# Deep Multiview Learning for Hyperspectral Image Classification

Bing Liu, Anzhu Yu, Xuchu Yu, Ruirui Wang, Kuiliang Gao, and Wenyue Guo

***Abstract*— Recently, the field of hyperspectral image (HSI) classification is dominated by deep learning-based methods. However, training deep learning models usually needs a large number of labeled samples to optimize thousands of parameters. In this article, a deep multiview learning method is proposed to deal with the small sample problem of HSI. First, two views of an HSI scene are constructed by applying principal component analysis to different bands. Second, a deep residual network is designed to embed the different views of a sample to a latent space. The designed deep residual network is trained by maximizing agreement between differently augmented views of the same data sample via a contrastive loss in the latent space. Note that the training procedure of the designed deep residual network does not use labeled information. Therefore, the proposed method belongs to the category of unsupervised learning, which could alleviate the lack of labeled training samples. Finally, a conventional machine learning method (e.g., support vector machine) is used to complete the classification task in the learned latent space. To demonstrate the effectiveness of the proposed method, extensive experiments are carried on four widely used hyperspectral data sets. The experimental results demonstrate that the proposed method could improve the classification accuracy with small samples.**

***Index Terms*— Deep learning, hyperspectral image (HSI) classification, multiview learning, small samples.**

## I. Introduction

HYPERSPECTRAL image (HSI) classification involves assigning a category tag to each sample according to its spectral information and spatial information [1], [2]. In this task, one of the greatest challenges is determining what types of features should be used as the input of a classifier. In HSI, each pixel can be regarded as a high-dimensional vector whose entries correspond to the spectral reflectance in a specific wavelength. Naturally, traditional classification methods focus on exploring the spectral signatures for the classification tasks. Thus, support vector machines (SVMs) [3], extreme learning machine (ELM) [4], random forest (RF) [5], sparse representation [6], and other pixelwise classifiers are used for HSI classification tasks. A disadvantage of these pixelwise classifiers is that it could not consider spatial information in the classification procedure. In this context, feature extraction methods that could include spatial information are introduced to improve the classification performance, e.g., Gabor filters [7], local binary patterns [8], morphological profiles [9]–[11], and wavelet [12]. A major limitation of these spatial features is that they require a great deal of tuning to get them to work well on a particular data set.

Deep learning can learn to extract features for classification from data without the artificial design of feature extraction rules. Thus, deep learning has gained more attention in the field of HSI classification [13]. Deep learning-based pixelwise classifiers for HSI include stacked autoencoder (SAE) [14], 1-D convolution neural network (1D-CNN) [15], deep belief network (DBN) [16], and recurrent neural network (RNN) [17]. To utilize spatial–spectral features to improve the classification accuracy, 2-D-CNN [18]–[22] and 3-D-CNN [23]–[26] are also designed for HSI classification. Moreover, a multiscale dense network [24] is designed to make full use of different scale information in the network structure and combine scale information throughout the network, which improves the training speed and accuracy for HSI classification. A deep learning ensemble framework [27] based on the integration of deep learning model and random subspace-based ensemble learning is proposed to further boost the classification performance. A cascaded RNN model is designed to fully explore the redundant and complementary information of the high-dimensional spectral signature in [28]. In addition to the traditional deep learning models (SAE, CNN, and RNN), some variants of deep learning are also applied to HSI classification, e.g., deep multigrained cascade forest [29] and capsule network [30].

The abovementioned supervised deep learning classifiers could achieve a higher classification accuracy than that of the traditional methods. However, the shortage of training samples remains one of the main obstacles in applying deep neural networks to the HSI classification tasks. Unsupervised learning can learn useful information from unlabeled data for subsequent tasks. Therefore, researchers have conducted some valuable explorations of unsupervised learning for HSI. Autoencoder is a common unsupervised framework and has been widely used for HSI [31]–[33]. Meanwhile, some improved methods of autoencoder are used in HSI classification, e.g., deep residual conv–deconv network [34] and 3-D convolutional autoencoder [35]. To further improve the classification accuracy, a self-taught learning framework

Manuscript received May 23, 2020; revised August 6, 2020 and September 23, 2020; accepted October 23, 2020. This work was supported by the National Natural Science Foundation of China under Grant 41201477. *(Corresponding author: Bing Liu.)*

Bing Liu, Anzhu Yu, Xuchu Yu, Kuiliang Gao, and Wenyue Guo are with PLA Strategic Support Force Information Engineering University, Zhengzhou 450001, China (e-mail: liubing220524@126.com).

Ruirui Wang is with the Institute of Surveying Mapping and Geo Information of Henan, Zhengzhou 450006, China.

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

Digital Object Identifier 10.1109/TGRS.2020.3034133

is designed in [36]. In addition, the generative adversarial network [37] is introduced to train a deep learning-based feature extractor in an unsupervised manner, which improves the results of unsupervised training. The majority of published unsupervised feature learning frameworks for HSI involve complex training procedure, and they could not deal with small sample problems.

In real-world applications, data are usually collected from diverse domains or obtained from various feature extractors and exhibit heterogeneous properties, as variables of each data example can be naturally partitioned into groups. Each variable group is referred to as a particular view. Multiview learning, aiming to learn one function to model each view and jointly optimize all the functions to improve the generalization performance, has made great progress and developments in recent years [38]. Notably, deep multiview learning has achieved great success in image classification and recognition. For example, a contrastive coding scheme is proposed in [39] and achieves state-of-the-art results on the Imagenet benchmark. Momentum contrast learning [40] has obtained positive results of unsupervised learning in a variety of computer vision tasks and data sets.

Bands of HSI can be naturally considered as different views of a scene, as different bands reflect the different properties of ground objects. Motivated by this and the recent success of multiview learning, a novel deep multiview learning method is proposed to deal with the classification issue of HSI with a small sample. Especially, a deep residual network, a variant of Resnet50 [41], is designed to map the different views to a latent space. The designed deep residual network is then trained by maximizing agreement between differently augmented views of the same data sample via a contrastive loss in the feature space. Unsupervised training manner of the proposed method ensures enough training data and alleviates the problem of lacking labeled training samples. More importantly, the learned view-invariant features could greatly improve the small sample classification accuracy of HSI.

The main contributions of this article can be summarized as follows.

1) A deep multiview learning method is proposed for HSI classification, which makes the deep network learn to extract view-invariant features. Extensive experiments and analysis on four benchmarks demonstrate the effectiveness of the proposed deep multiview learning method.
2) A deep residual network with 51 layers is designed to extract features from different views of HSI. Note that the 51-layer residual network is larger than the existing networks in the field of HSI classification. The depth of the designed network ensures the generalization ability of the learned features.
3) Two data augmentations are adopted to improve the unsupervised learning results. Experiments show that data augmentation can further improve the classification accuracy of the proposed method.

The remainder of this article is organized as follows. In Section II, the proposed classification framework is described in detail. In Section III, the experimental results and corresponding analysis are presented. In Section IV, this article concludes with some discussion.

## II. Proposed Method

In this section, the proposed method will be described in detail. First, we will give the architecture of deep multiview learning.

### A. Deep Multiview Learning

The traditional unsupervised loss function [e.g., mean square error (MSE)] calculates the distance between the predicted value and the original input. However, it is difficult to guarantee the effectiveness of the features only by optimizing the reconstruction error. In order to make the learned features more effective for classification tasks, we optimize the contrastive loss function to make the features from different views of the same sample consistent. This makes the features of the same class aggregate with each other, and the features of different classes are far away from each other. Therefore, the features obtained by optimizing the contrastive loss function of different views could effectively improve the classification accuracy. We use a deep CNN as the base feature extractor. We call this proposed method deep multiview learning.

The forward propagation of the proposed deep multiview learning illustrated in Fig. 1 mainly includes three steps: constructing two views of a sample, input each view into the networks $f(\cdot)$ and $g(\cdot)$ to generate the latent feature $\mathbf{h}$ and the output feature $\mathbf{z}$, and calculating contrastive loss function according to the output feature $\mathbf{z}$. Note that $f(\cdot)$ is a Resnet50 without the classification layer, and $g(\cdot)$ is a fully connected network to reduce the dimensions of output features.

A large number of studies show that considering the spatial neighborhood information in a neural network can improve the classification performance. Thus, a $m \times m \times b$ cube is used as the feature of a sample, where $m$ is the size of the neighborhood and $b$ is the number of bands. In fact, each band of an HSI could be treated as a view of a scene, as different bands reflect the different properties of ground objects. On the one hand, there is a strong correlation between the adjacent bands of HSI. On the other hand, there are many bands in HSI, which means a large number of views. A large number of views complicate the process of deep multiview learning. Taking these factors into account, we design a simple method to construct two views of an HSI. As shown in Fig. 1, the bands of HSI are divided into two groups. The first group band is transformed by PCA to generate the first view. Here, the first three principal components are taken as the first view. Similarly, the second group band is used to generate the second view. Taking the Indiana Pines data set as an example, there are 200 bands in this HSI. The first 100 bands are used to generate the first view. The remaining bands are used to generate the second view.

Let $\mathbf{x}_1^{(i)}$ and $\mathbf{x}_2^{(i)}$ represent the first view and the second view of a sample, respectively. The inputs $\mathbf{x}_1^{(i)}$ and $\mathbf{x}_2^{(i)}$ are mapped to latent vectors $\mathbf{h}_1^{(i)}$ and $\mathbf{h}_2^{(i)}$ through a nonlinear function $f(\cdot)$ parameterized by a set of weights. This embedding is defined such that the input view could be abstracted as a

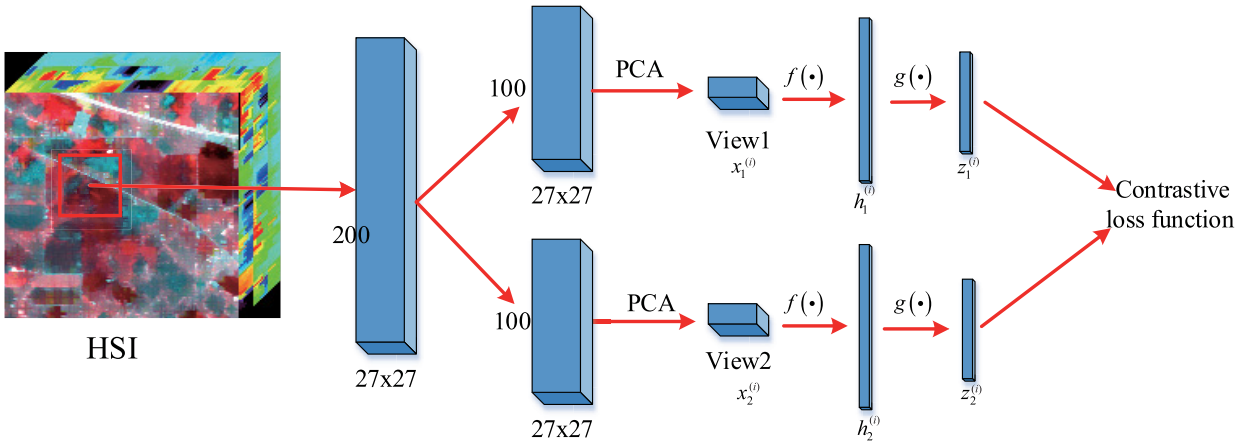



Fig. 1. Pipeline of the deep multiview learning for HSI.

