Self-Supervised Deep Subspace Clustering for Hyperspectral Images With Adaptive Self-Expressive Coefficient Matrix Initialization

Kun Li, Yao Qin, Qiang Ling, Yingqian Wang, Zaiping Lin, and Wei An

***Abstract*—Deep subspace clustering network has shown its effectiveness in hyperspectral image (HSI) clustering. However, there are two major challenges that need to be addressed: 1) lack of effective supervision for feature learning; and 2) negative effect caused by the high redundancy of the global dictionary atoms. In this article, we propose an end-to-end trainable network for HSI clustering. Specifically, to ensure the extracted features are well-suited to subsequent subspace clustering, the cluster assignments with high confidence are employed as pseudo-labels to supervise the feature learning process. Then, an adaptive self-expressive coefficient matrix initialization strategy is designed to reduce the dictionary redundancy, where the spectral similarity between each target sample and its neighbors is modeled via the *k*-nearest neighbor graph to guide the initialization. Experimental results on three public HSI datasets demonstrate the effectiveness of the proposed method. In particular, our method outperforms several state-of-the-art HSI clustering methods, and achieves overall accuracy of 100% on both SalinasA and Pavia University datasets.**

***Index Terms*—Deep subspace clustering (DSC), hyperspectral image (HSI), self-expressive, self-supervised, subspace clustering (SC).**

I. Introduction

SINCE hyperspectral images (HSIs) contain rich spatial and spectral information, they have been widely applied to different remote sensing applications, such as food safety [1], environmental monitoring [2], geological exploration [3], land-cover classification [4], [5], and hyperspectral unmixing [6]. Among these applications, HSI classification is a fundamental technique which aims to assign each pixel with a certain label [7]. Although supervised HSI classification methods have achieved state-of-the-art performance with the development of deep learning techniques [8], a large amount of labeled samples required in supervised HSI classification hinders its application. In contrast, HSI clustering has drawn much attention in recent years since it can automatically assign similar samples to a group in an unsupervised manner. However, clustering is still a challenging task due to the high dimensionality and complex spectral-spatial structures of HSI data [9].

Recently, subspace clustering (SC) [10] has been successfully applied to HSI clustering due to its capability to handle high-dimensional data and its effectiveness of capturing complex structures of HSI data [11]–[19]. These methods can be grouped into two categories, i.e., SC in original space and SC in feature space. The former ones construct affinity matrix from raw samples [11]–[16], whereas the latter ones construct affinity matrix from the features of samples [17]–[19]. Due to the inherent nonlinear structures of HSIs, SC in deep feature space can well capture the nonlinear characteristics of sample distribution [18], [19]. However, there exist two main problems that need to be tackled for these deep subspace clustering (DSC) methods. First, since affinity matrix learning and spectral clustering are performed independently in these methods, their feature learning lacks effective supervision. Therefore, the deep features extracted by the encoder cannot always suit for the subsequent SC [20]. Second, since these methods employ the global self-expressive dictionary to represent the features of samples, the high dictionary redundancy hinders the further improvement of the clustering performance [14].

To address the aforementioned issues, we propose a Self-supervised Deep Subspace Clustering method with Adaptive self-expressive coefficient matrix Initialization (SDSC-AI) for HSI clustering. Specifically, to learn discriminative features for the SC, we propose an end-to-end trainable network to combine affinity matrix learning and spectral clustering. In our network, fully connected layers are introduced on top of the encoder to serve as a classifier, which use the cluster assignments produced by spectral clustering as pseudo-labels to supervise the feature learning process. In this way, affinity matrix learning and spectral clustering are alternately performed and the whole model is trained in an end-to-end manner. Moreover, to obtain highly confident pseudo-labels, the samples closer to their cluster centers in spectral clustering are selected to train the encoder, and their cluster assignments are considered as highly confident.

To reduce the high redundancy of the global dictionary atoms, the correlated atoms need to be selected to express the target features of samples, while the uncorrelated atoms should be suppressed. However, existing DSC-based methods [18], [19] initialize the elements of the self-expressive coefficient matrix with the same non-zero values, which tends to induce all atoms to express the target features of samples. Therefore, the initialization

Manuscript received December 8, 2020; revised January 17, 2021; accepted February 26, 2021. Date of publication March 5, 2021; date of current version March 25, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 62002372 and Grant 61921001. *(Corresponding author: Yao Qin.)*

The authors are with the College of Electronic Science and Technology, National University of Defense Technology, Changsha 410073, China (e-mail: likun19@nudt.edu.cn; tsintuan@163.com; lq910131@163.com; wangyingqian16@nudt.edu.cn; linzaiping@sina.com; anwei@nudt.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2021.3063335

approach of self-expressive coefficient matrix in these methods cannot address the issue of high dictionary redundancy. Based on the fact that similar HSI samples are more likely lying in the same subspace [12], we construct k-nearest neighbor (KNN) graph to model the spectral similarity between each sample and its neighbors. The nonzero element in the binary adjacent matrix of KNN graph indicates that the corresponding two samples are similar. Therefore, these two samples and their corresponding features are likely lying in the same subspace. Moreover, since the weights of neural networks are generally initialized to small random values [21], [22], the nonzero elements in the binary adjacent matrix are updated by random values generated from a uniform distribution. Finally, the updated adjacent matrix is used to initialize the self-expressive coefficient matrix. In this way, the correlated atoms can be induced to express the target features, while the uncorrelated ones can be suppressed.

The main contributions of this article are summarized as follows.

1) We propose an end-to-end trainable network to combine the affinity matrix learning and spectral clustering. The cluster assignments with high confidence are used as pseudo-labels to supervise the feature learning process. To the best of our knowledge, this is the first attempt to introduce self-supervised learning for HSI clustering.
2) We proposed a spectral similarity based adaptive self-expressive coefficient matrix initialization strategy to reduce the high redundancy of global self-expressive dictionary atoms.
3) Experimental results on three benchmark HSI datasets demonstrate the superiority of our method as compared to several state-of-the-art clustering methods.

The rest of this article is organized as follows. Some related works are briefly reviewed in Section II. The proposed method is described in Section III. Section IV presents the experimental setup and results in detail. Section V concludes this article.

II. Related Works

In this section, we briefly review major works on HSI clustering, SC, and self-supervised learning.

A. HSI Clustering

HSI clustering methods have drawn much attention since they do not require any labeled samples during training phase. Generally, the existing HSI clustering algorithms can be divided into following categories [23]:

1) centroid-based methods;
2) density-based methods;
3) biological-based methods;
4) graph-based methods; and
5) deep learning-based methods.

Centroid-based methods such as k-means [24], fuzzy c-means (FCM) [25], and fuzzy c-means with spatial constraint (FCM_S) [26] iteratively update the cluster centers until the cluster centers remain unchanged. These methods are computationally efficient and easy to implement. However, they are sensitive to the initialization state [27]. Density-based methods such as clustering by fast search and find of density peaks [28] and its improved version [29] calculate the local density of each sample, and then select the samples both having high local density and large distance from samples with higher densities as cluster centers. The biological-based methods such as automatic fuzzy clustering method based on adaptive multiobjective differential evolution [30] employs the biological model to achieve HSI clustering, which transforms the clustering problem into a multiobject optimization problem. The graph-based methods such as spectral clustering [31], fast spectral clustering with anchor graph [16], and sparse subspace clustering (SSC) [11] construct graph to represent the similarity of each pair of samples, and then obtain the clustering results by applying spectral analysis to the similarity graph. The deep learning-based methods such as learning the deep embedding based on the set-to-set and sample-to-sample distances (LSSD) [23] embed the raw samples into low-dimensional feature space and group the deep representations to generate final clusters.

B. Subspace Clustering

Based on the fact that data in a high-dimensional space can be better represented as subspaces [32], SC-based methods obtain great research interests due to their capability to handle high-dimensional data and their effectiveness of capturing complex structures of HSI data [11]–[19]. These methods generally divide the task of clustering into two subproblems. The first one is to construct affinity matrix, and the second one is to apply spectral clustering on the affinity matrix. According to whether the affinity matrix is built in original space or not, these methods can be grouped into two categories, i.e., SC in original space and SC in feature space.

1) Subspace Clustering in Original Space: This type of methods [11]–[13] build affinity matrix from raw HSI samples based on the assumption that a sample in a union of subspaces can be expressed as a linear combination of other samples in the same subspace (i.e., self-expressiveness property of the data [11]). To construct informative affinity matrix, different regularization terms [10] are introduced to regularize the self-expressive coefficient matrix, e.g., the sparse affinity matrix induced by ℓ_1-norm [11], the low-rank affinity matrix induced by nuclear norm [33], [34]. In addition to these typical subspace learning methods, spectral-spatial sparse SC (S^4C) [12] and l_2-norm regularized SSC (l_2-SSC) [13] have been proposed for HSI clustering to better exploit both the spectral and spatial information.

