Self-supervised Learning with Adaptive Distillation for Hyperspectral Image Classification

Jun Yue, Leyuan Fang, *Senior Member, IEEE,* Hossein Rahmani, and Pedram Ghamisi, *Senior Member, IEEE*

***Abstract*—Hyperspectral image (HSI) classification is an important topic in the community of remote sensing, which has a wide range of applications on geoscience. Recently, deep learning-based methods have been widely used in HSI classification. However, due to the scarcity of labeled samples in HSI, the potential of deep learning-based methods has not been fully exploited. To solve this problem, a self-supervised learning (SSL) method with adaptive distillation is proposed to train the deep neural network with extensive unlabeled samples. The proposed method consists of two modules: adaptive knowledge distillation with spatial-spectral similarity and 3D transformation on HSI cubes. The SSL with adaptive knowledge distillation uses the self-supervised information to train the network by knowledge distillation, where self-supervised knowledge is the adaptive soft label generated by spatial-spectral similarity measurement. The SSL with adaptive knowledge distillation mainly includes the following three steps. First, the similarity between unlabeled samples and object classes in HSI is generated based on the spatial-spectral joint distance (SSJD) between unlabeled samples and labeled samples. Second, the adaptive soft label of each unlabeled sample is generated to measure the probability that the unlabeled sample belongs to each object class. Third, a progressive convolutional network (PCN) is trained by minimizing the cross entropy between the adaptive soft labels and the probabilities generated by the forward propagation of the PCN. The SSL with 3D transformation rotates the HSI cube in both the spectral domain and the spatial domain to fully exploit the labeled samples. Experiments on three public HSI datasets have demonstrated that the proposed method can achieve better performance than existing state-of-the-art methods.**

***Index Terms*—Deep neural network, hyperspectral image classification, self-supervised learning, knowledge distillation, spatial-spectral feature extraction.**

I. Introduction

HYPERSPECTRAL images (HSIs) usually contain hundreds of spectral channels, and each pixel in HSI has a corresponding spectral curve [1], [2]. Combining both spatial texture information and spectral reflectance information, HSIs have been widely used in many fields, e.g., agriculture, marine monitoring, geological exploration, environment, and ecology. HSI classification is utilized to determine the corresponding object category for each pixel (e.g., soil, trees, grass, building, roads, and river) which plays a key role in many remote sensing applications [3], [4].

In recent decades, many HSI classification methods have been proposed, including spectral matching-based methods, which determine the object types of samples based on the matching degree of spectral curves. Typical spectral matching-based methods include spectral angle mapper (SAM) [5], [6] and spectral information divergence (SID) [7]. Due to the high spectral dimensionality in HSIs, many statistical feature reduction methods have been proposed to transform spectral vector of HSI from high-dimensional feature space to low-dimensional feature space, including linear discriminant analysis [8], principal components analysis (PCA) [9], independent component analysis (ICA) [10] and minimum noise fraction (MNF) [11]. In addition, because of the nonlinear characteristics of HSI, many nonlinear feature extraction methods have been proposed in recent years, such as manifold coordinate representations [12], [13], [14], locality-preserving discriminant analysis [15], locality preserving projections (LPP) [16] and sparsity preserving projections (SPP) [17].

Due to the characteristics of spatial homogeneity and heterogeneity of HSI, it is hard to make full use of the features of HSI only by spectral feature extraction. Therefore, researchers have proposed a series of joint spatial-spectral feature extraction methods [18], [19]. The typical one is extended morphological profile (EMP) [20]. The first step of this method is to extract features by PCA, and then generate spatial features by morphological operations (opening or closing operations). Recently, morphological-based joint spatial-spectral feature extraction methods have been continuously improved, including extended attribute profile (EAP) [21], and directional morphological profiles (DMP) [22].

With the introduction and rapid development of deep learning [23], [24], [25], it has achieved great success in many tasks such as image classification [26], [27], [28], object detection [29], [30], [31], image segmentation [32], [33], [34], scene recognition [35], natural language processing [36], [37], optical character recognition [38] and so on. At the same time, researchers have proposed a series of deep learning-based methods for HSI classification, and the classification accuracy of HSI has been gradually improved [18]. To fully exploit spatial-spectral features of HSI, some joint spatial-spectral feature extraction methods with deep neural network have been proposed [39], including stacked autoencoders [40], deep

This work was supported by the National Natural Science Fund of China under Grant 61922029 and the Science and Technology Plan Project Fund of Hunan Province under Grant 2019RS2016. *(Corresponding author: Leyuan Fang.)*

J. Yue is with the Department of Geomatics Engineering, Changsha University of Science & Technology, Changsha, 410114, China (e-mail: jyue@pku.edu.cn).

L. Fang is with the College of Electrical and Information Engineering, Hunan University, Changsha, 410082, China, and also with the Peng Cheng Laboratory, Shenzhen, 518000, China (e-mail: fangleyuan@gmail.com).

H. Rahmani is with the School of Computing and Communications, Lancaster University, Lancaster, LA1 4YW, U.K. (e-mail: h.rahmani@lancaster.ac.uk).

P. Ghamisi is with the Helmholtz-Zentrum Dresden-Rossendorf (HZDR), Helmholtz Institute Freiberg for Resource Technology (HIF), Exploration, D-09599 Freiberg, Germany (e-mail: p.ghamisi@gmail.com).

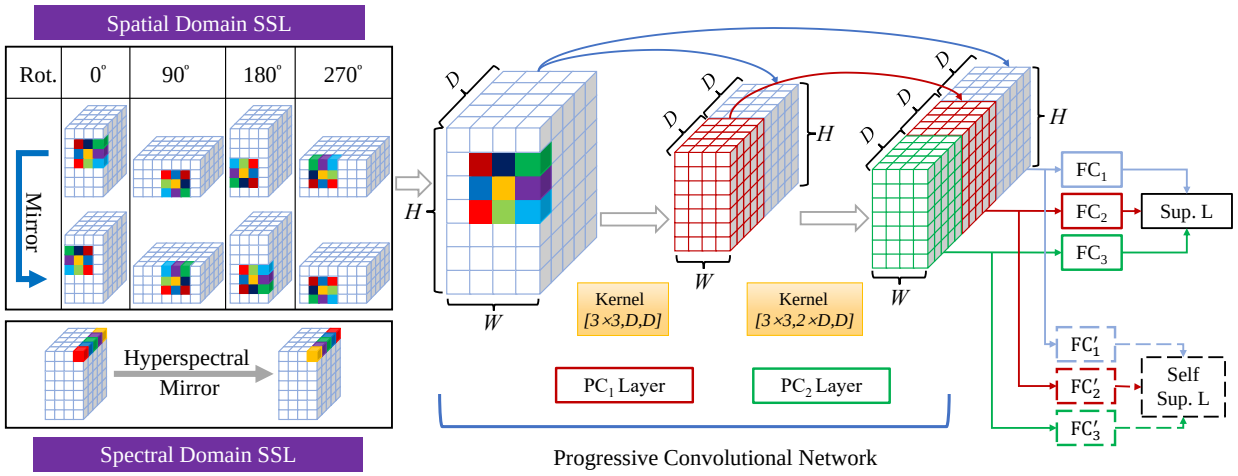

Fig. 1: Self-supervised learning with 3D transformation and progressive convolutional network, where PC and FC are abbreviations for progressive convolution and fully connected layer.

belief network [41], superpixel-based discriminative sparse model [42] and local covariance matrix representation [43].

However, the application of deep learning for HSI classification has not been well-addressed. One of the important reasons is that deep learning-based method requires a large number of labeled samples to obtain satisfactory accuracy. In the process of supervised HSI classification, there are usually two ways to obtain training samples: (1) field investigation; and (2) visual interpretation directly from high-resolution images. In particular, the training samples collected through field investigation can usually lead to higher classification accuracy [44]. However, compared with visual interpretation, field investigation is costly, complex and time-consuming, which greatly limits the number of training samples. In fact, it is difficult to obtain enough training samples to fully meet the training requirements of the deep neural network, which limits the researchers' attempts to further improve the classification accuracy of HSI by using deep neural network.

To solve this problem, several HSI classification methods based on few-shot learning have been proposed. Few-shot learning is an important branch of machine learning which is designed to address the problem of limited training samples. Usually, metric learning and meta learning strategies are used to learn the features of HSIs, and the absolute distance or relative distance between features is used to identify the object class of each unlabeled pixel in HSI [45], [46]. However, these methods still do not solve the problem of lack of samples very well. As a result, researchers began to explore some solutions to get supervision from the data itself, one of which is self-supervised learning (SSL). SSL mines its own supervised information from large-scale unsupervised data. In other words, the supervised information used in SSL does not need to be obtained by manual annotation, but the algorithm automatically constructs the supervised information from unsupervised data to train the model [47], [48]. The existing SSL methods include the method based on spatial relationship [49], [50], inpainting [51], image reconstruction [52], color transformation [53], super resolution [54] and spatial rotation transformation [55], [56], etc. In the field of HSI classification, some super-pixel based self-supervised learning methods have been proposed to fully exploit each labeled samples [57], [58]. However, none of the above methods generate corresponding soft labels for unlabeled samples and use them to train the network. From the perspective of training data transformation, the existing SSL-based methods for image classification are based on transformation in two-dimensional space. Since hyperspectral data is a three-dimensional cube with two-dimensional spatial dimension and spectral dimension, it is necessary to update the SSL with 2D transformation to SSL with 3D transformation.

In this paper, we propose a self-supervised learning method with adaptive distillation (SSAD) to train a deep neural network with a large number of unlabeled samples. The SSAD mainly consists of two modules: SSL with adaptive knowledge distillation and SSL with 3D transformation. In the adaptive knowledge distillation, the unlabeled samples are adaptively labeled based on the spatial-spectral distance measurement. By calculating the similarity between unlabeled samples and pre-defined object classes, the adaptive soft label of each unlabeled samples is generated. The adaptive soft label includes the probability that the sample belongs to each pre-defined object class. The proposed adaptive knowledge distillation method mainly includes the following steps. First, in order to measure the similarity between samples in an HSI, the spatial-spectral joint distance considering both spectral Kullback-Leibler (KL) divergence and spatial Euclidean distance is calculated. Second, based on the spatial-spectral joint distance (SSJD) between unlabeled samples and labeled samples, the distance between unlabeled samples and pre-defined object classes in HSI is generated. Third, the adaptive soft label of each unlabeled sample is generated by distance-probability transformation method. Fourth, the proposed progressive convolutional network (PCN) is trained by minimizing the cross entropy between the probabilities generated by PCN and the adaptive soft label of each unlabeled samples. In addition, SSL with 3D transformation further exploits each labeled sample

by transforming HSI in spectral and spatial domains.

The main contributions of this paper are threefold.