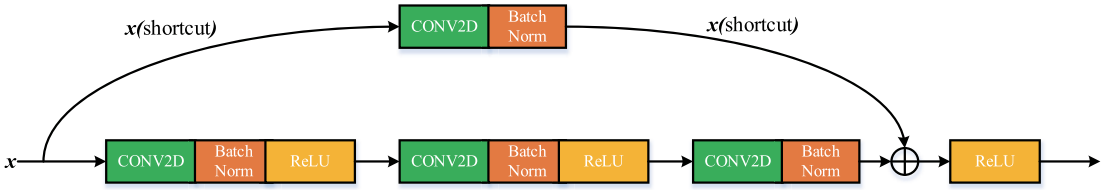



Fig. 2. Illustration of a standard residual block. CONV2D denotes a convolutional layer, BatchNorm denotes a batch normalization layer, and ReLU denotes a ReLU layer.

high-level feature vector. A multilayer perceptron with two fully connected layers $g(\cdot)$ is used to transform the latent features $\mathbf{h}_1^{(i)}$ and $\mathbf{h}_2^{(i)}$ into $\mathbf{z}_1^{(i)}$ and $\mathbf{z}_2^{(i)}$. Then, we define the contrastive loss on $\mathbf{z}_1^{(i)}$ and $\mathbf{z}_2^{(i)}$ rather than $\mathbf{h}_1^{(i)}$ and $\mathbf{h}_2^{(i)}$.

In the training procedure, a minibatch of $N$ samples is randomly selected as the training samples of one parameter update. The contrastive prediction task is defined on pairs of samples derived from the minibatch, resulting in $2N$ views. In $2N$ views, two views from the same sample are taken as a positive pair, and two views from different samples are taken as negative pairs. The contrastive loss is defined as

$$\zeta_{i,j} = -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j))}{\sum_{k=1}^{2N} l_{[k\neq i]} \text{sim}(\mathbf{z}_i, \mathbf{z}_k)} \tag{1}$$

where $l_{[k\neq i]}$ is an indicator function evaluating to 1 if $k \neq i$, and $\text{sim}(\mathbf{z}_i, \mathbf{z}_j) = \mathbf{z}_i^T \mathbf{z}_j / \|\mathbf{z}_i\| \|\mathbf{z}_j\|$ denotes the cosine similarity between two vectors $\mathbf{z}_i$ and $\mathbf{z}_j$. In a minibatch, the total loss is computed across all positive pairs. Our goal is to learn representations that capture information shared between multiple sensory views without human supervision. This loss is defined according to the similarity between views, which means that it does not need any human supervision information. In other words, it is an unsupervised method. More importantly, this loss ensures that the network learns to extract view-invariant features, which is a useful representation of the samples.

When the number of views is more than 3, the features of different views are combined in pairs. The contrastive loss is calculated respectively for the features of two combined views. Then, the sum of the contrastive loss calculated from different combined views is calculated as the final loss function.

### *B. Deep Residual Network*

The $f(\cdot)$ that extracts representation vectors from views could be various network architecture. Recent studies [41], [42] reveal that the classification performance benefits from bigger models. Residual learning has become a common method to improve the accuracy in natural image recognition and HSI classification [43], [44]. Therefore, a variant of Resnet50 [41] is used as the network that extracts representation vectors from views. Deep residual learning could make training deep network easier. Thus, it has been widely used in a variety of classification tasks. As shown in Fig. 2, the core idea of deep residual learning is to introduce a shortcut connection, which directly skips one or more layers. The deep residual network is based on residual block. As shown in Fig. 2, there are three convolutional layers in a standard residual block. Each convolution layer is followed by a batch normalization layer (BatchNorm) and a ReLU layer. The original input is then added with the output of the last convolutional layer as the output of a residual block, which is a shortcut operation. The output of a residual block is activated by a ReLU layer as the input of the later residual block. Note that the dimension of the input data may be different from the output dimension of the last convolutional layer of the residual block. When the dimension of the input and the last convolutional layer of the residual block is different, a $1 \times 1$ convolutional layer followed by a batch normalization layer is applied to the input to conduct a resample operation in order to ensure consistent data dimensions.

The original Resnet50 consists of one convolutional layer, 16 residual blocks, two pooling layers, and one classification

TABLE I
DETAILS OF DEEP RESIDUAL NETWORK USED AS THE BASE FEATURE EXTRACTOR

| | Stage1 | MaxPool | Stage2 | Stage3 | Stage4 | Stage5 | AVGPool |
|---|---|---|---|---|---|---|---|
| **Parameters** | $3\times3,\ 64$ | $2\times2$ max pooling | $\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$ | $\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times4$ | $\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times6$ | $\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times3$ | global average pooling |

**Algorithm 1** Minibatch Training Procedure

**Require:** Batch size $N$, deep residual network $f(\cdot)$, fully connected network $g(\cdot)$, data augmentation operation $\tau$
**for** sampled minibatch $\{\mathbf{x}_i\}_{i=1}^{N}$ **do**
  **for** $i$ in $\{1,\ldots,N\}$ **do**
    Draw two augmentation operations $\tau_1\sim\tau,\ \tau_2\sim\tau$
    # The first view
    $\mathbf{x}_{2k-1}=\tau_1(\mathbf{x}_1^{(i)})$
    $\mathbf{h}_{2k-1}=f(\mathbf{x}_{2k-1})$
    $\mathbf{z}_{2k-1}=g(\mathbf{h}_{2k-1)})$
    # The second view
    $\mathbf{x}_{2k}=\tau_1(\mathbf{x}_2^{(i)})$
    $\mathbf{h}_{2k}=f(\mathbf{x}_{2k})$
    $\mathbf{z}_{2k}=g(\mathbf{h}_{2k)})$
  **end for**
  **for** $i$ in $\{1,\ldots,2N\}$, $j$ in $\{1,\ldots,2N\}$ **do**
    $s_{i,j}=\mathbf{z}_i^T\mathbf{z}_j/\|\mathbf{z}_i\|\|\mathbf{z}_j\|$
  **end for**
  calculate the loss $\zeta_{i,j}=-\log\frac{\exp(s_{i,j})}{\sum_{k=1}^{2N}I_{[k\neq i]}\exp(s_{i,k})}$
  calculate the total loss $\zeta=\frac{1}{2N}\sum_{i=1}^{2N}\left[\zeta_{2i-1,2i}+\zeta_{2i,2i-1}\right]$
  update networks $f(\cdot)$ and $g(\cdot)$ to minimize $\zeta$
**end for**

layer (fully connected layer). The purpose of the training network is not to classify but to learn the representations of views. Therefore, the Resnet50 without classification layer is used as the base feature extractor $f(\cdot)$. As shown in Fig. 3, the Resnet50 $f(\cdot)$ actually consists of 49 convolutional layers, $1+3\times(3+4+6+3)=49$. The details of the deep residual network used as the base feature extractor $f(\cdot)$ are shown in Table I. Note that the output of the deep residual network is a 2048 vector. Subsequently, a multilayer perceptron with two fully connected layers $g(\cdot)$ is applied to the output vector of the Resnet50 $f(\cdot)$ to reduce the dimensions of output features. In fact, the network used to extract features includes 49 convolutional layers and two fully connected layers.

### C. Training and Testing Procedure

The contrastive loss is defined on the outputs of the multilayer perceptron. More specifically, the pseudocode for a training minibatch procedure is given in Algorithm 1.

Data augmentation is a common technique that can effectively improve the generalization ability of a model and has been widely used in supervised deep learning. However, data augmentation has not been used in the contrastive prediction task. Consequently, two data augmentations (random cropping and random Gaussian blur) are used to improve the robustness of network training.

In the testing procedure, the deep residual network trained on a specific HSI is used as a feature extractor. Then, all samples of this specific HSI pass through the deep residual network to output the corresponding feature vectors. So far, conventional machine learning methods could be applied to the extracted features to complete the classification task. Here, an SVM classifier and an RF classifier are used.

## III. EXPERIMENTAL RESULTS AND ANALYSIS

The proposed method is implemented by the PyTorch library. The results are generated on a PC equipped with an Intel Core i7-9750H with 2.6 GHz and an Nvidia GeForce RTX 2070M. The PC's memory is 16G.

### A. Data Sets

To demonstrate the effectiveness of the proposed method, the University of Pavia data set, the Indiana Pines data set, the Salinas data set, and the Houston data set are used to conduct classification experiments. In the feature learning procedure, 50% unlabeled samples are used as the training data, and the remaining 50% samples are used as the testing data. In each data set, five labeled samples per class are randomly selected as the training samples for the supervised classifier in the classification procedure.

The University of Pavia data set is acquired by the ROSIS sensor during a flight campaign over Pavia, Northern Italy. It has 103 spectral bands coverage from 0.43 to 0.86 $\mu$m and a geometric resolution of 1.3 m. The image size is $610\times340$ pixels. In this data set, 42 776 pixels with nine classes are labeled. Labels, the number of labeled training samples, and the number of testing samples are listed in Table II.

The second data set is the Indiana Pines data set. This data set is gathered by Airborne Visible Infrared Imaging Spectrometer (AVIRIS) sensor over the Indian Pines test site in Northwestern Indiana and consists of $145\times145$ pixels and 224 spectral reflectance bands in the wavelength range 0.4–2.5 $\mu$m; 24 bands covering the region of water absorption are removed, resulting in 200 bands for classification. This scene contains two-third agriculture and one-third forest or other natural perennial vegetation; 10 249 pixels with 16 classes are labeled. Labels, the number of labeled training samples, and the number of testing samples are listed in Table III.

