SDFL-FC: Semisupervised Deep Feature Learning With Feature Consistency for Hyperspectral Image Classification

Yun Cao, Yuebin Wang, *Member, IEEE*, Junhuan Peng, Chunping Qiu, Lei Ding, and Xiao Xiang Zhu, *Fellow, IEEE*

***Abstract*— Semisupervised deep learning methods (DLMs) can mitigate the dependence on large amounts of labeled samples using a small number of labeled samples. However, for semisupervised deep feature learning (SDFL), the quality of extracted features cannot be well ensured without a certain amount of labeled samples. To address this issue, we develop the SDFL method with feature consistency (SDFL-FC) for the hyperspectral image (HSI) classification. The SDFL-FC first adopts the convolutional neural network (CNN) to extract spectral–spatial features of HSI and then uses the fully connected layers (FCLs) to model the feature consistency. Moreover, two constraints that enforce both the feature consistency of single pixel (FCS) and feature consistency of group pixels (FCG) are introduced to obtain the representative and discriminative features. The FCS is achieved by the generative adversarial network (GAN) regularization, which can reconstruct the original data from extracted features. The FCG is based on the assumption that the features of group pixels should have similar characteristics within a superpixel, which is embedded in each FCL. The final FCL outputs the class labels, and the cross-entropy (CE) loss is calculated with the labeled samples, while the two losses of FCS and FCG are calculated with all the training samples (both labeled and unlabeled). SDFL-FC integrates the FCS, FCG, and CE loss into a unified objective function and uses a customized iterative optimization algorithm to optimize it. Experiments demonstrate that the SDFL-FC can outperform the related state-of-the-art HSI classification methods.**

***Index Terms*— Convolutional neural network (CNN), feature consistency, fully connected network, hyperspectral image (HSI) classification, optimization.**

Manuscript received October 5, 2020; revised November 15, 2020; accepted December 1, 2020. Date of publication December 24, 2020; date of current version November 24, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 41801241 and Grant 41711411, in part by the Fundamental Research Funds for the Central Universities under Grant 292018029 and Grant 375201906, in part by the Key Research and Development Projects of Shanxi Province under Grant 201903D121142, in part by the Open Fund of the State Key Laboratory of Remote Sensing Science under Grant OFSLRSS201923, and in part by the Guizhou Science and Technology Plan Project under Grant Qiankehezhicheng[2020] 4Y022. The work of Xiao Xiang Zhu was supported by the European Research Council (ERC) through the European Union's Horizon 2020 Research and Innovation Program (Acronym: So2Sat) under Grant ERC-2016-StG-714087 and in part by the German Federal Ministry of Education and Research (BMBF) in the framework of the International Future AI Lab "AI4EO" under Grant 01DD20001. *(Corresponding author: Yuebin Wang.)*

Yun Cao and Junhuan Peng are with the School of Land Science and Technology, China University of Geosciences, Beijing 100083, China (e-mail: cy12160019@163.com; pengjunhuan@163.com).

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

Chunping Qiu is with the Data Science in Earth Observation (SiPEO), Technical University of Munich (TUM), 80333 Munich, Germany (e-mail: chunping.qiu@outlook.com).

Lei Ding is with the Department of Information Engineering and Computer Science, University of Trento, 38123 Trento, Italy (e-mail: lei.ding@unitn.it).

Xiao Xiang Zhu is with the Remote Sensing Technology Institute (IMF), German Aerospace Center (DLR), 82234 Weßling, Germany, and also with the Data Science in Earth Observation (SiPEO), Technical University of Munich (TUM), 80333 Munich, Germany (e-mail: xiaoxiang.zhu@dlr.de).

Digital Object Identifier 10.1109/TGRS.2020.3044094

I. Introduction

HYPERSPECTRAL images (HSIs) can provide continuous observation bands and rich spectral information for each pixel in the remote sensing image, which can help us effectively identify different materials of interest [1]. HSI classification has always been one of the important topics in the field of HSI applications. Feature learning is an essential task for the HSI classification due to HSI's high dimensionality [2].

In the early stage of the studies on feature learning, many related methods extract features in a shallow manner [3], [4], such as the principal component analysis (PCA) [5], independent component analysis [6], and local linear embedding [7]. Bandos *et al.* [8] employed the regularized linear discriminant analysis for HSI classification in the case of a small ratio between the number of training samples and the number of spectral features. Li *et al.* [9] proposed a semisupervised learning algorithm to obtain the target's class label from a posterior distribution, which was built on the learned classification distributions and a Markov random field. In [10], the invariant attribute profiles locally extracted invariant features from HSI in both spatial and frequency domains. Villa *et al.* [11] proposed an independent component discriminant analysis for HSI classification. The independent component analysis was used to choose a transform matrix to make transformed components independent as soon as possible. Wang *et al.* [12] proposed a self-supervised low-rank representation method for HSI classification. In [13], a pixel- and superpixel-level aware subspace learning method was proposed to use the spectral information and spatial correlation among pixels effectively. However, these methods learn features in a shallow manner, whose abilities are limited to extract representative and discriminative features.

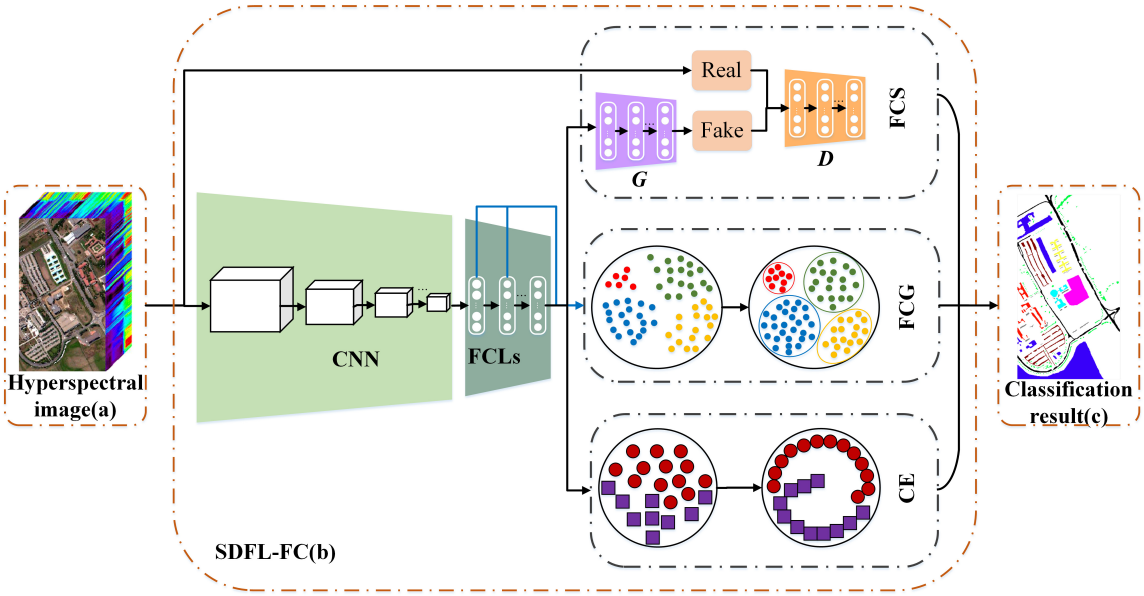

Fig. 1. Workflow of the SDFL-FC method. (a) HSI. (b) Learning process of SDFL-FC method. (c) Classification results. In (b), the CNN extracts the spectral–spatial features, whereas the final layer feature of CNN is transmitted to FCLs. Each FCL is constrained by the FCG (blue line), whereas the final feature of FCL is transmitted to FCS, FCG, and CE loss (black line). SDFL-FC integrates the FCS, FCG, and CE loss into a unified objective function.

Recently, deep learning methods (DLMs) that model deep feature presentations have achieved great success for the HSI classification [14]–[16]. In general, the DLMs can be divided into supervised, semisupervised, and unsupervised methods. The supervised DLMs have shown promising results for HSI classification with a certain amount of labeled training samples. In [17], a 3-D convolutional neural network (CNN)-based feature extraction model with L2 regularization and dropout was proposed to extract spectral–spatial features for HSI classification. Liu *et al.* [18] developed a supervised deep feature extraction model based on a siamese CNN to learn features. Hu *et al.* [19] employed deep CNNs for HSI classification in a pixel level. Zhao *et al.* [20] jointly used dimension reduction and deep CNNs to extract spectral–spatial features for HSI classification. A novel classification framework [21] is proposed that utilized deep CNN to learn pixel-pair features. The pixel pairs were constructed by combining the target pixel and its surrounding pixels. Li *et al.* [22] proposed a 3-D CNN that can simultaneously learn the spectral features and spatial features. A wider and deeper CNN that can optimally explore local contextual interactions was proposed in [23]. This method jointly exploited local spatial–spectral relationships of neighboring pixels by using a multiscale convolutional filter. Zhu *et al.* [24] explored the discriminator function of generative adversarial network (GAN) in a supervised way for HSI classification, where two GAN networks were designed: the discriminator as a spectral classifier and the discriminator as a spectral–spatial classifier. However, for supervised DLMs, the data instances may not be sufficient in real applications. It usually takes lots of time, labor to label the training samples.

Most unsupervised or semisupervised DLMs can extract deep features from an amount of unlabeled data, which can mitigate the dependence on the large amounts of labeled samples [25]–[27]. Self-supervised DLMs, as a form of unsupervised DLMs, play an essential role in learning from unlabeled or less labeled data. Wang *et al.* [28] proposed an HSI feature learning network to full advantage of the properties of subpixel, pixel, and superpixel levels by the self-supervised way. Simultaneously, the conditional random field framework is embedded into the network to improve the classification performance. The self-supervised fuzzy clustering network (SFCN) [29] is proposed to conduct retinal image classification, consisting of three main components: a feature learning module, reconstruction module, and a fuzzy self-supervision module. The loss function of SFCN is the weighted sum of three parts: reconstruction, self-supervision, and fuzzy supervision. The SFCN is optimized through a two-stage optimization algorithm. Zhang *et al.* [30] proposed the self-supervised convolutional subspace clustering network (S^2ConvSCN) to achieve simultaneous feature learning and subspace clustering. The S^2ConvSCN combines three parts into a joint optimization framework that are a ConvNet module (for feature learning), a self-expression module (for subspace clustering), and a spectral clustering module (for self-supervision). In [31], a more generalized embedding network with self-supervised learning is applied to incorporate with episodic task-based metalearning for few-shot image classification, where a metalearning is applied on top of a pretrained embedding network. Semisupervised DLMs takes a middle ground, which combines a small amount of labeled data with a large amount of unlabeled data. In [32], a 1-D GAN was proposed to construct a semisupervised DLM for HSI classification. The 1-D GAN is first trained with the unlabeled samples and then fine-tune its discriminator to classify HSIs with labeled samples. Wu and Prasad [33] proposed the semisupervised deep learning using pseudolabels (PL-SSDL) method. The PL-SSDL first applied the 1-D convolutional recurrent neural networks to pretrain the abundant unlabeled

data with pseudolabels and then fine-tune with the labeled data. The 1-D GAN and the PL-SSDL methods are pixel-based methods, where the useful spectral–spatial information of HSI is not well explored. Moreover, for these unsupervised or semisupervised DLMs, the image features' quality cannot be well ensured. Without the representative and discriminative features, the performance of HSI classification is limited. The challenge is to achieve appropriate representations by improving these approaches. Our method takes a hyperspectral data cube as input. The CNN extractor is used to exploit local spatial structures and spectral correlations. Moreover, the two constraints that enforce FCS and FCG are introduced to ensure the quality of extracted features, which can provide useful feedback information and obtain representative and discriminative features. With these features, the classification performance of HSI can be enhanced.

