element tree.

Contourlet transformation extracts the geometric information using multiscale and multidirection subbands to approximate the HSI, which is conductive to the detection of textures and edges. In this study, the NSCT decomposition of HSI is adopted as the geometric expression, in addition to the hyperspectral bands, to perform deep feature learning through unsupervised CNN for diverse representation. In order to highlight the main geometric structure and avoid noise interference, PCA is first employed to obtain the dimension-reduced data. The PCA algorithm sequentially finds a set of mutually orthogonal coordinate axes from the original data space to reduce HSI bands and minimize information loss. In this study, the first three principal components gained by PCA are adopted for the NSCT decomposition and unsupervised feature learning, which retains more than 98% of original HSI information.

Balancing the model complexity and texture redundancy of NSCT transformation for HSI expression, the decomposition of appropriate levels is performed on each principal component to obtain decomposed images, respectively. In this study, total 48 decomposed images of three principal components are stacked and input into the unsupervised CNN for deep geometric feature learning. Moreover, NSCT training patches are extracted based on the superpixels segmented via the SNIC algorithm using HSI bands. Similarly, there exist the multiscale characteristics of the geometric structure in NSCT decomposed images. Hence, the multiscale unsupervised feature learning is utilized for NSCT images, which can be represented as

$$F_{32\_n}^i = \mathrm{Encode}\left(I_{32\_n}^i\right) \tag{13}$$

$$F_{64\_n}^i = \mathrm{Encode}\left(I_{64\_n}^i\right) \tag{14}$$

$$F_{\mathrm{ms}\_n}^i = \left[F_{32\_n}^i, F_{64\_n}^i\right] \tag{15}$$

where $I_{32\_n}^i$ and $I_{64\_n}^i$ denote the NSCT patches of the ith superpixel within 32×32 and 64×64 contexts, respectively. Encode$(\cdot)$ means the feature transformation of the encoder, changing an NSCT patch into a vector. The unsupervised features $F_{32\_n}^i$ and $F_{64\_n}^i$ are concatenated to

obtain the multiscale NSCT deep feature $F^i_{\text{ms\_}n}$ for subsequent classification.

In order to integrate the advantages of hyperspectral and spatial information, the unsupervised deep features learned from HSI and NSCT patches, as well as raw spectral values, are assembled to achieve the multiscale and diverse features, which is expressed as

$$F^i_{\text{ms\_hn}} = \left[F^i_{\text{ms}}, F^i_{\text{ms\_}n}, F^i_{\text{hs}}\right] \tag{16}$$

where $F^i_{\text{ms\_hn}}$ is the fused feature to describe the ith superpixel from multiple perspectives of small- and large-scale contexts, as well as spectral and textural representation, and F^i_{hs} is raw spectral values. F^i_{ms} and $F^i_{\text{ms\_}n}$ mean the multiscale features of HSI and NSCT superpixels, respectively. The multiscale and diverse feature learning obtain abundant deep characteristics of superpixels through unsupervised spatial–spectral CNN, utilizing latent content of unlabeled samples to raise the performance.

E. Comprehensive Classification

In machine learning, RF is a widely employed classifier with stable performance for many applications. It contains multiple decision trees, and the output category is determined by the mode of individual tree outputs. For the construction of each tree, bootstrap sampling is utilized to take samples with replacement and form the training set, and partial input features are selected to determine the decision results of nodes. The RF classifier has suitable advantages for HSI classification, such as producing stable results, processing high-dimensional inputs, and maintaining performance with limited and unbalanced data. Therefore, RF is adopted as the comprehensive classifier to recognize land cover categories based on multiscale and diverse unsupervised feature learning.

In unsupervised CNN feature learning, the SNIC superpixels are employed to extract patches and produce the training dataset with meaningful contexts. In comprehensive classification, pixels are first used as the basic units to obtain initial classification results. Let φ_t denote the base learner of each decision tree, which is trained using the dataset D_t. The integrated classification can be expressed as

$$Z^j = \underset{y\in Y}{\operatorname{argmax}} \sum_{t=1}^{T} \prod \left(\varphi_t\left(F^j_{\text{ms\_hn}}\right) = y\right) \tag{17}$$

where $F^j_{\text{ms\_hn}}$ means the multiscale diverse feature of the jth pixel and Z^j is the corresponding RF predicted result. Y denotes the set of class labels, and y denotes the predicted label of each decision tree. T and t mean the complete set and the subset, respectively. For each decision tree, there is a separate subset of observations not used for training, called out-of-bag (OOB) observations, which can be employed as a testing set to evaluate the performance. The overall score of OOB observations is calculated to provide a single measure for RF performance as an alternative to cross-validation. The OOB error is the main basis to choose the optimal feature set for the RF classifier, which is calculated by

$$P^x_{\text{oob}} = \underset{y\in Y}{\operatorname{argmax}} \sum_{t=1}^{T} \prod \left(\varphi_t\left(F^x_{\text{ms\_hn}}\right) = y\right), \quad x \notin D_t \tag{18}$$

$$E^x_{\text{oob}} = \frac{1}{|D|} \sum_{(x,y)\in D} \prod \left(P^x_{\text{oob}} \neq y\right) \tag{19}$$

where $|D|$ represents the size of dataset D and x represents the samples not used for base learner training. $F^x_{\text{ms\_hn}}$, P^x_{oob}, and E^x_{oob} denote the multiscale and diverse feature, OOB prediction, and OOB error with samples not used for training, respectively.

Through the RF classifier with OOB estimation, multiscale and diverse features are fused and identified to obtain the initial land cover mapping, but there exists a salt and pepper effect in pixel classification results. Therefore, superpixel regularization is designed to optimize the pixel results, which assigns the internal pixels of each superpixel with the same labels by means of majority voting. In each superpixel, we count the classification labels of all internal pixels, then take the major label as the superpixel category, and update the internal pixels with other labels to the major label. Superpixel regularization is beneficial to eliminate the salt and pepper effect, reduce noise interference, and maintain the continuity of land cover distribution. Finally, the comprehensive classification results are evaluated qualitatively and quantitatively.

For the overall flow of UMsDFL, there are three main steps during training. The first is the SNIC segmentation and generation of HSI superpixels and NSCT superpixels. The second is the multiscale and diverse network training in parallel, namely, the small and large scales for HSI and the small and large scales for NSCT. For each unsupervised network, it is trained before with only the decoder branch and then fine-tuned later with both decoder and clustering branches. The third is the comprehensive classification through multiscale and diverse feature fusion and RF classifier training. For land cover prediction, contextual patches of each pixel should be first extracted and transformed by parallel CNNs to obtain multiscale and diverse feature vectors. Then, feature vectors are concatenated and input into the trained RF classifier to get pixel results. Finally, superpixel regularization is implemented on the pixel results to achieve the optimized classification maps.

III. Experimental Setup

A. Datasets

1) Houston Dataset: The dataset was acquired over the University of Houston campus and neighboring urban areas. The hyperspectral data were acquired by the ITRES Compact Airborne Spectrographic Imager 1500 (CASI-1500) sensor and provided by the 2013 IEEE Geoscience and Remote Sensing Society (GRSS) Data Fusion Competition. The data are in the size of 349 × 1905 pixels with 2.5-m spatial resolution, including 144 bands ranging from 380 to 1050 nm. The land cover is marked into 15 categories, and 15 029 labeled samples are given in the ground-truth image, as shown in Table I and Fig. 2.