(1) An unsupervised similarity measurement method between samples is proposed. This method considers both the spectral distance and the spatial distance between samples in HSI. Based on the SSJD, the similarity between each unlabeled sample and each pre-defined object class is generated.

(2) A self-supervised learning strategy with adaptive soft label is proposed. The adaptive soft label of each unlabeled sample measures the probability that each unlabeled sample belongs to each pre-defined object class. We add the adaptive soft label to the FCN model training process by calculating the cross entropy between the adaptive soft label and the output of the FCN model.

(3) A self-supervised learning strategy with 3D transformation is proposed by rotating the original HSI image in spectral domain and spatial domain.

This paper is organized as follows. In Section II, the proposed method is explained in detail, including spatial-spectral joint distance, adaptive soft label and adaptive knowledge distillation, self-supervised learning strategy with 3D transformation. In Section III, the experimental data and parameters are described. In Section IV, the results are shown and discussed, and the ablation study is analyzed. In Section V, the conclusions of this paper are summarized.

II. METHOD

In this paper, an SSL method is proposed, which consists of two main modules: SSL with 3D transformation and SSL with adaptive knowledge distillation. These two modules improve the existing SSL methods from the aspects of data input, self-supervised label generation, and self-supervised training.

A. Self-supervised learning with 3D transformation

The existing SSL methods include the approaches that are based on spatial relationship [49], [50], and the approaches that are based on spatial rotation transformation [55], [56], etc. But these methods are based on operations in two-dimensional space. This paper presents an SSL method for geometric transformation in three-dimensional space. Since HSIs are three-dimensional cubes with 2D spatial dimension and 1D spectral dimension, it can be rotated and flipped in three-dimensional space. The details of the method are shown in Fig. 1.

1) Spatial and spectral domain geometric transformation: Different from the nature images, the HSIs are rotational invariance and symmetry in spatial domain. In other words, the horizontal rotation in spatial domain does not change the predictions of the pixels in HSIs. Therefore, we rotate and mirror the HSI in spatial domain to promote the robustness of the HSI classification model. The cross entropy loss can be calculated by rotating the ground-truth map with the same horizontal rotation operation as the input HSI. In our implementation, We define four horizontal rotation operations, and the corresponding rotation angle set is $\mathcal{T} = \{90^\circ \cdot i \mid i \in [0, 1, 2, 3]\}$. Specifically, given an HSI $\mathcal{H}$, we first generate its four horizontal rotated replicas $\{\mathcal{H}^t \mid t \in \mathcal{T}\}$, where $\mathcal{H}^t$ is the replica of HSI $\mathcal{H}$ rotated by t degrees. We define mirroring in 2D spatial dimension as horizontal mirror. For each HSI $\mathcal{H}^t$, we perform a horizontal mirror operation as shown in Fig. 1 to generate the corresponding mirroring HSI $\bar{\mathcal{H}}^t$.

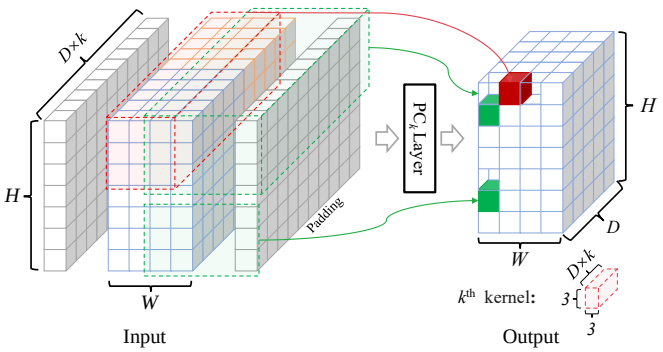

Fig. 2: The operations of the k^{th} progressive convolution layer. The size of the input matrix and the output matrix of the k^{th} PC layer are $[W, H, k \times D]$ and $[W, H, D]$, respectively.

Since HSIs have three dimensions (i.e., 2D spatial dimension and 1D spectral dimension), this paper proposes to rotate HSIs in the spectral domain and predict the order of spectral sequence. We define two kinds of spectral sequences. One is frequency from high to low and its label is defined as 1. The other is frequency from low to high and its label is defined as 0. We define mirroring in spectral dimension as hyperspectral mirror operation. Combining three operations (i.e., horizontal rotation, horizontal mirror and hyperspectral mirror), the training set can be denoted as $\{\mathcal{H}_1^t \cup \bar{\mathcal{H}}_1^t \cup \mathcal{H}_0^t \cup \bar{\mathcal{H}}_0^t \mid t \in \mathcal{T}\}$, where $\mathcal{H}_1^t$ is the HSI $\mathcal{H}$ rotated by t degrees, and its spectral sequence is arranged from high frequency to low frequency. $\mathcal{H}_0^t$ is the HSI $\mathcal{H}$ rotated by t degrees, and its spectral sequence is arranged from low frequency to high frequency. $\bar{\mathcal{H}}_1^t$ is the HSI $\bar{\mathcal{H}}$ rotated by t degrees, and its spectral sequence is arranged from high frequency to low frequency. $\bar{\mathcal{H}}_0^t$ is the HSI $\bar{\mathcal{H}}$ rotated by t degrees, and its spectral sequence is arranged from low frequency to high frequency.

2) Progressive convolutional network: In order to realize the geometric transformation in spatial and spectral domain, a fully convolutional neural network called progressive convolutional network (PCN) is designed. In PCN, the output of the forward propagation rotates with the HSI. The operations of the k^{th} progressive convolution (PC) layer are shown in Fig. 2. In the k^{th} PC layer, the input matrix size is $W \times H$ and the number of channels is $D \times k$, where W and H are the width and height of the HSI, respectively. The input matrix of the k^{th} PC layer is formed by concatenating the outputs of the 0^{th} to $(k-1)^{\text{th}}$ PC layer. First, we use the padding strategy around the input of the 0^{th} PC layer. Second, for the $(k-1)^{\text{th}}$ PC layer, we employ D PC filters to generate the feature matrix $\mathcal{O}_{k-1}$. Third, for the k^{th} PC layer, we concatenate the outputs across all the previous PC layers into a matrix with a size of $W \times H \times D \times k$ as the input of the k^{th} PC layer. The operations of the 0^{th} PC layer of PCN can be formulated as

follows:

$$\begin{cases} \mathcal{I}_0 = \mathcal{H} & , k = 0 \\ \mathcal{O}_0 = \mathrm{PC}_0(\mathcal{I}_0) & , k = 0 \end{cases} \tag{1}$$

where $\mathcal{O}_0$ represents the output of the 0^{th} PC layer. Then, the input and output of the k^{th} PC layer can be formulated as follows:

$$\begin{cases} \mathcal{I}_k = [\mathcal{O}_0, \mathcal{O}_1, \ldots, \mathcal{O}_{k-1}] & , k > 0 \\ \mathcal{O}_k = \mathrm{PC}_k(\mathcal{I}_k) & , k > 0 \end{cases} \tag{2}$$

where $\mathcal{I}_k$ and $\mathcal{O}_k$ is the input and output of the k^{th} PC layer, respectively. In each PC layer, the activation function ReLU (Rectified Linear Unit) [59] is applied to promote the learning ability of the PCN. In addition, the dropout strategy [60] is adopted to improve the generalization ability of the model and avoid over-fitting. In convolutional neural networks, the deeper the layers, the larger the receptive field. Therefore, the proposed PCN structure can perform convolution operation on multi-scale receptive fields and extract multi-scale features effectively.

3) Multi-PC layers fusion: The self-supervised learning with 3D transformation strategy enlarges the input hyperspectral data, and the corresponding classification results are expanded accordingly. In each PC layer, we introduce a fully-connected (FC) layer to transfer each pixel feature into class predictions (logits). The calculations are defined as follows:

$$\alpha_k = \mathrm{FC}_k(\mathcal{O}_k) = \mathcal{O}_k \cdot W_k + b_k \quad , k > 0 \tag{3}$$

where α_k is the prediction of the k^{th} PC layer. W_k and b_k are the parameters of the k^{th} FC layer. Thus, given the inputs $\{\mathcal{H}_1^t \cup \bar{\mathcal{H}}_1^t \cup \mathcal{H}_0^t \cup \bar{\mathcal{H}}_0^t \mid t \in \mathcal{T}\}$ of PCN, the output α_k of the k^{th} PC layer is a set of logits, i.e. $\mathcal{Q}_k = \{\alpha_k^t \cup \bar{\alpha}_k^t \mid t \in \mathcal{T}\}$. We then fuse the set of logits to generate the merged result of the k^{th} PC layer by

$$\tilde{\alpha}_k = \frac{1}{|\mathcal{Q}_k|} \sum \mathcal{L}(\dot{\alpha}_k^t), \forall \dot{\alpha}_k^t \in \mathcal{Q}_k \tag{4}$$

where $\mathcal{L}$ is the *softmax* operation and $|\mathcal{Q}_k|$ is the size of the logits set.

Furthermore, we fuse the output logits generated from multi-PC layers. Given the output logits set $\mathcal{R} = \{\tilde{\alpha}_k \mid k \in [1, 2, \ldots N_k]\}$ which were generated by the 1^{st} to the N_k^{th} PC layers, we use the *softmax* function to normalize each logits vector, and sum all normalized logits vectors from multi-layers as follows:

$$\alpha_o = \frac{1}{N_k} \sum_{k=1}^{N_k} \mathcal{L}(\tilde{\alpha}_k), \forall \tilde{\alpha}_k \in \mathcal{R} \tag{5}$$

where α_o is the final prediction of our proposed network, N_k is the total number of PC layers. Finally, we apply the *argmax* function on α_o and generate the HSI pixel-level labels with the maximum logit value.

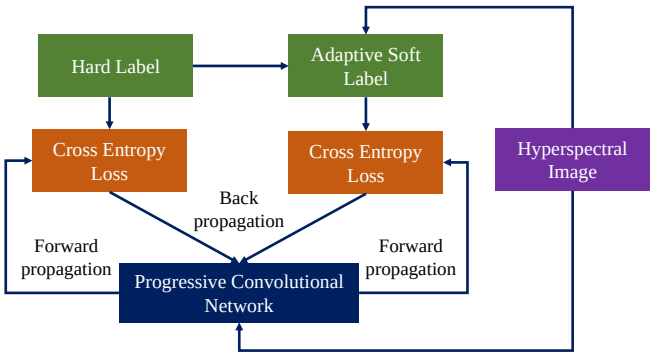

Fig. 3: Pipeline of SSL with adaptive knowledge distillation.

B. Self-supervised learning with adaptive knowledge distillation

Knowledge distillation is a training strategy by transferring the knowledge from one teacher network with a higher precision and larger model size to a student network with a smaller model size. Knowledge distillation uses a larger network to generate soft label, which is used to guide the training of the student network. In this paper, adaptive knowledge distillation is proposed by transferring the self-supervised knowledge to the network with adaptive soft label and hard label. The flow chart of SSAD distillation is shown in Fig. 3. The SSAD method consists of two steps. The first step is to generate adaptive soft labels for all unlabeled samples. The second step is to distill the self-supervised adaptive soft labels into the network.