2) Subspace Clustering in Feature Space: This type of methods [17]–[19], [35] maps the raw samples into feature space to better capture the nonlinear characteristics of sample distribution, and then construct the affinity matrix in the feature space. For instance, kernel SC [17] is proposed to implicitly map the HSI samples from original space to a kernelized space. However, this method empirically selects the optimal kernel and thus suffers from the degradation of different nonlinear kernels. Recently, DSC network [18] is proposed to nonlinearly map the raw samples to a latent space using deep convolutional autoencoders. Besides, DSC uses a novel self-expressive layer to

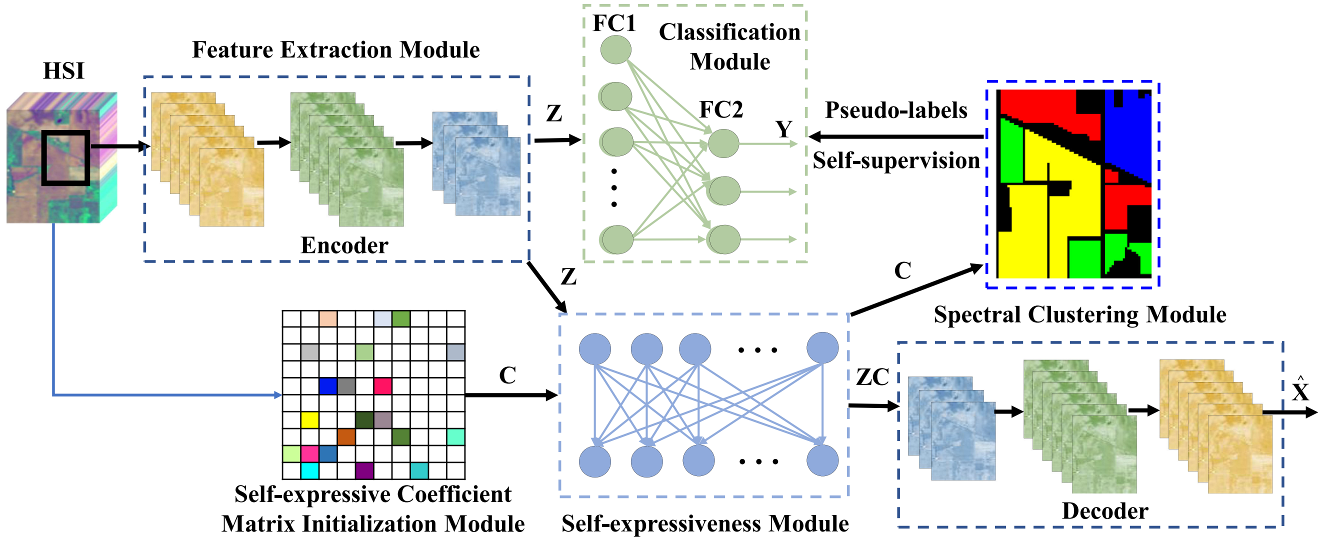

HSI
Feature Extraction Module
Encoder
Z
FC1
Classification Module
FC2
Y
Pseudo-labels
Self-supervision
Z
C
Spectral Clustering Module
C
Self-expressive Coefficient Matrix Initialization Module
Self-expressiveness Module
ZC
$\hat{X}$
Decoder

Fig. 1. Flowchart of our method. Our network consists of five modules: a) the feature extraction module is based on deep convolutional auto-encoders, where the encoder is used to extract features, and the decoder is used to reconstruct the raw input samples; b) the self-expressiveness module is used to learn the self-expressive coefficient matrix; c) the self-expressive coefficient matrix initialization module is used to provide a good initialized self-expressive coefficient matrix for the self-expressiveness module; d) self-supervised learning-based classification module classifies the features with the pseudo-labels generated from the spectral clustering module; and e) the spectral clustering module generates clustering results and provides pseudo-labels to supervise the feature learning (best viewed in color).

achieve self-expressiveness property of the features of samples. To preserve the cluster structure in data space, distribution-preserving subspace clustering [35] introduces a distribution consistency loss to guide the learning of distribution-preserving latent representation. Inspired by the success of the DSC method, Laplacian regularized deep subspace clustering [19] introduces Laplacian regularization to retain the manifold structure of HSI data.

C. Self-Supervised Learning

Self-supervised learning aims at learning general features without using any human-annotated labels [36]. To this end, self-supervised learning generally predefines a pretext task to learn feature representations of the unlabeled data using pseudo-labels that are automatically generated based on the attributes of unlabeled data. After pretext task training, the deep representations contain rich semantic information and then are transferred to downstream tasks by fine-tuning.

Since the pretext task plays a key role in self-supervised learning, several effective pretext tasks [36], [37] are designed to yield pseudo-labels from the unlabeled data to guide self-supervised learning. In [38], the prediction of the relative spatial position between the central image patch and its 8-neighbor image patches are used as a pretext task. In [39], pretext task is designed as the recovery of positions of spatially shuffled image patches. Prediction of geometrical image transformation such as rotations is also used as a pretext task [40]. Besides, mutual information (MI) maximization is a popular kind of pretext task in self-supervised learning [41], [42]. Recently, instance discrimination [37], [43], [44] is leveraged as a pretext task and achieves promising performance in downstream tasks.

Clustering is a natural pretext task since data are grouped according to their attributes and can be automatically assigned with clustering labels. DeepCluster [45] is a typical clustering-based self-supervised learning method whose training process includes two alternate steps, i.e., 1) train the encoder using cluster assignments as pseudo-labels, and 2) cluster the image features using k-means algorithm. In this article, we follow the clustering-based method to yield pseudo-labels.

III. Proposed Methodology

In this section, an SDSC-AI method is proposed for HSI clustering. As illustrated in Fig. 1, our method consists of feature extraction module, self-expressiveness module, adaptive self-expressive coefficient matrix initialization module, spectral clustering module, and self-supervised learning-based classification module. In this section, we first introduce each module of our network, and then introduce the implementation details.

A. Feature Extraction Module

The foundational component of our method is the feature extraction module, which nonlinearly maps the raw HSI samples into a latent space. To exploit the spatial information, the deep convolutional auto-encoders are used as the backbone network. Given HSI samples $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{m\times N}$ drawn from a union of n subspaces $\bigcup_{j=1}^{n}\{S_j\}$, where m, N, and n denote the spectral dimension, number of samples, and number of subspaces, respectively. Let $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N] \in \mathbb{R}^{p\times N}$ denotes the deep features of the input samples, i.e., the output of the encoder, where p is the dimension of the deep features. Then, $\mathbf{Z}$ is fed into the decoder to reconstruct the input samples $\mathbf{X}$. To ensure that the input samples $\mathbf{X}$ can be constructed by the

the auto-encoders, the loss function is defined as

$$\mathcal{L}_{DAE} = \frac{1}{2}\|\mathbf{X} - \hat{\mathbf{X}}\|_F^2 \tag{1}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, and $\hat{\mathbf{X}}$ denotes the samples reconstructed by the auto-encoders.

B. Self-Expressiveness Module

The self-expressiveness module is used to learn the self-expressive coefficient matrix, which introduces a fully connected (FC) layer (i.e., self-expressive layer [18]) without activation function and bias between the encoder and the decoder. The weights of the self-expressive layer can be considered as the self-expressive coefficient matrix. The loss of the self-expressiveness module is defined as

$$\|\mathbf{C}\|_\ell + \frac{1}{2}\|\mathbf{Z} - \mathbf{Z}\mathbf{C}\|_F^2, \text{ s.t. } \operatorname{diag}(\mathbf{C}) = \mathbf{0} \tag{2}$$

where $\mathbf{C} \in \mathbb{R}^{N\times N}$ denotes the parameters of the self-expressive layer. $\|\mathbf{C}\|_\ell$ is a regularization term to ensure that $\mathbf{C}$ have a block-diagonal structure. $\frac{1}{2}\|\mathbf{Z} - \mathbf{Z}\mathbf{C}\|_F^2$ is the self-expressiveness term. The constraint $\operatorname{diag}(\mathbf{C}) = \mathbf{0}$ denotes that the values of the diagonal elements of $\mathbf{C}$ are zeros, which is used to eliminate trivial solutions.

C. Self-Expressive Coefficient Matrix Initialization Module

This module is used to provide a good initialized self-expressive coefficient matrix $\mathbf{C}$ for self-expressive layer to address the issue of high dictionary redundancy. For this purpose, the correlated atoms need to be selected to represent the target features of samples, whereas the unrelated atoms should be well suppressed simultaneously. However, in recent DSC methods [19], [20], all elements of the self-expressive coefficient matrix are initialized with the same nonzero value, which tends to induce all the atoms in the global self-expressive dictionary to express each target feature. Consequently, the initialization approach of self-expressive coefficient matrix cannot address the problem of high redundancy of self-expressive dictionary atoms, which can degrade the clustering performance. This relationship can be clearly observed from the self-expressiveness property of the features:

$$\mathbf{z}_j = \sum_{i\neq j} \mathbf{C}_{i,j}\mathbf{z}_i + \mathbf{e}_j \tag{3}$$

where $\mathbf{z}_i$ and $\mathbf{z}_j$ are the ith and the jth columns of $\mathbf{Z}$ that denote the atom of self-expressive dictionary and the target feature, respectively. $\mathbf{e}_j$ denotes the noise. Since all the elements of C are initialized as $\mathbf{C}_{i,j} \neq 0$, $\mathbf{z}_j$ can be linearly expressed by all the atoms $\{\mathbf{z}_i \mid i = 1, \ldots, N,\ i \neq j\}$.