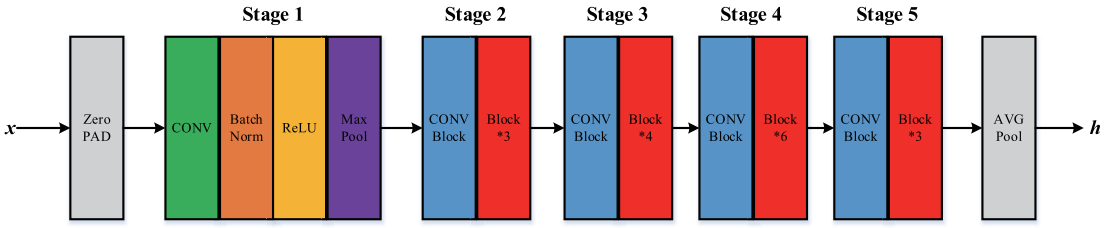



Fig. 3. Illustration of the deep residual network used as the base feature extractor. ZeroPAD denotes padding operation, CONV denotes a convolutional layer, BatchNorm denotes a batch normalization layer, ReLU denotes a ReLU layer, CONV Block denotes a residual block consisting of three convolutional layers, and MaxPool and AVGPool denote the max-pooling layer and the global average pooling layer, respectively. Block*$n$ represents the residual block $n$ times.

TABLE II

LABELS, THE NUMBER OF LABELED TRAINING SAMPLES, AND THE NUMBER OF TESTING SAMPLES FOR THE UNIVERSITY OF PAVIA DATA SET

| No. | Class | Training | Testing |
|---|---|---|---|
| 1 | Asphalt | 5 | 6631 |
| 2 | Meadows | 5 | 18649 |
| 3 | Gravel | 5 | 2099 |
| 4 | Trees | 5 | 3064 |
| 5 | Sheets | 5 | 1345 |
| 6 | Bare Soil | 5 | 5029 |
| 7 | Bitumen | 5 | 1330 |
| 8 | Bricks | 5 | 3682 |
| 9 | Shadows | 5 | 947 |
| | Total | 45 | 42776 |

TABLE III

LABELS, THE NUMBER OF LABELED TRAINING SAMPLES, AND THE NUMBER OF TESTING SAMPLES FOR THE INDIANA PINES DATA SET

| No. | Class | Training | Testing |
|---|---|---|---|
| 1 | Alfalfa | 5 | 46 |
| 2 | Corn-notill | 5 | 1428 |
| 3 | Corn-mintill | 5 | 830 |
| 4 | Corn | 5 | 237 |
| 5 | Grass-pasture | 5 | 483 |
| 6 | Grass-trees | 5 | 730 |
| 7 | Grass-pasture-mowed | 5 | 28 |
| 8 | Hay-windrowed | 5 | 478 |
| 9 | Oats | 5 | 20 |
| 10 | Soybean-notill | 5 | 972 |
| 11 | Soybean-mintill | 5 | 2455 |
| 12 | Soybean-clean | 5 | 593 |
| 13 | Wheat | 5 | 205 |
| 14 | Woods | 5 | 1265 |
| 15 | Buildings-Grass-Trees-Drives | 5 | 386 |
| 16 | Stone-Steel-Towers | 5 | 93 |
| | Total | 80 | 10249 |

The third data set is the Salinas data set gathered by the AVIRIS sensor in Northwestern Indiana. There are 224 spectral channels ranging from 0.4 to 2.5 $\mu$m with a spatial resolution of 3.7 m. The area covered comprises 512 $\times$ 217 pixels. As with the Indian Pines data set, 20 water absorption bands are discarded; 54 129 pixels with 16 classes are labeled. Labels, the number of labeled training samples, and the number of testing samples are listed in Table IV.

TABLE IV

LABELS, THE NUMBER OF LABELED TRAINING SAMPLES, AND THE NUMBER OF TESTING SAMPLES FOR THE SALINAS DATA SET

| No. | Class | Training | Testing |
|---|---|---|---|
| 1 | Brocoli_green_weeds_1 | 5 | 2009 |
| 2 | Brocoli_green_weeds_2 | 5 | 3726 |
| 3 | Fallow | 5 | 1976 |
| 4 | Fallow_rough_plow | 5 | 1394 |
| 5 | Fallow_smooth | 5 | 2678 |
| 6 | Stubble | 5 | 3959 |
| 7 | Celery | 5 | 3579 |
| 8 | Grapes_untrained | 5 | 11271 |
| 9 | Soil_vinyard_develop | 5 | 6203 |
| 10 | Corn_senesced_green_weeds | 5 | 3278 |
| 11 | Lettuce_romaine_4wk | 5 | 1068 |
| 12 | Lettuce_romaine_5wk | 5 | 1927 |
| 13 | Lettuce_romaine_6wk | 5 | 916 |
| 14 | Lettuce_romaine_7wk | 5 | 1070 |
| 15 | Vinyard_untrained | 5 | 7268 |
| 16 | Vinyard_vertical_trellis | 5 | 1807 |
| | Total | 80 | 54129 |

The fourth data set is the Houston data set gathered by The ITRES CASI-1500 sensor. This data set is composed of 349 $\times$ 1905 pixels with 144 spectral channels ranging from 364 to 1046 nm. There are 15 classes in this scene. Labels, the number of labeled training samples, and the number of testing samples are listed in Table V.

### *B. Parameter Setting and Analysis*

The neighborhood size is an important parameter that affects the classification performance. To analyze the influence of neighborhood size on classification accuracy, the neighborhood size is set to be 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, and 35, respectively. The classification results are shown in Fig. 4. According to the experimental results, we find that a small neighborhood size will reduce the classification accuracy, and the optimal neighborhood sizes of the four data sets are 25, 27, 35, and 27, respectively. However, as for the University of Pavia data set, setting neighborhood size to 25 or 27 has little effect on the final classification results. The Salinas data set has a similar situation. Considering the adapt-

TABLE V

LABELS, THE NUMBER OF LABELED TRAINING SAMPLES, AND THE NUMBER OF TESTING SAMPLES FOR THE HOUSTON DATA SET

| No. | Class | Training | Testing |
|---|---|---|---|
| 1 | Healthy grass | 5 | 1251 |
| 2 | Stressed grass | 5 | 1254 |
| 3 | Synthetic grass | 5 | 697 |
| 4 | Trees | 5 | 1244 |
| 5 | Soil | 5 | 1242 |
| 6 | Water | 5 | 325 |
| 7 | Residential | 5 | 1268 |
| 8 | Commercial | 5 | 1244 |
| 9 | Road | 5 | 1252 |
| 10 | Highway | 5 | 1227 |
| 11 | Railway | 5 | 1235 |
| 12 | Parking Lot 1 | 5 | 1233 |
| 13 | Parking Lot 2 | 5 | 469 |
| 14 | Tennis Court | 5 | 428 |
| 15 | Running Track | 5 | 660 |
| | Total | 75 | 15029 |

TABLE VI

DEPTH AND PARAMETERS OF DIFFERENT NETWORKS

| | LeNet | Resnet18 | Resnet34 | Resnet50 | SE+Resnet50 |
|---|---|---|---|---|---|
| Depth | 5 | 19 | 35 | 51 | 51 |
| Parameters | 0.07m | 10.97m | 20.60m | 23.48m | 26.09m |

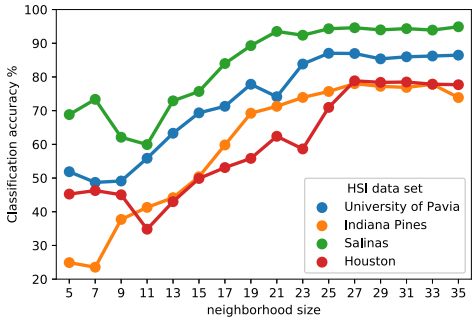



Fig. 4. Classification with different neighborhood sizes.

ability of parameters to different data sets, the neighborhood sizes of four data sets are set to be $27 \times 27$. Consequently, the dimension of each view for the four data sets is $27 \times 27 \times 3$.

In general, training a CNN requires setting the learning rate, the number of epochs, the optimizer, and the batch size. In this article, the widely used Adam [45] optimizer is used to optimize the designed deep residual network. The batch size is set to be 128. The training loss value is adopted as the evaluation index of the network training, as the training procedure is unsupervised. The learning rate is set to be 0.1, 0.01, and 0.001, respectively. The training loss values with different learning rates are shown in Fig. 5. From Fig. 5, we could find that a large learning rate (e.g., 0.1 and 0.01) is not conducive to the network training and would lead to a large loss function value. In contrast, a small learning rate (e.g., 0.001) could enable the network to be fully trained. Therefore, a small learning rate and a large number of epochs are used to ensure the convergence of the network. Finally, the learning rate is set to be 0.001, and the number of epochs is set to be 50.

The nonlinear SVM with radial basis function kernel is used as the supervised classifier in this section. The goal of this article is to deal with the problem of small sample classification. Therefore, only five labeled samples per class are randomly selected as the training samples of the SVM classifier to analyze the influence of parameters on classification accuracy. The SVM classifier with radial basis function kernel needs to set the parameters $C$ (a parameter that controls the amount of penalty during the SVM optimization) and $\gamma$ (spread of the RBF kernel). The optimal hyperplane parameters $C$ and $\gamma$ have been traced in the range of $C = 2^{-2}, 2^{-1}, \ldots, 2^{7}$ and $\gamma = 2^{-2}, 2^{-1}, \ldots, 2^{7}$ using fivefold cross validation [46].

To study the importance of data augmentation composition, the network is trained with applying augmentations individually or in pairs. The classification results are shown in Fig. 6. In Fig. 6, "None" represents that no data augmentation is applied, "RC" represents that only random cropping is applied, "RG" represents that only random Gaussian blur is applied, and "RC + RG" represents that random cropping and random Gaussian blur are applied. From the results of Fig. 6, we find that both random cropping and random Gaussian blur could improve the classification accuracy. In contrast, without data augmentation, the subsequent classification accuracy will be greatly reduced. Therefore, both random cropping and random Gaussian blur are used in the training procedure.