Algorithm 1 Adaptive soft label generation

Input:
The labels set $\mathcal{C}$
The labeled samples set $\mathcal{L}$
The unlabeled samples set $\mathcal{U}$
Output:
The adaptive soft labels set $\mathcal{S}$;

for $u_i \in \mathcal{U}$ **do**
  for $l_i \in \mathcal{L}$ **do**
    Compute the Euclidean Distance between u_i and l_i
    Compute the Kullback-Leibler divergence between u_i and l_i
    Compute SSJD(u_i, l_i) between u_i and l_i by Eq. (6)
  end for
end for
for $u_i \in \mathcal{U}$ **do**
  for $c_i \in \mathcal{C}$ **do**
    Compute the average SSJD(u_i, c_i) between u_i and class c_i by Eq. (8)
    Calculate the probability $P(\Phi(u_i) = c_i)$ that u_i belongs to class c_i by Eq. (9)
  end for
  Generate the adaptive soft label $\mathcal{S}(u_i)$ by Eq. (10)
  Add $\mathcal{S}(u_i)$ to adaptive soft label set $\mathcal{S}$
end for

1) Adaptive soft label generation: Given a hyperspectral image $\mathcal{H}$, we denote its labeled samples set, unlabeled samples set and labels set as $\mathcal{L}, \mathcal{U}$ and $\mathcal{C}$. In order to label all unlabeled samples based on spectral-spatial correlation between $\mathcal{L}$ and $\mathcal{U}$, we first define the spatial-spectral joint distance between samples (SSJD). SSJD is composed of spatial distance and spectral distance, which can be calculated by

$$\mathrm{SSJD}(u_i, l_i) = \sqrt{\mathrm{ED}(u_i, l_i) \cdot \mathrm{SID}(u_i, l_i)} \tag{6}$$

where $l_i \in \mathcal{L}$, $u_i \in \mathcal{U}$, $\mathrm{ED}(u_i, l_i)$ is the Euclidean Distance between l_i and u_i, $\mathrm{SID}(u_i, l_i)$ is the spectral information divergence [10] between u_i and l_i which can be formulated as:

$$\mathrm{SID}(u_i, l_i) = \mathrm{KL}(u_i \parallel l_i) + \mathrm{KL}(l_i \parallel u_i) \tag{7}$$

where $\mathrm{KL}(l_i \parallel u_i))$ denotes the relative entropy of u_i with respect to l_i, which is also known as Kullback–Leibler divergence. Based on SSJD between samples, we define the spatial-spectral joint distance between sample and class, which can be formulated as:

$$\mathrm{SSJD}(u_i, c_i) = \sum_{r=1}^{N_{c_i}} \frac{\mathrm{SSJD}(u_i, l_r)}{|\mathcal{C}|^{\mathrm{n}(u_i, l_r)}}, \forall \Phi(l_r) = c_i \tag{8}$$

where $c_i \in \mathcal{C}$, $\Phi(l_r)$ denotes the label of l_r, $|\mathcal{C}|$ is the size of the labels set, N_{c_i} is the total number of labeled samples which belongs to class c_i, $\mathrm{n}(u_i, l_r)$ is the ordinal by sorting all SSJDs between u_i and $l_r, \forall \Phi(l_r) = c_i$. Given the distance between sample and class, the probability that unlabeled sample u_i belonging to class c_i can be formulated as:

$$\mathrm{P}(\Phi(u_i) = c_i) = \frac{e^{-\mathrm{SSJD}(u_i, c_i)|\mathcal{C}|}}{\sum_{r=1}^{|\mathcal{C}|} e^{-\mathrm{SSJD}(u_i, c_r)|\mathcal{C}|}} \tag{9}$$

where e is the natural constant. Finally, the adaptive soft label of u_i can be generated by

$$\begin{aligned} \mathcal{S}(u_i) = [&\mathrm{P}(\Phi(u_i) = c_1), \mathrm{P}(\Phi(u_i) = c_2), ..., \\ &\mathrm{P}(\Phi(u_i) = c_{|\mathcal{C}|})] \end{aligned} \tag{10}$$

The procedure of the adaptive soft label generation is given in Algorithm 1.

2) Adaptive knowledge distillation: We design an optimization method called adaptive knowledge distillation to help PCN learn from both the hard labels and the adaptive soft labels. We first calculate the loss between the prediction α_k and the hard labels $\mathcal{G}$:

$$\mathcal{L}_k^{\mathcal{G}} = \Delta(\alpha_k, \mathcal{G}) = \Delta(\delta(\alpha_k), \mathcal{G}) \tag{11}$$

where Δ is *cross-entropy* function and δ is *softmax* operation.

Second, we calculate the loss between the prediction α_k and the adaptive soft labels $\mathcal{S}$ by

$$\mathcal{L}_k^{\mathcal{S}} = \Delta(\delta(\alpha_k), \mathcal{S}) \tag{12}$$

In addition, we introduce another FC layer after each PC layer to transfer each pixel feature into spectral sequence predictions. The forward propagation of spectral sequence prediction is formulated by

$$s_k^s = \mathrm{FC}_k^s(O_k) = O_k \cdot W_k^s + b_k^s \quad , k > 0 \tag{13}$$

where W_k^s and b_k^s are the weights and bias of the k^{th} FC layer for predicting the spectral sequence, respectively. The self-supervised loss on spectral domain can be formulated as follows:

$$\mathcal{L}_k^S = \Delta(\alpha_k^s, y) \tag{14}$$

where $y \in \{0, 1\}$ represents the order of the spectral sequence. $y = 1$ indicates that the frequency of the input hyperspectral data is from high to low. Otherwise, $y = 0$ indicates that the frequency of the input hyperspectral data is from low to high. Finally, the total loss for training PCN is defined as follows:

$$\mathcal{L} = \frac{1}{N_k} \sum_{k=1}^{N_k} (\mathcal{L}_k^G + \mathcal{L}_k^P + \mathcal{L}_k^S) \tag{15}$$

where N_K is the total number of PC layers.

TABLE I: The details of the HSI datasets, where W, H and N_B is the width, the height and the number of spectral channels of the HSI, respectively. N_C is the number of object categories and N_A is the number of activating pixels.

Datasets	W	H	N_B	N_C	N_A
IP	145	145	200	16	10249
UP	610	340	103	9	42776
HS	1905	349	144	15	15029

III. EXPERIMENTS

A. Datasets and evaluation

To demonstrate the effectiveness of the proposed method, experiments are conducted on three well-known HSI datasets including Indian Pines (IP), University of Pavia (UP) and Houston (HS) [1]. The details of the three datasets are listed in Table I.

(1) IP dataset: This dataset was collected by Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensors in northwestern Indiana. The spatial resolution of this image is 20 m covering spectral channels from 200 nm to 2400 nm. This image is consisting of 349 by 1905 pixels. Before data preprocessing, the spectral channels affected by noise and water absorption (104–108, 150- 163, and 220) are removed, and the remaining 200 spectral channels are involved in the experiment. The corresponding ground-truth of IP dataset contains 16 object categories. The false color composite image of IP dataset and its corresponding ground-truth map is shown in Fig. 4.

(2) UP dataset: This dataset was collected by the Reflective Optics Systems Imaging Spectrometer (ROSIS) sensor in Pavia, Italy. This image is mainly composed of urban features around the University of Pavia with 610 by 340 pixels. The

[1]We will release the source code on GitHub after the paper is accepted.

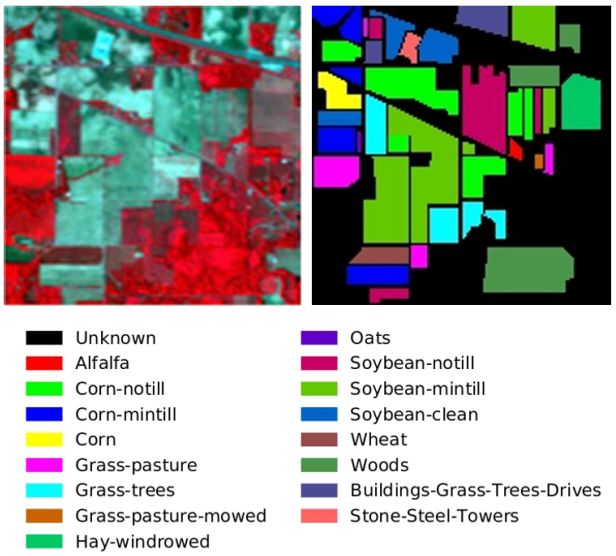

Fig. 4: The false color composite image and the corresponding ground-truth map of the IP dataset.

Fig. 5: The false color composite image and the corresponding ground-truth map of the UP dataset.

spatial resolution of this image is 1.3 m covering spectral bands from 430 nm to 860 nm. UP dataset contains 115 spectral bands, 12 of which are removed becasue of noise and water absorption, and the remaining 103 spectral channels are involved in the experiment. The corresponding ground-truth of UP dataset contains 9 object categories. The false color composite image and the corresponding ground-truth map of the UP dataset is shown in Fig. 5.

(3) HS dataset: This dataset was collected by the ITRES-CASI 1500 sensor [61]. This image is mainly composed of urban features around the University of Houston with 349 by 1905 pixels. The spatial resolution of this image is 2.5 m covering spectral bands from 380 nm to 1050 nm. HS dataset contains 144 spectral bands. The corresponding ground-truth of UP dataset contains 15 object categories. The false color image and the corresponding ground-truth map of HS dataset is shown in Fig. 6.

The proposed method was evaluated on several tasks. In each task, L labeled samples of each object category are randomly sampled for training, and the rest of the activating samples are used for testing. To report the classification results, we conduct 10 independent runs for each task and compute the mean OA (overall accuracy), AA (average accuracy) and kappa coefficient with standard deviation over the 10 runs.

B. Parameter settings

Defining the format of the parameters of the $k^{\text{th}}(k > 0)$ PC layer as $[kernel\ size, N_i, N_o, padding, stride]$, where N_i and N_o are the number of input filters and output filters, respectively. The proposed PCN is a fully convolutional neural network and the input size of the PCN is $W \times H \times N_B$ for each epoch. Specifically, the input sizes of IP, UP and HS datasets are $145 \times 145 \times 200$, $610 \times 340 \times 103$ and $1905 \times 349 \times 144$, respectively. We set the parameters of the k^{th} PC layer as $[3 \times 3, D, k \times D, 1, 1]$. As a result, the size of the input matrix and the output matrix of the k^{th} PC layer are $[W, H, k \times D]$ and $[W, H, D]$, respectively. In our proposed PCN, the embedding operation of each PC layer is transferring a matrix with N_i filters into a matrix with N_o filters and we set $D = N_B$ in our experiments. Thus, given an HSI with N_B bands and N_C categories, the parameters W and b of each FC are a matrix in $\mathbb{R}^{N_B \times N_C}$ and a vector in $\mathbb{R}^{N_C}$, respectively.