Based on the fact that similar HSI samples are more likely lying in the same subspace [12], the KNN graph is used to model the spectral similarity between each sample and its neighbors. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph, where $\mathcal{V}$ and $\mathcal{E}$ denote the set of nodes and edges, respectively. The adjacent matrix $\mathbf{A}$ of $\mathcal{G}$ is defined as

$$\mathbf{A}_{i,j} = \begin{cases} 1, & if\ \mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $\mathcal{N}_k(\mathbf{x}_i)$ denotes the set of neighbors of the sample $\mathbf{x}_i$. If $\mathbf{A}_{i,j} \neq 0$, sample i and sample j are similar and thus likely lying in the same subspace. Correspondingly, the features of these two samples are also likely lying in the same subspace. However, it is unreasonable to directly use the adjacent matrix to initialize self-expressive coefficient matrix since the self-expressive coefficients are generally smaller than 1. Moreover, the weights of neural networks are generally initialized to small random values [21], [22]. As a result, we update the adjacent matrix as follows:

$$\widetilde{\mathbf{A}}_{i,j} = \begin{cases} y, & if\ \mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where y is a small random value sampled from a uniform distribution $U[a, b]$. In this way, matrix $\widetilde{\mathbf{A}}$ not only retains the structure of adjacent matrix, but also meets the requirement that coefficients are smaller than 1. Consequently, $\widetilde{\mathbf{A}}$ can be used to induce the most correlated atoms to represent the target features. The flowchart of the initialization strategy is illustrated in Fig. 2.

D. Spectral Clustering Module

Spectral clustering module is used to generate clustering results. The parameters of the self-expressive layer (i.e., $\mathbf{C}$) are employed to construct an affinity matrix that is formulated as follows:

$$\mathbf{W} = \frac{1}{2}(|\mathbf{C}| + |\mathbf{C}|^T) \tag{6}$$

where $\mathbf{W} \in \mathbb{R}^{N\times N}$ is the affinity matrix with element $\mathbf{W}_{i,j}$ denoting the similarity between the ith and the jth samples. Then, the clustering results are produced by applying spectral clustering [46] to the affinity matrix.

E. Self-Supervised Learning-Based Classification Module

Since affinity matrix learning and spectral clustering DSC-based methods [18], [19] are independent, the features extracted from the convolutional encoder cannot be well-adopted to the subsequent spectral clustering due to the lack of effective supervision. To handle this problem, we use the cluster assignments generated from the spectral clustering as pseudo-labels to supervise the feature learning.

1) Self-Supervised Feature Learning: Inspired by [20], two FC layers are introduced on the top of the encoder as $p \times n_1 \times n_2 \times n$, where n_1 and n_2 are the numbers of neurons in the first and the second FC layers, respectively (see Fig. 1). The FC layers are served as a classifier that is trained with pseudo-labels and back-propagates to the encoder. The output of the FC layers is a multiple classification with a softmax function

$$P(Y = i \mid R, W, b) = \frac{e^{W_i R + b_i}}{\sum_j e^{W_j R + b_j}} \tag{7}$$

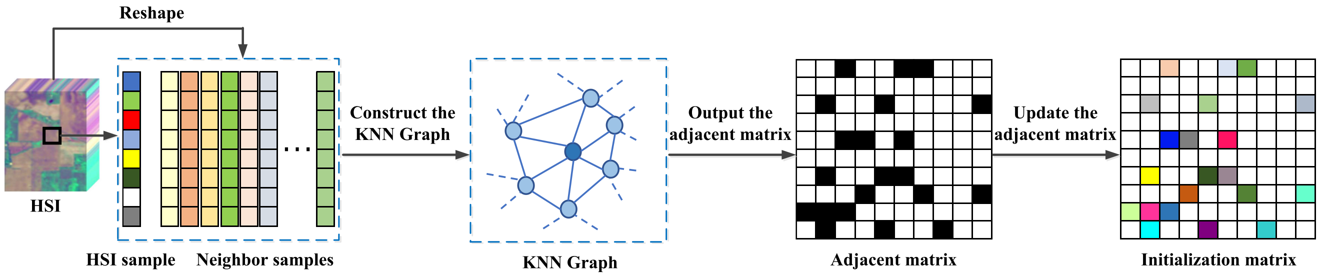


Fig. 2. Flowchart of the adaptive initialization of the self-expressive coefficient matrix. First, k-nearest neighbor (KNN) graph is constructed to model the spectral similarity between each target sample and its neighbors. Then, the adjacent matrix of KNN graph is updated to initialize the self-expressive coefficient matrix.

where R is the output of the encoder, W and b are the weights and biases of the FC layers, $P(\cdot)$ denotes the probability that the input belongs to the ith category.

Let $Q = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N] \in \{0,1\}^{n\times N}$ denote the output of the spectral clustering with each column $\mathbf{q}_i$ denoting the cluster assignments of the ith sample, and the output of the spectral clustering is fed into the FC layers and used as self-supervision information. We combine the cross-entropy loss and center loss to train the convolutional encoder. The loss function is defined as

$$\mathcal{L}_{CECT}(\mathbf{w}) = \frac{1}{N}\left(-\sum_{i=1}^{N}\mathbf{q}_i^T \log \mathbf{y}_i + \sum_{i=1}^{N}\|\mathbf{y}_i - \mathbf{c}_{\mathbf{y}_i}\|^2\right) \tag{8}$$

where $\mathbf{w}$ denotes the weights of the FC layers, $\mathbf{y}_i$ is the output of the FC layers at sample i calculated by (7), and $\mathbf{c}_{y_i}$ is the corresponding cluster center of $\mathbf{y}_i$ in the deep feature space. The first term of (8) denotes the cross-entropy loss that makes the features of different clusters separable, and the second term of (8) denotes center loss that minimizes the distances between the deep features and their cluster centers [47].

Highly confident pseudo-labels are useful for discriminative feature extraction and beneficial to the clustering [48]. However, some of the pseudo-labels produced by spectral clustering are incorrect and may misguide the feature learning. To handle this issue, it is necessary to introduce the confidence of samples to obtain highly confident pseudo-labels.

2) Selection of Samples With High Confidence: Inspired by [5], [49], we iteratively select highly confident pseudo-labels from each cluster to supervise the feature learning. Given the clustering results, the points in each cluster that are closer to their cluster center are assigned with high confidence. Let $\mathbf{U} \in \mathbb{R}^{N\times n}$ be the matrix containing the first n eigenvectors of graph Laplacian induced by the affinity matrix $\mathbf{W}$ as columns, and $\widetilde{U} = \{\mathbf{u}_j \mid j = 1, \ldots, n\}$ denotes the points consisting of the rows of the matrix $\mathbf{U}$. Then, k-means algorithm is applied to the points of $\widetilde{U}$ to obtain the clusters $A_1, \ldots, A_n$ and the corresponding cluster centers $\theta_1, \ldots, \theta_n$. The class-wise confidence is defined as

$$confidence_i = \rho \max_{\mathbf{u}_j \in A_i} \|\mathbf{u}_j - \theta_i\|_2^2 \tag{9}$$

where $\|\cdot\|_2^2$ denotes the Euclidean distance between point j and its cluster center. $\rho \in (0,1]$ is a parameter that controls the amount of selected samples. For each cluster A_i, the points with distances smaller than the $confidence_i$ have high confidence and their cluster assignments are considered to be highly confident. Correspondingly, the samples with high confidence are selected in the current clustering to train the encoder. Algorithm 1 illustrates the process of the sample selection in the current clustering. Note that, the samples selected in the current clustering are merged with the ones selected in the previous clustering, and no longer used as the candidates in the next clustering. Empirically, once the increment of selected samples is less than 0.5% of the number of total samples N, or the number of selected samples reaches 70% of N, the selection is forced to cease.

Algorithm 1: Selection of Samples With High Confidence in the Current Clustering

```
Input: $\widetilde{U}$, $S = \varnothing$
Perform $k$-means algorithm on $\widetilde{U}$.
for $i = 1, \ldots, n$ do
  Calculate the class-wise $confidence_i$ by (9).
  Calculate the distances between the $\theta_i$ and $A_i$.
  if $\|\mathbf{u}_j - \theta_i\|_2^2 \leq confidence_i$ do
    Sample $j$ is selected and $S \leftarrow S \cup \{j\}$.
  end if
end for
Output: Indexes of the selected samples $S$.
```

F. Implementation Details

The loss function of the proposed SDSC-AI method is defined as

$$\mathcal{L} = \mathcal{L}_{DAE} + \lambda_1\mathcal{L}_1 + \lambda_2\mathcal{L}_2 + \lambda_3\mathcal{L}_{CECT} \tag{10}$$

where $\mathcal{L}_1 = \|\mathbf{C}\|_\ell$, $\mathcal{L}_2 = \frac{1}{2}\|\mathbf{Z} - \mathbf{ZC}\|_F^2$. λ_1, λ_2, λ_3 are the weights to balance the contributions of different terms. Since the size of the datasets for HSI clustering is generally limited (e.g., in the order of thousands of samples), it is hard to directly train a network from scratch using these datasets. Therefore, the proposed network are trained following a two-stage pipeline: 1) pretrain the deep convolutional auto-encoders; and 2) train the whole network by alternately performing affinity matrix learning and spectral clustering.