To reduce the dependence on large amounts of labeled samples, in this article, we develop the semisupervised deep feature learning (SDFL) method with feature consistency (SDFL-FC) for HSI classification. First, with CNN feature extractor, we extract spectral–spatial features from a 3-D patch, which can exploit local spatial structures and spectral correlations. The final layer feature of CNN is transmitted to the fully connected layers (FCLs), which models the feature consistency. Moreover, two constraints that enforce the FCS and FCG are introduced to obtain the representative and discriminative features. The FCS is achieved by GAN regularization, which can reconstruct the original data from extracted features. It drives the extracted features to minimize the differences between the reconstructed data and the original data. The FCG is based on the assumption that the features of group pixels should have similar characteristics within a superpixel. To encode the features more accurately, each FCL is further constrained by the FCG to verify the similarity of features within a superpixel. The final FCL outputs the class labels and the cross-entropy (CE) loss is calculated with the labeled samples, while the two losses of FCS and FCG are calculated with all the training samples (both labeled and unlabeled). Then, SDFL-FC integrates the FCS, FCG, and CE loss into a unified objective function and uses a customized iterative optimization algorithm to optimize it. The results tested on three HSI data sets can validate that SDFL-FC outperforms the related state-of-the-art HSI classification methods. An overview of SDFL-FC for HSI classification is shown in Fig. 1.

The main contributions of this article are as follows.

1) SDFL-FC is developed to enforce the feature consistencies, which includes FCS and FCG designs. The FCS reconstructs the original data to minimize the differences between the reconstructed data and original data, whereas the FCG enforces the extracted features of group pixels to have similar characteristics within a superpixel.
2) SDFL-FC integrates the FCS, FCG, and CE loss into a unified objective function, which formulates an efficient end-to-end training framework. With the limited labeled samples, the features extracted from SDFL-FC can be ensured to be representative and discriminative.
3) SDFL-FC is optimized through a customized iterative algorithm. The results tested on three HSI data sets show that SDFL-FC outperforms the related state-of-the-art HSI classification methods.

For clarity, we illustrate important notations and definitions in Table I.

TABLE I
NOTATIONS AND DEFINITIONS

Notation	Definition
$\mathbf{X}$	Data matrix. $\mathbf{X} \in \mathbb{R}^{w\times w\times d\times n}$.
$\mathbf{C}^{(l_1)}$	The features of l_1 layer of CNN. $1 \le l_1 \le L_1$.
$\mathbf{f}^{(l_2)}$	The features of l_2 layer of FCLs. $1 \le l_2 \le L_2$.
$\mathbf{u}_i^{(l_2)}$	The average vector of features of pixel i in l_2 layer.
$\mathbf{h}_G^{(m_1)}$	The output of m_1 layer of G. $1 \le m_1 \le M_1$.
$\mathbf{h}_D^{(m_2)}$	The output of m_2 layer of D. $1 \le m_2 \le M_2$.
$\mathbf{W}^{(l_1)}$, $\mathbf{b}^{(l_1)}$	The weights and bias of CNN, respectively.
$\mathbf{W}^{(l_2)}$, $\mathbf{b}^{(l_2)}$	The weights and bias of FCLs, respectively.
$\mathbf{W}_G^{(m_1)}$, $\mathbf{W}_G^{(m_1)}$	The weights and bias of G.
$\mathbf{W}_D^{(m_2)}$, $\mathbf{W}_D^{(m_2)}$	The weights and bias of D.
K	The number of the superpixels.
m	The number of pixels within a superpixel.
ψ, φ	The activation functions.
λ_1, λ_2	Balance the corresponding terms.
d	The number of bands.
n	The number of samples.

II. Related Work

In this section, we introduce the related works about GAN and superpixel-based HSI classification.

A. GAN

GAN is a deep neural network proposed by Goodfellow *et al.* [34], which consists of two game participants: a generator G and a discriminator D. The applications of GAN have appeared in the fields of computer vision. One of the focuses of the studies on GAN is the ability to generate image samples that have the same distribution with the real samples. The SRGAN [35] was proposed for image super-resolution, which can recover the photorealistic textures from heavily downsampled images. The CycleGAN [36] introduced a cycle consistency loss to transform the images from the source domain to the target domain without matching pairs of images, which makes the data preparation much simpler. Another research direction is related to image analysis. An auxiliary classifier GAN [37] can be used in image classifications, whose discriminator D was modified to output the class labels for the training data. The weighted GAN [38] was proposed to transfer the clustering information of images to construct the hashing code for fast image retrieval.

With their success in computer vision, GAN quickly drew research interest in remote sensing domains. The GAN was combined with deep metric learning in [39] to regularize the high-level features extracted from the pretrained CNN

for high spatial resolution remote sensing images retrieval. Zhang *et al.* [40] optimized the objective function of the Wasserstein GAN to learn the features for HSI classification, where its discriminator consisted of convolutional layers is used to extract spatial–spectral features. In [32], a 1-D GAN was proposed to construct a semisupervised DLM for HSI classification, where its discriminator is also a spectral classifier to classify HSIs with labeled samples. When applied GAN to the HSI classification, attention is often paid to its discriminator function as a classifier or feature extractor. However, limited attention has been paid to its generator, which can reconstruct the original data to minimize the loss between the reconstructed data and the original data. In [41] and our SDFL-FC, the GAN generator's function is considered. In [41], the generator has two tasks: the reconstruction task and the classification task. Using an encoder and a decoder, the input HSI cube, which is also the input of the generator, is reconstructed. In our proposed SDFL-FC, the generator's function is explored when using GAN to achieve the FCS. The generator accepts the extracted features as inputs and reconstructs the corresponding HSI spectrum to ensure the feature consistency of single pixel. Moreover, the qualities of the extracted features can be evaluated.

B. Superpixel-Based HSI Classification

A superpixel in HSI collects spatially proximal and spectrally similar pixels that can preserve the object boundaries and describe the local structural information [42]. Superpixel can be segmented by using the graph-based algorithms, such as entropy rate (ER) superpixel segmentation [43] and normalized cuts [44], and the gradient-descent-based algorithms, such as mean shift [45], TurboPixels [46], and SLIC [47]. Superpixel-based classification methods have been successfully used for HSI classification. Fang *et al.* [48] employed the ER segmentation algorithm to obtain superpixel, and pixels within a superpixel were represented via the joint sparse regularization. In [49], a superpixel-level sparse representation classification framework with multitask learning was developed, which exploited the class-level sparsity before multiple-feature fusion and the correlation and distinctiveness of pixels in a local spatial region. Zhang *et al.* [50] proposed the multiscale superpixel-based sparse representation algorithm to obtain different structure information. Jiang *et al.* [51] used the superpixel-wise PCA to extract the intrinsic low-dimensional features. In [52], the multiscale superpixel features were captured, while the correlation among different scales was considered via the recurrent neural networks. Jia *et al.* [53] proposed the collaborative representation-based multiscale superpixel fusion method, where multiscale superpixels were generated from the extended multiattribute profiles to regularize the classification map. From a general perspective, existing works for superpixel-based HSI classification mostly employ the ER segmentation algorithm to get homogeneous nonoverlapping superpixels and then obtain the classification results based on the superpixel map. Building on the above studies, we first use the ER method to segment HSI and then propose a simple yet effective approach to regularize the FCG, which enforces the features within a superpixel to have similar characteristics.

III. Proposed Approach

In this section, a novel semisupervised method SDFL-FC is presented. First, we introduce the motivation of this article in Section III-A. Second, we introduce the CNN feature extractor in Section III-B. Third, the FCLs are described in Section III-C. Next, three loss functions are formulated in Section III-E, and the objective function of SDFL-FC is described in Section III-E. Moreover, we give the optimization of SDFL-FC in Section III-F and the implementation in Section III-G.

A. Motivation

Previous works have proved that supervised DLMs have great potentials for HSI classification with a certain number of labeled samples. However, data instances may not be sufficient. Semisupervised or unsupervised DMLs can extract features from an amount of unlabeled data. In this article, we propose the SDFL-FC method to alleviate the dependence on large amounts of labeled samples. In the SDFL-FC, the CNN feature extractor is adopted to extract spectral–spatial features. However, for the semisupervised or unsupervised DLMs, the quality of image features cannot be well ensured. Thus, in our proposed SDFL-FC, the final feature of CNN is transmitted to the FCLs, which models the feature consistency to ensure the quality of features. Feature consistencies include not only FCS but also FCG. The FCS reconstructs the original data to minimize the differences between the reconstructed data and the original data, whereas the FCG enforces the extracted features of group pixels to have similar characteristics within a superpixel. Moreover, the CE loss is introduced to enhance the feature learning. SDFL-FC integrates the FCS, FCG, and CE loss into a unified objective function, which formulates an efficient end-to-end trained framework. The framework of SDFL-FC is optimized with a customized iterative algorithm.

B. CNN Feature Extractor

Using the DLMs, spectral–spatial features are introduced to describe the high-dimensional HSI data. In this section, we describe the CNN feature extractor for extracting spectral-spatial features. The stride convolutional layers are employed to exploit local spatial structures and spectral correlations of the input 3-D patch. Then, the features extracted by CNN are flattened into a 1-D vector.

Given an HSI data set $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n)$, $\mathbf{X}_i \in \mathbb{R}^{w\times w\times d}$ where $w \times w$ represents the height and width of input, d is the number of HSI bands, and n is all samples of an HSI. The input 3-D patch is fed into stride convolutional layers, which can explore the useful information in the input 3-D patch

$$\mathbf{C}^{(l_1)} = \psi\left(\mathbf{W}^{(l_1)} \otimes \mathbf{C}^{(l_1-1)} + \mathbf{b}^{(l_1)}\right), \quad 1 \le l_1 \le L_1 \tag{1}$$

where ψ is ReLU activation function and $\otimes$ is the convolution operation. $\mathbf{C}^{(l_1)}(1 \le l_1 \le L_1)$ is the feature of l_1 layer. $\mathbf{W}^{(l_1)}$ and $\mathbf{b}^{(l_1)}$ are the weights and bias of l_1 layer, respectively.

The output of the CNN feature extractor $\mathbf{C}^{(L_1)}$ is flattened into a 1-D vector and then fed to the FCLs.