The studies have shown the advantages of large-scale deep CNNs (e.g., GoogLeNet, Resnet50) in natural image classification and recognition. However, large-scale deep CNNs have not been used in HSI classification tasks. In this article, a deep residual network with 51 layers derived from the standard Resnet50 is used as the feature extractor. To prove the necessity of using large-scale networks in multiview learning, the LeNet, Resnet18, and Resnet34 are also used as the feature extractors of four HSI data sets. We also test the channelwise attention Resnet [47] (SE + Resnet50). The depth and parameters of different networks are listed in Table V. The classification results are shown in Fig. 7. It is found that classification accuracy decreases with the decrease of the network scale. In addition, the introduction of channelwise attention has little effect on improving the final classification accuracy but will increase the training time. This is because the classification accuracy will be gradually stable or decrease with the increase in the network complexity. When the classification accuracy tends to be stable, increasing the network complexity (e.g., introducing channelwise attention) will further reduce the classification accuracy. Therefore, we finally use the Resnet50 as the base feature extractor. Note that Resnet50 is one of the most commonly used classical network models in computer vision tasks. Using Resnet50 enables us to reuse the classic network model. This not only saves the work of network design but also proves that a deep network model can be used to improve the classification accuracy in the HSI classification task.

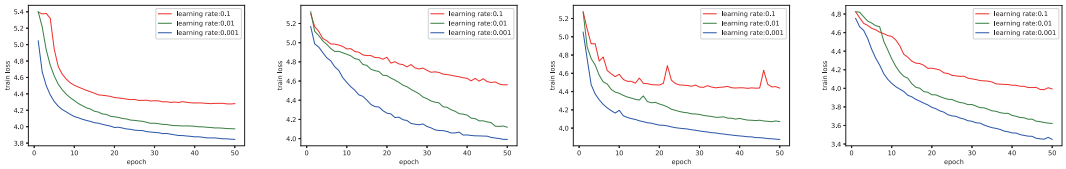



Fig. 5. Training loss values with different learning rate. (a) University of Pavia data set. (b) Indiana Pines data set. (c) Salinas data set. (d) Houston data set.

TABLE VII

CLASS-SPECIFIC ACCURACY, OA, AA, AND $\kappa$ OF DIFFERENT METHODS FOR THE UNIVERSITY OF PAVIA DATA SET (BOLD VALUES REPRESENT THE BEST ACCURACY AMONG THESE METHODS IN EACH CASE)

| Class No. | EMP+SVM | TSVM | JCR | 3DCAE | GAN | DFSL+SVM | Resnet50 | DMVL+SVM | DMVL+RF |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 72.16 | 69.78 | 74.08 | 63.26 | 60.91 | 92.76 | 71.85 | 57.80 | 55.86 |
| 2 | 86.13 | 66.25 | 60.28 | 90.02 | 77.47 | 95.90 | 80.72 | 98.32 | 95.37 |
| 3 | 80.75 | 65.03 | 75.08 | 60.60 | 69.56 | 45.90 | 77.93 | 84.37 | 98.86 |
| 4 | 98.40 | 62.14 | 81.63 | 80.00 | 92.46 | 98.54 | 28.58 | 56.01 | 55.87 |
| 5 | 99.26 | 100.0 | 100.0 | 99.85 | 98.36 | 99.41 | 70.30 | 100.0 | 99.41 |
| 6 | 16.88 | 84.97 | 70.77 | 12.79 | 48.98 | 77.91 | 76.74 | 100.0 | 100.0 |
| 7 | 96.54 | 98.27 | 88.42 | 83.23 | 95.94 | 46.24 | 89.91 | 99.85 | 87.07 |
| 8 | 8.23 | 80.07 | 84.06 | 43.10 | 55.43 | 48.11 | 59.56 | 97.23 | 93.05 |
| 9 | 99.79 | 54.17 | 97.57 | 97.04 | 99.89 | 92.40 | 52.30 | 27.35 | 45.51 |
| OA (%) | 70.77 | 71.62 | 70.90 | 70.85 | 72.06 | 81.92 | 68.85 | 86.96 | 85.70 |
| AA (%) | 73.13 | 75.63 | 81.32 | 69.99 | 77.67 | 77.46 | 67.54 | 80.10 | 81.22 |
| $\kappa$ | 61.35 | 64.71 | 63.94 | 61.01 | 64.03 | 76.46 | 58.23 | 82.46 | 81.07 |

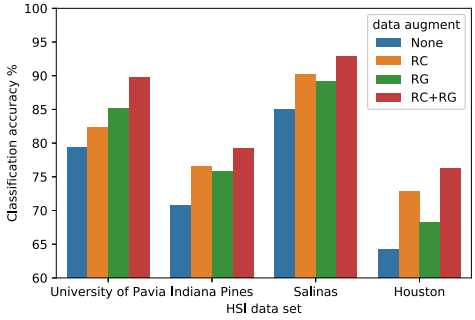



Fig. 6. Classification accuracy with different data augmentation strategies on the four HSI data sets. "None" represents that no data augmentation is applied, "RC" represents that only random cropping is applied, "RG" represents that only random gaussian blur is applied, and "RC + RG" represents that random cropping and random gaussian blur are applied.

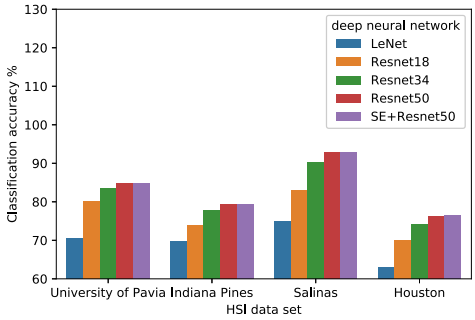



Fig. 7. Classification accuracy with different networks on the four HSI data sets.

To analyze the influence of the number of views on classification accuracy, we also divide the HSI into four views and eight views. The results are shown in Fig. 8. From the results of Fig. 8, we can find that more views (e.g., 4) could improve

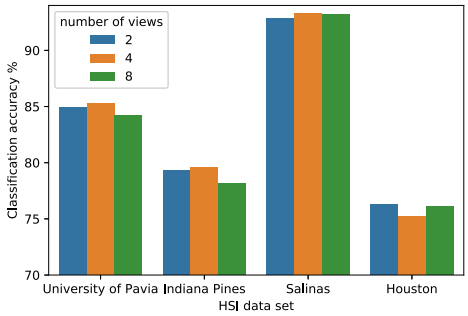



Fig. 8. Classification accuracy with different views on the four HSI data sets.

the classification accuracy slightly. However, more views will increase the complexity of the model, which will lead to a significant increase in training time. More importantly, only two views can achieve satisfactory classification results. Consequently, only two views are used to train the designed deep network in subsequent experiments.

In order to prove the necessity of using PCA, we test that the grouped HSI data cube (raw data) is directly input into the designed deep residual network for multiview learning. The results on four data set are shown in Fig. 9. It could be found that directly using raw data as input will greatly reduce the accuracy of subsequent classification. This is because PCA can not only retain the main information of each view but also increase the difference between the two views. Thus, it is helpful to use the contrast loss function to mine the features of HSIs.

### C. Comparison Results With the State-of-the-Art Methods

In this section, the performance of the proposed deep multiview learning (DMVL + SVM) is compared with several

TABLE VIII

CLASS-SPECIFIC ACCURACY, OA, AA, AND $\kappa$ OF DIFFERENT METHODS FOR THE INDIANA PINES DATA SET (BOLD VALUES REPRESENT THE BEST ACCURACY AMONG THESE METHODS IN EACH CASE)

| Class No. | EMP+SVM | TSVM | JCR | 3DCAE | GAN | DFSL+SVM | Resnet50 | DMVL+SVM | DMVL+RF |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 32.23 | 86.96 | 97.83 | 86.96 | 52.38 | 58.23 | 7.96 | 86.96 | 100.0 |
| 2 | 55.14 | 36.20 | 35.15 | 32.91 | 48.51 | 57.98 | 50.81 | 67.51 | 64.01 |
| 3 | 37.86 | 42.65 | 56.02 | 23.13 | 45.94 | 61.46 | 43.72 | 72.53 | 72.77 |
| 4 | 40.51 | 53.59 | 45.57 | 45.15 | 39.61 | 22.42 | 21.24 | 99.58 | 100.0 |
| 5 | 33.95 | 78.88 | 69.98 | 51.14 | 46.44 | 90.60 | 31.41 | 68.12 | 66.87 |
| 6 | 78.41 | 70.82 | 85.34 | 75.62 | 93.01 | 99.85 | 45.92 | 76.16 | 70.68 |
| 7 | 29.17 | 92.86 | 100.0 | 100.0 | 39.71 | 25.23 | 14.07 | 100.0 | 100.0 |
| 8 | 95.18 | 85.15 | 90.38 | 84.94 | 97.36 | 98.92 | 63.73 | 100.0 | 100.0 |
| 9 | 21.28 | 90.00 | 100.0 | 100.0 | 25.32 | 8.16 | 4.07 | 100.0 | 100.0 |
| 10 | 41.34 | 41.56 | 70.27 | 54.42 | 47.67 | 43.32 | 100.0 | 83.64 | 86.32 |
| 11 | 71.86 | 55.15 | 59.92 | 53.24 | 68.69 | 74.35 | 58.99 | 71.41 | 57.19 |
| 12 | 25.20 | 50.08 | 43.68 | 34.06 | 35.18 | 43.75 | 28.98 | 49.07 | 45.19 |
| 13 | 88.73 | 96.10 | 100.0 | 85.85 | 78.13 | 92.49 | 42.60 | 98.05 | 99.02 |
| 14 | 75.28 | 82.77 | 91.46 | 71.30 | 89.90 | 91.85 | 43.96 | 99.05 | 96.68 |
| 15 | 66.05 | 35.23 | 59.07 | 23.58 | 57.42 | 66.93 | 25.26 | 87.56 | 99.48 |
| 16 | 100.0 | 93.55 | 100.0 | 88.17 | 100.0 | 75.61 | 17.50 | 100.0 | 98.92 |
| OA (%) | 56.80 | 57.65 | 64.95 | 52.21 | 60.24 | 64.56 | 36.97 | 78.01 | 73.95 |
| AA (%) | 55.76 | 68.22 | 75.29 | 63.15 | 60.33 | 63.20 | 37.51 | 84.98 | 84.82 |
| $\kappa$ | 51.52 | 52.24 | 60.66 | 46.20 | 55.38 | 60.60 | 31.45 | 75.31 | 71.07 |