Algorithm 2: Training Process of the SDSC-AI Network

```
Input: $\mathbf{X}$, the cluster number $n$, parameters $\lambda_1$, $\lambda_2$, $\lambda_3$, $\rho$, $k$, $a$, $b$, maximum iteration $T_{\max}$, $T_0$, $S = \varnothing$ and $t = 1$.
Pre-train the deep convolutional autoencoders.
Calculate and update the adjacent matrix according to (4) and (5).
Initialize the self-expressive coefficient matrix $\mathbf{C}$.
Initialize the FC layers.
Train self-expressive layer and construct $\mathbf{W}$ according to (6).
Perform spectral clustering to obtain $\mathbf{U}$ and $Q$.
while $t \leq T_{\max}$ do
  Select samples by Algorithm 1.
  if $size(S_t) \leq 0.005N$ or $size(S) \leq 0.7N$ do
    $S_t \leftarrow S \cup S_t$ and $S \leftarrow S_t$.
  end if
  Fix $Q$ and update the remaining parts for $T_0$ epochs according to (10).
  Perform spectral clustering to update $\mathbf{U}$ and $Q$.
  $\widetilde{U} \leftarrow \{\mathbf{u}_j \mid j = 1, \ldots, n,\ j \notin S\}, t \leftarrow t + 1$
end while.
Output: HSI clustering results
```

1) *Pretrain Stage:* Pretrain aims to obtain good initialization weights and reduce the reconstruction difficulty in the later fine-tune stage. In the pretrain stage, we only use the reconstruction loss $\mathcal{L}_{DAE}$ to update the autoencoders.
2) *Fine-Tune Stage:* In this stage, we train our network end-to-end with all the losses defined in (10). The main step of the fine-tune stage are described as follows. First, the self-expressive coefficient matrix is initialized, and the self-expressive layer is trained to learn the self-expressive coefficient matrix. Second, the affinity matrix is constructed and spectral clustering is performed to get the cluster assignments. Third, the samples with high confidence and their corresponding cluster assignments are selected. Then, the spectral clustering module is fixed, and the remaining modules of the network are updated for T_0 epoches. Finally, the spectral clustering is performed to update cluster assignments. The affinity matrix learning and spectral clustering are iteratively performed. The training process of our network is illustrated in Algorithm 2.

IV. Experimental Results and Discussions

In this section, we extensively evaluate the performance of the proposed method on three widely used HSI datasets. The proposed method is implemented in TensorFlow on a PC with a 3.0 GHz Intel i7 CPU and 64-GB memory. The network is trained on an Nvidia GeForce RTX 2080 Ti GPU.

A. Experimental Setup

1) Datasets and Preprocessing: Three publicly available HSI datasets (i.e., Indian Pines, Pavia University, and Salinas) are employed to validate the effectiveness of the proposed method. To build affinity matrix for spectral clustering, most SC methods need to solve large-scale optimization problems, and calculate pairwise similarities among all the samples at one time. Therefore, these methods require $O(N^2)$ memory to store the affinity matrix and may suffer from an "out-of-memory" error in the training phase. Following [7], [12], [13], [17], [19], [23], [50], a subset of these datasets is used in our method for computational efficiency. Particularly, the subset taken from Salinas dataset is also known as Salinas-A dataset.

TABLE I
Number of Labeled Samples for IPS, PUS, and SA Datasets

IPS		SA		PUS	
Class	GT	Class	GT	Class	GT
Corn_n_t	1005	Brocoli_gw1	391	Asphalt	425
Grass	730	Corn_sgw	1343	Meadows	768
Soybean_n_t	732	Lettuce_r4	616	Tree	63
Soybean_m_t	1924	Lettuce_r5	1525	Metal sheet	1315
		Lettuce_r6	674	Bare soil	2559
		Lettuce_r7	799	Bitumen	860
				Bricks	94
				Shadows	361
4	4391	6	5348	8	6445

- The subset of the Indian Pines dataset obtained by the airborne visible infrared imaging spectrometer (AVIRIS) contains 200 spectral features with a size of 145 × 145 pixels. Four main land-cover classes are considered in this dataset, including *corn_n_t, grass, soybeans_n_t*, and *soybeans_m_t*.
- The subset of the Pavia University dataset collected by the reflective optics spectrographic image system (ROSIS) sensor contains 103 spectral reflectance bands with a size of 200×100 pixels. Eight classes are considered in this dataset, including *asphalt, meadows, tree, metal sheet, bare soil, bitumen, bricks*, and *shadows*.
- The Salinas-A dataset captured by AVRIS contains 224 bands with a size of 86×83 pixels. Six different classes of crops are considered in this dataset, including *Brocoli_gw1*, *Corn_sgw*, *Lettuce_r4*, *Lettuce_r5*, *Lettuce_r6*, and *Lettuce_r7*.

We briefly denote the three datasets as IPS, PUS, and SA, respectively. Table I reports the ground truth of the three datasets. For all datasets, principal component analysis is performed before the training process to reduce computational cost, and we fix the number of reduced spectral bands to 4. Furthermore, we use $15 \times 15, 9 \times 9, 9 \times 9$ as the size of the spatial window to obtain image patches for each dataset, respectively. The influence of different window size is analyzed in Section IV-F.

2) Compared Methods: We compare the proposed method with several existing HSI clustering methods, including FCM [25], SSC [11], LRSC [34], SSCS [12] and S^{4}C [12], DLSS [51], TV [14], RMMF [9], LSSD [23], DSC [18], and S^{2}CSC [20]. Since the results of some compared methods are difficult to reproduce on different datasets, we compare the proposed method with the corresponding state-of-the-art methods on different datasets (i.e., SSCS and S^{4}C methods on the IPS dataset; TV and LSSD methods on the PUS dataset; and DLSS

TABLE II
NETWORK SETTINGS FOR IPS, PUS, AND SA DATASETS

layers	kernel size	channels	BN	ReLU
encoder1	3×3	32	Yes	Yes
encoder2	3×3	32	Yes	Yes
encoder3	3×3	16	No	Yes
decoder1	3×3	16	Yes	Yes
decoder2	3×3	32	Yes	Yes
decoder3	3×3	32	No	Yes

and LSSD methods on the SA dataset). The clustering results of other compared methods (FCM, SSC, LRSC, RMMF, DSC, and S^{2}CSC) are reported on all three datasets. All parameters of the compared clustering algorithms are empirically tuned to their optimum.

Both visual results and quantitative metrics [i.e., the overall accuracy (OA), the average accuracy (AA), and Kappa coefficient (Kappa)] are used for performance evaluation. The OA is defined as follows:

$$\mathrm{OA} = \max_{\mathrm{map}} \frac{\sum_{i=1}^{N} \mathbf{1}\{\mathbf{g}_i = \mathrm{map}(\mathbf{y}_i)\}}{N} \tag{11}$$

where $\mathbf{g}_i$ is the ground-truth label, $\mathbf{y}_i$ is the cluster assignment of sample $\mathbf{x}_i$ generated by clustering algorithm, and map is a mapping function that ranges over all possible one-to-one mappings between cluster assignments and ground-truth labels, respectively. The optimal mapping function can be computed by Hungarian algorithm [52]. The implementation of mapping function can be referred to the publicly available code of the DSC method [18].[1] In addition, running time is reported to evaluate the efficiency of different methods.

3) Networks Architecture and Parameter Setting: Since the number of samples in each dataset for HSI clustering is limited (i.e., 4391, 6445, and 5348 for the IPS dataset, PUS dataset, and SA dataset, respectively), the proposed network is expected to have less parameters to avoid overfitting. Therefore, the channel numbers of the deep convolutional autoencoders are set to 32–32–16–16–32–32 (see Table II). The kernel size of all the convolutions in the autoencoders is set to 3×3. The rectified linear unit (ReLU) is used as activation function. Batch normalization is used after the convolutional layers except for the last layer of the encoder and the last layer of the decoder. The ADAM optimizer is used with the learning rate being set to 2×10^{-4}. We set the maximum number of training epochs $T_{\max} = 200$. We update the encoder, the decoder, the self-expressive layer, and the FC layers for $T_0 = 20$ epochs and then update the spectral clustering once. For FC layers, we set $n_1 = 1024$ and $n_2 = n$. We use all the samples in each dataset to generate a batch during training phase.

We use ℓ_1 norm to define the sparse regularization term $\|\mathbf{C}\|_p$ in all experiments. The values of trade-off parameters λ_1, λ_2, and λ_3 are set to 0.001, 100, and 2000, respectively, and ρ is set to 0.3, 0.1, and 0.1 for the IPS, PUS, and SA datasets, respectively. The number of nearest neighbors *k* is set to 120, 40, and 120 for the IPS, PUS, and SA datasets, respectively. The interval of the uniform distribution is set to $[0.001, 0.008]$.