C. Fully Connected Layer

Since the spectral–spatial features have been obtained, we further construct the FCLs. The FCLs can capture the diverse information from the spectral–spatial features and reduce the dimension. We feed the feature $\mathbf{C}^{(L_1)}$ into FCLs, which can be described as

$$\mathbf{f}^{(l_2)} = \psi\left(\mathbf{W}^{(l_2)} \cdot \mathbf{f}^{(l_2-1)} + \mathbf{b}^{(l_2)}\right), \quad 1 \le l_2 \le L_2 \tag{2}$$

where $\cdot$ is the fully connected operation and $\mathbf{f}^{(l_2)}(1 \le l_2 \le L_2)$ is the feature of the l_2 layer. For the first layer feature of the FCL, we assume $\mathbf{f}^{(l_2-1)} = \mathbf{f}^{(0)}$, which is equivalent to the input of FCL $\mathbf{C}^{(L_1)}$. $\mathbf{W}^{(l_2)}$ and $\mathbf{b}^{(l_2)}$ are the weights and bias of l_2 layer, respectively.

The features extracted from each FCL are segmented into K nonoverlapping homogeneous regions via the ER method [43]. Each superpixel in HSI corresponds to a group of similar features. We calculate the average vector of features within a superpixel to represent the characteristics of the superpixel

$$\mathbf{u}_i^{(l_2)} = \left(\sum_{i=1}^{m} \mathbf{f}_i^{(l_2)}\right)/m, \quad 1 \le l_2 \le L_2 \tag{3}$$

where m is the number of pixels within a superpixel. $\mathbf{u}_i^{(l_2)}$ is the average vector of features, including pixel i in the l_2 layer, and $\mathbf{f}_i^{(l_2)}$ is the features of pixel i in the l_2 layer.

We construct the FCG in each FCL. The FCG is based on the assumption that features within a superpixel are required to have similar characteristics. Each FCL is constructed by FCG, which can provide useful feedback information and alleviate the "salt-and-pepper" problem. The objective function of FCG for each superpixel is expressed as

$$\Theta_{\mathrm{FCG}} = \sum_{l_2=1}^{L_2}\left[\eta_{l_2}\sum_{i=1}^{n}\left\|\mathbf{f}_i^{(l_2)} - \mathbf{u}_i^{(l_2)}\right\|_2^2\right] \tag{4}$$

where η_{l_2} is a constant.

Since the features learned by FCLs are changeable during the iteration, the superpixels are segmented using the features learned from the previous iteration. The average vector of features within a superpixel is changing during iterations, which becomes more accurate with a number of iterations increasing.

D. Output Layer

The output layer computes three kinds of constraints: FCS, FCG, and CE loss. The constraints of FCS and FCG regularize the features with all the training data (both labeled and unlabeled), whereas the CE loss optimizes the features with the labeled data.

1) FCS: The FCS drives the extracted features to minimize the differences between the reconstructed data and the original data, where the reconstructed data are generated by the GAN from the extracted features. With the GAN regularization, the FCS is well preserved, which can ensure the quality of the extracted features. GAN consists of the generator and the discriminator. After the spectral–spatial features of HSI have been obtained, the features are embedded into the generator to reconstruct the corresponding HSI spectrum. Moreover, the reconstructed data and original data are fed into the discriminator, which can evaluate the quality of the reconstructed data. With the optimization of the GAN, the reconstructed data are more similar to the real data, and the features used to generate the reconstructed data are more representative and discriminative. The generator G maps the features $\mathbf{f}^{(L_2)}$ into fake pixels, which is represented by $G(\mathbf{f}^{(L_2)})$. The output $\mathbf{h}_G^{(m_1)}$ of m_1 $(1 \le m_1 \le M_1)$ layer can be computed as follows:

$$\mathbf{h}_G^{(m_1)} = \varphi\left(\mathbf{W}_G^{(m_1)} \cdot \mathbf{h}_G^{(m_1-1)} + \mathbf{b}_G^{(m_1)}\right), \quad 1 \le m_1 \le M_1 \tag{5}$$

where φ is the leaky ReLU activation function with the leaky rate 0.2. $\mathbf{W}_G^{(m_1)}$ and $\mathbf{b}_G^{(m_1)}$ are the weights and bias of G, respectively.

The discriminator D accepts both real data and fake data as inputs and classifies the input data into real or fake with a binary classifier. Let $D(\mathbf{X})$ be the probability over real data and $D(\mathbf{h}_G^{(M_1)})$ be the probability over fake data. The output $\mathbf{h}_D^{(m_2)}$ of m_2 $(1 \le m_2 \le M_2)$ layer is described as follows:

$$\mathbf{h}_D^{(m_2)} = \varphi\left(\mathbf{W}_D^{(m_2)} \cdot \mathbf{h}_D^{(m_2-1)} + \mathbf{b}_D^{(m_2)}\right), \quad 1 \le m_2 \le M_2 \tag{6}$$

where $\mathbf{W}_D^{(m_2)}$ and $\mathbf{b}_D^{(m_2)}$ are the weights and bias of D, respectively.

The objective function of GAN is to train with the minimax problem. Since the GAN is not easy to stabilize, we apply the Wasserstein GAN (WGAN) [54] in the GAN setup, whose loss function does not take log transformation

$$\Theta_{\mathrm{FCS}} = \min_G \max_D \Theta_{\mathrm{FCS}} = E_{\mathbf{X}\sim p(\mathbf{X}^1)}[D(\mathbf{X})] - E_{\mathbf{h}_G^{(M_1)}\sim p\left(\mathbf{h}_G^{(M_1)}\right)}\left[D\left(\mathbf{h}_G^{(M_1)}\right)\right] \tag{7}$$

where E represents the expectation operator, $p(\mathbf{X})$ represents the data generating distribution, and $p(\mathbf{h}_G^{(M_1)})$ represents the generative distribution.

2) FCG: FCG is introduced based on the assumption that the features from the same category should have similar characteristics and the features from the different classes should not be mixed. To achieve this, the superpixel is considered. Superpixel can reflect the homogeneous regularity of objects, which can exploit the contextual information among pixels. The spatially proximal and spectrally similar pixels are clustered via the superpixel segmentation methods in an unsupervised way. In order to offer feedback information about the features of group pixels, the FCG enforces the features of group pixels to have similar characteristics within a superpixel. With the optimization of the network, the FCG is well preserved, which can further ensure the quality of the extracted features. As described in Section III-B, the features and average vector of features within a superpixel have been obtained, and we construct the FCG in each FCL. The final layer feature of FCL $\mathbf{f}^{(L_2)}$ is also constructed by FCG. The objective function of FCG is described in (4).

3) CE Loss: To boost the classification performance, the softmax layer [55] is wired to the output of FCLs. There are s labeled HSI samples (X_1, y_1), $(\mathrm{X}_2, y_2), \ldots, (\mathrm{X}_s, y_s)$, and q unlabeled images, in which $y_1, y_2, \ldots, y_s$ are labels and $n = s + q$. The softmax layer transmits the features $\mathbf{f}^{(L_2)}$ extracted from the last FCL into their corresponding class label, which can be defined as follows:

$$p\left(\hat{y} = j|\mathbf{f}^{(L_2)}\right) = \frac{\exp\left(\mathbf{W}^{(L_3,j)} \cdot \mathbf{f}^{(L_2)} + \mathbf{b}^{(L_3,j)}\right)}{\sum_{c=1}^{C} \exp\left(\mathbf{W}^{(L_3,c)} \cdot \mathbf{f}^{(L_2)} + \mathbf{b}^{(L_3,c)}\right)} \tag{8}$$

where C is the number of HSI categories and $\hat{y}$ is the predicted class label possibility. $\mathbf{W}^{(L_3)}$ and $\mathbf{b}^{(L_3)}$ are the weights and bias of the softmax layer. $L_3 = L_1 + L_2 + 1$.

The CE loss is defined as

$$\Theta_{\mathrm{CE}} = -\frac{1}{s}\sum_{i=1}^{s}\sum_{j=1}^{C} I(j)\log\left(p\left(\hat{y}_i = j|\mathbf{f}^{(L_2)}\right)\right) \tag{9}$$

where the value of $I(j)$ is 1 when j equals the desired label y_i of pixel i ($1 \le i \le s$); otherwise, the value is 0.

E. SDFL-FC Loss

Considering the constraints of (4), (7) and (9), we formulate the joint objective function of SDFL-FC as follows:

$$\begin{aligned}\Theta &= \Theta_{\mathrm{FCS}} + \lambda_1\Theta_{\mathrm{FCG}} + \lambda_2\Theta_{\mathrm{CE}} \\ &= E_{\mathbf{X}\sim p(\mathbf{X})}[D(\mathbf{X})] - E_{\mathbf{h}_G^{(M_1)}\sim p\left(\mathbf{h}_G^{(M_1)}\right)}\left[D\left(\mathbf{h}_G^{(M_1)}\right)\right] \\ &\quad + \lambda_1\left[\sum_{l_2=1}^{L_2}\left[\eta_{l_2}\sum_{i=1}^{n}\left\|\mathbf{f}_i^{(l_2)} - \mathbf{u}_i^{(l_2)}\right\|_2^2\right]\right] \\ &\quad + \lambda_2\left[-\frac{1}{s}\sum_{i=1}^{s}\sum_{j=1}^{C} I(j)\log\left(p\left(\hat{y}_i = j|\mathbf{f}^{(L_2)}\right)\right)\right]\end{aligned} \tag{10}$$

where λ_1 and λ_2 are used to balance the importance of the corresponding terms.

F. Optimization of SDFL-FC

A customized iterative algorithm for SDFL-FC is summarized in algorithm 1. We iteratively minimize the objective function. The FCS is optimized alternately with the FCG and CE loss, whereas FCG and CE loss are optimized simultaneously. Moreover, we adopt the Adam stochastic gradient descent policy and the backpropagation learning framework.

1) Optimization for FCS: The optimization for FCS has two steps. First, fixing the parameters of G, the parameters of D are updated. Second, fixing the parameters of D, the parameters of G are updated.

a) Fixing G and updating D: We calculate the derivative of (10) with respect to $\mathbf{W}_D^{(m_2)}$ and $\mathbf{b}_D^{(m_2)}$ and would perform the update on each iteration

$$\mathbf{W}_D^{(m_2)} = \mathbf{W}_D^{(m_2)} - \alpha_1\frac{\partial\Theta}{\partial\mathbf{W}_D^{(m_2)}}, \quad \mathbf{b}_D^{(m_2)} = \mathbf{b}_D^{(m_2)} - \alpha_1\frac{\partial\Theta}{\partial\mathbf{b}_D^{(m_2)}} \tag{11}$$

where α_1 is the learning rate of D.

Algorithm 1 SDFL-FC

Input:

Training samples: $\mathbf{X}$; Parameters: η_{l_2}, λ_1, and λ_2; Initialize: $\mathbf{W}^{(l_1)}$, $\mathbf{b}^{(l_1)}$, $\mathbf{W}^{(l_2)}$, $\mathbf{b}^{(l_2)}$, $\mathbf{W}^{(L_3)}$, $\mathbf{b}^{(L_3)}$, $\mathbf{W}_G^{(m_1)}$, $\mathbf{b}_G^{(m_1)}$, $\mathbf{W}_D^{(m_2)}$, and $\mathbf{b}_D^{(m_2)}$. ($1 \le l_1 \le L_1$, $1 \le l_2 \le L_2$, $1 \le m_1 \le M_1$, $1 \le m_2 \le M_2$).