TABLE I
LAND-COVER CLASS IN THE HOUSTON DATASET

Class	Land Cover Type	No. of Samples
C1	Healthy grass	1251
C2	Stressed grass	1254
C3	Synthetic grass	697
C4	Tree	1244
C5	Soil	1242
C6	Water	325
C7	Residential	1268
C8	Commercial	1244
C9	Road	1252
C10	Highway	1227
C11	Railway	1235
C12	Parking lot 1	1233
C13	Parking lot 2	469
C14	Tennis court	428
C15	Running track	660
	Total	15029

TABLE II
LAND-COVER CLASSES IN THE PAVIA DATASET

Class	Land Cover Type	No. of Samples
C1	Water	65971
C2	Trees	7598
C3	Asphalt	3090
C4	Self-blocking bricks	2685
C5	Bitumen	6584
C6	Tiles	9248
C7	Shadows	7287
C8	Meadows	42826
C9	Bare soil	2863
	Total	148152

TABLE III
LAND-COVER CLASSES IN THE DIONI DATASET

Class	Land Cover Type	No. of Samples
C1	Dense urban fabric	1262
C2	Mineral extraction sites	204
C3	Non-irrigated arable land	614
C4	Fruit trees	150
C5	Olive groves	1768
C6	Coniferous forest	361
C7	Dense sclerophyllous vegetation	5035
C8	Sparce sclerophyllous vegetation	6374
C9	Sparcely vegetated areas	1754
C10	Rocks and sand	492
C11	Water	1612
C12	Coastal water	398
	Total	20024

2) Pavia Center Dataset: The dataset was acquired by the Reflective Optics Spectrographic Imaging System 03 (ROSIS-03) sensor over the center of Pavia, Italy, with 115 spectral bands. It is an image of 1096 × 1096 pixels, but some part contains no information and needs to be discarded. After processing the image and removing the noisy bands, HSI data are in the size of 1096 × 715 pixels with 1.3-m spatial resolution, including 102 bands. There are 148 152 labeled samples and nine classes of land cover categories in the ground-truth image, as shown in Table II and Fig. 3.

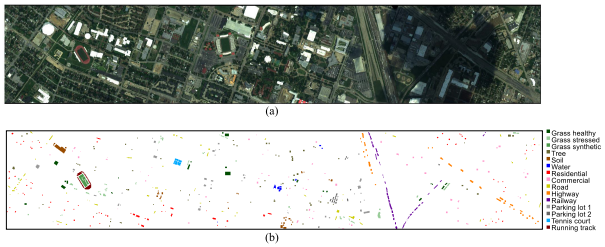

Fig. 2. (a) False color image and (b) ground-truth map of the Houston dataset (15 land-cover classes).

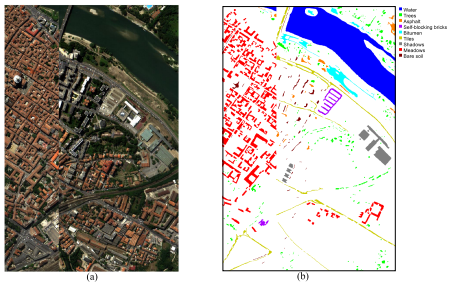

Fig. 3. (a) False color image and (b) ground-truth map of the Pavia dataset (nine land-cover classes).

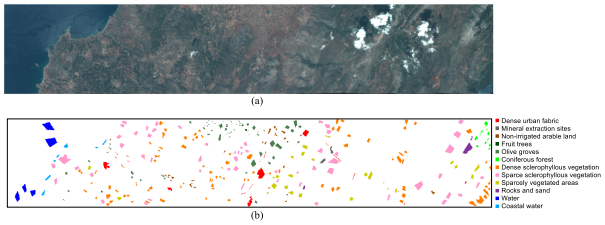

Fig. 4. (a) False color image and (b) ground-truth map of the Dioni dataset (12 land-cover classes).

3) Dioni Dataset: The HyRANK dataset was acquired by the Hyperion sensor on the Earth Observing-1 satellite, containing five HSIs, namely two training images (i.e., Dioni and Loukia) and three validating images (i.e., Erato, Kirki, and Nefeli). Among them, the Dioni is selected for experiments since it has a sample size of greater than 100 in each category. The data size is 250 × 1376 pixels with 30-m spatial resolution, including 176 bands. A total of 20 024 pixels are labeled in 12 classes in the ground-truth image, as shown in Table III and Fig. 4.

B. Evaluation Indices

In this study, we focus on improving the training and classification performance using small sample sets. Hence, a small number of labeled samples are selected from each land cover category for model training, and the remaining are used as a testing set. The numbers of training samples per class are set from 5 to 18 to demonstrate how the classification accuracy changes with increasing samples. Afterward, in order to evaluate the performance of different methods from multiple perspectives, the indicators of overall accuracy (OA), accuracy for each class (CA), and Kappa coefficient (Kappa)

are calculated and analyzed. The confusion matrix is a cross-tabulation of ground truth and predicted labels, organizing samples in a way that summarizes the classification results and quantifies the accuracy. The diagonal elements of the confusion matrix highlight the correct identification, and the nondiagonal elements show the missing and incorrect identification. Let Q_{ij} denote the pixel with ground truth i and classified as type j, and let $N = \sum_i \sum_j Q_{ij}$ denote the total number of all pixels in HSI. OA indicates the overall classification accuracy of all pixels, which is represented as

$$\mathrm{OA} = \frac{\sum_i Q_{ii}}{N} \tag{20}$$

where the range of i and j is $(1, 2, \ldots, K)$, and K is the number of land cover types. CA means the accuracy of each category related to classification maps and refers to the probability that predicted types are equal to true labels under the assumption of classification conditions. The CA of class i is calculated by

$$\mathrm{CA}_i = \frac{Q_{ii}}{\sum_j Q_{ij}} \tag{21}$$

where CA is also known as the user's CA. Kappa is employed for the consistency evaluation and calculated based on the confusion matrix. OA and CA reflect the proportion of correct classification but are not suitable for evaluating the unbalanced samples, whereas Kappa can describe the unbalanced confusion matrix. The Kappa is calculated by

$$\mathrm{Kappa} = \frac{\mathrm{OA} - Q_c}{1 - Q_c} \tag{22}$$

$$Q_c = \frac{\sum_k \left(\sum_j Q_{kj} \cdot \sum_i Q_{ik} \right)}{N \cdot N} \tag{23}$$

where the range of i, j, and k is $(1, 2, \ldots, K)$, and K is the number of categories.

C. Parameter Settings

In our proposed UMsDFL method, most of the parameters are set by default. Concretely, PCA operation reduces the high dimensions of HSI to three dimensions, which contains over 98% information of the original data. Considering to balance the model complexity and feature redundancy of decomposition, five-level NSCT is performed on each principal component, and 16 decomposed images are obtained, respectively. Each principal component is decomposed into the one-level image, two-level image, and three-level images in two directions, four-level images in four directions, and five-level images in eight directions. Total 48 decomposed images of three principal components are stacked and input into the unsupervised spatial–spectral CNN for deep geometric feature learning.

The numbers of convolution kernels at each layer of the unsupervised spatial–spectral CNN are significant parameters, which influences the ability of deep feature extraction. More convolution kernels can learn more groups of features, whereas more kernels will increase the training parameters, reduce the execution efficiency, and raise the overfitting risk. Therefore,

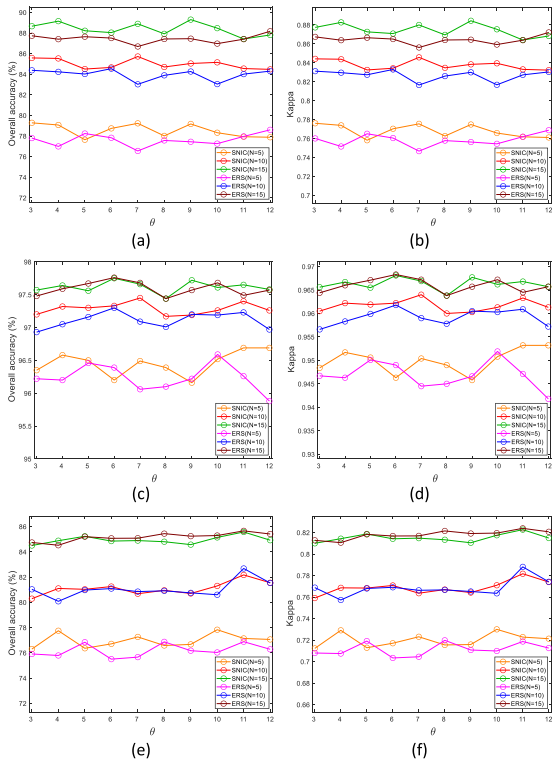

Fig. 5. Performance versus the value of θ with (a) OA and (b) Kappa on Houston, (c) OA and (d) Kappa on Pavia, and (e) OA and (f) Kappa on Dioni using five (N = 5), ten (N = 10), and 15 (N = 15) training samples per class, respectively.