TABLE IX

CLASS-SPECIFIC ACCURACY, OA, AA, AND $\kappa$ OF DIFFERENT METHODS FOR THE SALINAS DATA SET (BOLD VALUES REPRESENT THE BEST ACCURACY AMONG THESE METHODS IN EACH CASE)

| Class No. | EMP+SVM | TSVM | JCR | 3DCAE | GAN | DFSL+SVM | Resnet50 | DMVL+SVM | DMVL+RF |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 95.36 | 97.31 | 97.86 | 92.19 | 99.65 | 100.0 | 4.83 | 95.92 | 94.33 |
| 2 | 98.85 | 96.32 | 98.36 | 99.68 | 72.84 | 99.78 | 17.71 | 100.0 | 100.0 |
| 3 | 81.09 | 91.24 | 99.80 | 58.96 | 77.88 | 89.01 | 0.0 | 100.0 | 100.0 |
| 4 | 97.74 | 99.57 | 97.42 | 96.41 | 99.07 | 98.64 | 69.44 | 100.0 | 100.0 |
| 5 | 96.37 | 94.47 | 92.91 | 89.66 | 96.38 | 96.52 | 61.80 | 94.73 | 93.54 |
| 6 | 100.0 | 99.77 | 97.58 | 98.89 | 98.94 | 100.0 | 37.53 | 90.96 | 99.32 |
| 7 | 94.52 | 98.02 | 96.28 | 97.46 | 92.71 | 98.78 | 72.17 | 98.60 | 99.39 |
| 8 | 81.25 | 59.94 | 82.60 | 70.46 | 49.71 | 88.18 | 89.85 | 91.47 | 89.94 |
| 9 | 99.06 | 95.31 | 99.90 | 94.66 | 94.57 | 99.45 | 96.15 | 99.92 | 97.92 |
| 10 | 86.86 | 83.59 | 80.57 | 69.89 | 89.08 | 86.42 | 48.60 | 80.63 | 94.84 |
| 11 | 68.27 | 97.19 | 98.41 | 91.10 | 96.72 | 88.03 | 98.88 | 99.25 | 100.0 |
| 12 | 88.64 | 100.0 | 100.0 | 97.98 | 100.0 | 95.59 | 0.0 | 94.91 | 93.82 |
| 13 | 60.11 | 97.82 | 91.92 | 98.03 | 97.82 | 98.60 | 92.47 | 72.49 | 87.23 |
| 14 | 95.20 | 97.20 | 98.50 | 81.59 | 95.98 | 98.97 | 0.0 | 98.97 | 95.98 |
| 15 | 55.55 | 79.27 | 33.17 | 64.93 | 85.43 | 62.68 | 16.17 | 95.56 | 97.50 |
| 16 | 88.47 | 86.05 | 98.62 | 79.80 | 70.56 | 99.47 | 59.38 | 100.0 | 100.0 |
| OA (%) | 83.47 | 85.63 | 84.93 | 82.72 | 81.67 | 89.09 | 54.10 | 94.60 | 95.88 |
| AA (%) | 86.71 | 92.07 | 91.49 | 86.35 | 88.58 | 93.76 | 47.81 | 94.59 | 96.49 |
| $\kappa$ | 81.68 | 84.09 | 83.14 | 80.80 | 79.75 | 87.92 | 48.30 | 94.00 | 95.43 |

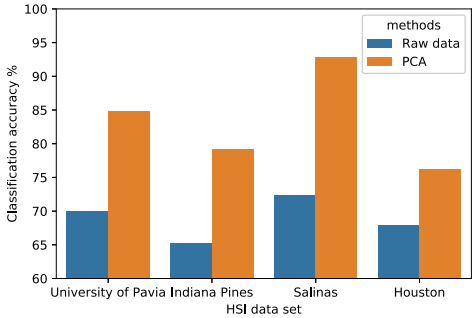



Fig. 9. Classification accuracy with raw data and PCA on the four HSI data sets.

state-of-the-art methods. The compared methods are listed as follows.

1) EMP + SVM [10] is a traditional spatial feature extraction method that could improve the classification accuracy. It has been widely used in the HSI classification task. As for EMP, two commonly used morphological filters based on square structure elements (opening and closing) are used to construct the morphological attribute profiles. The radius of structuring elements is set to be 1, 3, 5, 7, and 9, respectively. The optimal hyperplane parameters of the SVM classifier are determined by fivefold cross validation.
2) TSVM [48], [49] is a semisupervised method that could use unlabeled samples to improve classification accuracy. It also uses an SVM classifier with a radial basis function kernel. All unlabeled samples are used for training.
3) Joint within-class collaborative representation (JCR) [50] is a representation-based classification method. In this method, neighbors near the test pixel are simultaneously represented via linear combinations of available training samples. Several strategies of incorporating

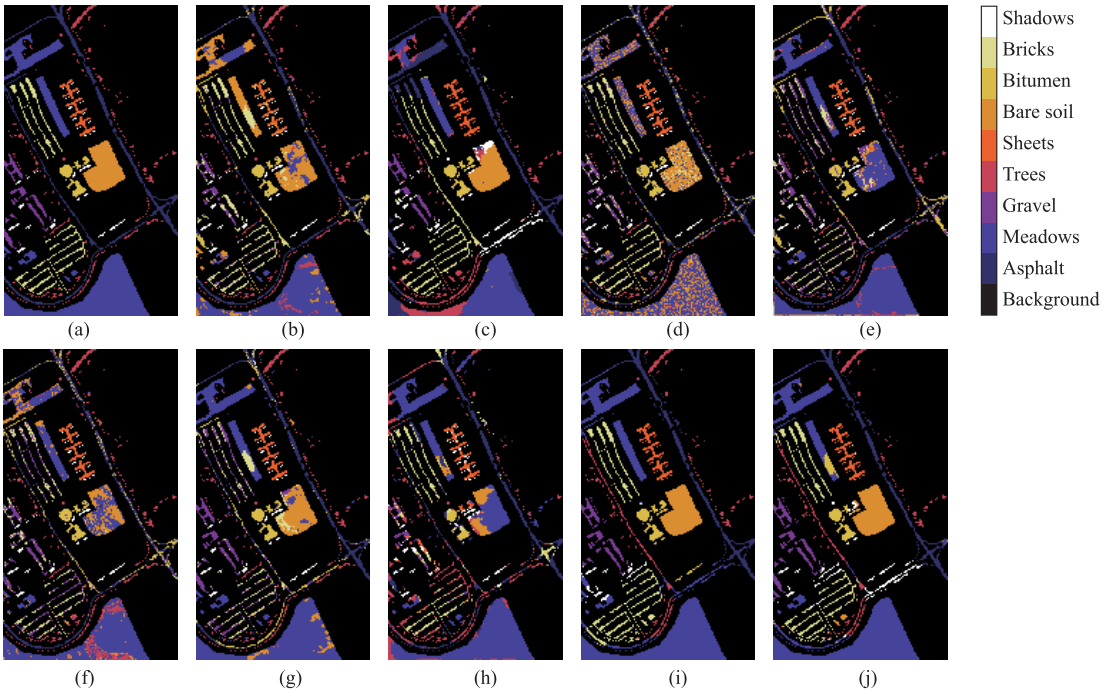



Fig. 10. Classification maps resulting from different methods for the University of Pavia data set. (a) Ground-truth map. (b) EMP + SVM (OA: 70.77%). (c) TSVM (OA: 71.62%). (d) JCR (OA: 70.90%). (e) 3DCAE (OA: 70.85%). (f) GAN (OA: 72.06%). (g) DFSL + SVM (OA: 81.92%). (h) Resnet50 (OA: 68.85%). (i) DMVL + SVM (OA: 86.96%). (j) DMVL + RF (OA: 85.70%).

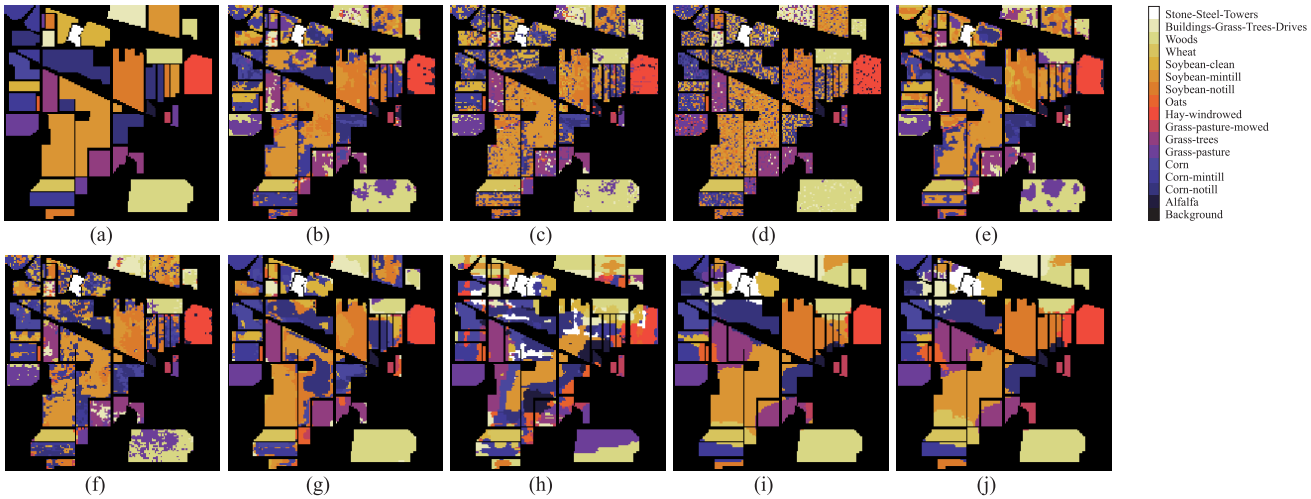



Fig. 11. Classification maps resulting from different methods for the Indiana Pines data set. (a) Ground-truth map. (b) EMP + SVM (OA: 56.80%). (c) TSVM (OA: 57.65%). (d) JCR (OA:64.95%). (e) 3DCAE (OA: 52.21%). (f) GAN (OA: 60.24%). (g) DFSL + SVM (OA: 64.56%). (h) Resnet50 (OA: 36.97%). (i) DMVL + SVM (OA: 78.01%). (j) DMVL + RF (OA: 73.95%).

contextual information are used to improve the classification performance.