Output:

Predicted class labels $\hat{y}$.

for number of training iterations **do**
Sample batch size of samples $\mathbf{X}$;
Compute features $\mathbf{f}^{(L_2)}$ from FCLs using Eq. (2);
Compute $\mathbf{u}_i^{(l_2)}$ using Eq. (3);
Generate fake samples using Eq.(5);
Compute the probability of real and fake data using Eq. (6);
Update $\mathbf{W}_D^{(m_2)}$ and $\mathbf{b}_D^{(m_2)}$ using Eq. (11);
Update $\mathbf{W}_G^{(m_1)}$ and $\mathbf{b}_G^{(m_1)}$ using Eq. (12);
Update $\mathbf{W}^{(l_1)}$, $\mathbf{b}^{(l_1)}$, $\mathbf{W}^{(l_2)}$, $\mathbf{b}^{(l_2)}$, $\mathbf{W}^{(L_3)}$, and $\mathbf{b}^{(L_3)}$ using Eq. (13);
Update $\mathbf{u}_i^{(l_2)}$ using Eq. (3);
end for

b) Fixing D and updating G: We calculate the derivative of (10) with respect to $\mathbf{W}_G^{(m_1)}$ and $\mathbf{b}_G^{(m_1)}$ and would perform the update on each iteration

$$\mathbf{W}_G^{(m_1)} = \mathbf{W}_G^{(m_1)} - \alpha_2\frac{\partial\Theta}{\partial\mathbf{W}_G^{(m_1)}}, \quad \mathbf{b}_G^{(m_1)} = \mathbf{b}_G^{(m_1)} - \alpha_2\frac{\partial\Theta}{\partial\mathbf{b}_G^{(m_1)}} \tag{12}$$

where α_2 is the learning rate of G.

2) Optimization for FCG and CE Loss: The parameters $\mathbf{W}^{(l_1)}$, $\mathbf{b}^{(l_1)}$, $\mathbf{W}^{(l_2)}$, $\mathbf{b}^{(l_2)}$, $\mathbf{W}^{(L_3)}$, and $\mathbf{b}^{(L_3)}$ ($1 \le l_1 \le L_1$, $1 \le l_2 \le L_2$) are updated by employing gradient descent method, and we would perform the update on each iteration

$$\begin{aligned}\mathbf{W}^{(L_3)} &= \mathbf{W}^{(L_3)} - \alpha_3\frac{\partial\Theta}{\partial\mathbf{W}^{(L_3)}}, \quad \mathbf{b}^{(L_3)} = \mathbf{b}^{(L_3)} - \alpha_3\frac{\partial\Theta}{\partial\mathbf{b}^{(L_3)}} \\ \mathbf{W}^{(l_2)} &= \mathbf{W}^{(l_2)} - \alpha_3\frac{\partial\Theta}{\partial\mathbf{W}^{(l_2)}}, \quad \mathbf{b}^{(l_2)} = \mathbf{b}^{(l_2)} - \alpha_3\frac{\partial\Theta}{\partial\mathbf{b}^{(l_2)}} \\ \mathbf{W}^{(l_1)} &= \mathbf{W}^{(l_1)} - \alpha_3\frac{\partial\Theta}{\partial\mathbf{W}^{(l_1)}}, \quad \mathbf{b}^{(l_1)} = \mathbf{b}^{(l_1)} - \alpha_3\frac{\partial\Theta}{\partial\mathbf{b}^{(l_1)}}\end{aligned} \tag{13}$$

where α_3 is the learning rate of FCG and CE.

G. Implementation

The architecture of the SDFL-FC is implemented using the PyTorch framework with an RTX 2080ti GPU. For learning the network, the Adam stochastic gradient descent policy with a batch size of 256 samples is used. The iteration number and the learning rate are set to 10 K and 1e−4, respectively. We crop each pixel and its surrounding 5×5 neighboring pixels as the input of the network. We also augment the samples by replacing the center pixel of 5×5 with its corresponding generated pixel, while the other pixels within 5×5 keep unchanged.

IV. Experiments

In this section, we evaluate the performance of the proposed method for HSI classifications. First, we briefly describe the

TABLE II
INDIAN PINES DATA SET

Class	Land Cover Type	Training (10%)	Labeling (5%)	Total
1	Alfalfa	5	2	46
2	Corn-notill	143	71	1,428
3	Corn-mintill	83	42	830
4	Corn	24	12	237
5	Grass-pasture	48	24	483
6	Grass-trees	73	36	730
7	Grass-pasture-mowed	3	1	28
8	Hay-windrowed	48	24	478
9	Oats	2	1	20
10	Soybean-notill	97	49	972
11	Soybean-mintill	245	123	2,455
12	Soybean-clean	59	29	593
13	Wheat	20	10	205
14	Woods	126	63	1,265
15	Bldg-grass-trees	39	20	386
16	Stone-Steel-Towers	9	5	93
	Total	1,024	512	10,249

used HSI data set. Then, we compare the classification results of the proposed method with those of related approaches.

A. Experimental Setup

The performance of the classification results is evaluated on three widely used HSI data sets: the Indian Pines data set, the University of Pavia data set (PaviaU), and the Houston data set (2018).

1) Indian Pines Data Set: The Indian Pines is a mixed vegetation site over the Indian Pines test area that is acquired by the AVIRIS in 1992. It consists of 145 × 145 pixels and 220 spectral bands ranging from 0.4 to 2.5 μm with a spatial resolution of 20 m. The 200 bands are preserved after removing 20 spectral bands (104–108, 150–163, and 220) due to the noise and water absorption. The data set has 16 classes and 10 249 labeled pixels.

2) PaviaU Data Set: The PaviaU data set is collected by the ROSIS sensor over the city of Pavia, Italy. The data set is composed of 610 × 340 pixels and 115 bands ranging from 0.43 to 0.86 μm with a high spatial resolution of 1.3 m. After removing 12 water absorption and noise bands, 103 bands are used in our experiment. The data set has nine classes and 42 776 labeled pixels.

3) Houston Data Set (2018): The Houston data set is collected by the Image Analysis and Data Fusion Technical Committee. The data set is composed of 601 × 2384 pixels and 48 bands with a spatial resolution of approximately 1.0 m. The data set has 20 classes and 504 172 labeled pixels.

For a fire comparison, we randomly select 5% per class as training samples (all labeled samples and unlabeled samples) for the PaviaU and Houston data sets and 10% for the Indian pines data set. The rest of the samples are used for testing. We further select 1% per class for the PaviaU and Houston data sets and 5% for the Indian pines data set as the labeled data. The labeled samples are chosen from the training samples. The detailed information with the classes and the numbers of the training and test samples of the three data sets are listed in Tables II–IV.

TABLE III
PAVIAU DATA SET

Class	Land Cover Type	Training (5%)	Labeling (1%)	Total
1	Asphalt	345	66	6,631
2	Meadows	917	186	18,649
3	Gravel	131	20	2,099
4	Trees	156	30	3,064
5	Metal sheets	77	13	1,345
6	Bare Soil	246	50	5,029
7	Bitumen	60	13	1,330
8	Bricks	181	36	3,682
9	Shadows	44	94	947
	Total	2,157	508	42,776

TABLE IV
HOUSTON DATA SET

Class	Land Cover Type	Training (5%)	Labeling (1%)	Total
1	Healthy Grass	490	98	9,799
2	Stressed Grass	1,625	325	32,502
3	Artificial Turf	34	7	684
4	Evergreen Trees	679	136	13,588
5	Deciduous Trees	252	50	5,048
6	Bare Earth	225	45	4,516
7	Water	13	2	266
8	Residential Buildings	1,988	397	39,762
9	Non-residential Buildings	11,184	2,237	223,684
10	Roads	2,290	458	45,810
11	Sidewalks	1,700	340	34,002
12	Crosswalks	75	15	1,516
13	Major Thoroughfares	2,318	463	46,358
14	Highways	492	98	9,849
15	Railways	347	69	6,937
16	Paved Parking Lots	574	115	11,475
17	Unpaved Parking Lots	7	1	149
18	Cars	333	65	6,578
19	Trains	267	53	5,365
20	Stadium Seats	346	68	6,824
	Total	25,239	5,042	504,712

B. Alternative Approaches

We adopted the following approaches to compare with our proposed SDFL-FC in terms of HSI classification accuracy.

1) *SVM With Radial Basis Function (SVM-RBF):* SVM-RBF is a classical supervised classification method. The same amount of training data are randomly selected to compare with other semisupervised methods.
2) *Active Labeling Method for Deep Learning (ALDL) [56]:* ALDL is a semisupervised active learning method, which is proposed for cost-effective selection of data to be labeled. ALDL provides three metrics for data selection, and we choose the entropy sampling.
3) *PCA + CE (PCA + CE) [5]:* The PCA was first applied to map the high-dimensionality data of HSI into a low-dimensionality domain. Then,

TABLE V
CLASSIFICATION RESULTS (%) WITH 10% TRAINING SAMPLES FOR THE INDIAN PINES DATA SET

Class	SVM-RBF	ALDL	PCA+CE	WGAN+CE	CAE+LGC	SSCNN	SESEMI	SDFL-FC
Alfalfa	0.00	39.53	36.43	40.00	93.47	32.25	81.95	93.02
Corn-notill	50.50	90.47	84.99	74.83	78.20	87.04	88.90	97.61
Corn-mintill	42.85	83.21	71.79	67.00	80.31	83.38	94.76	93.09
Corn	11.97	81.40	58.28	52.74	69.03	53.93	92.46	90.53
Grass-pasture	74.94	86.93	92.10	77.29	70.32	90.36	97.07	96.81
Grass-trees	93.13	90.19	98.78	86.36	96.89	99.38	97.63	98.72
Grass-pasture-mowed	0.00	56.04	78.97	25.38	29.05	65.38	94.93	85.21
Hay-windrowed	97.18	96.32	100.00	95.10	97.78	99.16	98.96	99.61
Oats	0.00	26.32	85.96	48.42	100.00	38.95	0.00	86.64
Soybean-notill	59.84	79.35	69.02	65.98	81.47	75.44	86.03	94.27
Soybean-mintill	92.13	90.00	85.55	82.28	88.30	89.73	92.39	98.03
Soybean-clean	23.64	70.13	64.97	49.30	85.46	54.70	91.53	97.32
Wheat	91.98	98.53	100.00	99.90	100.00	99.90	99.89	99.84
Woods	97.31	97.64	97.60	94.79	93.13	98.00	94.75	99.85
Bldg-grass-trees	31.30	83.88	72.81	41.49	43.80	74.48	83.00	89.70
Stone-Steel-Towers	50.60	100.00	95.53	92.35	95.27	94.54	92.05	96.68
OA	70.09 ± 0.94	87.86 ± 0.37	84.04 ± 0.34	76.57 ± 0.52	84.30 ± 0.73	86.06 ± 0.36	92.21 ± 0.42	**96.92** ± 0.27
AA	51.09 ± 1.84	79.37 ± 0.64	80.80 ± 1.54	68.33 ± 1.43	81.41 ± 1.06	77.29 ± 1.57	86.64 ± 0.68	**94.81** ± 0.39
Kappa	0.649 ± 0.012	0.861 ± 0.004	0.817 ± 0.004	0.733 ± 0.006	0.821 ± 0.008	0.840 ± 0.004	0.911 ± 0.005	**0.965** ± 0.003

The best results are highlighted in bold.