considering to balance the learning ability and model simplicity, the numbers of convolution kernels are set to 64, 64, 96, 128, and 128, respectively, for five blocks in the encoder. The structure of the decoder is symmetrical with the encoder, and the numbers of convolution kernels are set to 128, 96, 64, input dimension, and input dimension, respectively, for five blocks. The sizes of the convolution kernel and pooling window are set to 3×3 and 2×2, respectively. The unsupervised network is first trained through 300 epochs with only the decoder branch to obtain a stable initialization and second trained through 300 epochs with both decoder and clustering branches to enhance the feature differentiation in a batch size of 128. The NAdam (i.e., Adam with the Nesterov accumulation) optimizer and the Adam optimizer with default parameters are employed to adjust the learning rate during the first and second training processes, respectively. The mean square error is used as the evaluation index during the first training, and the mean square error and the categorical cross-entropy are used as the indexes for decoder and clustering branches during the second training, respectively. For the comprehensive classification based on multiscale and diverse feature learning, the number of decision trees in the RF classifier is set to 300 to simplify the model complexity and maintain the classification performance.

In order to balance the spectral similarity and spatial distance in SNIC segmentation, the compactness parameter related to weights ω_X and ω_S is set to 10, maintaining the performance of feature learning and classification regularization. The size of superpixels in the segmentation map is an

important parameter, which influences the learning efficiency and effect of unsupervised CNN. For feature learning and classification as in [60], large- and small-scale superpixels are merged through superpixel-level majority voting to retain accurate boundary information and spatial context information. For spatial postprocessing, superpixels can be utilized to optimize the class boundaries and correct the misclassified pixels, such as the conditional random field (CRF) method [61]. In this study, multiscale spatial context information is extracted via the unsupervised CNN feature learning in the structure of parallel network branches, centered on the segmented superpixels. Superpixels in an appropriate size are needed to maintain the accurate boundaries of ground objects and adapt to multiscale contextual feature learning. Since the image size, spatial resolution, and object distribution of each HSI are not consistent, it is not meaningful to directly set the same number of superpixels for all datasets. Generally, fewer pixels are needed to represent a ground object in the HSI with low spatial resolution, so each superpixel should contain fewer pixels to ensure cohesion and similarity. In contrast, more pixels are needed to represent a ground object in the HSI with high spatial resolution, so each superpixel should contain more pixels for the complete description. Therefore, a heuristic formula for adaptive calculation of superpixel size based on the HSI spatial resolution is proposed, which is expressed as

$$\mathrm{Ns} = \frac{100}{\mathrm{Res}^{1/\theta}} \tag{24}$$

where Ns denotes the number of pixels inside each superpixel in initial and Res means the spatial resolution of HSI (i.e., meters per pixel). The parameter θ controls the relationship between spatial resolution and superpixel size, and the sensitivity analysis of θ is performed and shown in Fig. 5. The number of pixels inside each superpixel is set to the value less than 100 with Res greater than 1 and is set to the value greater than 100 with Res less than 1. Among them, if θ is smaller, the superpixel size changes more drastically with the spatial resolution, and vice versa. In addition, the classification performance of different superpixel segmentation methods is also explored using SNIC [52] and entropy rate superpixel (ERS) [62] algorithms.

The classification performance of the UMsDFL method using different superpixel algorithms with various θ values is displayed in Fig. 5. The experiments are executed 20 times with randomly selected training samples, and the mean OA and Kappa are shown. The results illustrate that the SNIC algorithm has similar performance to the ERS on Pavia and Dioni datasets, and has a little better classification accuracy than the ERS on the Houston dataset. Therefore, the SNIC algorithm is adopted in this study to segment the HSI into superpixels for multiscale and diverse unsupervised feature learning. For the sensitivity analysis of θ, the overall OA and Kappa with different parameter values do not change much. On the one hand, the θ should not be too small, as it will cause the superpixel size to change drastically with spatial resolution, and θ should not be too large, as it will reduce the superpixel cohesion in the HSI with low spatial resolution. On the other hand, the θ value of 7 with the SNIC algorithm shows relatively better classification performance on Houston and Pavia datasets, and has stable performance on the Dioni dataset. Hence, θ is set to 7 in this study to decide the superpixel size during segmentation for a more reasonable representation of ground objects.

D. Ablation Study

In order to demonstrate the module effectiveness of the UMsDFL method, an ablation study is performed on the three datasets. Ablation analysis is carried out between single-scale and multiscale methods, and between single-type and diverse methods. USsFL_H and Unsupervised Large-scale Feature Learning (ULsFL)_H methods apply the RF classifier to small- and large-scale unsupervised HSI feature learnings, respectively, to demonstrate the ability of single-scale HSI features. USsFL_N and ULsFL_N methods apply the RF classifier to small- and large-scale unsupervised NSCT feature learnings, respectively, to demonstrate the ability of single-scale NSCT features. Moreover, UMsFL_H and UMsFL_N methods apply the RF classifier to the multiscale unsupervised feature learning of HSI and NSCT data, respectively, to demonstrate the ability of multiscale features. UMsFL_HSI unsupervised features and Spectral values (HS) and UMsFL_NSCT unsupervised features and Spectral values (NS) methods fuse the hyperspectral values based on UMsFL_H and UMsFL_N methods, respectively, to demonstrate the ability of multiscale features with spectral information. All results of these methods are optimized with superpixel regularization after the RF classification. To reduce the effect of random factors, training samples are selected randomly 20 times, and the experiments are carried out correspondingly. The mean OA and Kappa are shown in Fig. 6 for ablation analysis.

It is illustrated that the classification performance of OA and Kappa rises as the number of labeled samples increases due to more information and better fitting. In comparison between single-scale methods, the HSI-based method has similar accuracy to the NSCT-based, which both express the main information of HSI at a single scale. The difference is that small-scale feature learning is a little more discriminative than large scale, considering that small-scale learning pays more attention to local attributes of ground objects and avoids noise interference. Furthermore, combining the small-scale details and large-scale relationships, the multiscale feature learning (UMsFL_H and UMsFL_N) has a more distinctive capability for better classification. Multiscale feature learning is beneficial to represent the various ground objects in terms of spatial distribution and contextual patterns. On the other hand, based on the fusion of hyperspectral values and multiscale features, the OA and Kappa of UMsFL_HS and UMsFL_NS are improved and become more stable with different numbers of labeled samples. In order to integrate the spatial–spectral information and textural structure, the UMsDFL method fuses multiscale diverse features through comprehensive classification and achieves the best performance. Diverse feature learning is conductive to the deep mining and complementary description of object attributes from multiple perspectives. Therefore, our proposed method is superior to the single-scale or single-type feature learning methods.