4) 3DCAE [35] is an unsupervised spatial–spectral feature learning method based on 3-D convolutional autoencoder. It is very effective in extracting spatial–spectral features. The parameters are set the same as the paper.
5) GAN [51] is an unsupervised feature learning method based on a generative adversarial network. PCA is used to reduce the HSI to three dimensions. Then, a 2-D GAN is used to learn features. The neighborhood size is also set to be $28 \times 28$.
6) DFSL + SVM [52] is a transfer learning method. It trains a deep 3-D-CNN to learn a metric space on the data collected in advance. The trained network is then transferred to the target HSI. It achieves excellent results with small samples. The parameters are set the same as the paper.
7) In the case of only a single view, DMVL degenerates to a supervised classifier. In other words, a ResNet50 classifier is used for HSI classification. Therefore, we also test a supervised ResNet50 classifier. The learning rate is set to be 0.001, the batch size is set to be 5, and the number of epochs is set to be 300.

We also test the proposed DMVL with an RF classifier. Note that five labeled samples per class are randomly selected as the supervised samples and all labeled

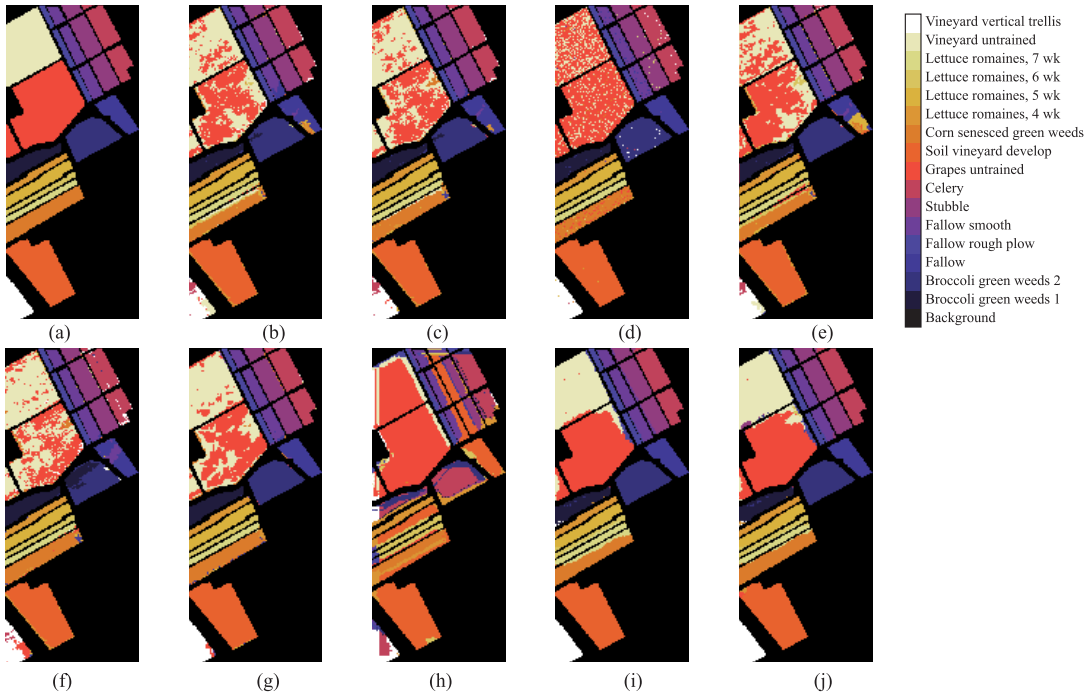



Fig. 12. Classification maps resulting from different methods for the Salinas data set. (a) Ground-truth map. (b) EMP + SVM (OA: 83.47%). (c) TSVM (OA: 85.63%). (d) JCR (OA:84.93%). (e) 3DCAE (OA:82.72%). (f) GAN (OA:81.67%). (g) DFSL + SVM (OA: 89.09%). (h) Resnet50 (OA: 54.10%). (i) DMVL + SVM (OA: 94.60%). (j) DMVL + RF (OA: 95.88%).

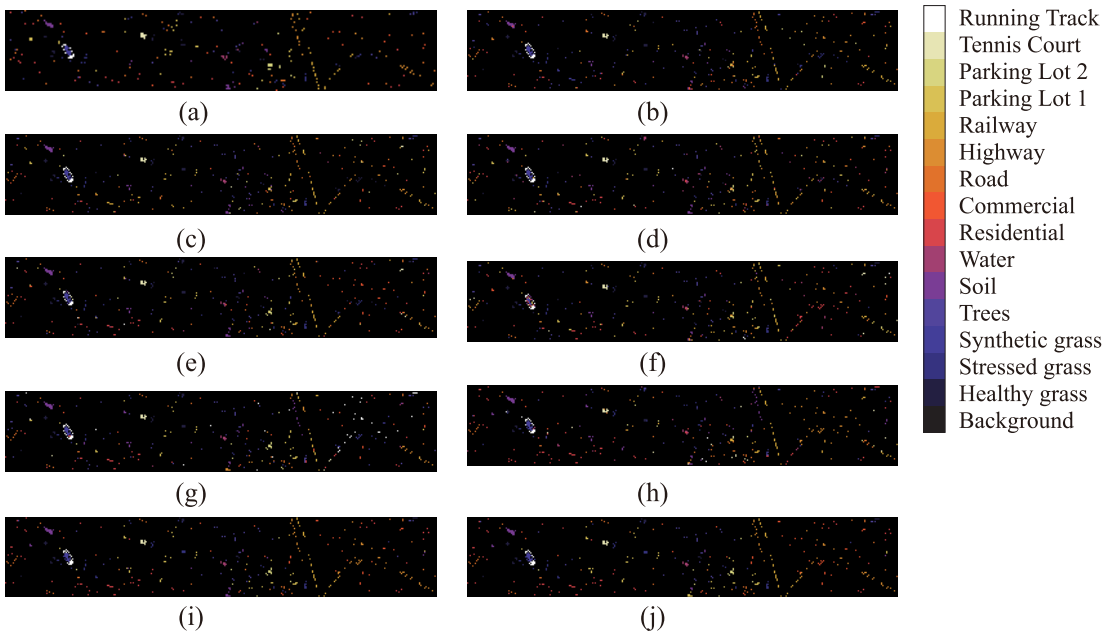



Fig. 13. Classification maps resulting from different methods for the Houston data set. (a) Ground-truth map. (b) EMP + SVM (OA: 67.02%). (c) TSVM (OA: 70.68%). (d) JCR (OA: 71.30%). (e) 3DCAE(OA: 74.36%). (f) GAN (OA: 58.60%). (g) DFSL + SVM (OA: 69.29%). (h) Resnet50 (44.72%). (i) DMVL + SVM (OA: 78.55%). (j) DMVL + RF (OA: 76.65%).

training samples for different methods are exactly the same.

The class-specific accuracy, overall accuracy (OA), average accuracy (AA), and $\kappa$ of different methods for four HSI data sets are listed in Tables VII–X. From these results, we can learn that the Resnet50 classifier has the worst classification accuracy when only five labeled samples are used in each class. This shows that training a deep network with a small sample will produce a serious overfitting problem, which leads to low classification accuracy. In contrast with the Resnet50 classifier, the proposed method DMVL combined with an SVM classifier or an RF classifier could achieve higher OA, AA, and $\kappa$ than the other compared methods. For example, in Table VII, DMVL + SVM (i.e., 86.96%) yields over

TABLE X

CLASS-SPECIFIC ACCURACY, OA, AA, AND $\kappa$ OF DIFFERENT METHODS FOR THE HOUSTON DATA SET (BOLD VALUES REPRESENT THE BEST ACCURACY AMONG THESE METHODS IN EACH CASE)

| Class No. | EMP+SVM | TSVM | JCR | 3DCAE | GAN | DFSL+SVM | Resnet50 | DMVL+SVM | DMVL+RF |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 85.29 | 95.20 | 71.30 | 83.37 | 84.01 | 74.74 | 36.61 | 50.52 | 50.52 |
| 2 | 94.90 | 85.49 | 85.25 | 81.98 | 66.99 | 46.97 | 38.44 | 68.74 | 66.35 |
| 3 | 99.57 | 99.00 | 92.97 | 99.43 | 41.75 | 95.12 | 94.69 | 86.94 | 90.82 |
| 4 | 93.17 | 91.40 | 89.31 | 69.69 | 87.78 | 86.41 | 25.80 | 63.42 | 63.67 |
| 5 | 85.43 | 97.26 | 97.18 | 97.83 | 87.04 | 78.82 | 85.27 | 87.84 | 63.53 |
| 6 | 76.62 | 92.62 | 85.85 | 73.85 | 40.31 | 73.54 | 73.54 | 75.38 | 75.38 |
| 7 | 58.99 | 43.61 | 69.09 | 75.39 | 54.97 | 61.36 | 82.02 | 92.35 | 91.96 |
| 8 | 25.32 | 34.41 | 24.12 | 60.21 | 32.23 | 56.35 | 15.76 | 64.31 | 63.67 |
| 9 | 71.09 | 53.75 | 61.02 | 66.29 | 16.29 | 78.59 | 63.02 | 81.87 | 82.51 |
| 10 | 29.58 | 57.78 | 56.89 | 44.09 | 60.39 | 39.77 | 31.38 | 100.0 | 100.0 |
| 11 | 70.69 | 71.74 | 60.32 | 78.79 | 49.31 | 38.06 | 57.98 | 88.34 | 88.34 |
| 12 | 18.33 | 51.26 | 78.26 | 58.56 | 51.26 | 85.00 | 1.46 | 68.45 | 71.29 |
| 13 | 34.54 | 17.70 | 30.06 | 74.41 | 36.03 | 91.47 | 11.51 | 95.74 | 95.74 |
| 14 | 99.07 | 96.50 | 84.58 | 99.77 | 89.72 | 98.60 | 78.50 | 100.0 | 100.0 |
| 15 | 98.79 | 97.88 | 99.85 | 82.12 | 73.33 | 92.88 | 86.21 | 82.73 | 80.45 |
| OA (%) | 67.02 | 70.68 | 71.30 | 74.36 | 58.60 | 69.29 | 44.72 | 78.55 | 76.65 |
| AA (%) | 69.42 | 72.37 | 72.40 | 76.39 | 58.09 | 73.18 | 52.15 | 80.44 | 78.95 |
| $\kappa$ | 64.42 | 68.33 | 69.04 | 72.32 | 55.25 | 66.97 | 48.72 | 76.81 | 74.75 |

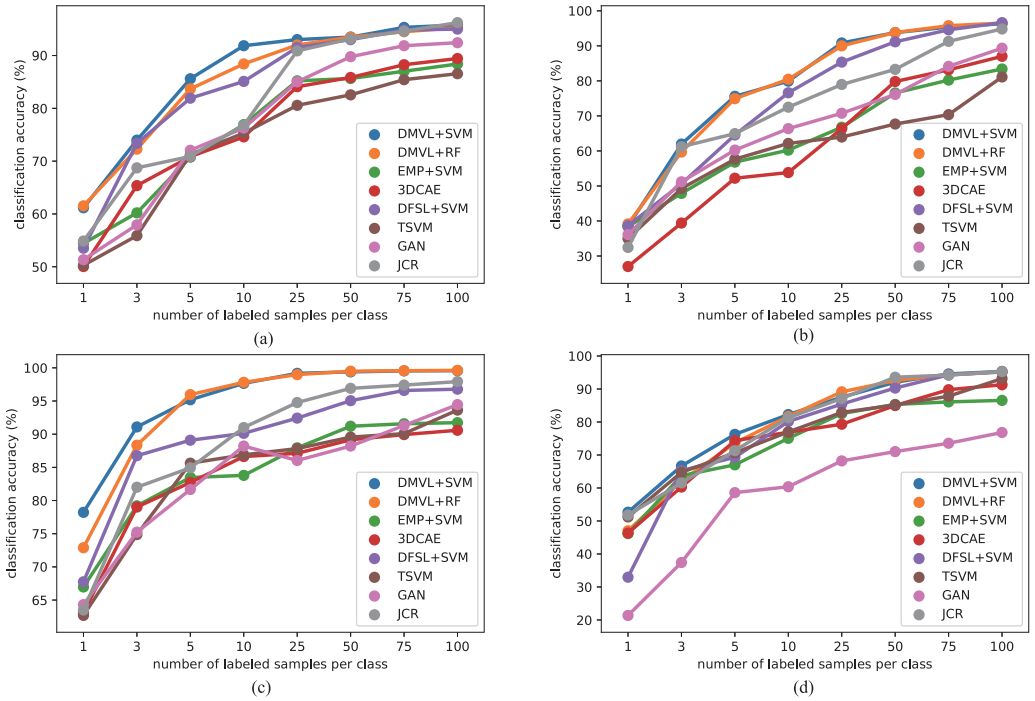



Fig. 14. Classification accuracy for the three HSI data sets with different sample number. (a) University of Pavia data set. (b) Indiana Pines. (c) Salinas. (d) Houston.

10% higher accuracy than EMP + SVM (i.e., 70.77) and approximately 5% higher accuracy than DFSL + SVM. Note that DFSL is a transfer learning method designed for small sample problems. Especially, in Table VIII, the OA of DMVL + SVM (78.01%) is over 20% higher than that of 3DCAE (52.21%).

In fact, the contrastive loss function is an unsupervised loss function. Thus, an unsupervised autoencoder (3DCAE) that adopts the traditional reconstruction error loss function is used as the compared method. Compared with the traditional reconstruction error, the contrastive loss function could greatly improve the performance of subsequent HSI classification tasks. In order to further illustrate the effectiveness of the algorithm, the proposed method is compared with a semisupervised loss function GAN. The experimental results show that the contrastive loss function is superior to GAN. In addition, the contrastive loss function is compared with the traditional cross-entropy loss function (ResNet50). The experimental results show that the classification accuracy of the contrastive loss function is much higher than that of the traditional cross-entropy loss function.

In order to better observe the classification results, the classification maps of different methods on four HSI data sets are shown in Figs. 10–13. To facilitate the comparison

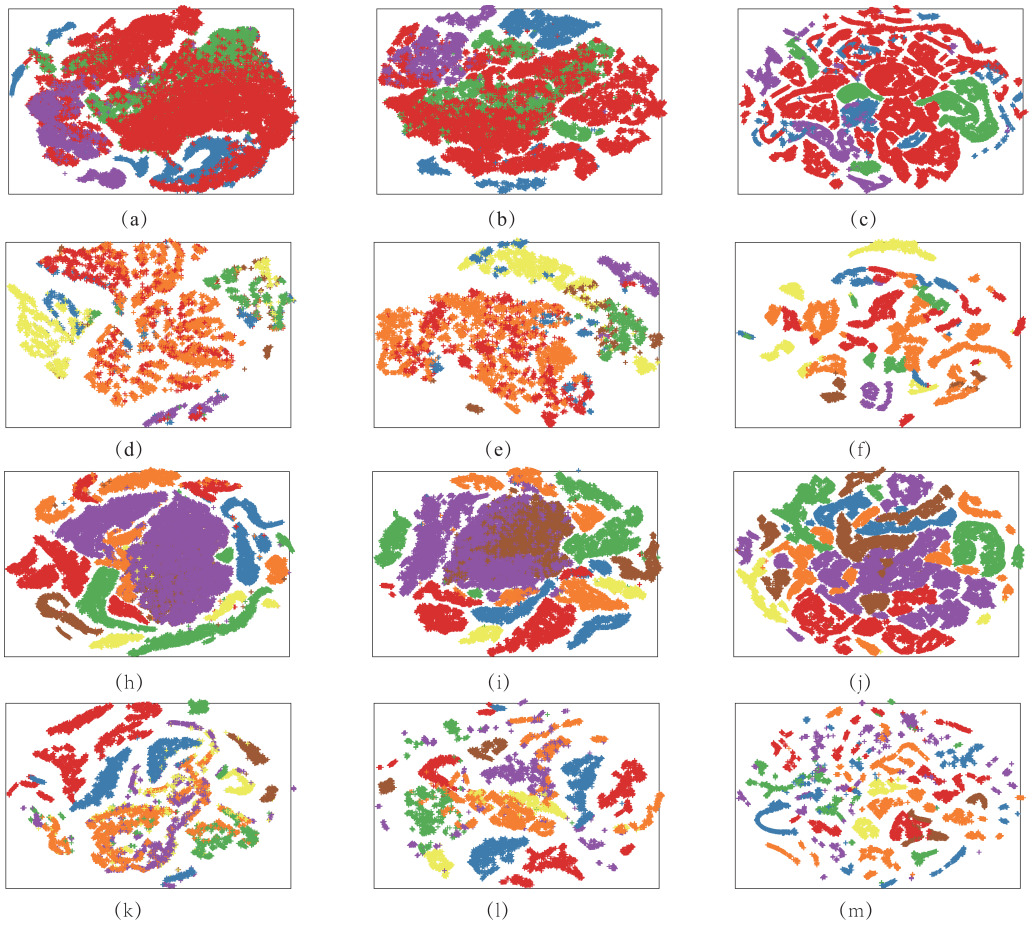



Fig. 15. Feature visualization results. The University of Pavia data set: (a) the spectral features, (b) the features obtained by the MSE loss function, (c) the features extracted by the DMVL, The Indiana Pines data set: (d) the spectral features, (e) the features obtained by the MSE loss function, (f) the features extracted by the DMVL,The Salinas data set: (g) the spectral features, (h) the features obtained by the MSE loss function, (i) the features extracted by the DMVL,The Houston data set: (j) the spectral features, (k) the features obtained by the MSE loss function, (l) the features extracted by the DMVL.

between different methods, the ground-truth maps are shown in Figs. 10–13. From these maps, we can learn that the compared methods exhibit higher classification errors than the proposed method.

The abovementioned experimental results have proved the effectiveness of the proposed method in the case of small samples. To further test the effectiveness of the proposed method, we change the number of labeled samples used for supervised training. The classification results are shown in Fig. 14. First, when the number of labeled samples is further reduced (e.g., one sample per class and three samples per class), the accuracy of all classification algorithms is greatly reduced. However, the proposed method could still achieve the highest classification accuracy. Second, the classification accuracy of all methods increases with the increase in the number of labeled samples. This is easy to understand because the overfitting problem is alleviated as the number of labeled samples increases. Third, DMVL + SVM, DMVL + RF, and DFSL + SVM generally outperform the other classification methods. More importantly, DMVL + SVM and DMVL + RF could obtain the best classification accuracy in most cases. This further demonstrates the proposed method can not only deal with the problem of small samples but also has good applicability to the number of labeled samples. That is to say, the proposed method can obtain higher classification accuracy than the other compared methods in the cases with a different number of labeled samples. In addition, with the increase in the number of labeled samples, the difference of classification accuracy between DMVL + SVM and DMVL + RF is less and less. Anyway, the proposed method combined with SVM and RF can achieve better classification results than the compared methods.

### *D. Feature Visual Analysis*

In order to analyze the effectiveness of the proposed method, we visualized the original spectral features, the features obtained by the MSE loss function, and the features extracted by the proposed method. The results of feature visualization are shown in Fig. 15. In Fig. 15, the first line is the feature visualization results obtained from the original spectral features of the University of Pavia data set, the MSE loss function results, and the contrastive loss function results. Similarly, lines 2–4 correspond to the visualization results of the Indiana Pines, Salinas, and Houston data sets, respectively. In Fig. 15, different colors represent the spatial distribution of different classes of samples. By observing the visualization results, we could see that the distribution of features extracted by the MSE loss function is not significantly improved compared with the distribution of the original spectral features. On the

TABLE XI
EXECUTION TIMES OF TRAINING AND TESTING PROCEDURES IN THE IP DATA SET

| University of Pavia Data set | | | | |
|---|---|---|---|---|
| | 3DCAE | GAN | DFSL | DMVL |
| Trianing (min) | 19.47 | 6.77 | 106.20 | 197.71 |
| Feature extraction (s) | 32.04 | 2.04 | 40.82 | 50.09 |
| Indiana Pines data set | | | | |
| | 3DCAE | GAN | DFSL | DMVL |
| Trianing (min) | 19.27 | 7.00 | 106.20 | 141.23 |
| Feature extraction (s) | 5.22 | 0.54 | 11.14 | 27.31 |
| Salinas data set | | | | |
| | 3DCAE | GAN | DFSL | DMVL |
| Trianing (min) | 19.32 | 12.34 | 106.20 | 243.17 |
| Feature extraction (s) | 26.4 | 2.66 | 52.31 | 77.74 |

contrary, the spatial distance of different classes of features extracted by the proposed method is significantly increased, which benefits from the contrastive loss function used in this article. Thus, the proposed method could effectively improve the classification accuracy of HSIs.