We set the range of the depth N_K of our proposed network from 1 to 15, and train these PCNs only under the *cross-entropy* loss by Eq. (11). During training the model, we remove the self-supervised learning with 3D transformation module. For each depths of PCN, we conduct 10 independent runs, and the average accuracy over 10 runs as well as the 70% confidence intervals is in shown in Fig. 7. For each run, we fuse the output predictions of all PC layers by Eq. (5). As depicted in Fig. 7, with the increasing of the depth of PCN, the average accuracy of the PCN rises first and falls later. As the depth of PCN increases, the receptive field increases and the oversized receptive field results in redundancy of input information. Therefore, we set the depth of PCN to be 3 since the PCN with 3 PC layers requires less computation and achieves better performance.

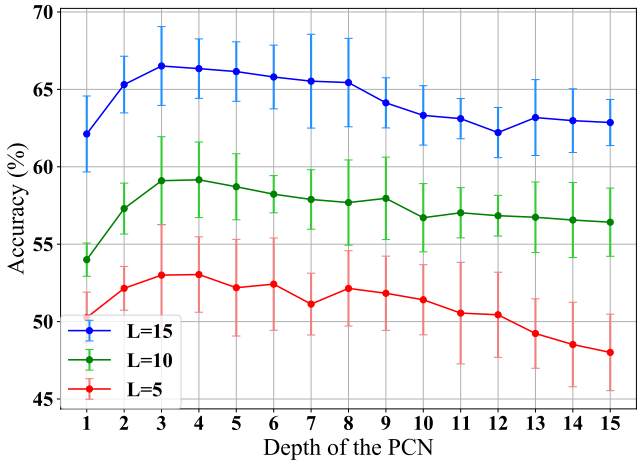

Fig. 7: The accuracies of PCN with different layers on IP dataset.

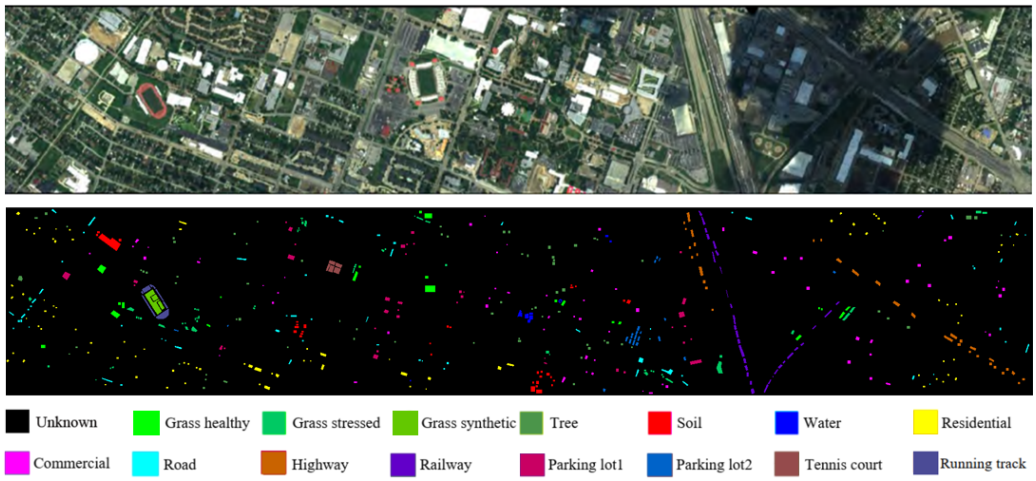

Fig. 6: The false color composite image and the corresponding ground-truth map of the HS dataset.

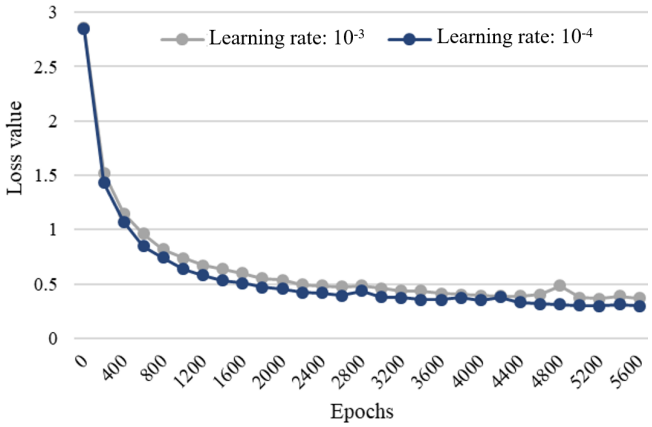

Fig. 8: Learning curves under different learning rates. The loss values are recorded every 200 epochs.

In the proposed PCN, the parameter of the dropout (dropout probability) is $\rho = 0.5$, and all the dropout layers are removed in the testing stage. To avoid outliers, for each unlabeled samples, we set the minimum SSJD to be no larger than 0.085. During the training stage, we train our network by Eq. (15) and use Adam to perform stochastic optimization. The learning rate determines the convergence of the model, which indirectly affects the classification performance. Referring to some relevant experiments [62], [63], we discussed the influence of learning rate on the loss value at 10^{-3} and 10^{-4}. As depicted in Fig. 8, the learning curves show that when the learning rate is 10^{-4}, a smaller and more stable loss value can be obtained.We set the initial learning rate and the weight decay to 10^{-4} and 10^{-5}, respectively. After each 1000 iterations, the learning rate is reduced by 0.1.

IV. RESULTS AND DISCUSSION

A. Accuracies

In this study, for the IP and the UP datasets, the experiments were conducted with the L number of supervised samples for classification as 5 and 10, respectively, due to the limited number of labeled samples for a class (i.e., 'Oats'). For the HS dataset, L was set to 50 and 80, respectively, because this dataset has relatively larger number of training samples. The classification maps are shown in Figs. 9-11.

We compare our method with state-of-the-art HSI classification ones under the small-scale training samples. The compared methods include 2D-CNN, 3D-CNN [64], DRNN (deep recurrent neural networks) [61], DBMA (double-branch multi-attention mechanism network) [65], MSDN (end-to-end 3-D dense convolutional network) [66] and DFSL (deep few-shot learning for HSI classification) [45]. The comparisons of IP, UP, and HS datasets are shown in the Table II-IV, respectively.

B. Ablation study

The proposed method in this paper mainly consists of two strategies, i.e., SSL with 3D transformation and adaptive knowledge distillation. To demonstrate the effectiveness of each strategy, an ablation study was performed on IP dataset. In other words, the accuracy of the proposed method was tested when the two strategies were removed.

1) Effectiveness of SSL with 3D transformation: In this ablation study, the SSL with 3D transformation is used in both the training and the testing stages. Thus, the output of PCN with 3 PC layers is a set of logits: $\mathcal{Q} = \{\alpha_k^t \cup \bar{\alpha}_k^t \mid t \in \mathcal{T}, k \in [1, 2, 3]\}$. Then, the output logits are fused by Eq. (4) to generate the prediction $\tilde{\alpha}_k$ of the k^{th} PC layer. The accuracies are shown in Table V, where 'w/ SSL-3DT' means the network was trained with the 3D transformation strategy (3DT). The 'Baseline' method is a pure 3 layer PCN without the 3D transformation strategy. Compared with the results generated by the 'Baseline' method, the results generate by PCN with the 3D transformation strategy perform better in all settings. Meanwhile, the results generated by the deeper PC layer are higher than these generated by previous PC layers. Finally, we fuse the output logits of all the PC layers by Eq. (5) to generate the fused results: α_o. As shown in Table V, the fused results α_o perform the best accuracies. These improvements indicate the usefulness of SSL with 3D transformation.

TABLE II: Classification results (%) by using 5 and 10 labeled samples for each class on the IP dataset (the bold value is the best accuracy in each case).

Class	2D-CNN		3D-CNN		DRNN		DBMA		MSDN		DFSL		SSAD(Ours)	
	L=5	L=10	L=5	L=10	L=5	L=10	L=5	L=10	L=5	L=10	L=5	L=10	L=5	L=10
1	73.3±3.1	77.3±2.7	80.8±2.9	88.9±2.7	79.5±3.9	87.0±3.5	91.8±3.2	92.5±4.4	85.0±4.7	95.7±1.3	83.3±1.6	92.7±1.8	97.3±1.8	99.3±0.3
2	62.2±9.6	71.0±8.7	61.3±8.7	69.4±9.2	71.0±7.6	79.3±5.3	59.7±11.9	67.3±10.6	54.8±12.0	70.5±5.6	53.1±7.1	64.1±5.9	70.4±4.1	80.9±2.5
3	55.1±9.2	61.0±8.2	67.4±6.7	65.3±9.1	62.2±8.6	67.3±7.6	64.9±10.9	73.5±9.6	61.4±10.9	74.7±4.8	63.7±7.3	70.7±5.4	75.1±3.8	79.7±2.3
4	63.0±5.2	65.7±7.6	69.0±8.1	73.5±1.6	67.6±3.7	74.9±4.9	89.3±2.1	79.8±2.6	81.1±7.3	85.3±4.7	90.1±5.6	90.7±3.9	79.8±6.7	83.5±3.4
5	60.6±9.1	63.6±9.1	70.4±7.9	68.7±8.2	65.2±8.7	71.9±8.4	66.8±8.3	74.8±12.7	60.2±7.2	75.3±8.0	65.0±10.0	71.0±5.4	75.0±1.9	82.1±2.3
6	62.4±7.1	66.0±9.3	72.4±6.4	70.6±9.5	66.4±7.5	73.7±6.6	66.5±13.1	78.7±11.8	65.1±15.6	80.0±6.6	71.7±8.3	69.3±7.9	81.6±6.2	83.0±2.4
7	70.9±4.5	70.2±4.4	82.8±5.1	75.1±5.1	74.5±4.8	81.9±4.9	91.3±1.3	73.1±2.5	83.9±2.5	84.8±0.0	90.7±3.1	100.0±0.0	92.6±3.0	89.4±3.6
8	66.0±6.1	78.0±5.3	80.7±9.5	86.8±4.0	78.0±4.4	84.1±5.0	90.4±3.7	90.7±3.8	84.2±6.0	92.4±2.8	91.9±3.5	90.5±3.4	95.3±4.5	97.8±2.5
9	80.3±4.6	89.6±2.5	83.6±2.3	100.0±0.0	89.3±3.3	98.5±1.2	85.9±7.1	84.6±6.9	81.1±13.9	90.8±1.4	94.7±1.8	89.5±4.9	95.1±7.2	96.0±1.4
10	53.0±9.4	60.8±9.2	57.5±8.3	67.6±11.0	59.8±8.1	65.3±8.9	69.2±15.6	74.6±13.7	70.7±15.5	77.7±6.7	68.6±8.5	75.4±7.4	76.3±4.2	80.4±2.4
11	63.5±8.4	68.7±8.7	59.2±7.5	68.8±7.7	71.0±8.4	73.3±7.7	61.8±9.9	75.1±16.0	63.4±14.2	75.2±8.0	60.9±10.2	71.2±7.4	81.1±2.5	82.9±1.9
12	52.5±7.7	68.7±6.1	52.9±9.5	67.2±4.3	65.0±6.6	72.3±7.2	71.9±5.2	74.8±4.5	64.2±8.3	77.2±3.4	66.2±4.2	77.0±2.5	75.2±2.6	81.2±2.1
13	66.7±7.5	72.1±4.3	75.5±1.4	77.3±2.5	73.0±4.2	71.6±4.2	88.9±4.4	82.8±4.9	86.8±2.9	88.5±1.2	85.2±1.6	92.4±1.7	93.9±1.6	91.5±2.6
14	73.7±4.8	79.7±4.6	81.6±5.9	78.8±3.1	80.6±3.9	88.3±2.6	85.5±3.1	85.1±2.6	79.8±4.7	88.8±1.7	86.2±2.2	86.6±2.1	88.6±3.1	91.8±2.1
15	66.9±6.2	70.2±2.5	79.4±3.1	73.5±2.1	71.7±3.5	79.8±3.6	89.3±2.4	80.2±3.5	83.9±5.3	87.5±1.1	94.8±1.4	90.9±2.7	88.4±3.5	87.1±2.3
16	67.0±8.7	70.9±7.1	91.0±2.2	83.6±1.2	72.0±3.5	80.5±2.2	90.7±3.1	94.8±3.1	85.4±6.0	99.0±0.4	84.9±1.1	82.5±2.7	89.8±2.4	97.7±2.0
OA	62.4±5.5	69.1±4.8	66.6±4.0	71.3±3.5	69.8±5.8	75.5±4.3	70.5±5.9	76.7±5.4	67.7±6.3	78.9±2.4	69.4±3.1	75.4±2.9	**80.3±2.5**	**84.3±2.0**
AA	64.8±4.9	70.8±4.6	72.8±3.7	75.9±2.5	71.7±4.4	78.1±3.4	79.0±4.3	80.2±2.4	74.4±3.9	84.0±2.2	78.2±2.3	82.2±2.4	**84.7±2.3**	**87.8±1.6**
κ	61.8±4.7	69.0±5.1	64.7±4.3	69.1±3.6	68.6±5.1	75.6±3.9	70.5±6.2	75.3±5.5	62.6±7.1	75.6±2.6	68.7±3.2	74.4±3.3	**78.8±3.1**	**79.9±2.0**