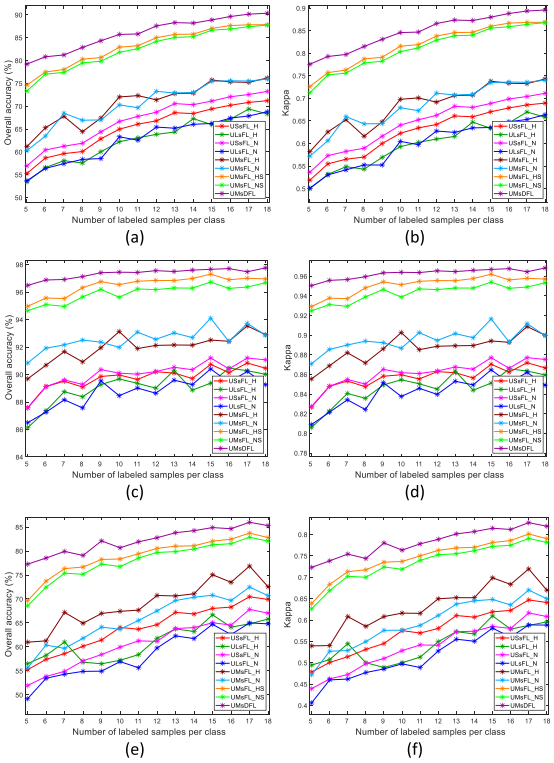

Fig. 6. Performance versus the number of labeled samples per class with (a) OA and (b) Kappa on Houston, (c) OA and (d) Kappa on Pavia, and (e) OA and (f) Kappa on Dioni using different ablation methods.

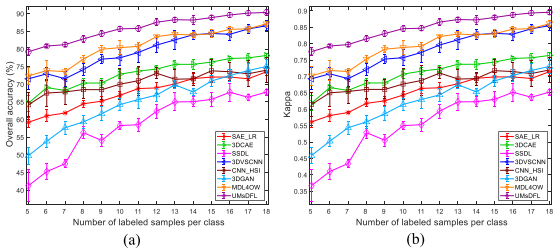

Fig. 7. Houston dataset. (a) OA and (b) Kappa as functions of the number of labeled samples per class.

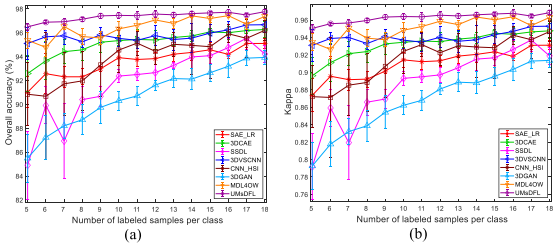

Fig. 8. Pavia dataset. (a) OA and (b) Kappa as functions of the number of labeled samples per class.

IV. Experimental Results

A. Comparison Methods

To verify the effectiveness of the UMsDFL method for HSI land cover classification, a series of experimental tests are carried out. Our proposed approach is compared with the state-of-the-art methods in the following. SAE_Logistic Regression (LR) [26] is the first DL attempt to employ the autoencoder for HSI classification using a greedy layerwise strategy to pretrain each layer and then fine-tune the classifier. 3-DCAE [37] is an unsupervised spatial–spectral feature learning method proposed for HSI classification based on the 3-D convolutional autoencoder. After pretraining, the SVM is utilized to classify the hidden features on the top. Low Spatial Sampling Distance (SSDL) [39] is a framework to merge spatial and spectral features via SAE and deep CNN, respectively, followed by the spatial pyramid pooling and LR classifier. The features of 7×7 neighbor regions are learned through the autoencoder pretraining with 80% of data. 3-DVSCNN [38] is a method that provides the valuable sample set and employs the CNN to extract deep features for HSI classification. An active learning strategy is adopted to construct the valuable training set by iteratively selecting the most uncertain samples through SVM, and 80% of samples are picked to form the set. CNN_HSI [36] is a model that combines the multilayer 2-D convolutions and local response normalization for HSI classification using dropout strategy and data augmentation. 3-DGAN [32] is a method built upon CNN and GAN structure, containing a generative network and a discriminative network in competition. Two CNNs are designed to generate the so-called fake inputs and discriminate the inputs, respectively. MDL4OW [63] is a multitask DL framework that simultaneously conducts the classification and reconstruction with probable unknown classes in HSI. Two strategies for few- and many-shot scenarios with the extreme value theory are proposed to improve the performance.

To reduce the effect of random factors, training samples are selected randomly 20 times on each dataset, and classification tests are carried out correspondingly with various numbers of labeled samples. Both the means and standard deviations of OA, CA, and Kappa indicators are calculated in the experimental analysis.

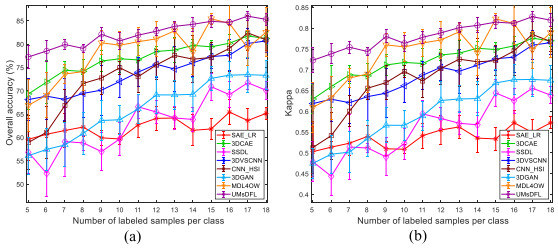

Fig. 9. Dioni dataset. (a) OA and (b) Kappa as functions of the number of labeled samples per class.

B. Classification Results

To show the effectiveness of the proposed method, we quantitatively and qualitatively evaluate the classification performance by comparing UMsDFL with the aforementioned baseline methods. Figs. 7–9 show the OA and Kappa of 8 methods (i.e., SAE_LR, 3-DCAE, SSDL, 3-DVSCNN, CNN_HSI, 3-DGAN, MDL4OW, and our UMsDFL) when varying the number of training samples per class from 5 to 18. In general, as the number of samples increases, the

TABLE IV

CLASSIFICATION ACCURACY (%) AND KAPPA MEASURE USING SAE_LR, 3-DCAE, SSDL, 3-DVSCNN, CNN_HSI, 3-DGAN, MDL4OW, AND UMSDFL FOR THE HOUSTON DATASET WITH FIVE LABELED SAMPLES PER CLASS AS TRAINING SET

Class	SAE_LR	3DCAE	SSDL	3DVSCNN	CNN_HSI	3DGAN	MDL4OW	UMsDFL
C1	86.75±8.47	89.41±6.01	70.95±8.88	90.31±6.61	88.60±6.86	64.08±14.30	84.28±13.06	**90.95±6.68**
C2	58.17±7.98	78.99±9.32	47.00±26.92	74.83±14.65	81.15±11.02	28.26±10.18	53.31±24.05	**83.03±11.48**
C3	96.53±3.52	93.74±4.42	68.29±20.12	88.34±5.95	94.84±6.97	73.29±11.28	89.25±8.12	**100.00±0.00**
C4	87.17±7.63	92.46±1.45	48.70±17.16	89.08±6.73	91.79±3.16	31.58±12.20	66.40±33.71	**95.31±2.52**
C5	79.23±14.31	85.39±6.70	54.61±19.57	**98.22±2.47**	93.09±7.40	69.40±13.62	90.26±11.43	94.98±3.56
C6	73.29±4.07	76.86±9.64	50.71±27.14	84.00±7.30	81.91±7.62	70.74±10.68	76.34±8.15	**91.69±4.53**
C7	42.00±9.43	48.21±9.98	37.59±18.46	55.55±9.74	47.69±8.74	37.40±14.01	44.86±13.95	**86.53±3.02**
C8	26.59±12.71	30.70±10.06	14.05±17.51	**45.72±7.38**	31.56±12.50	36.42±10.00	43.25±8.78	39.27±7.27
C9	57.22±14.71	61.14±15.52	29.09±15.21	61.28±9.62	51.02±22.30	21.37±8.88	48.41±23.66	**87.97±2.59**
C10	33.01±9.47	36.72±11.84	30.73±10.80	61.30±11.57	26.63±19.44	**71.96±16.36**	60.73±16.68	51.20±6.02
C11	52.79±18.95	47.81±18.46	27.58±12.95	51.94±12.93	39.64±27.33	54.57±15.58	59.06±14.98	**64.12±11.42**
C12	24.46±12.97	31.11±10.03	32.52±22.62	58.30±16.63	40.61±18.13	47.29±10.27	49.12±10.23	**59.70±8.49**
C13	33.82±8.09	58.53±9.31	21.54±8.77	74.61±7.87	56.03±26.56	40.17±13.14	**75.67±26.16**	73.45±4.83
C14	97.10±1.81	92.71±3.83	78.64±5.52	91.10±12.12	99.16±1.16	90.33±10.82	93.55±7.97	**100.00±0.00**
C15	90.24±6.37	97.28±2.31	35.03±20.34	87.55±12.40	99.73±0.19	63.80±5.26	83.85±19.67	**100.00±0.00**
OA(%)	59.23±1.41	64.67±2.24	41.29±4.50	71.56±3.06	64.21±3.40	49.79±2.37	72.39±2.48	**79.23±1.20**
Kappa	0.561±0.016	0.619±0.024	0.367±0.048	0.693±0.033	0.615±0.037	0.459±0.025	0.702±0.030	**0.776±0.013**