TABLE VI
CLASSIFICATION RESULTS (%) WITH 5% TRAINING SAMPLES FOR THE PAVIAU DATA SET

Class	SVM-RBF	ALDL	PCA+CE	WGAN+CE	CAE+LGC	SSCNN	SESEMI	SDFL-FC
Asphalt	89.41	90.96	93.39	89.92	90.03	95.68	86.57	96.61
Meadows	98.16	98.76	99.00	98.14	98.16	99.69	99.49	99.72
Gravel	63.46	78.89	82.51	77.77	79.80	74.60	69.37	90.85
Trees	87.86	90.20	86.91	85.69	96.74	88.54	95.21	93.51
Metal sheets	86.00	95.31	93.57	93.07	99.93	93.35	98.80	99.29
Bare Soil	58.93	74.40	86.53	81.19	77.65	93.34	49.82	96.39
Bitumen	74.69	86.37	80.13	78.15	91.81	77.05	92.73	93.48
Bricks	88.18	83.47	86.34	81.26	86.78	84.37	92.88	92.40
Shadows	98.88	99.54	96.65	95.33	99.05	93.31	99.51	99.63
OA	87.79 ± 0.23	91.31 ± 0.15	93.09 ± 0.12	90.68 ± 0.51	92.38 ± 0.35	93.93 ± 0.22	89.06 ± 0.45	**97.13** ± 0.20
AA	82.84 ± 0.37	88.66 ± 0.26	89.45 ± 0.19	86.73 ± 0.69	91.10 ± 0.61	88.88 ± 0.39	87.15 ± 0.74	**95.76** ± 0.31
Kappa	0.835 ± 0.003	0.884 ± 0.002	0.908 ± 0.002	0.876 ± 0.007	0.898 ± 0.005	0.919 ± 0.003	0.852 ± 0.006	**0.962** ± 0.003

The best results are highlighted in bold.

we extracted the neighborhood region of the pixel in the low-dimensionality domain and fed into a two-layer CNN. The spectral–spatial features are classified using the CE loss with the labeled data.

4) *Wasserstein GAN + CE (WGAN + CE) [54]:* WGAN extracts the spectral–spatial features of training samples using the adversarial loss. The CE loss is trained with labeled data simultaneously to correctly assign labels for the features.
5) *Convolutional AE + Local and Global Consistency (CAE + LCG) [57], [58]:* CAE extracts the spectral–spatial features using the reconstruction loss, whereas the semisupervised LCG method maps out-of-sample data.
6) *Semisupervised CNN (SSCNN) [59]:* The SSCNN is a semisupervised network that can automatically learn spectral–spatial features from HSIs. The skip connection parameters are added between the encoder layer and the decoder layer. SSCNN is trained with minimizing the sum of supervised and unsupervised cost functions.
7) *SESEMI [60]:* The SESEMI introduces a self-supervised loss term to enhance the semisupervised image classification. The supervised CE loss is computed using the ground-truth labels and self-supervised CE loss is computed using the proxy labels. The SESEMI is learned by minimizing the weighted sum of supervised and self-supervised loss components.

In the LN, PCA + CE, WGAN + CE, CAE + LCG, SSCNN, SESEMI, and our SDFL-FC, the number of the feature dimensions in each layer is selected from {32, 64, 128, 256, 512}. The number of batch size is selected from {16, 32, 64, 128, 256, 512}, whereas the learning rate is selected from {0.00001, 0.00005, 0.0002, 0.002, 0.02}. For the Indian pines and PaviaU data set, the value of L_1 is 3,

TABLE VII
CLASSIFICATION RESULTS (%) WITH 5% TRAINING SAMPLES FOR THE HOUSTON DATA SET

Class	SVM-RBF	ALDL	PCA+CE	WGAN+CE	CAE+LGC	SSCNN	SESEMI	SDFL-FC
Healthy Grass	93.74	89.39	88.21	85.55	93.62	89.66	37.62	93.37
Stressed Grass	95.02	93.10	93.99	93.85	90.73	91.67	95.77	95.26
Artificial Turf	100.00	89.14	99.36	95.13	97.49	99.10	0.00	99.56
Evergreen Trees	93.55	92.96	96.03	93.53	94.86	94.69	82.69	97.44
Deciduous Trees	55.06	62.71	70.10	64.86	63.46	67.69	34.34	77.05
Bare Earth	79.24	80.12	92.76	85.08	83.47	92.88	76.13	98.39
Water	71.90	68.20	82.76	84.79	84.03	82.79	65.98	92.02
Residential Buildings	75.67	75.40	79.61	75.16	71.86	80.09	80.81	88.15
Non-residential Buildings	93.29	93.66	95.60	93.73	94.42	94.26	88.87	97.12
Roads	53.60	63.08	66.07	59.28	62.91	62.01	59.16	71.06
Sidewalks	40.30	56.62	61.40	56.08	57.18	56.31	29.57	68.24
Crosswalks	0.00	11.56	5.34	7.13	3.6	1.01	0.00	13.47
Major Thoroughfares	55.15	71.91	65.52	67.13	67.24	60.51	50.95	78.52
Highways	60.82	74.84	82.61	82.22	77.27	74.33	40.93	88.86
Railways	85.57	92.51	96.99	94.98	96.53	94.72	66.36	99.30
Paved Parking Lots	63.92	78.75	85.30	75.71	80.59	83.32	9.31	91.84
Unpaved Parking Lots	0.00	85.90	28.68	26.53	74.15	22.09	0.00	97.28
Cars	1.23	44.26	63.04	38.81	52.1	56.28	0.52	76.27
Trains	38.93	86.47	85.83	81.28	80.14	81.50	31.94	91.00
Stadium Seats	70.96	84.56	92.31	85.14	87.51	86.83	62.16	96.21
OA	77.05 ± 0.16	82.55 ± 0.18	84.74 ± 0.11	81.77 ± 0.81	82.43 ± 0.15	82.37 ± 0.16	71.08 ± 0.44	**89.17** ± 0.08
AA	61.40 ± 0.28	74.56 ± 1.04	76.58 ± 1.16	72.3 ± 1.16	75.66 ± 0.80	73.59 ± 1.44	45.65 ± 1.10	**85.52** ± 0.26
Kappa	0.696 ± 0.002	0.772 ± 0.003	0.801 ± 0.002	0.761 ± 0.012	0.769 ± 0.005	0.770 ± 0.002	0.616 ± 0.007	**0.859** ± 0.001

The best results are highlighted in bold.

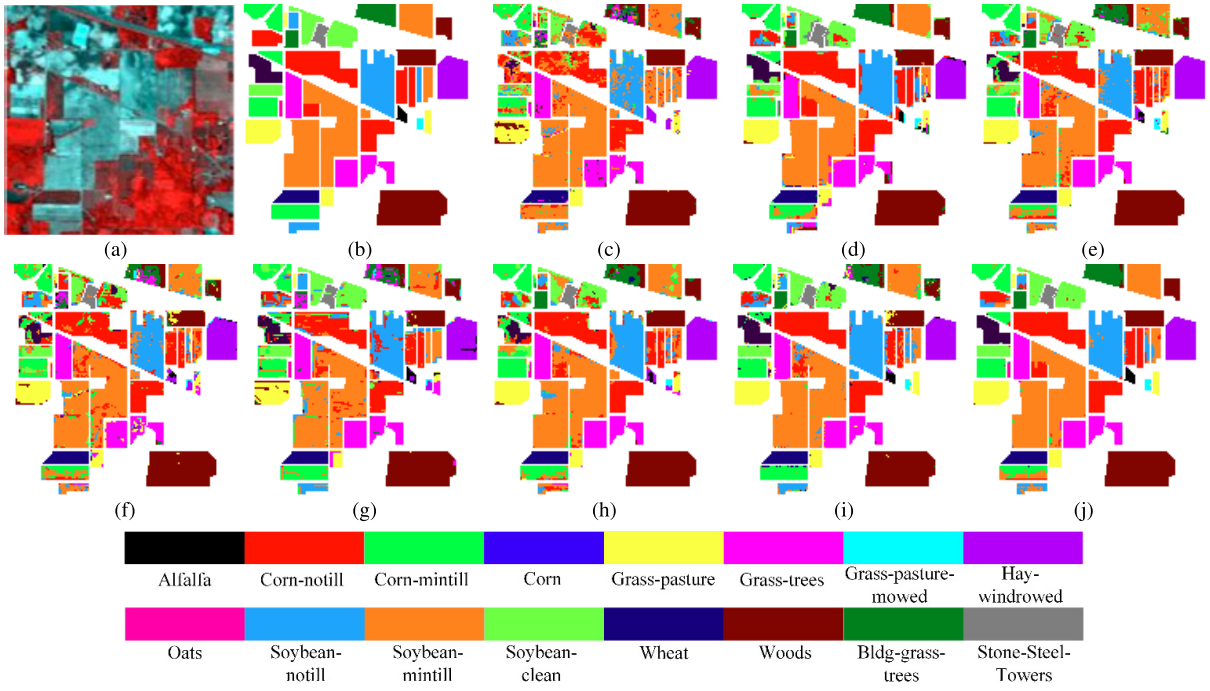

Fig. 2. Classification maps of the different methods with 10% training samples for the Indian Pines data set. (a) False color. (b) Ground truth. (c) SVM-RBF. (d) ALDL. (e) PCA + CE. (f) WGAN + CE. (g) CAE + LGC. (h) SSCNN. (i) SESEMI. (j) SDFL-FC.

and L_2 equals 2. For the Houston data set, the values of L_1 and L_2 are both 2. For the three data sets, the values of M_1 and M_2 are both 4. The number of nonoverlapping superpixel regions K is selected from {100, 300, 500, 700, 900, 1100}. The balancing parameters λ_1, λ_2, and η_{l_2} are selected from {0.01, 0.1, 1, 10, 100}. Since the size of the Houston data set is too large, we first crop the Houston data set into 20 parts, and then, the 20 parts are segmented using the ER method. For the CE loss, we randomly select s training samples as the labeled data. For the Indian pines data set, the labeled data are set to 5%, 6%, 7%, 8%, 9%, and 10%. For the PaviaU and Houston data sets, the labeled data are set to 1%, 2%, 3%, 4%, and 5%. The remaining data are unlabeled.