TABLE III
QUANTITATIVE EVALUATION OF DIFFERENT CLUSTERING ALGORITHMS ON THE IPS DATASET

Class	SSC	LRSC	FCM	SSCS	S^{4}C	RMMF	DSC	S^{2}CSC	SDSC-AI
Corn_n_t	1.49	45.47	44.78	58.05	61.00	30.05	88.06	94.53	**92.04**
Grass	48.63	86.44	99.86	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
Soybean_n_t	0.82	9.02	62.43	68.85	65.30	40.71	**100.00**	**100.00**	**100.00**
Soybean_m_t	89.91	57.38	59.98	64.76	65.28	93.40	52.96	50.57	**98.28**
OA (%)	47.96	51.42	63.54	68.12	70.08	71.21	76.66	77.09	**97.43**
AA (%)	35.21	49.58	66.76	72.92	72.90	66.04	85.26	86.27	**97.58**
Kappa (%)	15.76	31.45	49.18	55.45	58.25	56.09	68.88	69.16	**96.31**

TABLE IV
QUANTITATIVE EVALUATION OF DIFFERENT CLUSTERING ALGORITHMS ON THE PUS DATASET

Class	SSC	LRSC	FCM	TV	RMMF	LSSD	DSC	S^{2}CSC	SDSC-AI
Asphalt	0.00	3.53	0.00	98.82	0.00	0.00	97.88	97.88	**100.00**
Meadows	0.00	0.00	82.49	99.61	0.00	36.74	**100.00**	**100.00**	**100.00**
Tree	0.00	68.25	0.00	66.67	0.00	**100.00**	**100.00**	**100.00**	**100.00**
Metal Sheet	24.79	27.68	37.41	**100.00**	99.54	**100.00**	**100.00**	**100.00**	**100.00**
Bare soil	**100.00**	70.69	43.96	68.58	96.99	94.83	87.77	**100.00**	**100.00**
Bitumen	**100.00**	36.05	99.88	97.33	97.56	**100.00**	**100.00**	99.77	**100.00**
Bricks	0.00	3.19	34.04	0.00	0.00	97.87	0.00	11.70	**100.00**
Shadows	59.56	67.59	**100.00**	0.28	92.80	**100.00**	**100.00**	**100.00**	**100.00**
OA (%)	61.44	43.26	54.34	79.67	77.04	83.79	93.48	98.54	**100.00**
AA (%)	35.54	34.62	49.72	66.41	48.36	78.68	85.71	88.67	**100.00**
Kappa (%)	48.83	25.49	45.58	74.59	68.04	83.56	91.61	98.08	**100.00**

B. Results and Analyses

1) Qualitative Results: The cluster maps generated by different clustering methods are visualized in Figs. 3–5. Compared with the other methods, cluster maps produced by our method are very close to the ground-truth map, which clearly validates the effectiveness of our method. Taking the IPS dataset as an example, most methods cannot separate the four classes and generate many misclassifications due to the similar spectral signatures of the land-cover classes. In contrast, our method can better distinguish the four kinds of land-covers. Particularly, the "Grass" and "Soybean_n_t" classes are completely separated. For both PUS and SA datasets, there are no misclassifications in the cluster maps generated by our method.

2) Quantitative Results: The quantitative results achieved by different methods are summarized in Tables III–V, respectively, where the best results of each row are highlighted in italic. The following conclusions can be drawn from these results.

- The proposed method achieves the best clustering in terms of OA, AA, and Kappa on all three datasets. It should be noticed that OAs, AAs, and Kappas achieved by our method are 100% on both PUS and SA datasets, which outperforms all the compared methods by a notable margin.

[1] https://github.com/panji1990/Deep-subspace-clustering-networks.

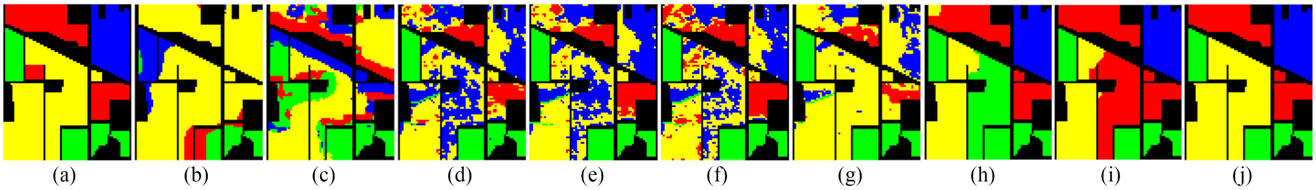


Fig. 3. Clustering maps generated by different methods on the IPS dataset. (a) GT. (b) SSC. (c) LRSC. (d) FCM. (e) SSCS. (f) S^{4}C. (g) RMMF. (h) DSC. (i) S^{2}CSC. (j) Our SDSC-AI.

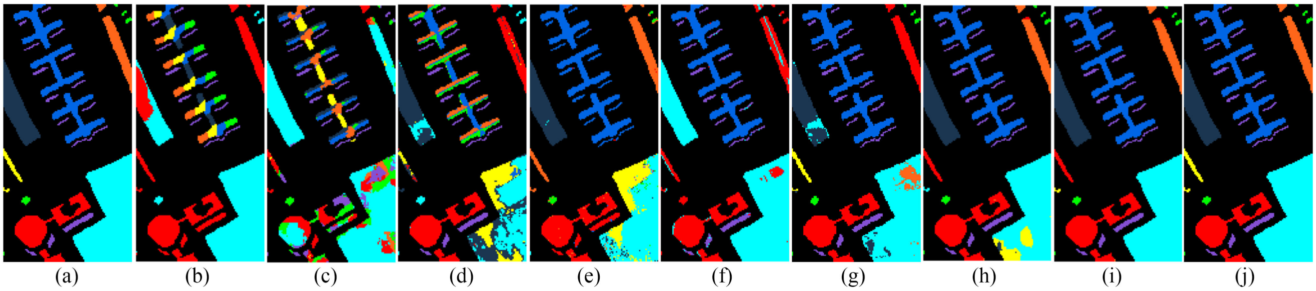


Fig. 4. Clustering maps generated by different methods on the PUS dataset. (a) GT. (b) SSC. (c) LRSC. (d) FCM. (e) TV. (f) RMMF. (g) LSSD. (h) DSC. (i) S^{2}CSC. (j) Our SDSC-AI.

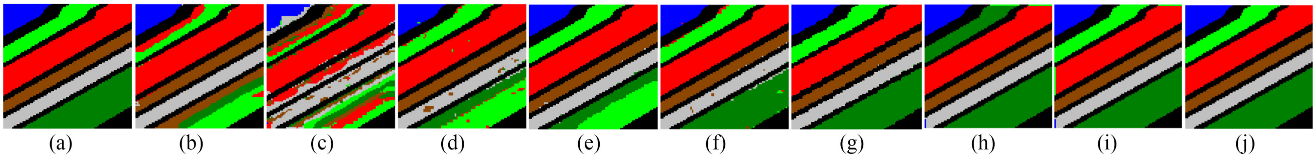


Fig. 5. Clustering maps generated by different methods on the SA dataset. (a) GT. (b) SSC. (c) LRSC. (d) FCM. (e) DLSS. (f) RMMF. (g) LSSD. (h) DSC. (i) S^{2}CSC. (j) Our SDSC-AI.

TABLE V
QUANTITATIVE EVALUATION OF DIFFERENT CLUSTERING ALGORITHMS ON THE SA DATASET

Class	SSC	LRSC	FCM	DLSS	RMMF	LSSD	DSC	S^{2}CSC	SDSC-AI
Brocoli_gw1	99.74	70.84	99.49	**100.00**	99.74	**100.00**	**100.00**	**100.00**	**100.00**
Corn_sgw	25.54	24.27	36.63	40.43	96.72	**100.00**	**100.00**	**100.00**	**100.00**
Lettuce_r4	63.31	30.52	96.27	**100.00**	94.16	**100.00**	3.25	**100.00**	**100.00**
Lettuce_r5	99.67	95.60	99.80	99.87	**100.00**	**100.00**	98.95	97.70	**100.00**
Lettuce_r6	97.77	10.53	**100.00**	99.11	99.26	99.85	**100.00**	**100.00**	**100.00**
Lettuce_r7	**100.00**	85.36	93.87	99.12	99.12	98.77	99.25	99.25	**100.00**
OA (%)	76.68	56.13	82.65	84.76	98.20	99.80	88.44	99.23	**100.00**
AA (%)	81.01	52.86	87.68	89.76	98.11	99.77	83.58	99.49	**100.00**
Kappa (%)	71.40	44.87	78.81	81.39	97.75	99.86	85.27	99.04	**100.00**