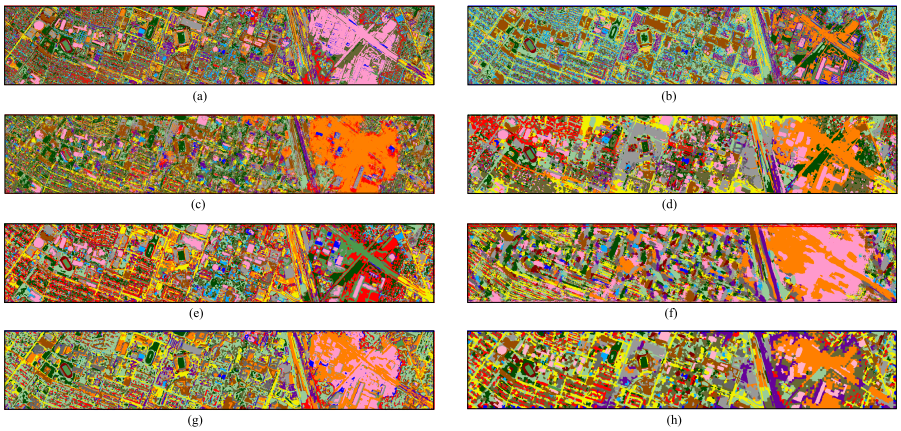

Fig. 10. Classification maps of the Houston dataset obtained by (a) SAE_LR, (b) 3-DCAE, (c) SSDL, (d) 3-DVSCNN, (e) CNN_HSI, (f) 3-DGAN, (g) MDL4OW, and (h) UMsDFL when the number of training samples is five per class (the percentage in the brackets is the corresponding accuracy).

TABLE V

CLASSIFICATION ACCURACY (%) AND KAPPA MEASURE USING SAE_LR, 3-DCAE, SSDL, 3-DVSCNN, CNN_HSI, 3-DGAN, MDL4OW, AND UMSDFL FOR THE PAVIA DATASET WITH FIVE LABELED SAMPLES PER CLASS AS TRAINING SET

Class	SAE_LR	3DCAE	SSDL	3DVSCNN	CNN_HSI	3DGAN	MDL4OW	UMsDFL
C1	98.51±1.07	99.17±0.43	98.08±1.30	99.84±0.45	96.28±1.97	99.39±0.37	99.26±0.39	**99.85±0.18**
C2	80.75±11.66	84.47±4.55	80.85±7.38	84.83±11.12	77.69±16.51	41.95±21.34	77.54±14.81	**87.77±5.00**
C3	85.12±11.11	**94.59±3.79**	76.89±9.33	90.03±5.90	81.29±18.89	66.25±14.71	75.99±22.10	83.59±5.84
C4	58.00±35.87	76.30±13.77	67.88±15.77	88.17±10.00	87.94±14.97	84.88±13.23	88.82±12.86	**98.77±1.02**
C5	64.11±34.54	72.79±10.09	55.95±13.53	83.37±7.10	66.95±10.26	62.84±9.12	71.65±11.44	**97.09±2.01**
C6	82.28±21.59	92.18±7.02	74.60±12.85	87.66±12.17	92.38±8.46	22.86±6.65	67.74±36.75	**95.18±4.31**
C7	79.82±8.98	80.35±3.79	76.50±8.14	**84.39±5.69**	72.02±8.91	75.01±12.16	82.78±6.26	83.66±4.19
C8	91.42±3.87	89.56±5.39	74.90±8.69	96.03±3.04	91.73±9.00	94.46±3.77	**97.70±1.20**	96.60±0.63
C9	95.46±5.64	98.31±1.87	87.61±9.39	89.63±8.52	98.36±1.80	49.33±13.51	77.02±28.00	**98.54±1.68**
OA(%)	91.02±1.32	92.58±1.54	84.91±2.86	95.10±1.34	90.85±2.64	85.50±2.01	95.37±0.68	**96.49±0.32**
Kappa	0.874±0.019	0.896±0.021	0.792±0.038	0.931±0.019	0.872±0.035	0.793±0.027	0.935±0.010	**0.950±0.004**

classification accuracy gradually improves due to more sufficient samples and richer features. First, the SAE_LR, SSDL, and 3-DGAN methods produce relatively worse classification results with lower accuracy and consistency. The SAE_LR and SSDL methods learn features via CNN and SAE, respectively, whereas the performance is limited by the network structure and training skills to be improved. The 3-DGAN method generates fake inputs and discriminates the images, but the parameters of generation and discrimination networks are hard to learn well using limited samples. Second, the CNN_HSI and 3-DCAE methods output classification results with moderate accuracy. The CNN_HSI method gets poor performance with fewer samples, and the accuracy improves significantly as the number of samples increases, reflecting

TABLE VI

CLASSIFICATION ACCURACY (%) AND KAPPA MEASURE USING SAE_LR, 3-DCAE, SSDL, 3-DVSCNN, CNN_HSI, 3-DGAN, MDL4OW, AND UMSDFL FOR THE DIONI DATASET WITH FIVE LABELED SAMPLES PER CLASS AS TRAINING SET

Class	SAE_LR	3DCAE	SSDL	3DVSCNN	CNN_HSI	3DGAN	MDL4OW	UMsDFL
C1	12.69±25.83	40.03±9.95	27.59±24.38	61.51±16.91	53.13±12.87	47.36±14.04	58.00±16.65	**69.24±15.95**
C2	20.69±25.41	73.11±11.86	49.22±19.21	83.63±16.67	86.03±9.80	82.70±18.51	81.91±18.26	**90.60±9.32**
C3	64.43±41.51	64.97±12.11	30.62±11.34	63.86±12.52	63.31±11.50	35.47±12.47	52.44±25.78	**68.78±4.28**
C4	22.40±29.03	87.81±8.23	52.67±24.68	75.40±11.47	77.27±13.59	47.07±17.13	68.60±22.63	**89.11±11.12**
C5	15.14±23.26	59.92±10.23	16.73±8.28	56.99±15.27	42.58±8.14	44.85±13.47	**68.96±15.18**	63.09±8.50
C6	8.70±13.82	97.88±1.37	71.86±12.69	**98.95±1.42**	95.96±4.90	73.68±11.05	89.25±14.79	97.23±7.03
C7	79.34±8.74	83.54±9.69	56.63±26.07	**86.17±10.54**	62.09±13.73	51.33±17.54	67.02±16.80	74.72±7.62
C8	75.65±12.51	66.06±12.16	71.48±15.07	56.65±14.13	54.37±15.80	52.44±18.47	55.02±11.58	**81.13±5.84**
C9	17.10±24.05	48.92±13.68	29.79±33.64	49.10±13.83	45.84±11.47	42.29±15.40	44.66±12.13	**52.99±13.53**
C10	51.26±49.30	91.52±4.96	69.59±24.40	92.85±4.38	90.47±6.71	92.20±4.73	93.46±4.53	**96.85±4.12**
C11	**100.00±0.00**	81.25±15.79	99.74±0.58	75.69±15.68	85.46±21.59	99.99±0.02	90.38±9.28	**100.00±0.00**
C12	6.53±14.61	62.24±17.85	57.74±9.14	84.50±14.54	32.74±33.49	96.81±9.49	84.62±12.34	**100.00±0.00**
OA(%)	59.62±4.48	69.32±3.27	56.83±4.55	68.12±4.48	58.92±5.88	56.10±3.97	66.90±5.57	**77.26±2.48**
Kappa	0.503±0.060	0.630±0.036	0.479±0.048	0.619±0.047	0.513±0.062	0.475±0.037	0.610±0.061	**0.723±0.030**