### *E. Execution Time Analysis*

The training time of a deep neural network is mainly affected by the number of samples, the dimension of inputs, and the network parameters. The training time and feature extraction time for different methods are listed in Table XI. As for DFSL, the model for different HSIs is trained on the same data. Thus, the training time is the same for three HSIs. The number of labeled testing samples is different on three HSIs, which leads to different feature extraction time for three HSIs. On the one hand, all unlabeled samples are used to train the deep residual network. On the other hand, a deep residual network with 51 layers is used to extract features. Note that the 51-layer residual network is larger than the existing networks in the field of HSI classification. Therefore, compared with similar algorithms, our method is more time-consuming. It takes about 3 min to train at an epoch on the Indiana Pines data set. The number of epochs is set to be 50, resulting in approximately 150 min of the training procedure on the Indiana Pines data set. The training procedure takes more time, which is a disadvantage of the method proposed in this article. However, the increase in time is acceptable. More importantly, the proposed method could greatly improve the classification accuracy in the case of small samples.

## IV. CONCLUSION

Recently, deep learning-based methods have been widely explored in HSI classification. However, training a deep-learning classifier notoriously requires hundreds or thousands of labeled samples. Therefore, training the models to learn the useful representations of HSIs in an unsupervised manner is the Holy Grail for researchers. In this article, we proposed the deep multiview learning method for HSI classification. By training the network to learn view-invariant features, the proposed method could greatly improve classification accuracy, especially in the case of small samples. Moreover, we first explore using a deep residual network with 51 layers in the HSI field. Experiments demonstrate the necessity of using bigger models. Although this method has achieved excellent classification performance, the improvement of classification accuracy is based on the premise of sacrificing training time. The time-consuming training procedure is a disadvantage of the proposed method. In order to simplify the process, we have only constructed two views. In the future, we will construct more views to further improve classification performance. Finally, the proposed method is easy to combine with the existing supervised classifiers. We only test the SVM classifier and RF classifier. In the future, we would test more classifiers.

## ACKNOWLEDGMENT

The authors thank Prof. Mei Shaohui for providing the codes of 3DCAE. They also thank Prof. Wei Li for providing the codes of JCR.

## REFERENCES

[1] 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.

[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] 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.

[4] D. B. Heras, F. Argüello, and P. Quesada-Barriuso, "Exploring elm-based spatial–spectral classification of hyperspectral images," *Int. J. Remote Sens.*, vol. 35, no. 2, pp. 401–423, 2014.

[5] Y. Zhang, G. Cao, X. Li, and B. Wang, "Cascaded random forest for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 4, pp. 1082–1094, Apr. 2018.

[6] C. Li, Y. Ma, X. Mei, C. Liu, and J. Ma, "Hyperspectral image classification with robust sparse representation," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 5, pp. 641–645, May 2016.

[7] 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.

[8] W. Li, C. Chen, H. Su, and Q. Du, "Local binary patterns and extreme learning machine for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3681–3693, Jul. 2015.

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

[10] M. D. Mura, J. A. Benediktsson, and L. Bruzzone, "Classification of hyperspectral images with extended attribute profiles and feature extraction techniques," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGRSS)*, Jul. 2010, pp. 76–79.

[11] B. Liu *et al.*, "Morphological attribute profile cube and deep random forest for small sample classification of hyperspectral image," *IEEE Access*, vol. 8, pp. 117096–117108, 2020.

[12] Y. Qian, M. Ye, and J. Zhou, "Hyperspectral image classification based on structured sparse logistic regression and three-dimensional wavelet texture features," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 4, pp. 2276–2291, Apr. 2013.

[13] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Deep learning classifiers for hyperspectral imaging: A review," *ISPRS J. Photogramm. Remote Sens.*, vol. 158, pp. 279–317, Dec. 2019.

[14] C. Xing, L. Ma, and X. Yang, "Stacked denoise autoencoder based feature extraction and classification for hyperspectral images," *J. Sensors*, vol. 2016, pp. 1–10, Nov. 2016.

[15] 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.

[16] Y. Chen, X. Zhao, and X. Jia, "Spectral–spatial classification of hyperspectral data based on deep belief network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2381–2392, Jun. 2015.

[17] 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.

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

[19] 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, Sep. 2016.

[20] Y. Chen, L. Zhu, P. Ghamisi, X. Jia, G. Li, and L. Tang, "Hyperspectral images classification with Gabor filtering and convolutional neural network," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2355–2359, Dec. 2017.

[21] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. J. Plaza, and F. Pla, "Deep pyramidal residual networks for spectral–spatial hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, pp. 740–754, Feb. 2019.

[22] 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.

[23] 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.

[24] C. Zhang, G. Li, and S. Du, "Multi-scale dense networks for hyperspectral remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 9201–9222, Nov. 2019.

[25] H. Zhang, Y. Li, Y. Jiang, P. Wang, Q. Shen, and C. Shen, "Hyperspectral classification based on lightweight 3-D-CNN with transfer learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5813–5828, Aug. 2019.

[26] B. Liu, X. Yu, P. Zhang, X. Tan, R. Wang, and L. Zhi, "Spectral–spatial classification of hyperspectral image using three-dimensional convolution network," *J. Appl. Remote Sens.*, vol. 12, no. 1, pp. 1–18, 2018.

[27] Y. Chen, Y. Wang, Y. Gu, X. He, P. Ghamisi, and X. Jia, "Deep learning ensemble for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 6, pp. 1882–1897, Jun. 2019.

[28] 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.

[29] X. Liu, R. Wang, Z. Cai, Y. Cai, and X. Yin, "Deep multigrained cascade forest for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 8169–8183, Oct. 2019.

[30] H.-C. Li, W.-Y. Wang, L. Pan, W. Li, Q. Du, and R. Tao, "Robust capsule network based on maximum correntropy criterion for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 738–751, 2020.

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

[32] 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.

[33] S. Koda, F. Melgani, and R. Nishii, "Unsupervised spectral–spatial feature extraction with generalized autoencoder for hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 3, pp. 469–473, Mar. 2020.

[34] 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.

[35] 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.

[36] R. Kemker and C. Kanan, "Self-taught feature learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2693–2705, May 2017.

[37] 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.

[38] J. Zhao, X. Xie, X. Xu, and S. Sun, "Multi-view learning overview: Recent progress and new challenges," *Inf. Fusion*, vol. 38, pp. 43–54, Nov. 2017.

[39] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," pp. 1–16, 2019, *arXiv:1906.05849*. [Online]. Available: https://arxiv.org/abs/1906.05849v4

[40] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. Comput. Vis. Pattern Recognit.*, 2019, pp. 9729–9738.

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

[42] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. CVPR*, 2017, pp. 2261–2269.

[43] W. Song, S. Li, L. Fang, and T. Lu, "Hyperspectral image classification with deep feature fusion network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3173–3184, Jun. 2018.

[44] 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.

[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[46] 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.

[47] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[48] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. Int. Conf. Mach. Learn.*, vol. 99, 1999, pp. 200–209.

[49] L. Bruzzone, M. Chi, and M. Marconcini, "Transductive SVMs for semisupervised classification of hyperspectral data," in *Proc. IGARSS*, vol. 1, 2005, p. 4.

[50] W. Li and Q. Du, "Joint within-class collaborative representation for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2200–2208, Jun. 2014.

[51] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: http://arxiv.org/abs/1511.06434

[52] B. Liu, X. Yu, A. Yu, P. Zhang, G. Wan, and R. Wang, "Deep few-shot learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2290–2304, Apr. 2019.


**Bing Liu** received the B.S. degree in measurement and control engineering, the M.S. degree in pattern recognition and intelligent system, and the Ph.D. degree in surveying and mapping science and technology from Information Engineering University, Zhengzhou, China, in 2013, 2016, and 2019, respectively.

He is working at Information Engineering University as a Lecturer. His research interests include machine learning, pattern recognition, and signal processing in Earth observation.

Dr. Liu is an Active Reviewer of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE ACCESS, the *International Journal of Remote Sensing*, *Remote Sensing Letter*, and the *Journal of Applied Remote Sensing*.

**Anzhu Yu** received the B.S. degree in remote sensing science and technology and the M.S. degree in photogrammetry and remote sensing from PLA Strategic Support Force Information Engineering University, Zhengzhou, China, in 2011 and 2014, respectively, and the Ph.D. degree from the Institute of Surveying and Mapping, PLA Strategic Support Force Information Engineering University, in 2018.

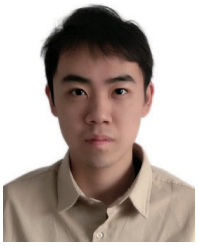
He is working at the Institute of Surveying and Mapping, PLA Strategic Support Force Information Engineering University, as an Associate Professor. His research interest includes signal processing in Earth observation.

**Xuchu Yu** received the Ph.D. degree from the Institute of Surveying and Mapping, Zhengzhou, China, in 1997.

He is working at Information Engineering University, Zhengzhou, as a Professor and Doctoral Supervisor. His research interests include photogrammetry, remote sensing, and pattern recognition.

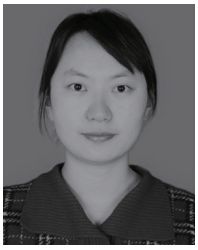
**Ruirui Wang** received the B.S. degree in surveying and mapping engineering from Xuchang University, Xuchang, China, in 2014, and the B.S. degree in photogrammetry and remote sensing from Information Engineering University, Zhengzhou, China, in 2017.

She is working at the Institute of Surveying Mapping and Geo Information of Henan, Zhengzhou, as an Assistant Engineer. Her research interest includes machine learning and feature extraction.



**Kuiliang Gao** received the B.S. degree in remote sensing science and technology from Information Engineering University, Zhengzhou, China, in 2019, where he is pursuing the M.S. degree.

His research interests include hyperspectral image processing, pattern recognition, and deep learning.



**Wenyue Guo** received the bachelor's and master's degrees in cartography and geographic information engineering and the Ph.D. degree in surveying and mapping from Information Engineering University, Zhengzhou, China, in 2012, 2015, and 2018, respectively.

She is working at PLA Strategic Support Force Information Engineering University, Zhengzhou, as a Lecturer. Her research interests include geographic information science and graph representation.