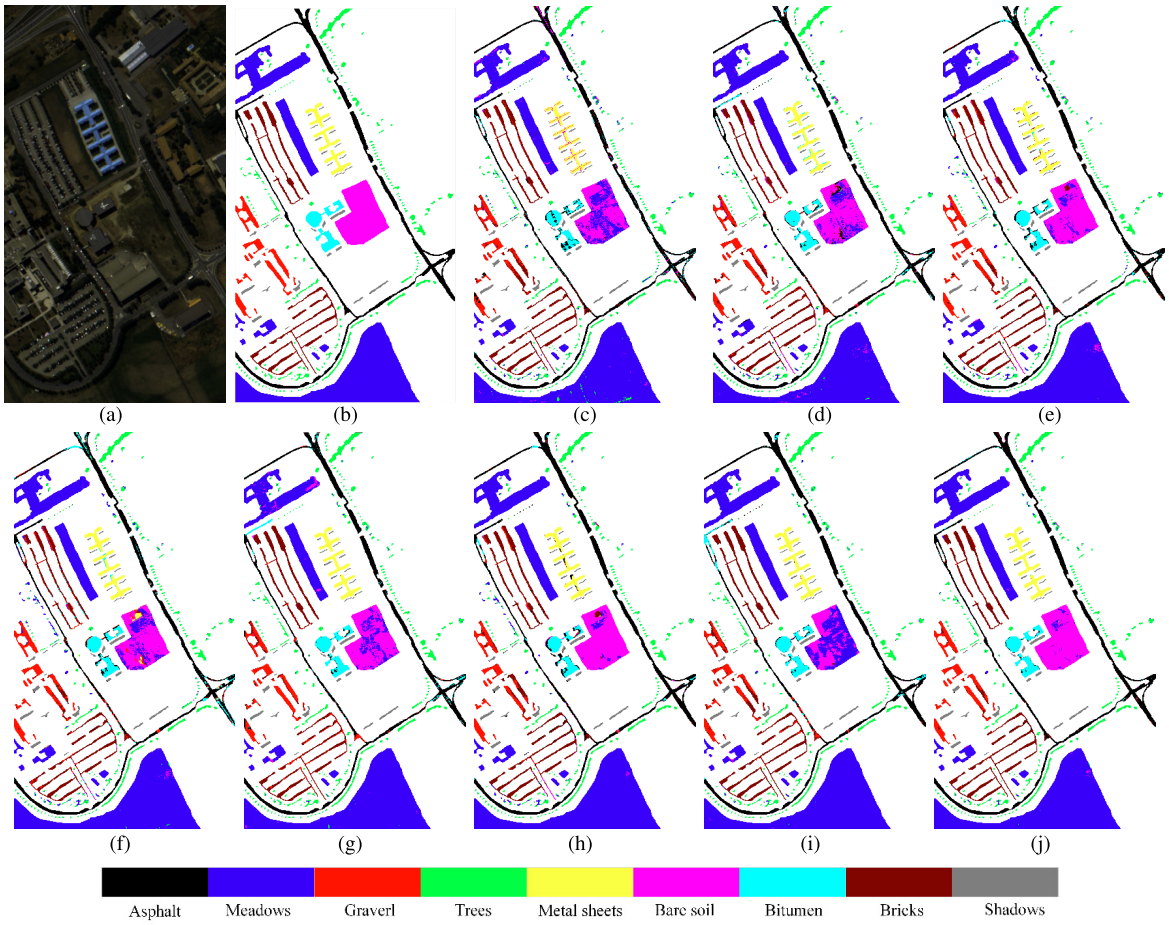

Fig. 3. Classification maps of the different methods with 5% training samples for the PaviaU data set. (a) False color. (b) Ground truth. (c) SVM-RBF. (d) ALDL. (e) PCA + CE. (f) WGAN + CE. (g) CAE + LGC. (h) SSCNN. (i) SESEMI. (j) SDFL-FC.

TABLE VIII
PERFORMANCE COMPARISON (OA%) WITH DIFFERENT PERCENTAGES TRAINING SAMPLES

Dataset	Method	5%	10%	15%	20%
Indian pines	CAE+LGC	83.32±0.49	84.30±0.73	85.31±0.87	86.21±0.54
	SDFL-FC	**95.78**±0.19	**96.92**±0.23	**97.72**±0.19	**98.49**±0.08
PaviaU	CAE+LGC	92.38±0.35	93.18±0.69	93.85±0.16	94.21±0.07
	SDFL-FC	**97.13**±0.20	**97.97**±0.07	**98.45**±0.07	**98.70**±0.06
Houston	CAE+LGC	82.43±0.15	83.43±0.15	84.19±0.10	85.25±0.08
	SDFL-FC	**89.17**±0.08	**89.93**±0.07	**90.91**±0.03	**91.53**±0.13

The best results are highlighted in bold.

C. Classification Results

Three indicators, including overall accuracy (OA), average accuracy (AA), and Kappa coefficient, are used to compute and compare the HSI classification performances of the different methods. The training and testing samples are randomly taken 15 times, and the classification results are evaluated on testing data with reporting the average and standard deviation of three evaluation indicators. The classification results of the three data sets are shown in Tables V–VIII and Figs. 2–6. We further have the following observations.

1) Compared with the related methods, the SDFL-FC can achieve the best classification performances using the testing HSI samples for the three data sets. The SDFL-FC has the best classification accuracies than those that are obtained by other methods. It demonstrates that SDFL-FC can extract more representative and discriminative features with the help of FCS and FCG.
2) From Figs. 2–4, the classification map obtained by the SDFL-FC is more compact than those obtained by other methods using the three data sets. The main reason is that FCS and FCG can provide useful feedback information for feature learning, which is useful and beneficial for HSI classification.
3) In Fig. 5, with the number of labeled samples increasing, the classification performances of all methods are improved. Moreover, the overall classification accuracies achieved by the SDFL-FC are higher than other methods with different numbers of the labeled samples. It further validates that SDFL-FC outperforms the compared methods. In the SDFL-FC, FCS, FCG, and CE loss are integrated to form a unified objective function. The FCS reconstructs the original data to minimize the differences between the reconstructed data and the original data, whereas the FCG enforces the features of group pixels to have similar characteristics within a superpixel. Therefore, the HSI classification results of SDFL-FC are much better than the results of other methods.

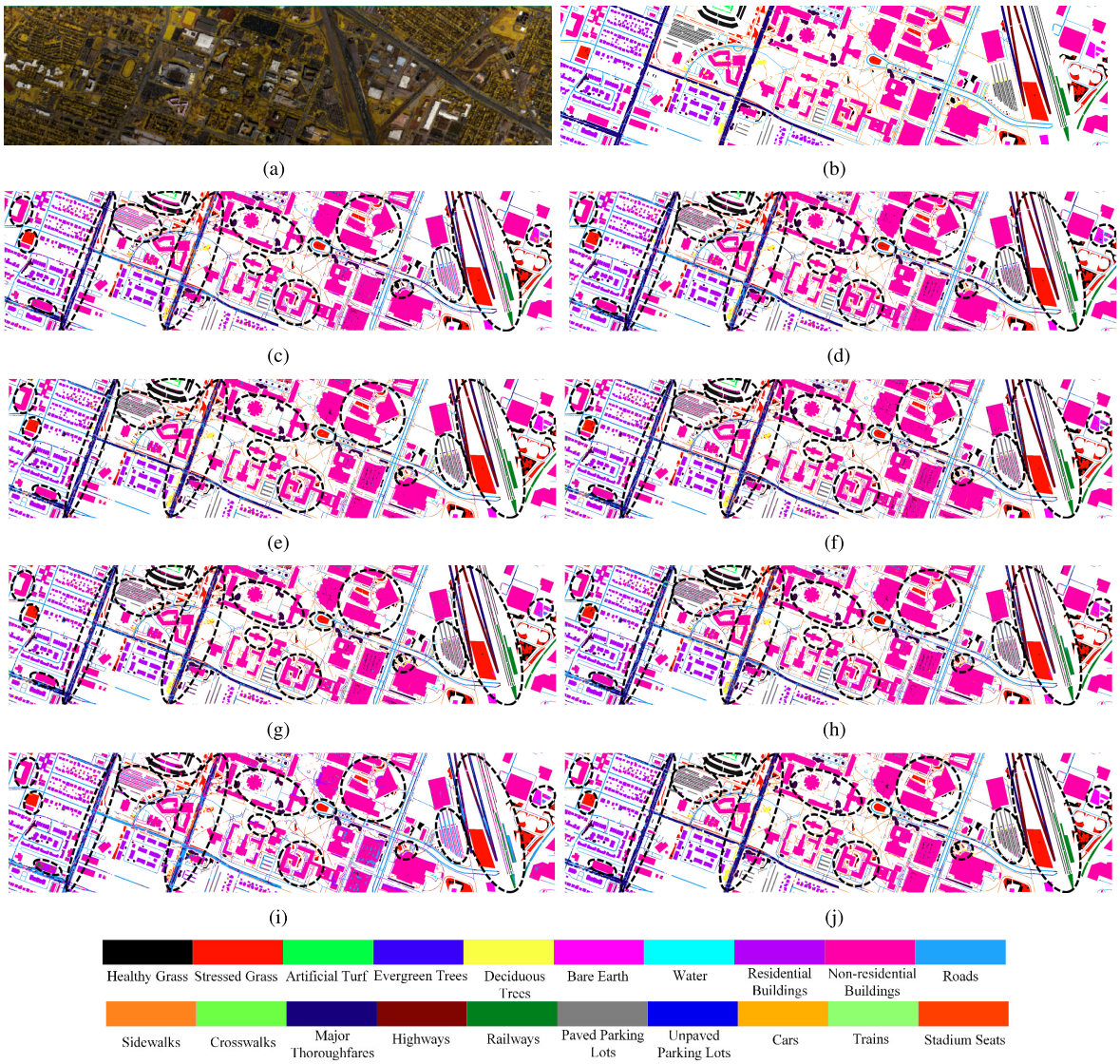

Fig. 4. Classification maps of the different methods with 5% training samples for the Houston data set. The regions are circled in red dotted line to show the differences. (a) False color. (b) Ground truth. (c) SVM-RBF. (d) ALDL. (e) PCA + CE. (f) WGAN + CE. (g) CAE + LGC. (h) SSCNN. (i) SESEMI. (j) SDFL-FC.

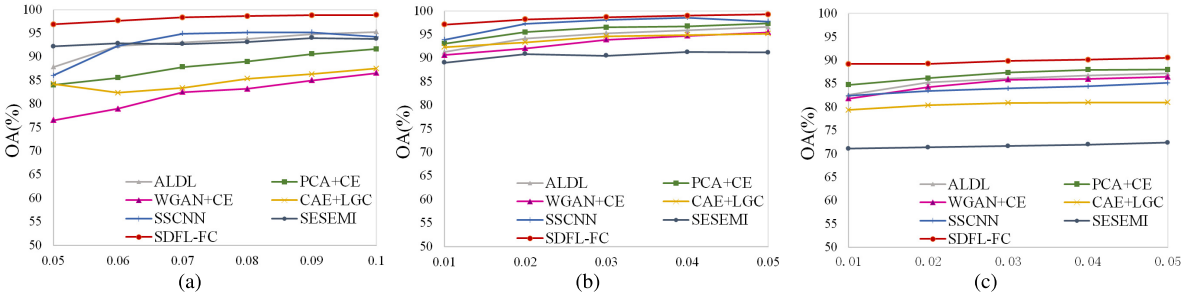

Fig. 5. Classification results of the semisupervised methods: ALDL, PCA + CE, WGAN + CE, CAE + LGC, SSCNN, SESEMI, and SDFL-FC with different percentages of labeled samples. (a) Indian Pines data set. (b) PaviaU data set. (c) Houston data set.