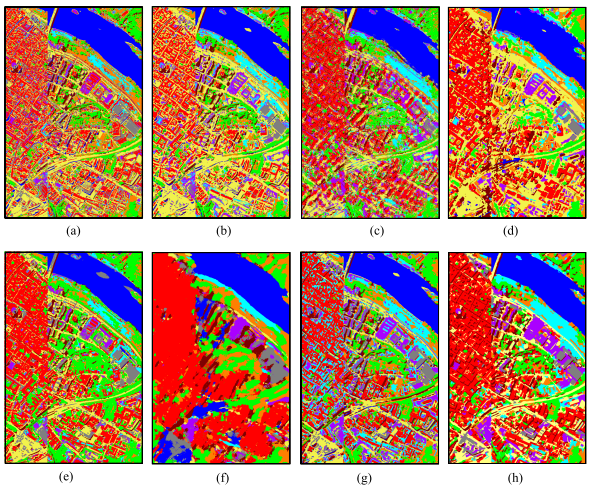

Fig. 11. Classification maps of the Pavia dataset obtained by (a) SAE_LR, (b) 3-DCAE, (c) SSDL, (d) 3-DVSCNN, (e) CNN_HSI, (f) 3-DGAN, (g) MDL4OW, and (h) UMsDFL when the number of training samples is five per class (the percentage in the brackets is the corresponding accuracy).

the feature learning patterns of supervised CNN. The 3-DCAE method employs a 3-D convolutional autoencoder to enhance the feature expression and recognize the objects better than SAE_LR, but the lack of discriminative constraints reduces the accuracy. Third, the 3-DVSCNN and MDL4OW methods obtain relatively stable precision for land cover identification. The 3-DVSCNN method is trained with more valuable samples through active learning, whereas the CNN has disadvantages of fixed-size receptive fields and single feature type. The MDL4OW method estimates the unknown score with all data using the statistical model in a multitask framework, but the singular feature learning of CNN limits the performance.

Finally, as expected, the UMsDFL method achieves the best performance in most cases, especially using a small sample set. This is reasonable since the complementary information of multiple scales and diverse features is helpful for the recognition of object attributes and spatial relationships. The unsupervised spatial–spectral CNN with clustering and multilayer combination is conductive to extracting the discriminative deep features from unlabeled training patches. Moreover, the multiscale contextual information is beneficial to identify land cover in different sizes and conditions, and the consideration of diverse features is useful to distinguish complicated and confusing objects. For Houston and Dioni datasets, the classification gap among various methods is more obvious than that for Pavia, and the overall precision of the Pavia dataset is relatively higher considering there are fewer categories and clearer discrimination.

C. Performance Under Limited Samples

The quantitative results acquired by various methods using five samples per class are presented in Tables IV–VI, and the highest record in each row is highlighted in bold. The classification performance per class is evaluated by the CA indicator, and there are obvious differences among the eight methods. Using a small sample set for training, it is shown that our proposed UMsDFL method has the greatest capability for feature learning and land cover classification, achieving the best OA and Kappa with the highest CA for most classes. For the Houston dataset, the C3 (synthetic grass), C14 (tennis court), and C15 (running track) classes are easy to distinguish, and most methods have good recognition results. The UMsDFL method correctly identifies these three classes with the CA of 100%, whereas the SSDL method shows especially inferior performance on C15. The C12 (parking lot 1) and C13 (parking lot 2) classes are confusing land cover types with similar characteristics, and the UMsDFL has the best CA for C12 and stable CA for C13, respectively. Moreover, the UMsDFL method also achieves the highest CA for C9 (road) class, which is linearly distributed with other objects and hard to classify. In contrast, the CA of C8 (commercial) and C13 in SSDL, C12 in SAE_LR, C10 (highway) in CNN_HSI, and C9 in 3-DGAN are relatively worse.

For the Pavia dataset, the classification accuracy and consistency are generally better than those of the Houston and Dioni datasets, containing more discriminative ground objects. The C1 (water) class distributes continuously and is easy to identify with distinct features, and UMsDFL and 3-DVSCNN methods have the best and second best CAs for water, respectively. On the contrary, the C2 (trees) class distributes discretely

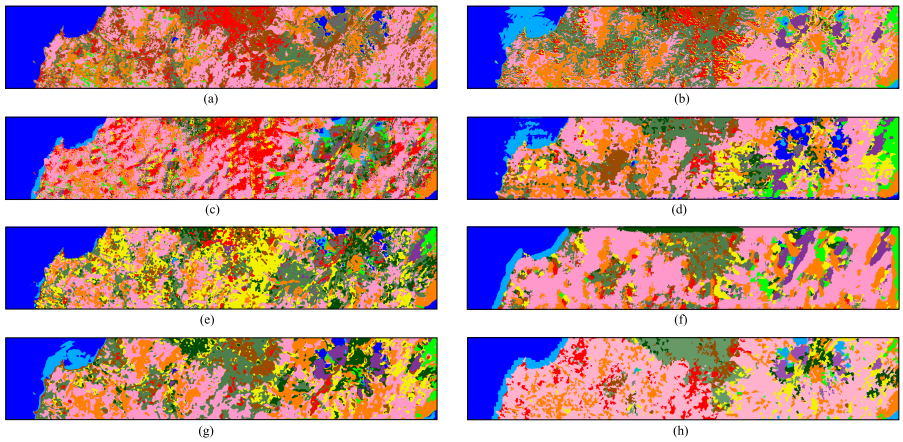

Fig. 12. Classification maps of the Dioni dataset obtained by (a) SAE_LR, (b) 3-DCAE, (c) SSDL, (d) 3-DVSCNN, (e) CNN_HSI, (f) 3-DGAN, (g) MDL4OW, and (h) UMsDFL when the number of training samples is five per class (the percentage in the brackets is the corresponding accuracy).

along with other objects that tend to be mixed, and UMsDFL and 3-DGAN methods obtain the best and worst results, respectively. Furthermore, the C4 (self-blocking bricks), C5 (bitumen), and C6 (tiles) classes are artificial ground objects with similar attributes and appearance, and UMsDFL method has superior ability to distinguish them. The CA of C2, C6, and C9 (bare soil) in 3-DGAN, C5 in SSDL, and C4 in SAE_LR are relatively worse.

For the Dioni dataset, there exists the contiguous water (C11) with coastal water (C12) along the edge, and UMsDFL and 3-DGAN methods obtain the best and second best results for them, respectively, whereas the SAE_LR method has extremely high and low accuracy for C11 and C12, respectively. The C7 (dense sclerophyllous vegetation) and C8 (sparse sclerophyllous vegetation) classes are similar types of vegetation with different densities, occupying a relatively large proportion of labeled samples, and the UMsDFL method achieves the best CA for C8 and stable results for C7, respectively. In addition, the C1 (dense urban fabric) class is the man-made surface that distributes discretely, and the UMsDFL method identifies C1 better than other comparison methods with diverse feature representation. Comparatively, the SAE_LR method shows great unbalance in CA with poor precision for C1, C5 (olive groves), C6 (coniferous forest), and C12 classes.

D. Visual Comparison

The visual and qualitative comparison among the eight methods using five samples per class is made in form of classification maps, as shown in Figs. 10–12. In general, there exists a salt and pepper effect in the result maps of SSDL and SAE_LR, which uses pixels or discrete units for feature extraction and classification. Although the 3-DGAN method does not perform badly with the salt and pepper effect, the classification boundaries do not match ground objects and present the distribution of bars. The 3-DCAE and CNN_HSI methods produce results with moderate appearance and consistency, and the CNN relatively maintains the continuity of pixel predictions within receptive fields and has limited identification accuracy. The 3-DVSCNN and MDL4OW methods obtain basically clear classification maps, but the boundaries of confusing objects are worse in complicated conditions. As expected, the classification regions of the UMsDFL method are more coherent and complete with continuous boundaries, which adopts the multiscale and diverse features to describe spatial relationships and geometric textures. Most objects are classified more correctly in UMsDFL maps than the others due to the strategies of effective unsupervised spatial–spectral feature learning and comprehensive classification.

For the Houston dataset, the road, highway, and railway are similar objects with linear distribution characteristics, and most comparison methods are prone to confuse them. The UMsDFL method obtains better connectivity for road recognition, and the main highway is identified badly by SAE_LR and CNN_HSI methods. The 3-DGAN method losses a lot of classification details in complicated regions although it keeps the relatively linear shape of railways and highways. For the Pavia dataset, the precision of the water class is high for most methods, but there exist discontinuous regions in 3-DCAE, 3-DVSCNN, CNN_HSI, and MDL4OW methods, considering that the accumulation of sediment in the river causes the misclassification. As one of the main land cover types, the UMsDFL method recognizes meadows more completely and continuously, whereas the meadows are confused with bitumen and tiles intricately in comparison maps. For the Dioni dataset, artificial surface objects of dense urban fabric have a poor appearance in comparison results, and some regions of fruit trees and sparse vegetation are misclassified as urban. The 3-DCAE, 3-DVSCNN, and MDL4OW methods have confusing results for water and coastal water, and SAE_LR and CNN_HSI methods tend to misclassify coastal water as water. Due to the influence of the cloud and its shadow, the corresponding ground objects are generally misclassified as various types. The UMsDFL method achieves the best classification boundaries and the least confusing area, especially for regions of water and coastal water.

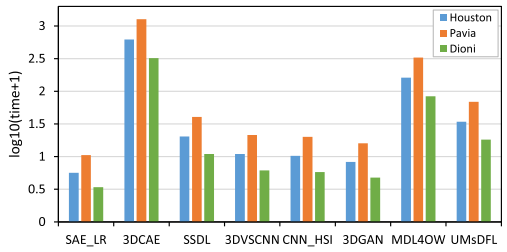

Fig. 13. Total inferring time (seconds) of the compared methods on Houston, Pavia, and Dioni datasets.

E. Efficiency Evaluation

The inferring time of various methods are concerned, as illustrated in Fig. 13. All the experiments are carried out on a Tesla P100 graphic processing unit with 16 GB of memory, and the inferring time on full HSI is recorded. It is obvious that the inferring time rises as the size of HSI increases. The SAE_LR method is the fastest due to the simple network structure and single feature learning, but its classification performance is relatively worse. In contrast, the 3-DCAE method is the slowest since the implementation efficiency of the framework is not high and the 3-D convolutional calculation is a bit time-consuming. The 3-DVSCNN and MDL4OW methods have relatively stable classification results on three datasets, but the MDL4OW method costs more execution time and has lower efficiency. Although the UMsDFL method does not take the least time for inferring, it achieves the best classification results, considering it transforms multiscale and diverse features from HSI. The UMsDFL method builds the unsupervised spatial–spectral CNN in an appropriately designed structure, which is suitable and effective for hyperspectral and textural feature learning. It extracts comprehensive features via parallel network branches and produces the best classification maps using small sample sets to obtain superior performance with a bit more but acceptable inferring time.

V. Conclusion

A UMsDFL approach for HSI classification has been proposed in this article. In detail, after applying the modified SNIC to HSI with the heuristic calculation of superpixel size, the deep features of superpixels are learned through the unsupervised spatial–spectral CNN. The network is designed with the convolutional encoder and decoder, the clustering branch, and the multilayer feature combination. Then, the object characteristics and spatial relationships in multiscale contexts are extracted through unsupervised CNN, and diverse features of hyperspectral information and NSCT textures are extracted collaboratively. Finally, we utilize the RF classifier to fuse multiscale and diverse features, and obtain comprehensive classification maps using the small sample set. The superpixel regularization is adopted to optimize the pixel classification results and achieve good performance.

In summary, the main contributions of this article are proposals of the unsupervised spatial–spectral CNN with clustering and multilayer combination, multiscale and diverse feature learning, and comprehensive classification with limited labeled samples. Compared with the SAE_LR, 3-DCAE, SSDL, 3-DVSCNN, CNN_HSI, 3-DGAN, and MDL4OW methods, the experimental results consistently show that the unsupervised spatial–spectral feature learning of UMsDFL can efficiently improve the feature expression and HSI classification accuracy. The unsupervised CNN exhibits the excellent ability of feature extraction, and multiscale and diverse feature learning is conductive to raising the performance. In future works, considering that the multiple features are distinct and interconnected, we will further explore the attention mechanism to make full use of complementary advantages. For the comprehensive classification, an advanced feature fusion strategy will also be explored to promote the classification results.

References

[1] N. Audebert, B. L. Saux, and S. Lefevre, "Deep learning for classification of hyperspectral data: A comparative review," *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 2, pp. 159–173, Jun. 2019.

[2] P. Ghamisi, J. Plaza, Y. Chen, J. Li, and A. J. Plaza, "Advanced spectral classifiers for hyperspectral images: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 1, pp. 8–32, Mar. 2017.

[3] 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.

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

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

[6] L. Shi, L. Zhang, J. Yang, L. Zhang, and P. Li, "Supervised graph embedding for polarimetric SAR image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 2, pp. 216–220, Mar. 2013.

[7] Y. Y. Tang, H. Yuan, and L. Li, "Manifold-based sparse representation for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 12, pp. 7606–7618, Dec. 2014.

[8] W. Sun, G. Yang, B. Du, L. Zhang, and L. Zhang, "A sparse and low-rank near-isometric linear embedding method for feature extraction in hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 4032–4046, Jul. 2017.

[9] L. Ma, M. M. Crawford, X. Yang, and Y. Guo, "Local-manifold-learning-based graph construction for semisupervised hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2832–2844, May 2015.

[10] 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, Jul. 2010.

[11] 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.

[12] G. Moser and S. B. Serpico, "Combining support vector machines and Markov random fields in an integrated framework for contextual image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 5, pp. 2734–2752, May 2013.

[13] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 11, pp. 3804–3814, Nov. 2008.

[14] M. Dalla Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone, "Morphological attribute profiles for the analysis of very high resolution images," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 10, pp. 3747–3762, Oct. 2010.

[15] D. Hong, X. Wu, P. Ghamisi, J. Chanussot, N. Yokoya, and X. X. Zhu, "Invariant attribute profiles: A spatial-frequency joint feature extractor for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 3791–3808, Jun. 2020.

[16] S. Jia, K. Wu, J. Zhu, and X. Jia, "Spectral–spatial Gabor surface feature fusion approach for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 1142–1154, Feb. 2019.

[17] Y. Y. Tang, Y. Lu, and H. Yuan, "Hyperspectral image classification based on three-dimensional scattering wavelet transform," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2467–2480, May 2015.

[18] L. Li, L. Ma, L. Jiao, F. Liu, Q. Sun, and J. Zhao, "Complex contourlet-CNN for polarimetric SAR image classification," *Pattern Recognit.*, vol. 100, Apr. 2020, Art. no. 107110.

[19] J. Yang, Y.-Q. Zhao, and J. C.-W. Chan, "Learning and transferring deep joint spectral–spatial features for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4729–4742, Aug. 2017.

[20] 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.

[21] M. Zhang, W. Li, and Q. Du, "Diverse region-based CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2623–2634, Jun. 2018.

[22] C. Yu, R. Han, M. Song, C. Liu, and C.-I. Chang, "A simplified 2D-3D CNN architecture for hyperspectral image classification based on spatial–spectral fusion," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 2485–2501, Apr. 2020.

[23] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.

[24] X. Zhang, Y. Sun, K. Jiang, C. Li, L. Jiao, and H. Zhou, "Spatial sequential recurrent neural network for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 4141–4155, Nov. 2018.

[25] Q. Liu, F. Zhou, R. Hang, and X. Yuan, "Bidirectional-convolutional LSTM based spectral–spatial feature learning for hyperspectral image classification," *Remote Sens.*, vol. 9, no. 12, p. 1330, Dec. 2017.

[26] 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.

[27] X. Sun, F. Zhou, J. Dong, F. Gao, Q. Mu, and X. Wang, "Encoding spectral and spatial context information for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2250–2254, Dec. 2017.

[28] P. Zhou, J. Han, G. Cheng, and B. Zhang, "Learning compact and discriminative stacked autoencoder for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4823–4833, Jul. 2019.

[29] S. Wan, C. Gong, P. Zhong, S. Pan, G. Li, and J. Yang, "Hyperspectral image classification with context-aware dynamic graph convolutional network," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 597–612, Jan. 2021.

[30] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5966–5978, Jul. 2021.

[31] S. Wan, C. Gong, P. Zhong, B. Du, L. Zhang, and J. Yang, "Multiscale dynamic graph convolutional network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3162–3177, May 2020.

[32] L. Zhu, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Generative adversarial networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5046–5063, Sep. 2018.

[33] X. Wang, K. Tan, Q. Du, Y. Chen, and P. Du, "Caps-TripleGAN: GAN-assisted CapsNet for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 7232–7245, Sep. 2019.

[34] J. Wang, F. Gao, J. Dong, and Q. Du, "Adaptive DropBlock-enhanced generative adversarial networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 5040–5053, Jun. 2021.

[35] M. Zhang, M. Gong, Y. Mao, J. Li, and Y. Wu, "Unsupervised feature extraction in hyperspectral images based on Wasserstein generative adversarial network," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 5, pp. 2669–2688, May 2018.

[36] S. Yu, S. Jia, and C. Xu, "Convolutional neural networks for hyperspectral image classification," *Neurocomputing*, vol. 219, pp. 88–98, Jan. 2017.

[37] S. Mei, J. Ji, Y. Geng, Z. Zhang, X. Li, and Q. Du, "Unsupervised spatial–spectral feature learning by 3D convolutional autoencoder for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6808–6820, Sep. 2019.

[38] L. Hu, X. Luo, and Y. Wei, "Hyperspectral image classification of convolutional neural network combined with valuable samples," *J. Phys., Conf. Ser.*, vol. 1549, no. 5, Jun. 2020, Art. no. 052011.

[39] J. Yue, S. Mao, and M. Li, "A deep learning framework for hyperspectral image classification using spatial pyramid pooling," *Remote Sens. Lett.*, vol. 7, no. 9, pp. 875–884, Jun. 2016.

[40] S. Hao, W. Wang, Y. Ye, T. Nie, and L. Bruzzone, "Two-stream deep architecture for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2349–2361, Apr. 2018.

[41] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2018.

[42] R. Li, S. Zheng, C. Duan, Y. Yang, and X. Wang, "Classification of hyperspectral image based on double-branch dual-attention mechanism network," *Remote Sens.*, vol. 12, no. 3, p. 582, Feb. 2020.

[43] W. Wang, S. Dou, Z. Jiang, and L. Sun, "A fast dense spectral–spatial convolution network framework for hyperspectral images classification," *Remote Sens.*, vol. 10, no. 7, p. 1068, Jul. 2018.

[44] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019.

[45] M. Imani and H. Ghassemian, "An overview on spectral and spatial information fusion for hyperspectral image classification: Current trends and challenges," *Inf. Fusion*, vol. 59, pp. 59–83, Jul. 2020.

[46] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 132–149.

[47] X. Zhan, J. Xie, Z. Liu, Y.-S. Ong, and C. C. Loy, "Online deep clustering for unsupervised representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 6688–6697.

[48] S. Zhang *et al.*, "EMMCNN: An ETPS-based multi-scale and multi-feature method using CNN for high spatial resolution image land-cover classification," *Remote Sens.*, vol. 12, no. 1, p. 66, Dec. 2019.

[49] M. Längkvist, A. Kiselev, M. Alirezaie, and A. Loutfi, "Classification and segmentation of satellite orthoimagery using convolutional neural networks," *Remote Sens.*, vol. 8, no. 4, p. 329, Apr. 2016.

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

[51] R. A. Ansari and K. M. Buddhiraju, "Textural classification based on wavelet, curvelet and contourlet features," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 2753–2756.

[52] R. Achanta and S. Susstrunk, "Superpixels and polygons using simple non-iterative clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4651–4660.

[53] 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.

[54] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.

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

[56] M. N. Do and M. Vetterli, "The contourlet transform: An efficient directional multiresolution image representation," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2091–2106, Dec. 2005.

[57] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. COM-31, no. 4, pp. 532–540, Apr. 1983.

[58] M. N. Do and M. Vetterli, "Framing pyramids," *IEEE Trans. Signal Process.*, vol. 51, no. 9, pp. 2329–2342, Sep. 2003.

[59] A. L. da Cunha, J. Zhou, and M. N. Do, "The nonsubsampled contourlet transform: Theory, design, and applications," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 3089–3101, Oct. 2006.

[60] J. Cheng, F. Zhang, D. Xiang, Q. Yin, and Y. Zhou, "PolSAR image classification with multiscale superpixel-based graph convolutional network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022.

[61] J. Cheng, F. Zhang, D. Xiang, Q. Yin, Y. Zhou, and W. Wang, "PolSAR image land cover classification based on hierarchical capsule network," *Remote Sens.*, vol. 13, no. 16, p. 3132, Aug. 2021.

[62] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 2097–2104.

[63] S. Liu, Q. Shi, and L. Zhang, "Few-shot hyperspectral image classification with unknown classes using multitask deep learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 5085–5102, Jun. 2021.

Shuyu Zhang received the B.E. and Ph.D. degrees from the College of Earth Sciences, Zhejiang University, Hangzhou, China, in 2015 and 2020, respectively.

She is a Post-Doctoral Researcher with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. Her research interests include hyperspectral image classification and deep learning.

Meng Xu (Member, IEEE) received the B.S. and M.E. degrees in electrical engineering from the Ocean University of China, Qingdao, China, in 2011 and 2013, respectively, and the Ph.D. degree from the University of New South Wales, Canberra, ACT, Australia, in 2017.

She is an Associate Research Fellow with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. Her research interests include cloud removal and remote sensing image processing.

Jun Zhou (Senior Member, IEEE) received the B.S. degree in computer science and the B.E. degree in international business from the Nanjing University of Science and Technology, Nanjing, China, in 1996 and 1998, respectively, the M.S. degree in computer science from Concordia University, Montreal, QC, Canada, in 2002, and the Ph.D. degree in computing science from the University of Alberta, Edmonton, AB, Canada, in 2006.

He was a Research Fellow with the Research School of Computer Science, The Australian National University, Canberra, ACT, Australia, and a Researcher with the Canberra Research Laboratory, National Information and Communications Technology Australia, Canberra. In 2012, he joined the School of Information and Communication Technology, Griffith University, Nathan, QLD, Australia, where he is an Associate Professor. His research interests include pattern recognition, computer vision, and spectral imaging and their applications in remote sensing and environmental informatics.

Sen Jia (Senior Member, IEEE) received the B.E. and Ph.D. degrees from the College of Computer Science, Zhejiang University, Hangzhou, China, in 2002 and 2007, respectively.

Since 2008, he has been with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China, where he is a Full Professor. His research interests include hyperspectral image processing, signal and image processing, and machine learning.