TABLE III: Classification results (%) by using 5 and 10 labeled samples for each class on the UP dataset (the bold value is the best accuracy in each case).

Class	2D-CNN		3D-CNN		DRNN		DBMA		MSDN		DFSL		SSAD(Ours)	
	L=5	L=10	L=5	L=10	L=5	L=10	L=5	L=10	L=5	L=10	L=5	L=10	L=5	L=10
1	64.6±6.0	74.9±5.6	65.6±8.5	78.9±3.3	62.4±5.1	83.5±3.4	73.9±5.8	82.0±3.5	65.9±5.9	68.5±7.8	72.8±6.1	84.1±4.5	82.5±2.2	86.6±1.9
2	68.9±4.4	76.2±3.3	71.9±4.1	78.5±4.3	74.2±3.8	78.9±3.1	79.9±9.8	79.5±4.5	75.8±5.3	78.8±5.3	80.3±5.1	80.1±5.6	80.3±2.8	84.5±2.4
3	61.8±5.5	75.4±3.8	63.4±7.2	80.8±3.4	55.4±4.9	86.4±3.2	66.4±8.5	75.7±3.7	63.4±6.5	66.1±7.4	56.3±6.2	76.7±4.8	85.0±2.3	82.2±2.0
4	76.6±4.6	85.0±3.6	81.7±6.5	79.4±3.2	82.3±3.6	81.9±2.0	90.0±5.8	88.2±3.6	86.0±4.9	87.6±6.5	96.7±2.5	89.6±4.6	83.6±2.5	91.6±1.9
5	79.0±3.6	86.8±2.8	83.8±3.1	79.4±4.3	86.2±2.2	81.2±2.7	91.0±5.9	91.2±4.4	91.2±2.0	91.3±2.4	97.9±2.4	90.1±2.6	83.9±2.8	93.9±2.4
6	68.6±5.4	71.1±4.2	67.2±6.8	80.9±3.8	71.4±5.3	73.1±4.8	75.3±6.0	85.0±3.9	81.8±3.2	84.3±6.2	73.8±6.4	85.4±7.0	84.2±2.4	89.4±2.1
7	73.6±4.6	81.3±3.5	81.5±5.6	80.8±3.9	80.1±3.9	78.8±2.6	83.0±5.1	88.4±4.1	88.5±4.5	84.0±3.4	85.8±4.8	89.4±3.7	83.9±2.6	91.8±2.2
8	68.7±6.7	75.9±4.9	71.4±8.2	81.0±4.5	76.0±6.2	79.1±4.5	80.5±5.5	84.4±4.5	86.0±3.7	89.4±4.2	80.1±8.4	86.1±5.2	85.1±2.8	89.0±2.4
9	77.5±3.5	86.3±2.9	81.1±2.7	79.7±3.4	84.4±2.2	81.3±2.1	93.3±3.2	90.9±3.5	89.1±2.8	87.9±2.4	92.4±2.3	93.2±2.2	84.8±2.2	90.2±1.9
OA	69.0±4.9	76.7±4.5	71.5±4.2	79.4±3.7	72.6±4.1	79.7±4.1	79.3±4.1	82.3±4.1	77.1±3.8	79.5±3.1	79.3±3.3	83.3±2.9	**82.3±2.5**	**86.8±2.3**
AA	71.0±4.3	79.2±4.1	74.2±3.8	79.9±3.7	74.7±3.8	80.5±3.9	81.5±4.0	85.0±3.9	80.9±3.4	82.0±2.9	81.8±3.2	86.1±2.8	**83.7±2.6**	**88.8±2.1**
κ	68.4±5.1	74.8±4.7	67.9±4.8	76.2±4.2	71.9±4.3	77.3±3.8	75.1±4.6	79.3±4.4	73.9±4.2	76.0±3.2	75.4±3.7	80.0±3.3	**81.2±2.8**	**83.4±2.4**

TABLE IV: Classification results (%) by using 50 and 80 labeled samples for each class on the HS dataset (the bold value is the best accuracy in each case).

Class	2D-CNN		3D-CNN		DRNN		DBMA		MSDN		DFSL		SSAD(Ours)	
	L=50	L=80	L=50	L=80	L=50	L=80	L=50	L=80	L=50	L=80	L=50	L=80	L=50	L=80
1	77.1±2.7	80.6±4.6	84.6±3.5	86.8±1.6	83.7±2.5	84.2±4.2	84.7±1.6	90.3±2.8	84.0±1.5	85.5±3.8	87.6±3.0	90.3±1.7	89.0±2.1	86.3±1.4
2	77.8±4.1	79.7±5.0	81.4±4.1	87.4±1.1	78.0±4.6	81.8±3.0	85.6±1.0	87.9±2.6	84.9±1.0	82.4±2.4	86.5±1.9	84.5±1.5	87.8±1.9	89.0±1.9
3	81.5±1.8	83.8±1.5	85.5±1.8	88.8±0.7	82.9±2.0	91.0±2.5	86.6±0.5	95.9±1.1	85.9±1.7	91.3±0.4	91.6±0.4	99.9±0.1	96.9±1.3	92.6±0.7
4	75.3±4.1	72.0±3.3	80.6±4.8	85.4±1.5	74.2±4.2	79.7±4.7	84.5±1.3	85.0±3.2	84.0±1.4	80.9±3.3	84.9±2.5	80.1±1.6	87.6±2.0	90.0±1.9
5	77.7±3.8	84.9±2.9	80.0±2.8	87.2±1.1	80.4±4.0	86.4±2.9	84.7±1.0	88.8±2.0	84.1±1.1	84.0±2.8	86.8±2.0	87.3±1.5	85.9±1.9	91.9±0.7
6	80.9±3.9	84.9±3.5	86.4±3.4	87.4±1.3	82.3±1.7	88.1±2.3	85.7±1.4	93.4±2.3	85.0±1.7	87.4±2.4	89.1±1.9	95.8±1.3	88.4±1.6	90.8±1.2
7	70.0±3.8	71.4±5.1	73.6±4.6	77.3±3.1	77.1±5.3	77.6±5.9	74.5±2.8	77.7±4.0	74.2±3.0	80.1±1.4	79.2±2.4	76.6±2.4	85.3±3.1	87.0±2.0
8	67.0±4.3	76.6±5.6	67.9±4.7	77.5±2.4	66.9±6.5	76.1±4.4	73.5±3.2	76.6±3.8	72.5±3.9	69.0±4.2	73.8±3.5	74.0±2.2	72.2±2.9	84.1±2.0
9	66.2±7.3	80.0±4.2	68.4±4.7	72.8±4.6	72.9±3.3	84.9±5.4	68.4±5.1	72.2±5.7	68.5±5.2	74.3±3.1	72.8±4.8	72.8±4.1	83.7±5.3	84.4±2.6
10	71.5±4.3	76.5±3.7	77.1±3.8	80.8±3.5	81.5±4.2	84.2±3.3	78.2±3.0	85.5±3.9	77.7±3.1	81.5±1.4	82.0±2.7	88.3±2.4	90.2±3.1	89.8±1.8
11	68.5±5.4	71.3±2.9	71.9±4.0	75.4±3.4	73.7±4.0	80.9±4.9	71.5±3.9	79.5±4.4	71.8±4.1	76.7±2.0	75.4±3.3	85.0±2.9	89.0±3.8	87.3±2.3
12	66.5±6.1	68.2±3.7	67.9±4.6	74.0±4.4	70.3±4.2	77.5±4.4	69.9±4.0	74.1±4.6	69.8±4.4	76.3±3.9	74.7±4.3	74.7±3.4	81.5±4.4	89.0±1.6
13	57.8±6.2	88.0±4.1	64.2±5.0	65.9±4.2	81.9±5.9	88.0±4.1	57.7±4.9	68.0±5.5	58.3±5.2	73.6±5.3	66.2±5.6	77.7±3.6	88.4±4.6	85.0±2.8
14	79.9±3.7	85.9±3.3	83.7±2.2	88.6±0.7	83.7±2.8	90.3±2.5	86.6±0.7	95.0±1.5	85.9±1.2	89.0±2.4	90.4±1.6	98.1±1.3	87.1±1.7	92.3±0.7
15	79.1±4.8	84.1±3.4	81.0±2.2	88.8±1.1	81.6±3.5	89.3±3.4	86.5±1.0	92.7±1.8	85.7±1.0	87.9±3.3	89.9±2.3	95.1±1.5	85.1±1.9	92.2±0.8
OA	72.5±4.7	77.5±3.4	76.2±3.7	81.1±3.5	76.9±3.8	82.6±3.5	78.2±2.5	83.0±3.1	77.7±2.6	80.2±3.3	81.3±2.4	83.3±2.9	**85.9±2.2**	**88.3±1.7**
AA	73.1±3.8	79.2±3.2	76.9±2.1	81.6±2.3	78.1±3.5	84.0±2.9	78.6±2.4	84.2±2.5	78.2±2.6	81.3±2.8	82.1±1.7	85.3±3.8	**86.5±1.8**	**88.8±1.6**
κ	71.2±4.1	77.4±4.0	75.0±3.9	80.3±3.9	75.9±3.7	81.4±3.4	77.3±3.0	81.8±3.3	77.0±2.8	79.2±3.3	80.3±2.8	81.6±2.7	**84.9±2.5**	**87.9±1.8**

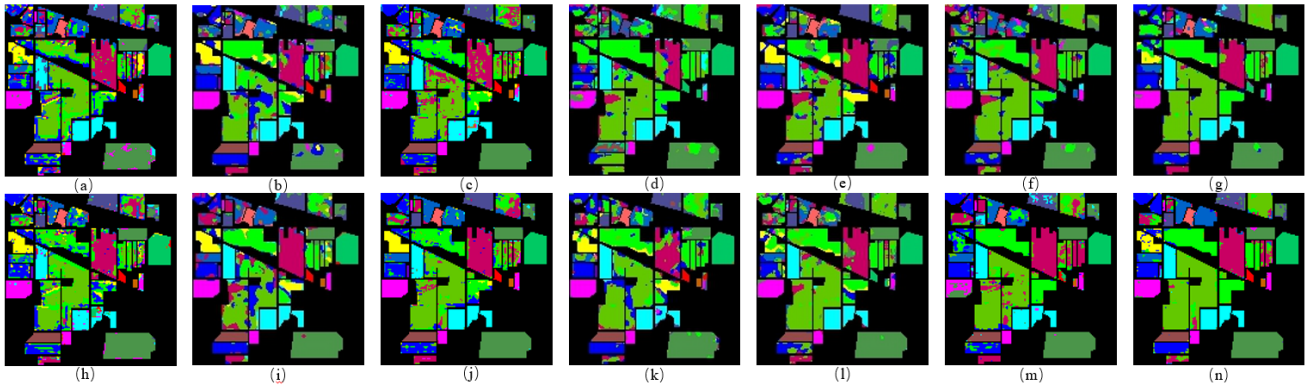

Fig. 9: The classification map of the IP dataset. (a) 2D-CNN (L=5). (b) 3D-CNN (L=5). (c) DRNN (L=5). (d) DBMA (L=5). (e) MSDN (L=5). (f) DFSL (L=5). (g) The proposed method (L=5). (h) 2D-CNN (L=10). (i) 3D-CNN (L=10). (j) DRNN (L=10). (k) DBMA (L=10). (l) MSDN (L=10). (m) DFSL (L=10). (n) The proposed method (L=10).

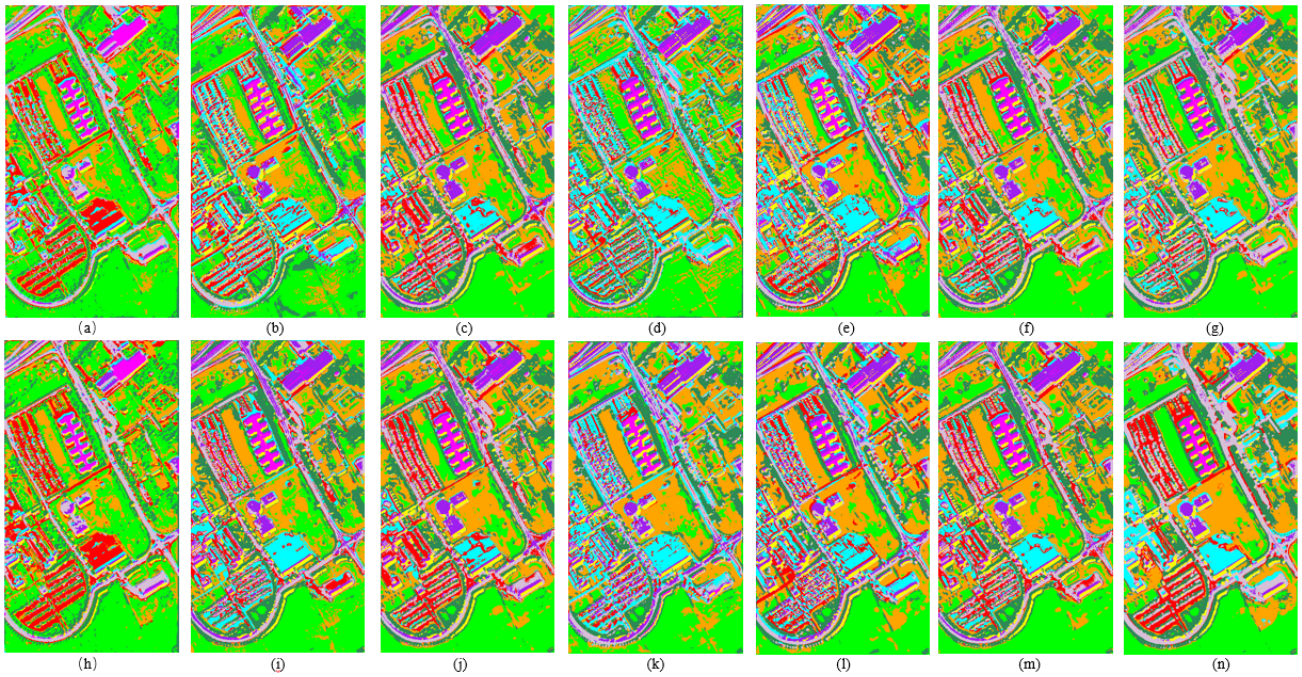

Fig. 10: The classification map of the UP dataset. (a) 2D-CNN (L=5). (b) 3D-CNN (L=5). (c) DRNN (L=5). (d) DBMA (L=5). (e) MSDN (L=5). (f) DFSL (L=5). (g) The proposed method (L=5). (h) 2D-CNN (L=10). (i) 3D-CNN (L=10). (j) DRNN (L=10). (k) DBMA (L=10). (l) MSDN (L=10). (m) DFSL (L=10). (n) The proposed method (L=10).

TABLE V: The effectiveness of SSL with 3D transformation strategy.

Methods	$L = 5$	$L = 10$	$L = 15$
Baseline	53.0 ± 3.1	59.1 ± 2.7	66.5 ± 2.4
$\tilde{\alpha}_1$ w/ SSL-3DT	57.7 ± 2.4	61.1 ± 2.9	67.4 ± 2.5
$\tilde{\alpha}_2$ w/ SSL-3DT	58.5 ± 3.9	64.6 ± 1.8	68.1 ± 1.6
$\tilde{\alpha}_3$ w/ SSL-3DT	60.7 ± 2.9	66.3 ± 2.8	70.1 ± 1.9
α_o w/ SSL-3DT	**69.3** ± 3.0	**74.7** ± 1.8	**78.1** ± 1.6

TABLE VI: The effectiveness of SSL with adaptive knowledge distillation strategy.

Methods	$L = 5$	$L = 10$	$L = 15$
Baseline	53.0 ± 3.1	59.1 ± 2.7	66.5 ± 2.4
α_1 w/ SSL-AKD	59.3 ± 3.7	60.3 ± 2.8	63.7 ± 2.3
α_2 w/ SSL-AKD	71.1 ± 3.2	77.2 ± 3.3	78.0 ± 2.5
α_3 w/ SSL-AKD	72.1 ± 3.5	80.3 ± 2.7	82.2 ± 2.7
α_o w/ SSL-AKD	**73.2** ± 3.9	**83.5** ± 2.1	**84.6** ± 2.3

2) Effectiveness of SSL with Adaptive Knowledge Distillation: In this ablation study, we train the PCN with ataptive knowledge distillation but without 3D transformation. The classification results are shown in Table VI, where 'w/ SSL-AKD' means the PCN trained under the strategy of SSL with adaptive knowledge distillation (AKD). The 'Baseline'

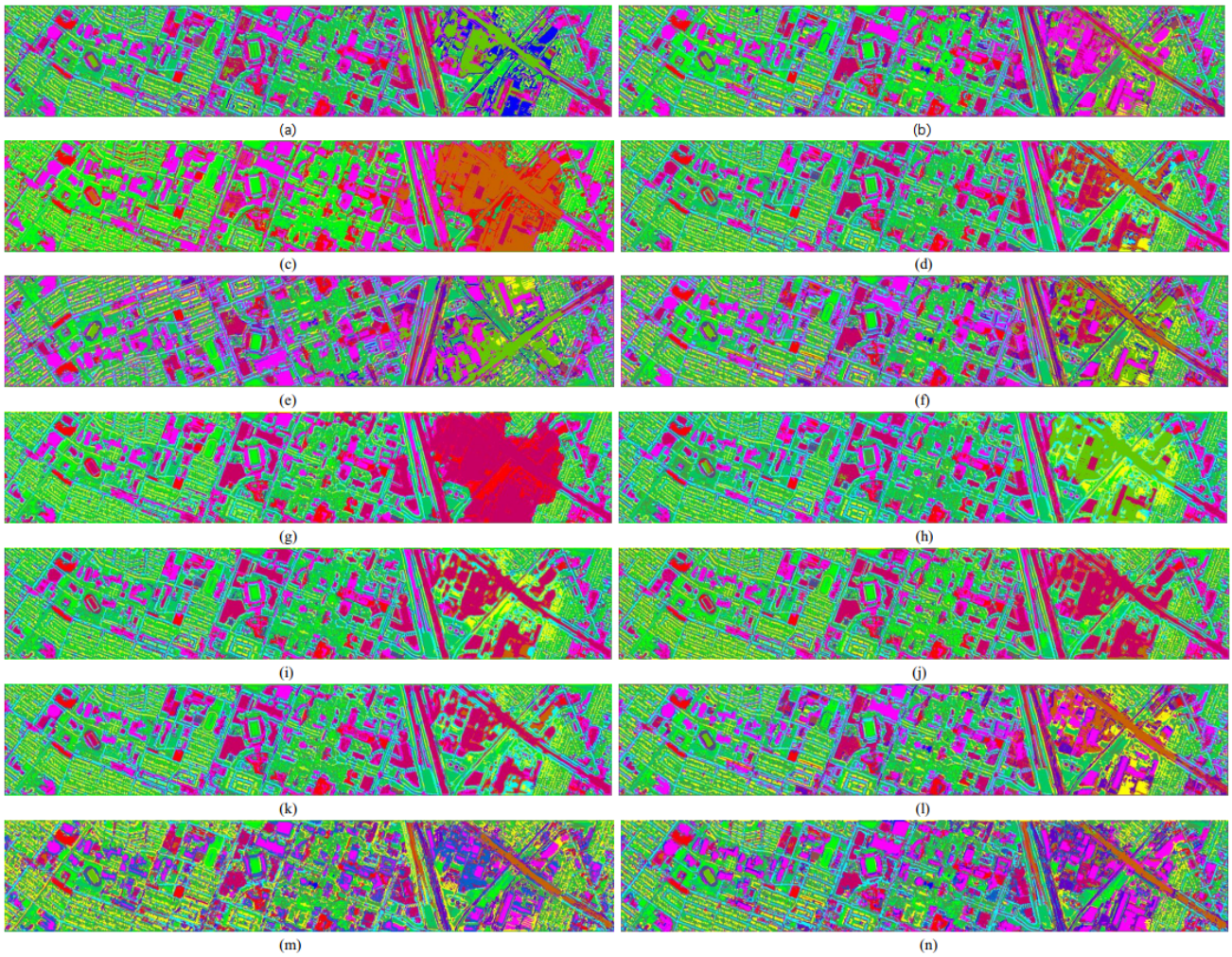

Fig. 11: The classification map of the HS dataset. (a) 2D-CNN (L=50). (b) 2D-CNN (L=80). (c) 3D-CNN (L=50). (d) 3D-CNN (L=80). (e) DRNN (L=50). (f) DRNN (L=80). (g) DBMA (L=50). (h) DBMA (L=80). (i) MSDN (L=50). (j) MSDN (L=80). (k) DFSL (L=50). (l) DFSL (L=80). (m) The proposed method (L=50). (n) The proposed method (L=80).

TABLE VII: The effectiveness of the proposed method.

Methods	$L = 5$	$L = 10$	$L = 15$
Baseline	53.0 ± 3.1	59.1 ± 2.7	66.5 ± 2.4
$\tilde{\alpha}_1$ w/ 3DT&AKD	67.4 ± 3.4	73.3 ± 2.6	73.4 ± 2.2
$\tilde{\alpha}_2$ w/ 3DT&AKD	74.5 ± 2.5	82.7 ± 2.7	81.7 ± 2.2
$\tilde{\alpha}_3$ w/ 3DT&AKD	76.5 ± 3.0	83.5 ± 1.9	83.9 ± 2.3
α_o w/ 3DT&AKD	**80.3** ± 2.5	**84.3** ± 2.0	**86.4** ± 1.5

TABLE VIII: The comparisons of different components in the proposed method.

Methods	$L = 5$	$L = 10$	$L = 15$
Baseline	53.0 ± 3.1	59.1 ± 2.7	66.5 ± 2.4
α_o w/ SSL-3DT	69.3 ± 3.0	74.7 ± 1.8	78.1 ± 1.6
α_o w/ SSL-AKD	73.2 ± 3.9	83.5 ± 2.1	84.6 ± 2.3
α_o w/ 3DT&AKD	**80.3** ± 2.5	**84.3** ± 2.0	**86.4** ± 1.5

method is a pure 3 layer PCN without the ataptive knowledge distillation strategy. Similar to the performances in the Table V, the results generated by different PC layers are higher than those generated by the 'Baseline' model. The fused results show better performance than the results generated by the 'Baseline' method and each single PC layer. These improvements indicate the usefulness of ataptive knowledge distillation.

3) Effectiveness of SSL with both 3D transformation and adaptive knowledge distillation: In this ablation study, the PCN was trained under adaptive knowledge distillation with the 3D transformation strategy. The results of different PC layers ($\{\tilde{\alpha}_k\}_{k=1}^{3}$) and the fused result ($s_o$) over all PC layers are generated by Eq. (4) and Eq. (5), respectively. The results are shown in Table VII, where 'w/ 3DT&AKD' means the PCN is trained under adaptive knowledge distillation with 3D transformation strategy. Compared with the results generated by 'Baseline' method, the improvements of our final fusion results for $L = 5$, 10, and 15 are 27.3%, 25.2%, and 19.9%, respectively, showing the effectiveness of the proposed method.

TABLE IX: The testing time of the proposed method and comparison methods.

Methods	3D-CNN	DBMA	MSDN	DFSL	Ours
Time(s)	13.67	13.56	12.46	11.13	10.82

To clearly show the effectiveness of different strategies in our proposed method, the final results of the PCN trained with different strategies are listed in Table VIII. The results show that the combination of these strategies helps the PCN to achieve the best results over all settings. This ablation study illustrates the complementarity between adaptive soft labels and 3D transformation strategy.

C. Time Consumption

The time consumption of the proposed method and the comparison methods was tested on the IP dataset. The detailed results are shown in Table IX. The computer environment for time testing in this paper is shown as follows: the processor is "Intel(R) Xeon(R) Gold 5118"; the graphics card is "NVIDIA GeForce RTX 2080 Ti" with "CUDA version 10.0.130"; the programming language and the deep learning platform are Python (version 3.6.2) and PyTorch (version 1.2.0), respectively. Table IX shows that the time consumption for testing the whole IP dataset of the proposed method is similar with those of other methods under the same computer environment.

V. Conclusion

This study proposed a self-supervised learning method with adaptive distillation, including the strategies of adaptive knowledge distillation, data transformation in spectral domain and spatial domain, for HSI classification with a small number of labeled samples. The experiments were conducted for verification, and some main conclusions were reached as follows.

(1) The proposed method outperforms existing popular HSI classification methods in presence of small annotated training samples. The comparative analysis of results suggest that the proposed method achieves state-of-the-art result for the few-shot hyperspectral image classification.

(2) The ablation study demonstrates that both modules (SSL with adaptive knowledge distillation and SSL with 3D transformation) are effective for improving the accuracy.

(3) The fusion strategies improve the performance of pixel-level classification under the limited labeled training samples.

The SSL method proposed in this paper boosts the existing SSL methods by two modules: SSL with 3D transformation and SSL with adaptive knowledge distillation. Next, we will explore the SSL method based on nonlinear transformation and generative adversarial network.

Acknowledgment

The authors gratefully acknowledge the National Center for Airborne Laser Mapping (NCALM), the University of Houston and the IEEE Image Analysis and Data Fusion Technical Committee for providing the Houston dataset, Pavia University and Grupo de Inteligencia Computacional (GIC) for providing the Indian Pines dataset and the University of Pavia dataset.

References

[1] A. Plaza, J. A. Benediktsson, J. W. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri, M. Marconcini, J. C. Tilton, and G. Trianni, "Recent advances in techniques for hyperspectral image processing," *Remote Sensing of Environment*, vol. 113, pp. S110 – S122, 2009, imaging Spectroscopy Special Issue. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0034425709000807

[2] Z. Zhong, B. Fan, K. Ding, H. Li, S. Xiang, and C. Pan, "Efficient multiple feature fusion with hashing for hyperspectral imagery classification: A comparative study," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4461–4478, 2016.

[3] J. A. Benediktsson and P. Ghamisi, *Spectral-spatial classification of hyperspectral remote sensing images*. Artech House, 2015.

[4] P. Ghamisi, J. Plaza, Y. Chen, J. Li, and A. J. Plaza, "Advanced spectral classifiers for hyperspectral images: A review," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 1, pp. 8–32, 2017.

[5] C. Hecker, M. Van, der Meijde, H. Van, der Werff, and F. D. Van, der Meer, "Assessing the influence of reference spectra on synthetic sam classification results," *IEEE Transactions on Geoscience & Remote Sensing*, vol. 46, no. 12, pp. 4162–4172, 2008.

[6] F. A. Kruse, A. B. Lefkoff, J. W. Boardman, K. B. Heidebrecht, A. T. Shapiro, P. J. Barloon, and A. F. H. Goetz, "The spectral image processing system (sips)—interactive visualization and analysis of imaging spectrometer data," *Remote Sensing of Environment*, vol. 44, no. 2-3, pp. 145–163, 1993.

[7] C. I. Chang, "An information-theoretic approach to spectral variability, similarity, and discrimination for hyperspectral image analysis," *Information Theory IEEE Transactions on*, vol. 46, no. 5, pp. 1927–1932, 2000.

[8] Q. Du and H. Ren, "Real-time constrained linear discriminant analysis to target detection and classification in hyperspectral imagery," *Pattern Recognition*, vol. 36, no. 1, pp. 1–12, 2003.

[9] S. Prasad and L. M. Bruce, "Limitations of principal components analysis for hyperspectral target recognition," *IEEE Geoscience & Remote Sensing Letters*, vol. 5, no. 4, pp. 625–629, 2008.

[10] A. Villa, J. A. Benediktsson, J. Chanussot, and C. Jutten, "Hyperspectral image classification with independent component discriminant analysis," *IEEE Transactions on Geoscience & Remote Sensing*, vol. 49, no. 12, pp. 4865–4876, 2012.

[11] J. Harris, D. Rogge, R. Hitchcock, O. Ijewliw, and D. Wright, "Mapping lithology in canada's arctic: application of hyperspectral data using the minimum noise fraction transformation and matched filtering," *Canadian Journal of Earth Sciences*, vol. 42, no. 12, pp. 2173–2193, 2005.

[12] C. M. Bachmann, T. L. Ainsworth, and R. A. Fusina, "Exploiting manifold geometry in hyperspectral imagery," *IEEE Transactions on Geoscience & Remote Sensing*, vol. 43, no. 3, pp. 441–454, 2005.

[13] C. M. Bachmann, T. L. Ainsworth, and R. A. Fusina, "Improved manifold coordinate representations of large-scale hyperspectral scenes," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 10, pp. 2786–2803, 2006.

[14] B. Raducanu and F. Dornaika, "A supervised non-linear dimensionality reduction approach for manifold learning," *Pattern Recognition*, vol. 45, no. 6, pp. 2432–2444, 2012.

[15] W. Li, S. Prasad, J. E. Fowler, and L. M. Bruce, "Locality-preserving discriminant analysis in kernel-induced feature spaces for hyperspectral image classification," *IEEE Geoscience & Remote Sensing Letters*, vol. 8, no. 5, pp. 894–898, 2011.

[16] Y. W. Chen and X. H. Han, *Classification of High-Resolution Satellite Images Using Supervised Locality Preserving Projections*. Springer Berlin Heidelberg, 2008.

[17] Qiao, Lishan, Chen, Songcan, Tan, and Xiaoyang, "Sparsity preserving projections with applications to face recognition," *Pattern Recognition*, vol. 43, no. 1, pp. 331–341, 2010.

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

[19] B. Rasti, D. Hong, R. Hang, P. Ghamisi, X. Kang, J. Chanussot, and J. A. Benediktsson, "Feature extraction for hyperspectral imagery: The evolution from shallow to deep (overview and toolbox)," *IEEE Geoscience and Remote Sensing Magazine*, 2020.

[20] A. Plaza, P. Martinez, J. Plaza, and R. Perez, "Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations," *IEEE Transactions on Geoscience & Remote Sensing*, vol. 43, no. 3, pp. 466–479, 2005.

[21] P. R. Marpu, M. Pedergnana, M. D. Mura, J. A. Benediktsson, and L. Bruzzone, "Automatic generation of standard deviation attribute profiles for spectral-spatial classification of remote sensing data," *IEEE Geoscience & Remote Sensing Letters*, vol. 10, no. 2, pp. 293–297, 2013.

[22] W. Liao, R. Bellens, A. Pizurica, W. Philips, and Y. Pi, "Classification of hyperspectral data over urban areas using directional morphological profiles and semi-supervised feature extraction," *IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing*, vol. 5, no. 4, pp. 1177–1190, 2012.

[23] Y. Bengio, "Learning deep architectures for ai," *Foundations & Trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[24] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. Red Hook, NY, USA: Curran Associates Inc., 2012, p. 1097–1105.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[27] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[28] G. S. Xie, Z. Zhang, L. Liu, F. Zhu, X. Y. Zhang, L. Shao, and X. Li, "Srsc: Selective, robust, and supervised constrained feature representation for image classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 10, pp. 4290–4302, 2020.

[29] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, 2016.

[30] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.

[31] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[32] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[33] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2017.

[34] Z. Tian, H. Zhao, M. Shu, Z. Yang, R. Li, and J. Jia, "Prior guided feature enrichment network for few-shot segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[35] G. Xie, X. Zhang, S. Yan, and C. Liu, "Hybrid cnn and dictionary-based models for scene recognition and domain adaptation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 6, pp. 1263–1274, 2017.

[36] B. Wang, Z. Ou, and Z. Tan, "Learning trans-dimensional random fields with applications to language modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 876–890, 2018.

[37] B. Zhang, D. Xiong, and J. Su, "Neural machine translation with deep attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 1, pp. 154–163, 2020.

[38] D. Keysers, T. Deselaers, H. A. Rowley, L. Wang, and V. Carbune, "Multi-language online handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1180–1194, 2017.

[39] P. Ghamisi, E. Maggiori, S. Li, R. Souza, Y. Tarablaka, G. Moser, A. D. Giorgi, L. Fang, Y. Chen, M. Chi, S. B. Serpico, and J. A. Benediktsson, "New frontiers in spectral-spatial hyperspectral image classification: The latest advances based on mathematical morphology, markov random fields, segmentation, sparse representation, and deep learning," *IEEE Geoscience and Remote Sensing Magazine*, vol. 6, no. 3, pp. 10–43, 2018.

[40] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2094–2107, 2014.

[41] Y. S. Chen, X. Zhao, and X. P. Jia, "Spectral-spatial classification of hyperspectral data based on deep belief network," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 2381–2392, 2015.

[42] L. Fang, S. Li, X. Kang, and J. A. Benediktsson, "Spectral–spatial classification of hyperspectral images with a superpixel-based discriminative sparse model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 8, pp. 4186–4201, 2015.

[43] L. Fang, N. He, S. Li, A. J. Plaza, and J. Plaza, "A new spatial–spectral feature extraction method for hyperspectral images using local covariance matrix representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 6, pp. 3534–3546, 2018.

[44] C. Zhang, J. Yue, and Q. Qin, "Deep quadruplet network for hyperspectral image classification with a small number of samples," *Remote Sensing*, vol. 12, no. 4, p. 647, 2020.

[45] B. Liu, X. Yu, A. Yu, P. Zhang, G. Wan, and R. Wang, "Deep few-shot learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 4, pp. 2290–2304, 2019.

[46] C. Zhang, J. Yue, and Q. Qin, "Global prototypical network for few-shot hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pp. 1–1, 2020.

[47] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[48] X. Liu, F. Zhang, Z. Hou, Z. Wang, L. Mian, J. Zhang, and J. Tang, "Self-supervised learning: Generative or contrastive," *arXiv preprint arXiv:2006.08218*, vol. 1, no. 2, 2020.

[49] I. Misra and L. v. d. Maaten, "Self-supervised learning of pretext-invariant representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6707–6717.

[50] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1422–1430.

[51] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2536–2544.

[52] L. Dinh, D. Krueger, and Y. Bengio, "Nice: Non-linear independent components estimation," *arXiv preprint arXiv:1410.8516*, 2014.

[53] G. Larsson, M. Maire, and G. Shakhnarovich, "Learning representations for automatic colorization," in *European conference on computer vision*. Springer, 2016, pp. 577–593.

[54] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[55] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *ICLR 2018*, 2018.

[56] L. Jing and Y. Tian, "Self-supervised spatiotemporal feature learning by video geometric transformations," *CoRR*, vol. abs/1811.11387, 2018. [Online]. Available: http://arxiv.org/abs/1811.11387

[57] Y. Wang, J. Mei, L. Zhang, B. Zhang, A. Li, Y. Zheng, and P. Zhu, "Self-supervised low-rank representation (sslrr) for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 10, pp. 5658–5672, 2018.

[58] Y. Wang, J. Mei, L. Zhang, B. Zhang, P. Zhu, Y. Li, and X. Li, "Self-supervised feature learning with crf embedding for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 5, pp. 2628–2642, 2019.

[59] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML'10. Madison, WI, USA: Omnipress, 2010, p. 807–814.

[60] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[61] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3639–3655, 2017.
[62] K. Gao, B. Liu, X. Yu, J. Qin, P. Zhang, and X. Tan, "Deep relation network for hyperspectral image few-shot classification," *Remote Sensing*, vol. 12, no. 6, p. 923, Mar 2020. [Online]. Available: http://dx.doi.org/10.3390/rs12060923
[63] K. Gao, W. Guo, X. Yu, B. Liu, A. Yu, and X. Wei, "Deep induction network for small samples classification of hyperspectral images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 3462–3477, 2020.
[64] A. B. Hamida, A. Benoit, P. Lambert, and C. B. Amar, "3-d deep learning approach for remote sensing image classification," *IEEE Transactions on geoscience and remote sensing*, vol. 56, no. 8, pp. 4420–4434, 2018.
[65] W. Ma, Q. Yang, Y. Wu, W. Zhao, and X. Zhang, "Double-branch multi-attention mechanism network for hyperspectral image classification," *Remote Sensing*, vol. 11, no. 11, p. 1307, 2019.
[66] B. Fang, Y. Li, H. Zhang, and J. C.-W. Chan, "Hyperspectral images classification based on dense convolutional networks with spectral-wise attention mechanism," *Remote Sensing*, vol. 11, no. 2, p. 159, 2019.

Jun Yue received the B.Eng. degree in geodesy from Wuhan University, Wuhan, China, in 2013 and the Ph.D. degree in GIS from Peking University, Beijing, China, in 2018.

He is currently an Assistant Professor with the Department of Geomatics Engineering, Changsha University of Science & Technology. His research interests include satellite image understanding, pattern recognition, and few-shot learning. Dr. Yue serves as a reviewer for IEEE Transactions on Geoscience and Remote Sensing, ISPRS Journal of Photogrammetry and Remote Sensing, IEEE Geoscience and Remote Sensing Letters, IEEE Transactions on Biomedical Engineering, IEEE Journal of Biomedical and health Informatics, Information Fusion, Information Sciences, International Journal of Remote Sensing, Remote Sensing Letters, etc.

Leyuan Fang (S'10-M'14-SM'17) received the Ph.D. degree from the College of Electrical and Information Engineering, Hunan University, Changsha, China, in 2015.

From September 2011 to September 2012, he was a visiting Ph.D. student with the Department of Ophthalmology, Duke University, Durham, NC, USA, supported by the China Scholarship Council. From August 2016 to September 2017, he was a Postdoc Researcher with the Department of Biomedical Engineering, Duke University, Durham, NC, USA. He is currently a Professor with the College of Electrical and Information Engineering, Hunan University. His research interests include sparse representation and multi-resolution analysis in remote sensing and medical image processing. He is the associate editors of IEEE Transactions on Image Processing, IEEE Transactions on Geoscience and Remote Sensing, IEEE Transactions on Neural Networks and Learning Systems, and Neurocomputing. He was a recipient of one 2nd-Grade National Award at the Nature and Science Progress of China in 2019.

Hossein Rahmani received the B.Sc. degree in computer software engineering from the Isfahan University of Technology, Isfahan, Iran, in 2004, the M.Sc. degree in software engineering from Shahid Beheshti University, Tehran, Iran, in 2010, and the Ph.D. degree from The University of Western Australia, in 2016. He has published several papers in top conferences and journals such as CVPR, ICCV, ECCV, and the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.

He is currently an Associate Professor (Lecturer) with the School of Computing and Communications, Lancaster University. Before that, he was a Research Fellow at the School of Computer Science and Software Engineering, The University of Western Australia. His research interests include computer vision, action recognition, 3D shape analysis, and machine learning.

Pedram Ghamisi (S'12-M'15-SM'18) received the B.Sc. degree in civil (survey) engineering from the Tehran South Campus of Azad University, Tehran, Iran, the M.Sc. degree (Hons.) in remote sensing from the K. N. Toosi University of Technology, Tehran, in 2012, and the Ph.D. degree in electrical and computer engineering from the University of Iceland, Reykjavik, Iceland, in 2015. In 2013/2014, he spent seven months at the School of Geography, Planning and Environmental Management, The University of Queensland, Brisbane, QLD, Australia. After Ph.D., he subsequently worked as a Post-Doctoral Research Fellow at the University of Iceland. In 2015, he won the prestigious "Alexander von Humboldt Fellowship" and started his work as a Post-Doctoral Research Fellow at the Technical University of Munich (TUM), Munich, Germany, and Heidelberg University, Heidelberg, Germany, since October 2015. He was also a Research Scientist at German Aerospace Center (DLR), Remote Sensing Technology Institute (IMF), Oberpfaffenhofen, Germany, from October 2015 to May 2018. In 2018, he won the prestigious "High Potential Program" and started his work as Head of Machine Learning Group, Helmholtz-Zentrum Dresden-Rossendorf (HZDR), Freiberg, Germany. His research interests involve interdisciplinary research on remote sensing and machine (deep) learning, image and signal processing, and multisensory data fusion.

Dr. Ghamisi serves as an Associate Editor for the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS (GRSL) and Remote Sensing.