4) In Table VIII, we compare the OA performance of SDFL-FC with CAE + LGC to analyze the effects of training samples as the percentage of the training data set is changed: 5%, 10%, 15%, and 20%. With the number of training samples increases, the values of OA obtained by SDFL-FC increase. The SDFL-FC provides higher accuracy than CAE + LGC using different percentages of training samples.
5) In Fig. 6, we provide the classification maps on the whole image of SDFL-FC and SSCNN. It can be observed that both SDFL-FC and SSCNN can maintain the edge information in most cases, whereas other meth-

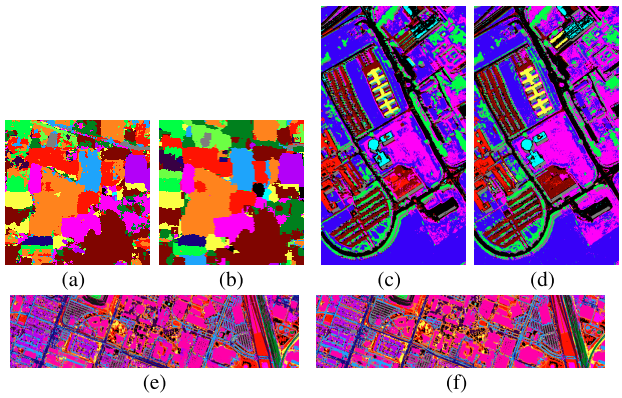

Fig. 6. Classification maps on the whole image of the SSCNN and SDFL-FC for the three data sets. (a) and (b) Maps of SSCNN and SDFL-FC for the Indian pines data set, respectively. (c) and (d) Maps of SSCNN and SDFL-FC for the PaviaU data set, respectively. (e) and (f) Maps of SSCNN and SDFL-FC for the Houston data set, respectively.

ods have the limited performances of HSI classification. Overall, it is concluded that our proposed SDFL-FC is more effective than SSCNN.

D. Parameter Analysis

In our method, five parameters need to be tuned, including λ_1, λ_2, η_1, η_2, and K. We discuss the influences of these parameters on the classification results. The parameters λ_1 and λ_2 are related to the terms of FCG and CE terms, respectively. The parameters η_1 and η_2 represent the balancing constant corresponding to the FCG of the first FCL and second FCL, respectively. The parameter K represents the number of superpixels in the superpixel segmentation. Fig. 7 shows the OA results of the tuned parameters. Every data set has its distinctive data structure. Thus, the parameters that achieve the best performances would be different for each data set. We acquire the initial values of parameters of λ_1, λ_2, η_1, and η_2 according to the magnitude among different parts. The best values of these parameters should refer to the classification results of the testing data.

For the Indian pines data set, the optimal parameters λ_1 and λ_2 are close to 0.01 and 0.01, respectively, which indicates that the FCG and CE regularizations are equally important. For the PaviaU data set, the best λ_1 and λ_2 are to be 0.1 and 10.0, respectively, which indicates that CE loss plays the more important role. For the Houston data set, the optimal parameters λ_1 and λ_2 are close to 1.0 and 0.01, respectively, which indicates that FCG regularization plays the more important role.

For the Indian pines data set, the optimal parameters η_1 and η_2 are close to 0.1 and 0.01, respectively. For the PaviaU data set, the best η_1 and η_2 are close to 0.01 and 0.01, respectively. The values of η_1 and η_2 are nearly the same, which proves that the influences of FCG of the first and second FCL are equally important for Indian pines and PaviaU data set, respectively. For the Houston data set, the best η_1 and η_2 are to be 10.0 and 0.01, which proves that the FCG of the first FCL is more important than that of the second FCL.

The best classification results for three data sets are achieved when the values of K are close to 300, 900, and 1100 (for each

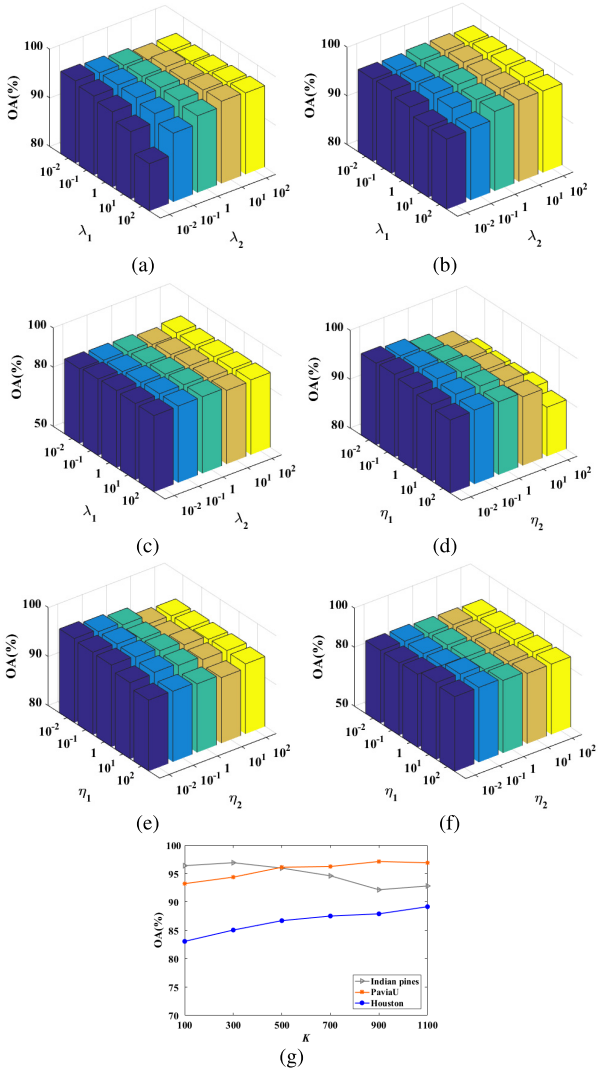

Fig. 7. Influences of the different parameter values on the HSI classification results. (a)–(c) Influences of λ_1 and λ_2 on the classification results for the Indian Pines, PaviaU, and Houston data sets, respectively. (d)–(f) Influences of η_1 and η_2 on the classification results for the Indian Pines, PaviaU, and Houston data sets, respectively. (g) Influences of different values of K on the classification results for Indian Pines, PaviaU, and Houston data sets.

segmented parts), respectively. The sizes of the three data sets are 145 × 145, 610 × 340, and 601 × 2384, respectively. For the PaviaU and Houston data sets, the image sizes are larger than the Indian pines data set, and the data structure is more complicated than the Indian pines data set. Therefore, the optimal K tends to increase as the size of HSI increases and the data structure becomes complex.

E. Discussion

The HSI classification results using different components in (10) are discussed. The effectiveness of the input sizes is also discussed.

1) Independent Analysis of the Regularization Terms: To verify the contribution of FCS and FCG term in the objective function (10), we compare the independent term and the joint terms for HSI classification. Since the CE is adopted to classify with the labeled samples, the CE loss cannot be removed.

TABLE IX
CLASSIFICATION RESULTS (OA%) WITH DIFFERENT COMPONENTS ON THE THREE DATA SETS

Dataset	Without FCS	Without FCG	SDFL-FC
Indian Pines	93.87 ± 0.22	94.71 ± 0.10	**96.92** ± 0.23
PaviaU	95.30 ± 0.12	95.46 ± 0.11	**97.13** ± 0.20
Houston	86.43 ± 0.16	84.18 ± 0.49	**89.17** ± 0.08

The best results are highlighted in bold.

TABLE X
CLASSIFICATION RESULTS (OA%) FROM DIFFERENT INPUT SIZES ON THE THREE DATA SETS

Dataset	1×1	3×3	5×5	7×7
Indian Pines	75.12 ± 0.67	91.61 ± 0.32	95.77 ± 0.13	**96.92** ± 0.27
PaviaU	92.84 ± 0.17	93.05 ± 0.21	**97.13** ± 0.20	96.59 ± 0.14
Houston	73.48 ± 0.53	86.38 ± 0.21	88.87 ± 0.06	**89.17** ± 0.08

The best results are highlighted in bold.

As follows, the objective function can be divided into two learning models.

a) Without FCS: It does not consider the FCS term and is defined with the following equations:

$$\lambda_1\Theta_{\mathrm{FCG}}+\lambda_2\Theta_{\mathrm{CE}} = \lambda_1\left[\sum_{l_2=1}^{L_2}\left[\eta_{l_2}\sum_{i=1}^{n}\left\|\mathbf{f}_i^{(l_2)}-\mathbf{u}_i^{(l_2)}\right\|_2^2\right]\right] + \lambda_2\left[-\frac{1}{s}\sum_{i=1}^{s}\sum_{j=1}^{C}I(j)\log\left(p\left(\hat{y}_i=j|\mathbf{f}^{(L_2)}\right)\right)\right]. \tag{14}$$

b) Without FCG: It does not consider the FCG term and is defined with the following equations:

$$\Theta_{\mathrm{FCS}}+\lambda_2\Theta_{\mathrm{CE}} = E_{\mathbf{X}\sim p(\mathbf{X})}[D(\mathbf{X})]-E_{\mathbf{h}_G^{(M_1)}\sim p\left(\mathbf{h}_G^{(M_1)}\right)}\left[D\left(\mathbf{h}_G^{(M_1)}\right)\right] + \lambda_2\left[-\frac{1}{s}\sum_{i=1}^{s}\sum_{j=1}^{C}I(j)\log\left(p\left(\hat{y}_i=j|\mathbf{f}^{(L_2)}\right)\right)\right]. \tag{15}$$

From Table IX, we have the following observations. First, for the Indian pines and PaviaU data sets, the method without FCG performs better than the method without FCS. The reason is that FCS reconstructs the original data to minimize the loss between the reconstructed data and the original data. The reconstructed data as the augmented samples also enhance the classification performances. For the Houston data set, the method without FCS performs better than the method without FCG. For the Houston data set, the number of samples of different categories is not balanced. For example, if taking 1% labeled samples, the number of labeled samples for Unpaved Parking Lots is only 1, whereas the number of labeled samples for Nonresidential Buildings is 2237. The FCG can help the samples with a small number of tags get more clustering information because FCG enforces the features within a superpixel to have similar characteristics. Therefore, compared to the method without FCG, the method without FCS can achieve better classification results for the Houston data set. Second, both the FCS and FCG designs are considered in the SDFL-FC, so the SDFL-FC outperforms FCS and FCG designs.

2) Effectiveness of the Input Sizes: To evaluate the effectiveness of the input sizes in the proposed method, we compare the SDFL-FC with different input sizes, which consists of 1×1, 3×3, 5×5, and 7×7. In Table X, for the three data sets, the OAs of input patches 5×5 and 7×7 outperform that of input patches 1×1 and 3×3. The reason is that the networks with 1×1 and 3×3 input patches fail to exploit the spatial information effectively.