- Compared with the original S^{2}CSC method [20], the proposed method significantly improves the clustering performance on all three datasets. Specifically, the OA, AA, and Kappa on the IPS dataset are improved by our method by 20.34%, 11.31%, and 27.15%, respectively. That is because, the proposed method can precisely represent each target feature with the correlated atoms, and select highly confident pseudo labels to facilitate the self-supervised learning to extract discriminative features.
- The proposed method is robust to class unbalance. Note that, the class distribution is unbalanced on all three datasets, e.g., the numbers of the "tree" and "bricks" classes are 63 and 94 on the PUS dataset, respectively, which makes the clustering more challenging. Most methods cannot perform well on the two land-cover classes. For example, the DSC method [20] misclassifies the "Bricks" class into the "Asphalt" and "Bare soil" classes. In contrast, our method accurately recognizes these three classes.
- Compared with the clustering methods that only use the spectral information (FCM [25], SSC [11], LRSC [34], TV [14], DLSS [51]), the spectral-spatial clustering methods (SSCS [12], S^{4}C [12], LSSD [23], DSC [18], S^{2}CSC [20], SDSC-AI) can achieve better performance in terms of OA, AA, and Kappa. Taking the IPS dataset as an example, the SC-based methods such as SSC and LRSC achieve relatively low accuracies especially for the "Corn_no_till" class and the "Soybeans_n_t" class. This is

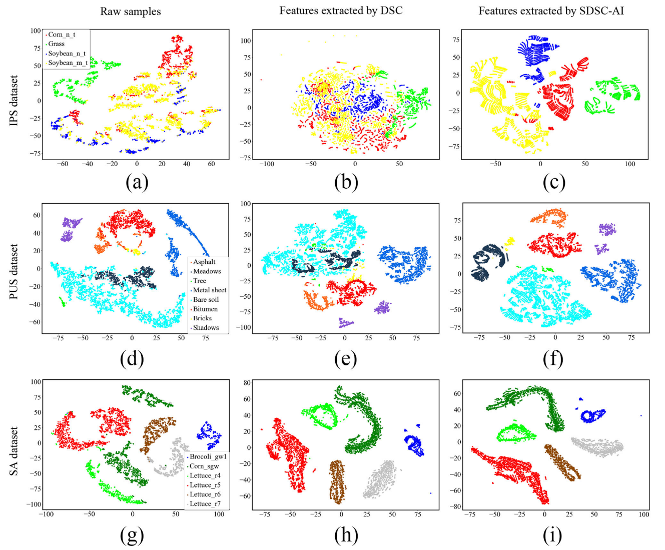


Fig. 6. Visualization of the raw HSI samples, the features extracted by the DSC method, and the features extracted by the SDSC-AI method with t-SNE [53].

because these methods only focus on the spectral information while the spectral signatures of the four land-cover classes are very similar and difficult to distinguish. In contrast, the SSCS, S^4C, DSC, S^2CSC, and our method significantly improve the clustering accuracy by incorporating the spatial neighborhood information.

- Generally, deep learning-based methods perform better than the conventional methods in terms of OA, AA, and Kappa due to their powerful capability of nonlinear feature extraction. For instance, the OAs achieved by the deep learning-based methods (LSSD [23], DSC [18], S^2CSC [20], SDSC-AI) are higher than those achieved by the conventional methods (SSC [11], LRSC [34], FCM [25], TV [14], RMMF [9]) on the PUS dataset.

C. *t-SNE Feature Visualization*

To investigate whether the features extracted by the proposed method are discriminative and benefcial to the SC, we use the t-distributed stochastic neighbor embedding (t-SNE) [53] approach to visualize the raw samples, the features produced by the DSC method [18], and the features produced by the SDSC-AI method on all datasets for comparison. First, as shown in Fig. 6(a), (d), and (g), the raw samples are mixed on the three datasets (especially on the IPS dataset) due to their similar spectral signatures. Therefore, it is difficult to separate the land-cover classes in original space. Second, although the interclass boundaries are apparent in the raw samples distribution, the distances between intraclass samples and their cluster centers are large (e.g., the “Corn_sgw” class, “Lettuce_r4” class, and “Lettuce_r5” class on the SA dataset). Third, although the features extracted by the DSC method can be separated on the SA dataset, they cannot be separated on both the IPS and PUS datasets [see Fig. 6(b), (e), and (h)] due to the large spectral variability on these two datasets. In contrast, the features extracted by our method are well separated on all three datasets [see Fig. 6(c), (f), and (i)] by introducing self-supervised learning. Particularly, the features are completely separated on both the PUS and SA datasets. It can be also observed that the features learned by our method are both interclass dispense and intraclass compact since the center loss penalizes the distances between the deep features and their cluster centers [47].

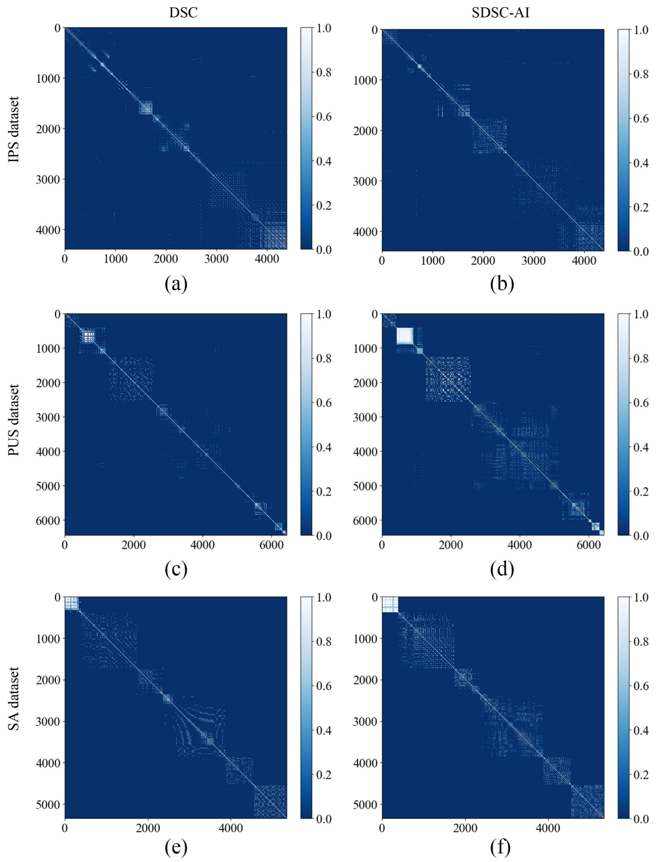


Fig. 7. Visualization of the affinity matrix obtained by the DSC method and our SDSC-AI method.

D. *Affinity Matrix Visualization*

Generally, affinity matrix represents the similarity between each pair of samples. The affinity matrix is constructed by the self-expressive coefficient matrix. If a group of samples lies in the same subspace, the corresponding self-expressive coefficients are nonzero, otherwise, they will be zero. Hence, an ideal affinity matrix is sparse and block-diagonal with each block signifying a land-cover class. To further demonstrate the effectiveness of the proposed method, we visualize the affinity matrices learned by both the DSC method [18] and the proposed method on all three datasets. As shown in Fig. 7, for all datasets, we can clearly observe that the affinity matrices obtained from the proposed method are superior to the ones obtained from the DSC method due to their sparsity and apparent block-diagonal structure. It clearly demonstrates that the proposed method can accurately represent each feature with the correlated atoms in the same subspace, and the deep features learned from the convolutional encoder benefit the affinity matrix learning.

TABLE VI
ABLATION STUDY OF THE PROPOSED METHOD

Dataset	S^{2}CSC	S^{2}CSC-AI	SDSC	SDSC-AI
IPS (%)	77.09	95.45	77.59	97.43
PUS (%)	98.54	99.98	99.98	100.00
SA (%)	99.23	99.98	99.70	100.00

Note that, "-AI" represents methods trained with adaptive self-expressive coefficient initialization, and "SDSC" represents methods trained with self-supervised learning using selected samples.

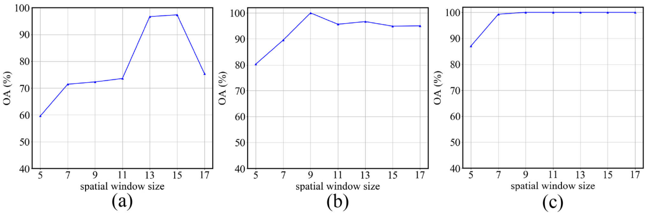


Fig. 8. Clustering OA with respect to different settings of spatial window size. (a) IPS dataset. (b) PUS dataset. (c) SA dataset.

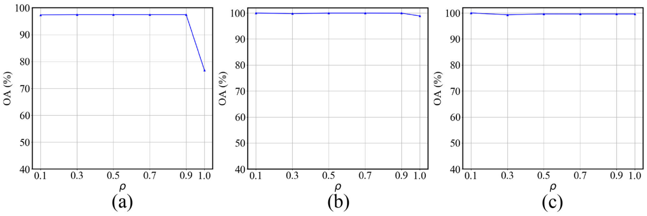


Fig. 9. Clustering OA with respect to different settings of ρ. (a) IPS dataset. (b) PUS dataset. (c) SA dataset.

TABLE VII
IMPACT OF THE INTERVAL OF UNIFORM DISTRIBUTION ON IPS, PUS, AND SA DATASETS

Dataset	[-0.04, 0.001]	[0.001, 0.008]	[0.003, 0.01]	[0.01, 0.05]	[0.05, 1.0]
IPS (%)	73.95	97.43	97.36	62.49	61.60
PUS (%)	70.24	100.00	85.82	55.84	54.45
SA (%)	73.74	100.00	99.96	64.17	64.23

E. Ablation Study

To evaluate the impact of adaptive self-expressive coefficient matrix initialization and sample selection in self-supervised learning, we perform ablation study on all three datasets. The S^{2}CSC method [20] is used as a baseline method. As shown in Table VI, the adaptive self-expressive coefficient matrix initialization can significantly improve the clustering accuracies on all three datasets. Specifically, 18.36%, 1.44%, and 0.75% improvements in OA metric can be achieved on the IPS, PUS, and SA datasets by using adaptive self-expressive coefficient matrix initialization. Compared with original S^{2}CSC method, the sample selection in self-supervised learning can also improve the clustering accuracies on the three datasets. Specifically, 0.5%, 1.44%, and 0.47% improvements in OA metric can be achieved on the IPS, PUS, and SA datasets. Note that, the clustering OA achieved by using sample selection method on the IPS dataset is relatively low. This is because the pseudo-labels obtained by spectral clustering are of low confidence due to the low-clustering OA achieved at the beginning of the training process. Therefore, it is difficult to select highly confident pseudo-labels. With the help of adaptive self-expressive coefficient matrix initialization, the clustering accuracies are significantly improved and the pseudo-labels are enhanced to be highly confident.

F. Parameter Sensitivity Analyses

In this section, we conduct experiments to investigate the influence of the important parameters on the clustering performance.

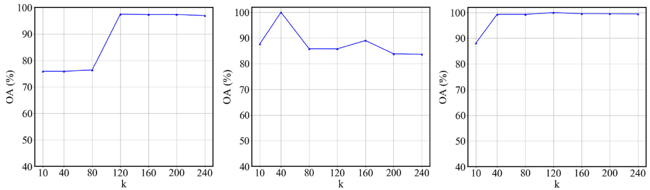
1) Impact of Spatial Window Size: Since spatial information of the center pixel is crucial to the spectral-spatial feature extraction [54], the size of spatial window can influence the clustering performance. The impact of the size of spatial window on the clustering results is presented in Fig. 8. It can be observed that the highest OAs are achieved when the window size is 15 and 9 on the IPS and PUS datasets, respectively. When the window size is larger than 9, the OA achieved by our method is 100% on the SA dataset. Generally, the image patch can exploit more spatial information with an increased window size. However, large window size will increase the computational burden and introduce noise [55]. Therefore, we set the window size to 15, 9, 9 for all the experiments on the three datasets, respectively.

2) Impact of ρ: Parameter ρ controls the number of selected samples in the self-supervised feature learning. Fig. 9 illustrates the impact of ρ on the clustering performance in which we set $\rho = \{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$. It can be clearly observed that the proposed method is insensitive to parameter ρ since the clustering results are stable with respect to different values of ρ. The proposed method achieves the highest OAs when $\rho = 0.3, 0.5, 0.7$, $\rho = 0.1$, $\rho = 0.1$ on the IPS, PUS, and SA datasets, respectively. Note that the proposed method uses all the cluster assignments to supervise the training of feature extraction in the case of $\rho = 1.0$, and the clustering performance degrades by different degrees on the three datasets. This is because, by setting $\rho = 1.0$, some low-confident pseudo-labels are adopted to supervise the network training. Hence, the selection of highly confident cluster assignments as pseudo-labels is important to the self-supervised learning.

3) Impact of Distribution Interval: We randomly selected several distribution intervals to investigate the influence of the distribution interval $[a, b]$ on the clustering performance. As shown in Table VII, for all datasets, the proposed method achieves the highest OAs when the distribution interval falls into the range of [0.001, 0.008]. Moreover, when the interval covers the value of zero (e.g., [−0.04, 0.001]), the clustering performance degrades since the structure of the initialized self-expressive coefficient matrix is changed. Finally, it can be seen that the clustering OAs significantly degrade when $a \geq 0.01$. Therefore, we set the distribution interval to [0.001, 0.008] for all the experiments on the three datasets.

TABLE VIII
RUNNING TIME OF THE DIFFERENT CLUSTERING METHODS ON THE IPS, PUS, AND SA DATASETS

Dataset	SSC	LRSC	FCM	SSCS/TV/DLSS	RMMF	S^4C/LSSD	DSC	S^{2}CSC	SDSC-AI
IPS (s)	593.24	2427.91	1.56	1032.70 (SSCS)	1.92	1567.90 (S^4C)	121.58	463.65	471.61
PUS (s)	1124.60	8363.19	1.74	6933.00 (TV)	2.60	674.58 (LSSD)	228.54	928.00	1037.16
SA (s)	756.35	4537.13	3.12	8.03 (DLSS)	2.40	578.32 (LSSD)	144.04	629.89	666.46



(a) (b) (c)

Fig. 10. Clustering OA with respect to different settings of k. (a) IPS dataset. (b) PUS dataset. (c) SA dataset.

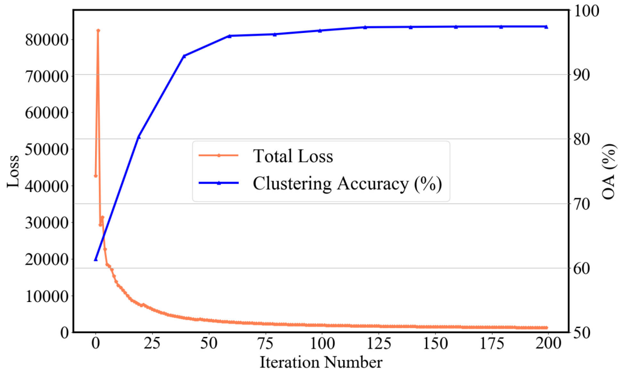


Fig. 11. Loss and clustering OA of our method during training phase on the IPS dataset.

4) Impact of k: Since the number of nearest neighbors k plays an important role in constructing the KNN graph and controls the numbers of the correlated atoms of each target feature, we set $k = \{10, 40, 80\ 120\ 160, 200\ 240\}$ as in [14] and conduct k-sensitivity experiments on the three datasets. The influence of the number of nearest neighbors k on the clustering results is shown in Fig. 10. It can be seen that the optimal value of k varies for different datasets. Moreover, we can further observe that the cluster performance tends to be saturated with an increasing k. However, since a large k will increase the computational burden and dictionary redundancy, we set $k = 120$, $k = 40$, $k = 120$ for all the experiments on the IPS, PUS, and SA datasets, respectively.

G. Convergence Analysis

To show the convergence of our network, we conduct convergence experiment on the IPS dataset. The maximum number of training iterations is set to 200. The clustering OA is computed every 20 iterations. As shown in Fig. 11, the loss values are fluctuant at the early training phase. Then, they decrease rapidly and tend to be stable. Meanwhile, the clustering OA increases gradually, and then tends to be saturated when the number of iteration is larger than 120. Therefore, the network can well converge within 200 iterations on the IPS dataset. In all experiments, we report the cluster results of the last iteration.

H. Running Time

We investigate the running time of different HSI clustering methods. As reported in Table VIII, conventional SC-based methods (SSC [11], LRSC [34], TV [14], SSC-S, and S^4C [12]) take more time than other clustering methods since they iteratively compute the representation coefficient matrices. Compared with deep learning-based methods (LSSD [23] and DSC [18]), our method takes more running time since it takes most computational time on the iterative self-supervised learning process and initialization. Moreover, compared with S^{2}CSC [20], our method takes more time in sample selection and initialization. Although FCM [25], DLSS [51], and RMMF [9] methods are very efficient, they achieve lower accuracies than our method. To sum up, our method achieves a good tradeoff between clustering performance and computational efficiency.

V. CONCLUSION

In this article, we propose an end-to-end trainable network named SDSC-AI for HSI clustering. Specifically, we introduce self-supervised learning for feature extraction to make sure that the learned features are well-adapted to subsequent SC. Moreover, we design a spectral similarity based adaptive self-expressive coefficient matrix initialization strategy to enhance the clustering performance. The experimental results demonstrate the superiority of the proposed method as compared to several state-of-the-art HSI clustering methods. To build the affinity matrix for spectral clustering, the proposed method needs to integrate all samples in one batch to train the network, which makes it difficult to scale for large HSI data. The scalability problem will be studied in our future work.

REFERENCES

[1] M. Berman, P. Connor, L. Whitbourn, D. Coward, B. Osborne, and M. Southan, "Classification of sound and stained wheat grains using visible and near infrared hyperspectral image analysis," *J. Near Infrared Spectrosc.*, vol. 15, no. 6, pp. 351–358, 2007.

[2] P. Ghamisi *et al.*, "Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 37–78, Dec. 2017.

[3] F. A. Kruse, J. W. Boardman, and J. F. Huntington, "Comparison of airborne hyperspectral data and eo-1 hyperion for mineral mapping," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 6, pp. 1388–1400, Jun. 2003.

[4] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.

[5] Y. Qin, L. Bruzzone, B. Li, and Y. Ye, “Cross-domain collaborative learning via cluster canonical correlation analysis and random walker for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3952–3966, Jun. 2019.

[6] L. Zhou *et al.*, “Subspace structure regularized nonnegative matrix factorization for hyperspectral unmixing,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 4257–4270, Jul. 2020.

[7] Y. Cai, Z. Zhang, Z. Cai, X. Liu, and Q. Yan, “Graph convolutional subspace clustering: A robust subspace clustering framework for hyperspectral image,” *IEEE Trans. Geosci. Remote Sens.*, early access, Aug. 31, 2020, doi: 10.1109/TGRS.2020.3018135.

[8] 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.

[9] L. Zhang, L. Zhang, B. Du, J. You, and D. Tao, “Hyperspectral image unsupervised classification by robust manifold matrix factorization,” *Inf. Sci.*, vol. 485, pp. 154–169, 2019.

[10] R. Vidal, “Subspace clustering,” *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, Feb. 2011.

[11] E. Elhamifar and R. Vidal, “Sparse subspace clustering: Algorithm, theory, and applications,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

[12] H. Zhang, H. Zhai, L. Zhang, and P. Li, “Spectral-spatial sparse subspace clustering for hyperspectral remote sensing images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3672–3684, Jun. 2016.

[13] H. Zhai, H. Zhang, L. Zhang, P. Li, and A. Plaza, “A new sparse subspace clustering algorithm for hyperspectral remote sensing imagery,” *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 1, pp. 43–47, Jan. 2016.

[14] H. Zhai, H. Zhang, L. Zhang, and P. Li, “Total variation regularized collaborative representation clustering with a locally adaptive dictionary for hyperspectral imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 166–180, Jan. 2018.

[15] S. Huang, H. Zhang, and A. Pižurica, “Joint sparsity based sparse subspace clustering for hyperspectral images,” in *Proc. 25th IEEE Int. Conf. Image Process.*, 2018, pp. 3878–3882.

[16] R. Wang, F. Nie, and W. Yu, “Fast spectral clustering with anchor graph for large hyperspectral images,” *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 11, pp. 2003–2007, Nov. 2017.

[17] H. Zhai, H. Zhang, X. Xu, L. Zhang, and P. Li, “Kernel sparse subspace clustering with a spatial max pooling operation for hyperspectral remote sensing data interpretation,” *Remote Sens.*, vol. 9, no. 4, pp. 335–349, 2017.

[18] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, “Deep subspace clustering networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 24–33.

[19] M. Zeng, Y. Cai, X. Liu, Z. Cai, and X. Li, “Spectral-spatial clustering of hyperspectral image based on Laplacian regularized deep subspace clustering,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 2694–2697.

[20] J. Zhang *et al.*, “Self-supervised convolutional subspace clustering network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5473–5482.

[21] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.

[23] Y. Qin, L. Bruzzone, and B. Li, “Learning discriminative embedding for hyperspectral image clustering based on set-to-set and sample-to-sample distances,” *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 473–485, Jan. 2019.

[24] S. Lloyd, “Least squares quantization in PCM,” *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

[25] W. Pedrycz, “Fuzzy sets in pattern recognition: Methodology and methods,” *Pattern Recognit.*, vol. 23, no. 1–2, pp. 121–146, 1990.

[26] S. Chen and D. Zhang, “Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure,” *IEEE Trans. Syst. Man Cybern. Part B*, vol. 34, no. 4, pp. 1907–1916, Aug. 2004.

[27] 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.

[28] A. Rodriguez and A. Laio, “Clustering by fast search and find of density peaks,” *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[29] H. Xie *et al.*, “Unsupervised hyperspectral remote sensing image clustering based on adaptive density,” *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 4, pp. 632–636, Apr. 2018.

[30] Y. Zhong, S. Zhang, and L. Zhang, “Automatic fuzzy clustering based on adaptive multi-objective differential evolution for remote sensing imagery,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 5, pp. 2290–2301, Oct. 2013.

[31] U. V. Luxburg, “A tutorial on spectral clustering,” *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

[32] L. Zhou, X. Bai, X. Liu, J. Zhou, and E. R. Hancock, “Learning binary code for fast nearest subspace search,” *Pattern Recognit.*, vol. 98, 2020, Art. no. 107040.

[33] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, “Robust recovery of subspace structures by low-rank representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2012.

[34] X. Zhu, S. Zhang, Y. Li, J. Zhang, L. Yang, and Y. Fang, “Low-rank sparse subspace for spectral clustering,” *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 8, pp. 1532–1543, Aug. 2018.

[35] L. Zhou *et al.*, “Latent distribution preserving deep subspace clustering,” in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 4440–4446.

[36] L. Jing and Y. Tian, “Self-supervised visual feature learning with deep neural networks: A survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, May 04, 2020, doi: 10.1109/TPAMI.2020.2992393.

[37] X. Liu *et al.*, “Self-Supervised learning: Generative or contrastive,” 2020, *arXiv:2006.08218*.

[38] C. Doersch, A. Gupta, and A. A. Efros, “Unsupervised visual representation learning by context prediction,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1422–1430.

[39] M. Noroozi and P. Favaro, “Unsupervised learning of visual representations by solving jigsaw puzzles,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 69–84.

[40] S. Gidaris, P. Singh, and N. Komodakis, “Unsupervised representation learning by predicting image rotations,” in *Proc. Int. Conf. Learn. Representation*, 2018.

[41] R. D. Hjelm *et al.*, “Learning deep representations by mutual information estimation and maximization,” in *Proc. Int. Conf. Learn. Representation*, 2019.

[42] P. Bachman, R. D. Hjelm, and W. Buchwalter, “Learning representations by maximizing mutual information across views,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 15535–15545.

[43] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.

[44] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[45] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, “Deep clustering for unsupervised learning of visual features,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 132–149.

[46] A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 849–856.

[47] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, “A discriminative feature learning approach for deep face recognition,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 499–515.

[48] J. Wu *et al.*, “Deep comprehensive correlation mining for image clustering,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8150–8159.

[49] Y. Zou, Z. Yu, B. Vijaya Kumar, and J. Wang, “Unsupervised domain adaptation for semantic segmentation via class-balanced self-training,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 289–305.

[50] Y. Qin, B. Li, W. Ni, S. Quan, P. Wang, and H. Bian, “Affinity matrix learning via non-negative matrix factorization for hyperspectral imagery clustering,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 402–415, Jan. 2021, doi: 10.1109/JSTARS.2020.3040218.

[51] J. M. Murphy and M. Maggioni, “Unsupervised clustering and active learning of hyperspectral images with nonlinear diffusion,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1829–1845, Mar. 2018.

[52] J. Munkres, “Algorithms for the assignment and transportation problems,” *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, 1957.

[53] L. v. d. Maaten and G. Hinton, “Visualizing data using t-SNE,” *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[54] 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.

[55] 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.

Kun Li received the B.S degree in computer science and technology from the China University of Mining and Technology, Xuzhou, China, in 2007, and the M.S degree in computer application technology from the Tianjin University of Technology, Tianjin, China, in 2010. He is currently working toward the Ph.D. degree at the College of Electronic Science and Technology, National University of Defense Technology, Changsha, China.

His research interests include pattern recognition and hyperspectral image processing.

Yao Qin received the B.S degree in information engineering from Shanghai Jiaotong University, Shanghai, China, in 2013, and the M.S and Ph.D. degrees in information and communication engineering from the National University of Defense Technology, Changsha, China, in 2015 and 2019, respectively.

He was a Visiting Ph.D. student at the Remote Sensing Laboratory, Department of Information Engineering and Computer Science, University of Trento, Italy. His research interests include infrared small target detection, hyperspectral image classification and clustering, and domain adaptation.

Qiang Ling received the B.Eng. degree in measurement engineering and the M.Eng. degree in control science and engineering from Air Force Engineering University, Shanxi, China, in 2013 and 2016, and the Ph.D. degree in information and communication engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2019.

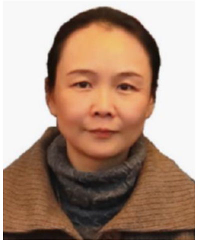
He is currently a Lecturer with the College of Electronic Science and Technology, NUDT. His research interests include pattern recognition and hyperspectral image processing.

Yingqian Wang received the B.E. degree in electrical engineering from Shandong University (SDU), Jinan, China, in 2016, and the M.E. degree in information and communication engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2018. He is currently working toward the Ph.D. degree at the College of Electronic Science and Technology, NUDT.

He has authored a number of papers in journals and conferences such as TPAMI, TIP, CVPR, and ECCV. His research interests include low-level vision, particularly on light field imaging, and image super-resolution.

Zaiping Lin received the B.Eng. and Ph.D. degrees from the National University of Defense Technology (NUDT), Changsha, China, in 2007 and 2012, respectively.

He is currently an Associate Professor with the College of Electronic Science and Technology, NUDT. His research interests include infrared image processing and signal processing.



Wei An received the Ph.D. degree from the National University of Defense Technology (NUDT), Changsha, China, in 1999.

In 2016, she visited the University of Southampton, Southampton, U.K., as a Senior Visiting Scholar. She is currently a Professor with the College of Electronic Science and Technology, NUDT. She has authored or co-authored over 100 journal and conference publications. Her research interests include signal processing and image processing.