V. CONCLUSION

In this article, a novel semisupervised method called SDFL-FC is proposed to reduce the dependence of the labeled samples, in which the FCS, FCG, and CE loss are integrated into a unified objective function. The FCS is achieved by GAN regularization, which can reconstruct the original data from extracted features. It is achieved via minimizing the differences between the reconstructed data and the original data. The FCG is based on the assumption that that the features of group pixels should have similar characteristics within a superpixel, which is also embedded into each FCL. In this way, SDFL-FC can extract more representative and discriminative features in a semisupervised way to mitigate the need for large amount of labeled samples. The SDFL-FC is optimized using a customized iterative optimization algorithm. Experimental results on three HSI data sets demonstrate the effectiveness of the SDFL-FC. In the future, the SDFL-FC will be integrated into other deep learning frameworks, such as graph convolutional networks, to improve the performance of the HSI classification.

REFERENCES

[1] C. Chang, *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*, vol. 1. Dordrecht, The Netherlands: Kluwer, 2003.
[2] 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.
[3] K.-H. Liu, Y.-Y. Lin, and C.-S. Chen, "Linear spectral mixture analysis via multiple-kernel learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 2254–2269, Apr. 2015.
[4] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "CoSpace: Common subspace learning from hyperspectral-multispectral correspondences," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4349–4359, Jul. 2019.
[5] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.
[6] N. Falco, J. A. Benediktsson, and L. Bruzzone, "Spectral and spatial classification of hyperspectral images based on ICA and reduced morphological attribute profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 11, pp. 6223–6240, Nov. 2015.
[7] J. Wang, F. Wang, C. Zhang, H. C. Shen, and L. Quan, "Linear neighborhood propagation and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 9, pp. 1600–1615, Sep. 2009.
[8] T. V. Bandos, L. Bruzzone, and G. Camps-Valls, "Classification of hyperspectral images with regularized linear discriminant analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 3, pp. 862–873, Mar. 2009.
[9] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4085–4098, Nov. 2010.

[10] 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.

[11] A. Villa, J. A. Benediktsson, J. Chanussot, and C. Jutten, "Hyperspectral image classification with independent component discriminant analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 12, pp. 4865–4876, Dec. 2011.

[12] Y. Wang *et al.*, "Self-supervised low-rank representation (SSLRR) for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 5658–5672, Oct. 2018.

[13] J. Mei *et al.*, "PSASL: Pixel-level and superpixel-level aware subspace learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4278–4293, Jul. 2019.

[14] D. Hong, N. Yokoya, J. Chanussot, J. Xu, and X. X. Zhu, "Joint and progressive subspace analysis (JPSA) with spatial-spectral manifold alignment for semisupervised hyperspectral dimensionality reduction," *IEEE Trans. Cybern.*, early access, Nov. 11, 2020, doi: 10.1109/TCYB.2020.3028931.

[15] R. Hang, Z. Li, P. Ghamisi, D. Hong, G. Xia, and Q. Liu, "Classification of hyperspectral and LiDAR data using coupled CNNs," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 7, pp. 4939–4950, Jul. 2020.

[16] R. Hang, Q. Liu, D. Hong, and P. Ghamisi, "Cascaded recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5384–5394, Aug. 2019.

[17] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

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

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

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

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

[22] Y. Li, H. Zhang, and Q. Shen, "Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sens.*, vol. 9, no. 1, p. 67, Jan. 2017.

[23] 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.

[24] 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.

[25] L. Mou, P. Ghamisi, and X. X. Zhu, "Unsupervised spectral–spatial feature learning via deep residual conv-deconv network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 391–406, Jan. 2018.

[26] D. Hong, N. Yokoya, N. Ge, J. Chanussot, and X. X. Zhu, "Learnable manifold alignment (LeMA): A semi-supervised cross-modality learning framework for land cover and land use classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 147, pp. 193–205, Jan. 2019.

[27] Y. Cao *et al.*, "SLCRF: Subspace learning with conditional random field for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, early access, Sep. 11, 2020, doi: 10.1109/TGRS.2020.3011429.

[28] Y. Wang *et al.*, "Self-supervised feature learning with CRF embedding for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 5, pp. 2628–2642, May 2019.

[29] Y. Luo, J. Pan, S. Fan, Z. Du, and G. Zhang, "Retinal image classification by self-supervised fuzzy clustering network," *IEEE Access*, vol. 8, pp. 92352–92362, 2020.

[30] J. Zhang *et al.*, "Self-supervised convolutional subspace clustering network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5468–5477.

[31] D. Chen, Y. Chen, Y. Li, F. Mao, Y. He, and H. Xue, "Self-supervised learning for few-shot image classification," 2019, *arXiv:1911.06045*. [Online]. Available: http://arxiv.org/abs/1911.06045

[32] Y. Zhan, D. Hu, Y. Wang, and X. Yu, "Semisupervised hyperspectral image classification based on generative adversarial networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 212–216, Feb. 2018.

[33] H. Wu and S. Prasad, "Semi-supervised deep learning using pseudo labels for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1259–1270, Mar. 2018.

[34] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[35] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 105–114.

[36] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2242–2251.

[37] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, Sydney, NSW, Australia, Aug. 2017, pp. 2642–2651.

[38] Y. Wang, L. Zhang, F. Nie, X. Li, Z. Chen, and F. Wang, "WeGAN: Deep image hashing with weighted generative adversarial networks," *IEEE Trans. Multimedia*, vol. 22, no. 6, pp. 1458–1469, Jun. 2020.

[39] Y. Cao *et al.*, "DML-GANR: Deep metric learning with generative adversarial network regularization for high spatial resolution remote sensing image retrieval," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 12, pp. 8888–8904, Dec. 2020.

[40] 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 2019.

[41] R. Hang, F. Zhou, Q. Liu, and P. Ghamisi, "Classification of hyperspectral images via multitask generative adversarial networks," *IEEE Trans. Geosci. Remote Sens.*, early access, Jun. 25, 2020, doi: 10.1109/TGRS.2020.3003341.

[42] R. Nagar and S. Raman, "SymmSLIC: Symmetry aware superpixel segmentation and its applications," 2018, *arXiv:1805.09232*. [Online]. Available: http://arxiv.org/abs/1805.09232

[43] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *Proc. CVPR*, Jun. 2011, pp. 2097–2104.

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

[45] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[46] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "TurboPixels: Fast superpixels using geometric flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2290–2297, Dec. 2009.

[47] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "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.

[48] L. Fang, S. Li, X. 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.

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

[50] S. Zhang, S. Li, W. Fu, and L. Fang, "Multiscale superpixel-based sparse representation for hyperspectral image classification," *Remote Sens.*, vol. 9, no. 2, p. 139, Feb. 2017.

[51] J. Jiang, J. Ma, C. Chen, Z. Wang, Z. Cai, and L. Wang, "SuperPCA: A superpixelwise PCA approach for unsupervised feature extraction of hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4581–4593, Aug. 2018.

[52] C. Shi and C.-M. Pun, "Multiscale superpixel-based hyperspectral image classification using recurrent neural networks with stacked autoencoders," *IEEE Trans. Multimedia*, vol. 22, no. 2, pp. 487–501, Feb. 2020.

[53] S. Jia, X. Deng, J. Zhu, M. Xu, J. Zhou, and X. Jia, "Collaborative representation-based multiscale superpixel fusion for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7770–7784, Oct. 2019.

[54] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 214–223.

[55] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org
[56] D. Wang and Y. Shang, "A new active labeling method for deep learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2014, pp. 112–119.
[57] G. Abdi, F. Samadzadegan, and P. Reinartz, "Spectral-spatial feature learning for hyperspectral imagery classification using deep stacked sparse autoencoder," *Proc. SPIE*, vol. 11, no. 4, Aug. 2017, Art. no. 042604.
[58] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 16. Cambridge, MA, USA: MIT Press, 2003, pp. 321–328.
[59] B. Liu, X. Yu, P. Zhang, X. Tan, A. Yu, and Z. Xue, "A semi-supervised convolutional neural network for hyperspectral image classification," *Remote Sens. Lett.*, vol. 8, no. 9, pp. 839–848, Sep. 2017.
[60] P. V. Tran, "Exploring self-supervised regularization for supervised and semi-supervised learning," 2019, *arXiv:1906.10343*. [Online]. Available: http://arxiv.org/abs/1906.10343

Yun Cao is pursuing the Ph.D. degree with the School of Land Science and Technology, China University of Geosciences, Beijing, China.

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

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

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

Junhuan Peng received the Ph.D. degree in geodesy from Wuhan University, Wuhan, China, in 2003.

He is a Professor with the School of Land Science and Technology, China University of Geosciences (Beijing), Beijing, China. His research interests include temporal–spatial data analysis, surveying adjustment, applied statistics, and their associated application in surveying engineering, image geodesy, remote sensing, and satellite geodesy.

Chunping Qiu received the B.Sc. and M.Sc. degrees in photogrammetry and remote sensing from the Zhengzhou Institute of Surveying and Mapping, China, in 2013 and 2016, respectively. She is pursuing the Ph.D. degree with the Technical University of Munich (TUM), Munich, Germany.

In 2019, she was a Guest Researcher with the Telecommunications and Remote Sensing Laboratory, University of Pavia, Pavia, Italy. Her research interest is focused on deep learning and remote sensing data fusion with an application on urban land cover classification and analysis.

Lei Ding received the B.S. degree in measurement and control engineering and the M.S. degree in photogrammetry and remote sensing from the University of Information Engineering, Zhengzhou, China, in 2013 and 2016, respectively. He is pursuing the Ph.D. degree with the Remote Sensing Laboratory, Department of Information Engineering and Computer Science, University of Trento, Trento, Italy.

His research interests are related to remote sensing image processing and machine learning.

Xiao Xiang Zhu (Fellow, IEEE) received the M.Sc., Dr.-Ing., and Habilitation degrees in signal processing from the Technical University of Munich (TUM), Munich, Germany, in 2008, 2011, and 2013, respectively.

Since 2019, she has been a Co-Coordinator with the Munich Data Science Research School. Since 2019, she also heads the Helmholtz Artificial Intelligence—Research Field Aeronautics, Space and Transport. Since May 2020, she has been the Director of the International Future AI lab AI4EO—Artificial Intelligence for Earth Observation: Reasoning, Uncertainties, Ethics and Beyond, Munich, Germany. Since October 2020, she also serves on the board of directors of the Munich Data Science Institute (MDSI), TUM. She was a Guest Scientist or Visiting Professor with the Italian National Research Council (CNR-IREA), Naples, Italy, Fudan University, Shanghai, China, the University of Tokyo, Tokyo, Japan, and the University of California at Los Angeles, Los Angeles, CA, USA, in 2009, 2014, 2015, and 2016, respectively. She is the Professor for Data Science in Earth Observation, TUM, and also the Head of the Department EO Data Science, Remote Sensing Technology Institute, German Aerospace Center (DLR), Weßling, Germany. Her main research interests are remote sensing and earth observation, signal processing, machine learning, and data science, with a special application focus on global urban mapping.

Dr. Zhu is a member of the young academy (Junge Akademie/Junges Kolleg) at the Berlin-Brandenburg Academy of Sciences and Humanities and the German National Academy of Sciences Leopoldina and the Bavarian Academy of Sciences and Humanities. She is an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING.