Article

Deep Spatial-Spectral Subspace Clustering for Hyperspectral Images Based on Contrastive Learning

Xiang Hu ¹ , Teng Li ^{2,3}, Tong Zhou ^{1,2} and Yuanxi Peng ^{1,*}

¹ The State Key Laboratory of High-Performance Computing, College of Computer, National University of Defense Technology, Changsha 410073, China; huxiang@nudt.edu.cn (X.H.); zhouotong09@nudt.edu.cn (T.Z.)

² Beijing Institute for Advanced Study, National University of Defense Technology, Beijing 100020, China; liteng09@nudt.edu.cn

³ College of Advanced Interdisciplinary Studies, National University of Defense Technology, Changsha 410073, China

* Correspondence: pyx@nudt.edu.cn

Abstract: Hyperspectral image (HSI) clustering is a major challenge due to the redundant spectral information in HSIs. In this paper, we propose a novel deep subspace clustering method that extracts spatial-spectral features via contrastive learning. First, we construct positive and negative sample pairs through data augmentation. Then, the data pairs are projected into feature space using a CNN model. Contrastive learning is conducted by minimizing the distances of positive pairs and maximizing those of negative pairs. Finally, based on their features, spectral clustering is employed to obtain the final result. Experimental results gained over three HSI datasets demonstrate that our proposed method is superior to other state-of-the-art methods.

Keywords: hyperspectral image clustering; deep subspace clustering; deep learning; spectral clustering

Citation: Hu, X.; Li, T.; Zhou, T.; Peng, Y. Deep Spatial-Spectral Subspace Clustering for Hyperspectral Images Based on Contrastive Learning. *Remote Sens.* **2021**, *13*, 4418. <https://doi.org/10.3390/rs13214418>

Academic Editors: Claude Cario and Steven Le Moan

Received: 26 September 2021

Accepted: 29 October 2021

Published: 3 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Hyperspectral remote sensing has been widely used in many different fields [1–3]. Hyperspectral image (HSI) classification is a fundamental issue and a hot topic in hyperspectral remote sensing. HSIs can provide rich spectral and spatial information, which improves the utility of HSIs in various applications. However, the abundant spectral information also causes a low classification accuracy, which is called the Hughes phenomenon. Moreover, the limited number of labeled hyperspectral samples also causes difficulties in hyperspectral image classification. In the real world, more and more hyperspectral data are becoming available with the development of information acquisition technology. However, most of these data are unlabeled, and labeling the data is an extremely laborious and time-consuming process. Nevertheless, HSI clustering focus on achieving a good classification performance without training labels. Thus, HSI clustering has attracted increasing levels of attention in recent years.

Some traditional methods used for natural images have been applied in the study of HSI clustering [4–8]. The complex characteristics of HSIs strongly reduce their accuracy. Subsequently, more and more HSI clustering methods have been proposed. These methods can be divided into two main groups: spectral-only methods and spatial-spectral methods. Spectral-only methods ignore the spatial information of HSIs, which limits the performance of these methods. To improve accuracy, some spatial-spectral clustering methods have been proposed [9–12].

Additionally, to solve problems relating to high dimensionality, some methods based on sparse subspace clustering (SSC) [13] have been proposed. Those methods rely on clustering HSI data in the low-dimensional subspace. However, the subspace that HSI data exists in is usually non-linear. This limits the performance of these clustering methods.

Recently, deep learning has achieved great success in the computer vision field [14–17]. To handle the challenge of nonlinearity, many deep learning-based methods have been

proposed. Zhong et al. [18] proposed a spectral–spatial residual network (SSRN) based on ResNet [19]. Inspired by DenseNet [20], Wang et al. [21] designed a fast dense spectral–spatial convolution network (FDSSC). Ma et al. [22] adopted a two-branch architecture and proposed a double-branch multi-attention mechanism network (DBMA). Li et al. [23] introduced the self-attention mechanism to their double-branch dual-attention mechanism network (DBDA).

For HSI clustering, most of the existing deep-learning-based clustering methods can be divided into two steps: feature extraction via deep learning models and traditional clustering. Auto-encoders are used in deep clustering as feature extractors under unsupervised conditions. By encoding images into features and reconstructing images from the features, the model can extract features from HSIs without labels. Based on these features, traditional clustering methods or classification layers can be used to obtain the clustering result. For example, Zeng et al. [24] proposed a Laplacian regularized deep subspace clustering method (LRDSC) for HSI clustering. In this method, a 3D auto-encoder network with skip connections is used to extract spatial–spectral features. Lei et al. [25] designed a multi-scale auto-encoder to obtain spatial–spectral information for HSI clustering. Inputs at different scales can provide different types of information, but can increase the computation significantly.

However, the auto-encoder used for HSI processing requires an inordinate amount of computational resources due to the need to reconstruct the input data. Recently, contrastive learning was proposed as a means to extract features under unsupervised conditions. Unlike autoencoders, contrastive learning models operate on different augmented views of the same input image. Since these methods do not require image reconstruction, they require fewer computational resources. Li et al. [26] proposed a clustering method based on contrastive learning.

To the best of our knowledge, there has been little research on contrastive learning methods for HSI processing. The contrastive learning methods used for typical RGB images can not be applied directly to HSI processing because some typical RGB image augmentation methods are not available for HSIs. For example, color distortion for typical RGB images will destroy spectral information when used on HSIs. We explore HSI augmentation by removing the spectral information of some non-central pixels. Different methods of selecting pixels to remove spectral information can be considered as different HSI augmentation methods.

In this paper, we propose a clustering method for HSIs based on contrastive learning. Firstly, we use contrastive learning methods to train a CNN model to extract features from HSIs. Then, we apply a spectral clustering algorithm to these features. The main contributions of our study are summarized as follows.

1. Inspired by DBMA and DBDA, we designed a double-branch dense spectral–spatial network for HSI clustering. These two branches can extract spectral and spatial features separately, avoiding the huge computation caused by multi-scale inputs. To reduce the computational load further, we remove the attention blocks in DBDA and DBMA.
2. We use contrastive learning to explore spatial–spectral information. We augment the image by removing the spectral information of some non-central pixels. Different methods of selecting pixels to remove spectral information can provide different augmented views of the HSI block.
3. The experimental results obtained over three publicly available HSI datasets demonstrate the superiority of our proposed method compared to other state-of-the-art methods.

The rest of this paper is organized as follows. A brief overview of related work is presented in Section 2. Our proposed method is described in Section 3. Sections 4 and 5 provide an analysis of the results and a discussion.

2. Related Works

2.1. Traditional Clustering for HSIs

Spectral-only methods only use spectral information. For example, Paoli et al. [27] proposed a method for estimating the class number, extracting features, and performing clustering simultaneously. Zhong et al. [28] introduced an artificial immune network for HSI clustering. However, the absence of spatial information affects the accuracy of these methods.

Spatial-spectral clustering methods based on both spatial information and spectral information can provide a higher accuracy than spectral-only methods. Chen et al. [10] proposed a spatial constraint based fuzzy C-means method for HSI clustering. Murphy and Magioni [12] combined spatial-spectral information and diffusion-inspired labeling to create a diffusion learning-based spatial-spectral clustering method (DLSS).

Many sparse subspace clustering (SSC) [13]-based methods have also been proposed for HSI clustering. Zhai et al. [29] proposed a band selection method. Tian et al. [30] applied Gaussian kernels and proposed a kernel spatial-spectral-based multi-view low-rank sparse subspace clustering method. Zhang et al. [31] designed a spectral-spatial sparse subspace clustering (S^4C) algorithm that utilizes the spectral similarity of a local neighborhood. However, these methods cannot handle the problem of the non-linear subspace structure of HSIs, which decreases their accuracy enormously.

2.2. Deep Clustering for HSIs

Many deep learning-based clustering methods have been proposed recently. A study proposing a deep embedded clustering (DEC) [32] method was the first to propose using deep networks to learn feature representations and cluster assignments simultaneously. Chang et al. [33] designed a deep adaptive image clustering (DAC) method using a binary constrained pairwise-classification model for clustering. Fard et al. [34] proposed a novel approach for addressing the problem of joint clustering and learning representations. Barthakur and Sarma [35] proposed a deep learning-based method for the semantic segmentation of satellite images in a complex background. Sodjinou et al. [36] proposed a deep semantic segmentation-based algorithm to segment crops and weeds in agronomic color images. Based on SSC, Ji et al. [37] used convolutional autoencoders to map data into a latent space and achieved a more robust clustering result than could be gained using traditional clustering methods. A generative adversarial network (GAN) [38,39] was also used to cluster normal images.

For HSI clustering, Egaña et al. [40] proposed a novel methodology for geometallurgical sample characterization based on HSI data. Xu et al. [41] proposed a novel context-aware unsupervised discriminative ELM method for HSI clustering. Zeng et al. [24] applied skip connections and proposed a Laplacian regularized deep subspace clustering (LRDSC) method for HSI clustering. Lei et al. [25] designed a multi-scale 3D auto-encoder network for HSI clustering. Different input sizes can encourage the model to extract features from different scales. However, these methods aim to reconstruct data, which greatly increases the amount of computation required. Moreover, using a multi-scale network further increases the amount of computation. We used a two-branch CNN model in our method. One branch is used to extract spectral information and the other is used to extract spatial information. We believe that this can play the same role as multi-scale inputs without imposing the same computational burden.

2.3. Contrastive Learning

As a recently proposed unsupervised learning method, contrastive learning has achieved a promising performance. Different from autoencoder and GAN, the contrastive learning method does not focus on generating data. Instead, it maps the data to a feature space by maximizing the distances of negative pairs and minimizing the distances of positive pairs. The positive pair contains two different augmented views of the same sample and the other pairs between different samples are regarded as negative. Several contrastive

learning methods have been proposed for normal images, such as similar contrastive learning (SimCLR) [42], momentum contrast for unsupervised visual representation learning (MoCo) [43], and bootstrap your own latent (BYOL) [44].

For clustering, Li et al. [26] proposed an online clustering method named Contrastive Clustering (CC) that can explicitly perform instance- and cluster-level contrastive learning. Inspired by CC, we used the contrastive clustering method to train the CNN model. Then, we adopted a traditional spectral clustering algorithm rather than a simple layer to obtain the clustering result.

3. Method

Our proposed method consists of two stages: training and testing. Firstly, we used two augmented versions of HSI to train our CNN model. After training, we used the CNN model to obtain the features. Finally, we applied the spectral clustering algorithm based on the features to obtain the clustering result.

3.1. Augmentation in Our Experimental Method

We use two different composite methods to augment the HSI image. The augmentation methods are based on two steps. First, we use horizontal flip or vertical flip as the preliminary augmentation method. Then, we select some non-central pixels in the input blocks to remove spectral information. The different ways in which these pixels are selected can result in different augmentation methods, as illustrated in Algorithms 1 and 2, and Figure 1. The size of the rectangular area in Algorithm 1 is not fixed.

Figure 1. The augmentation methods used in our proposed method.

Algorithm 1 Selecting Random Rectangular Area to Remove Spectral Information.

- 1: **Input:** input image I ; image size $w \times h \times c$.
 - 2: **Output:** augmented image I^* .
 - 3: Generate a matrix of the size $(w \times h)$ using 1
 - 4: Select a random submatrix in this matrix and change the elements inside to 0
 - 5: **if** the center point of the matrix is in the submatrix **then**
 - 6: change the element of that point to 1
 - 7: **end if**
 - 8: **for** $i = 1$ to c **do**
 - 9: multiply the image in the i th channel by this matrix to obtain the augmented image I^*
 - 10: **end for**
 - 11: Return the augmented image I^*
-

Algorithm 2 Selecting Discrete Points to Remove Spectral Information.

```

1: Input: input image  $I$ ; image size  $w \times h \times c$ 
2: Output: augmented image  $I^*$ 
3: Use 0 and 1 with the same probability to generate a random matrix of the size  $(w \times h)$ 
4: if the center point of the matrix is 0 then
5:     change the element of that point to 1
6: end if
7: for  $i = 1$  to  $c$  do
8:     multiply the image in the  $i$ th channel by this matrix to obtain the augmented image  $I^*$ 
9: end for
10: Return the augmented image  $I^*$ 

```

3.2. Architectures of Our Experimental Models

Our proposed method is illustrated in Figure 2. We use a two-branch CNN model as the backbone model. The double-branch architecture can reduce the interference between spectral and spatial features. The backbone of the CNN model is shown in Figure 3. To keep the network architecture the same for different hyperspectral images with different bands, we use the PCA method to reduce the dataset dimension to 100. The parameters of the 3D convolutions and batchnorms in our model are illustrated in Table 1. A detailed introduction of these datasets is presented in Section 4.1. The two MLPs in our method are shown in Figure 4. The parameters of these MLPs can be seen in Table 2. For MLP II, the final output dimension is equal to the cluster number.

Table 1. Parameters of the 3D convolutions and batchnorms in our model.

Layer	Input Shape	Output Shape	Parameters	Padding	Kernel_Size	Stride
Conv11	[1,9,9,100]	[24,9,9,47]	192	(0,0,0)	(1,1,7)	(1,1,2)
Conv12	[24,9,9,47]	[12,9,9,47]	2028	(0,0,3)	(1,1,7)	(1,1,1)
Conv13	[36,9,9,47]	[12,9,9,47]	3036	(0,0,3)	(1,1,7)	(1,1,1)
Conv14	[48,9,9,47]	[12,9,9,47]	4044	(0,0,3)	(1,1,7)	(1,1,1)
Conv15	[60,9,9,47]	[60,9,9,1]	169,260	(0,0,0)	(1,1,47)	(1,1,1)
Conv21	[1,9,9,100]	[24,9,9,1]	2424	(0,0,0)	(1,1,100)	(1,1,1)
Conv22	[24,9,9,1]	[12,9,9,1]	2604	(1,1,0)	(3,3,1)	(1,1,1)
Conv23	[36,9,9,1]	[12,9,9,1]	3900	(1,1,0)	(3,3,1)	(1,1,1)
Conv24	[48,9,9,1]	[12,9,9,1]	5196	(1,1,0)	(3,3,1)	(1,1,1)
Layer	Input Shape	Output Shape	Parameters	eps	Momentum	Affine
BN11	[24,9,9,47]	[24,9,9,47]	48	0.001	0.1	True
BN12	[36,9,9,47]	[36,9,9,47]	72	0.001	0.1	True
BN13	[48,9,9,47]	[48,9,9,47]	96	0.001	0.1	True
BN14	[60,9,9,47]	[60,9,9,47]	120	0.001	0.1	True
BN21	[24,9,9,1]	[24,9,9,1]	48	0.001	0.1	True
BN22	[36,9,9,1]	[36,9,9,1]	72	0.001	0.1	True
BN23	[48,9,9,1]	[48,9,9,1]	96	0.001	0.1	True
BN3	[120,9,9,1]	[120,9,9,1]	240	0.001	0.1	True

Total params: 193,476

Trainable params: 193,476

Non-trainable params: 0

Total mult-adds (M): 50.02

Input size (MB): 0.03

Forward/backward pass size (MB): 6.84

Params size (MB): 0.74

Estimated Total Size (MB): 7.61

Table 2. Parameters of the two MLPs.

MLP I			MLP II		
Layer	Output Shape	Parameter	Layer	Output Shape	Parameter
Linear	[120]	14,520	Linear	[120]	14,520
Relu	[120]	0	Relu	[120]	0
Linear	[256]	30,976	Linear	[4]	484
			Softmax	[4]	0
Total params: 45,496			Total params: 15,004		
Trainable params: 45,496			Trainable params: 15,004		
Non-trainable params: 0			Non-trainable params: 0		
Total mult-adds (M): 0.09			Total mult-adds (M): 0.03

Figure 2. The overall architecture of our proposed method. **Figure 3.** The architecture of our backbone CNN model.

Figure 4. The architecture of our MLPs.

3.3. Summary of Our Experimental Method

The overall architecture of our proposed method is shown in Algorithm 3 and Figure 3. Firstly, we use different augmentations to generate different views of input. Then, we train the CNN model. After training, we can obtain the features of input HSIs via the CNN model. Finally, we use the spectral clustering algorithm based on the features to obtain the clustering result.

Algorithm 3 Our proposed clustering algorithm.

- 1: **Input:** dataset I ; pixel block size $w \times h \times c$; training epochs E ; batch size N .
 - 2: **Output:** cluster assignments.
 - 3: Sample pixel block of size $w \times h \times c$ from the dataset I
 - 4: //training
 - 5: **for** epoch = 1 to E **do**
 - 6: compute instance-level contrastive loss L_{ins}
 - 7: compute cluster-level contrastive loss L_{clu}
 - 8: compute overall contrastive loss L_{all}
 - 9: update the network
 - 10: **end for**
 - 11: //test
 - 12: Extract features using the CNN model
 - 13: Use spectral clustering algorithm to obtain the clustering result
-

We utilize overall contrastive loss to guide the training process. The overall contrastive loss L_{all} consists of two parts: instance-level contrastive loss L_{ins} and cluster-level contrastive loss L_{clu} .

In this paper, the mini-batch size is N . After two types of image augmentations on each input image x_i , our proposed method works based on $2N$ samples $\{x_1^a, \dots, x_N^a, x_1^b, \dots, x_N^b\}$. For a specific sample x_i^a , there are a positive pair $\{x_i^a, x_i^b\}$ and $2N - 2$ negative pairs between this sample with the augmented visions of other input images. We can obtain $\{z_1^a, \dots, z_N^a, z_1^b, \dots, z_N^b\}$ using MLP I. The instance-level contrastive loss is calculated based on the cosine similarity of each pair. The similarity is computed by

$$D(z_i^{k_1}, z_j^{k_2}) = \frac{(z_i^{k_1})^\top (z_j^{k_2})}{\|z_i^{k_1}\| \|z_j^{k_2}\|}, \quad (1)$$

where $k_1, k_2 \in \{a, b\}$ and $i, j \in \{x \in \mathbb{N} : 1 \leq x \leq K\}$. The cluster-level contrastive loss L_{ins} is calculated using the following equations.

$$\ell_i^a = -\log \frac{\exp(D(z_i^a, z_i^b)/\tau_{II})}{\sum_{j=1}^M [\exp(D(z_i^a, z_j^a)/\tau_{II}) + \exp(D(z_i^a, z_j^b)/\tau_{II})]}, \quad (2)$$

$$L_{\text{ins}} = \frac{1}{2N} \sum_{i=1}^N (\ell_i^a + \ell_i^b), \quad (3)$$

where τ_I is the instance-level temperature parameter. ℓ_i^a is the loss for the sample x_i^a and ℓ_i^b is the loss for the sample x_i^b .

For cluster-level contrastive loss L_{clu} , we use the MLP II output $y^a \in \mathcal{R}^{N \times K}$, $y^b \in \mathcal{R}^{N \times K}$. a, b are the two types of image augmentations, N is the batch size, and K is the cluster number. y_i^a is the i th column of Y^a , which is the representation of cluster i under the data augmentation a . There is one positive pair $\{y_i^a, y_i^b\}$ and $2K - 2$ negative pairs. The cluster-level contrastive loss is calculated based on the cosine similarity of each pair. The similarity is computed by

$$D(y_i^{k_1}, y_j^{k_2}) = \frac{(y_i^{k_1})(y_j^{k_2})^\top}{\|y_i^{k_1}\| \|y_j^{k_2}\|}, \quad (4)$$

where $k_1, k_2 \in \{a, b\}$ and $i, j \in \{x \in \mathbb{N} : 1 \leq x \leq N\}$. The instance-level contrastive loss L_{ins} is calculated using the following equations.

$$\ell_i^a = -\log \frac{\exp(D(y_i^a, y_i^b)/\tau_I)}{\sum_{j=1}^N [\exp(D(y_i^a, y_j^a)/\tau_I) + \exp(D(y_i^a, y_j^b)/\tau_I)]}, \quad (5)$$

$$P(y_i^k) = \sum_{t=1}^N Y_{ti}^k / \|Y^k\|_1, k \in \{a, b\}, \quad (6)$$

$$H(Y) = -\sum_{i=1}^K [P(y_i^a) \log P(y_i^a) + P(y_i^b) \log P(y_i^b)], \quad (7)$$

$$L_{\text{clu}} = \frac{1}{2K} \sum_{i=1}^K (\ell_i^a + \ell_i^b) - H(Y), \quad (8)$$

where τ_{II} is the cluster-level temperature parameter. ℓ_i^a is the loss for the sample x_i^a and ℓ_i^b is the loss for the sample x_i^b . $H(Y)$ prevents most instances from being assigned to the same cluster.

The overall contrastive loss L_{all} is calculated using the following equation:

$$L_{\text{all}} = L_{\text{ins}} + L_{\text{clu}} \quad (9)$$

After training, we can use the model to extract features. Then, we use the spectral clustering algorithm to obtain the final clustering result. To the best of our knowledge, we are the first to propose a contrastive learning-based HSI clustering method. Moreover, we explore the HSI augmentation method that we apply to our proposed clustering method.

4. Experiments

4.1. Experimental Datasets

We conducted experiments using three real HSI datasets: Indian Pines, University of Pavia, and Salinas. For computational efficiency, we used three subsets of these datasets for experiments and analyses, as stated in Figure 5. The details of the three subsets are presented in Table 3. The false-color images were acquired by the Spectral python library using the default library.

Figure 5. (a–c) False-color images of the Indian Pines, University of Pavia, and Salinas data sets.

Table 3. Summary of the experimental subsets.

Datasets	Indian Pines	Pavia University	Salinas
Location	[30–115, 24–94]	[150–350, 100–200]	[0–140, 50–200]
Channels	200	103	204
Clusters	4	8	6
Samples	4391	6445	15,428

The Indian Pines image was acquired by the AVIRIS sensor over northwestern Indiana. The image has a size of $145 \times 145 \times 220$. Due to the water absorption effect, 20 bands were removed.

The University of Pavia dataset was collected by the ROSIS sensor over Pavia, northern Italy. The image has 610×340 pixels with 103 bands.

The Salinas dataset was gathered by the AVIRIS sensor over Salinas Valley, California. The image consists of 512×217 pixels. As with the Indian Pines scene, 20 water absorption bands were discarded. The remaining 204 bands are available for processing.

4.2. Evaluation Metrics

We used three metrics—overall accuracy (OA), average accuracy (AA), and kappa coefficient (KAPPA)—to evaluate the performances of all the experimental methods. These metrics vary in $[0,1]$. The higher the values are, the better the clustering result is.

4.3. Experimental Parameter

We performed all the experiments on a server with four Titan-RTX GPUs and a 125 G memory. Because our proposed method does not require much GPU memory, we only used one Titan-RTX GPU throughout the whole experiment. According to Table 1, the CNN model consumes 7.61 M GPU memory for an input patch. The model was implemented using the Pytorch framework. We used the PCA to reduce the raw data dimension to 100. The input size was $9 \times 9 \times 100$. We set the batch size as 128. The learning rate was set to 0.00003. We trained the CNN model for 15 epochs and chose the model with the least training loss for the test. The instance-level temperature parameter τ_I was 1. The cluster-level temperature parameter τ_{II} was 0.5. The spectral clustering algorithm was carried out using the scikit-learn python library. We only set the cluster number. Since the kmeans label assignment strategy is unstable, we set the label assignment strategy to discretize. The remaining parameters of the spectral clustering algorithm were the default ones.

4.4. Comparison Methods

To validate the effectiveness of our proposed method, we compared it with several clustering methods, including traditional clustering methods and state-of-the-art methods.

Traditional clustering methods are k-means [5], sparse subspace clustering (SSC) [13], elastic net subspace clustering (EnSC) [45], and sparse subspace clustering by orthogonal matching pursuit (SSC-OMP) [46]. The state-of-the-art methods include spectral–spatial sparse subspace clustering [31], spectral–spatial diffusion learning (DLSS) [12], Laplacian regularized deep subspace clustering (LRDSC) [24], and deep spatial–spectral subspace clustering network (DS³CNet) [25]. As far as we know, DS³CNet is the most recent method based on deep learning for HSI clustering. The results of SSC, S⁴C, DLSS, LRDSC, and DS³CNet were gained from the published literature [25]. The k-means clustering was conducted using the scikit-learning python library. We used the public code to implement the EnSC and SSC-OMP methods.

4.5. Result Analysis

4.5.1. Indian Pines

The clustering result gained for the Indian Pines dataset is shown in Table 4 and Figure 6. The spectral information of the Indian Pines dataset is stated in Figure 7. From the table and the figure, we can easily conclude that our proposed method achieved the highest clustering accuracy. Moreover, three deep-learning-based methods, LRDSC, DS³CNet, and our proposed method, performed much better than other traditional clustering methods. Furthermore, the spatial–spectral-based clustering methods, including S⁴C, DLSS, and the three deep-learning-based methods, achieved a higher accuracy than the spectral-only clustering methods. As can be seen from the table, our proposed method had an at least 15.72% accuracy increase for the Corn-notill class. From Figures 7 and 8, we found that the spectral characteristics of Corn-notill were similar to those of Soybean-mintill. Using our CNN model, it is much easier to cluster the features of Corn-notill and Soybean-mintill.

Figure 6. The clustering results achieved by different methods on the Indian Pines dataset.

Table 4. The clustering results of the Indian Pines dataset. The best results are highlighted in bold.

Class	Number	Methods								
		k-Means	SSC	EnSC	SSC-OMP	S ⁴ C	DLSS	LRDSC	DS ³ CNet	Proposed
Corn-notill	1005	0.4328	0.4935	0.7452	0.1034	0.6100	0.4418	0.5970	0.5184	0.9203
Grass-trees	730	0.9958	0.9958	0.6616	0.0000	1.0000	0.9763	0.8883	1.0000	0.9986
Soybean-notill	732	0.5737	0.6694	0.1489	0.0204	0.6530	0.4980	0.7031	0.9784	1.0000
Soybean-mintill	1924	0.6351	0.6410	0.4069	0.9968	0.6528	0.7508	0.7767	0.8933	0.9381
OA		0.6386	0.6701	0.4837	0.4639	0.7008	0.6736	0.7410	0.8388	0.9545
AA		0.6594	0.6999	0.4907	0.2802	0.7290	0.6667	0.7413	0.8475	0.9642
Kappa		0.4911	0.5988	0.2731	0.0593	0.5825	0.5833	0.6777	0.7989	0.9353

Figure 7. The spectral information of Indian Pines dataset. **Figure 8.** Visualization of data points of the Indian Pines dataset. Using t-SNE, we reduced the feature dimensionality to 2.

4.5.2. University of Pavia

The clustering result gained for the University of Pavia dataset is indicated in Table 5 and Figure 9. The spectral information of the University of Pavia dataset is stated in Figure 10. It can be seen that our proposed method obtained the highest clustering accuracy. Moreover, similar to the results of the Indian Pines dataset, three deep-learning-based methods—LRDSC, DS³CNet, and our proposed method—performed much better than the

other traditional clustering methods, while the spatial–spectral-based clustering methods (including S^4C , DLSS, and three deep-learning-based methods) achieved a higher accuracy than the spectral-only clustering methods. As can be seen from the table, in some areas our proposed method achieved a 100% accuracy for the University of Pavia dataset.

It should, however, be noted that for the University of Pavia dataset, our proposed method obtained a 0% accuracy for some classes. We think the reason for is that the pixel number was too low. In fact, trees, self-blocking bricks, and shadows were the three least numerous sample types. According to the Figures 10 and 11, the spectral characteristics of trees are very different to those of other sample types. Taking these two factors together, our proposed method only achieved an accuracy of 49.2% for trees, while the accuracy was 0% for self-blocking bricks and shadows.

Figure 9. The clustering results of different methods on the Pavia University dataset.

Table 5. The clustering results of the University of Pavia dataset. The best results are highlighted in bold.

Class	Number	Methods								
		k-Means	SSC	EnSC	SSC-OMP	S ⁴ C	DLSS	LRDSC	DS ³ CNet	Proposed
Asphalt	425	0.0000	0.9540	0.6541	0.1882	0.8730	0.6522	0.4658	1.0000	1.0000
Meadows	768	0.8476	0.0280	0.9062	0.3333	0.6064	0.9907	0.8785	0.0000	1.0000
Trees	63	0.0000	0.4853	0.7777	0.0317	0.9861	0.4559	0.0000	0.0000	0.4920
Painted metal sheet	1315	0.3680	0.9976	0.7171	0.7893	0.9909	0.0000	0.7784	0.9953	0.6410
Bare soil	2559	0.4060	0.3264	0.5291	0.4028	0.3193	0.7023	0.8942	0.9610	1.0000
Bitumen	860	0.9988	0.0000	0.4430	0.7104	0.0000	1.0000	0.4891	0.0024	0.9930
Self-Blocking Bricks	94	0.3510	0.6000	0.0000	0.1489	0.9837	0.7343	0.9940	1.0000	0.0000
Shadows	361	1.0000	1.0000	1.0000	0.2493	0.9909	0.5956	0.9363	0.5873	0.9944
OA		0.5317	0.5655	0.6303	0.4844	0.6509	0.6250	0.8117	0.8687	0.9060
AA		0.4964	0.5489	0.6284	0.3567	0.7188	0.6414	0.6795	0.5682	0.7650
Kappa		0.4449	0.5641	0.5590	0.3732	0.5852	0.6242	0.8111	0.8685	0.8784

Figure 10. The spectral information of the University of Pavia dataset.

Figure 11. Visualization of the data points of the University of Pavia dataset. Using t-SNE, we reduced the feature dimensionality to 2.

4.5.3. Salinas

The clustering result of the Salinas dataset is presented in Table 6 and Figure 12. The spectral information of the Salinas dataset is illustrated in Figure 13. Our proposed method obtained the highest clustering accuracy. This is different from the results of the Indian Pines dataset and the University of Pavia dataset, where many methods, including all spatial–spectral methods and one spectral-only method, SSC-OMP, achieved an OA higher than 80%. From Figures 13 and 14, we can see that the spectral characteristics of Fallow_rough_plow, Fallow_smooth, Stubble, and Celery are easy to cluster. However, the spectral characteristics of Grapes_untrained and Vineyard_untrained are very similar. Moreover, the pixels belonging to these two categories are distributed in the neighboring areas. All these methods used for comparison with our proposed method achieved a high accuracy for Grapes_untrained but a very low accuracy for Vineyard_untrained. Considering that the sample number of each class is quite close, we think that this phenomenon dramatically affects the overall accuracy.

Figure 12. The clustering results of different methods on the Salinas dataset.

Table 6. The clustering results achieved for the Salinas dataset. The best results are highlighted in bold.

Class	Number	Methods								
		k-Means	SSC	EnSC	SSC-OMP	S ⁴ C	DLSS	LRDSC	DS ³ CNet	Proposed
Fallow_rough_plow	1229	0.9910	0.3318	0.0000	0.9780	0.9959	0.9930	0.9558	0.9971	1.0000
Fallow_smooth	2441	0.9946	0.7461	0.2494	0.9631	0.9926	0.9935	0.9919	1.0000	0.9983
Stubble	3949	0.6920	0.6571	0.6505	0.8465	0.9977	0.9970	0.9997	1.0000	1.0000
Celery	3543	0.9937	1.0000	0.3211	0.9960	0.9984	0.9946	0.9804	1.0000	1.0000
Grapes_untrained	2198	0.9986	1.0000	0.8999	0.9126	1.0000	0.9969	0.9946	0.9843	0.6974
Vineyard_untrained	2068	0.0000	0.0000	0.0483	0.0415	0.0000	0.0000	0.0000	0.0879	1.0000
OA		0.7840	0.6481	0.4144	0.8113	0.8631	0.8564	0.8474	0.8698	0.9566
AA		0.7783	0.6225	0.3615	0.7896	0.8307	0.8292	0.8204	0.8449	0.9493
Kappa		0.7367	0.6438	0.2969	0.7682	0.8312	0.8562	0.8473	0.8696	0.9466

Figure 13. The spectral information of the Salinas dataset.

Figure 14. Visualization of the data points of the Salinas dataset. Using t-SNE, we reduced the feature dimensionality to 2.

From Figures 8, 11 and 14, we can see that the features show better clustering characteristics than the original data. After training, the CNN model can extract the features under unsupervised conditions efficiently. For example, in the Indian Pines image, Corn-notill, Soybean-notill, and Soybean-mintill are difficult to cluster, as these three kinds of samples have similar spectral characteristics. Using the CNN model to obtain the features, it can be seen that these three kinds of features are easier to cluster. For the University of Pavia dataset, meadows, bare soil, asphalt, and bitumen are easy to cluster; for the Salinas dataset, Grapes_untrained and Vinyard_untrained are easy to cluster. These samples are also easier to cluster when the CNN model is used to obtain the features.

5. Discussion

5.1. Influence of Patch Size

The input patch size is important for the 3D CNN for HSI classification. We set the input patch size to 7×7 , 9×9 , 11×11 , and 13×13 . The classification result is shown in Table 7. From the results, we can see that 9×9 is the best patch size for our proposed method.

Table 7. Accuracy with different input patch sizes. The best value in a row is bolded.

Dataset	Metric	7×7	9×9	11×11	13×13
Indian Pines	OA	0.6955	0.9545	0.6807	0.7335
	AA	0.7642	0.9642	0.7835	0.6481
	Kappa	0.5805	0.9353	0.5870	0.5961
University of Pavia	OA	0.8740	0.9060	0.7626	0.7845
	AA	0.7777	0.7650	0.6764	0.6978
	Kappa	0.8424	0.8784	0.7168	0.7301
Salinas	OA	0.9564	0.9566	0.9561	0.9542
	AA	0.9490	0.9493	0.9487	0.9466
	Kappa	0.9464	0.9466	0.9460	0.9436

5.2. Influence of Data Augmentation Methods

To find the best augmentation method for HSI clustering, we conducted several experiments. We used no flip, only selected discrete points, only selected random rectangular areas, and used rotation instead of flips and compared the performance. The results are presented in Table 8. From the results, we can see our proposed method did not achieve the best accuracy over the Indian Pines dataset and Salinas datasets. However, the differences are very small. Moreover, selecting only discrete points or rectangular areas can provide very different results in different datasets. These two methods are weakly robust.

Table 8. Accuracy obtained with different augmentation methods. The best value in a row is bolded.

Dataset	Metric	No Flip	Only Point	Only Rectangle	Rotation	Proposed
Indian Pines	OA	0.9549	0.6101	0.9679	0.9508	0.9545
	AA	0.9645	0.4810	0.9704	0.9622	0.9642
	Kappa	0.9359	0.3723	0.9541	0.9302	0.9353
University of Pavia	OA	0.8794	0.8808	0.8009	0.8836	0.9060
	AA	0.7794	0.7797	0.6687	0.7801	0.7650
	Kappa	0.8488	0.8505	0.7544	0.8539	0.8784
Salinas	OA	0.9567	0.9569	0.8503	0.9568	0.9566
	AA	0.9493	0.9496	0.7499	0.9494	0.9493
	Kappa	0.9467	0.9469	0.8147	0.9468	0.9466

5.3. Influence of Spectral Clustering

K-means and spectral clustering are two commonly used clustering methods. Here, we compare the performance of our proposed method based on spectral clustering and our method based on K-means clustering. The results are shown in Table 9. As shown in Table 9, our proposed method based on spectral clustering surpasses the performance of our method based on K-means clustering.

Table 9. Accuracy with K-means clustering and spectral clustering. The best results obtained for each dataset are bolded.

Metric	Indian Pines		University of Pavia		Salinas	
	K-Means	Spectral	K-Means	Spectral	K-Means	Spectral
OA	0.6809	0.9545	0.5600	0.9060	0.6803	0.9566
AA	0.7287	0.9642	0.5322	0.7650	0.6443	0.9493
Kappa	0.5654	0.9353	0.4887	0.8784	0.6187	0.9466

5.4. Running Time and Complexity

The running time of our proposed method is presented in Table 10. From the table, we can see that training the CNN model consumes most of the time. Since the input patch size for different datasets is the same, we believe that the computational complexity of training the model is $O(n)$. As for spectral clustering, the computational complexity is $O(n^3)$ [47], and the space complexity is $O(n^2)$ [48]. Because of the space complexity, we cannot conduct our proposed method on the complete hyperspectral images.

Table 10. The running time of our proposed method.

Time(s)	Indian Pines	University of Pavia	Salinas
Training CNN	74.53	99.08	235.45
Getting features	0.55	0.82	1.96
Spectral clustering	25.14	41.44	172.73
Total	102.22	141.34	410.14

6. Conclusions and Future Research

In this paper, we proposed a contrastive learning method for HSI clustering. The contrastive learning method extracts spatial-spectral information based on different augmented views of HSI. We removed the spectral information of some non-central pixels to augment the HSIs. Different methods of selecting the pixels to remove spectral information can be regarded as different augmentation methods. Based on the augmented views of samples, the CNN model was trained under supervision using instance-level and cluster-level contrastive loss. After training, the CNN model was used to extract features from input pixel blocks. Finally, according to the features, we conducted spectral clustering to obtain the clustering result. The experimental results achieved on three public datasets

confirmed the superiority of our proposed method. However, our proposed method also has some disadvantages. Because spectral clustering has the computational complexity of $O(n^3)$ and the space complexity of $O(n^2)$, it is not suitable for use on large datasets.

In the future, we will focus on HSI data augmentation. More augmentation methods for use on HSIs will be studied, such as rotation, GAN-based augmentation, and so on. We will also try to find a more effective method for selecting non-central pixels to remove the corresponding spectral information. Moreover, we will try to study our proposed method under more challenging conditions, such as luminosity, atmospheric conditions, spatial data sparsity, and noisy spectral data.

Author Contributions: X.H. and T.Z. implemented the algorithms, designed the experiments, and wrote the paper; X.H. performed the experiments; Y.P. and T.L. guided the research. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially supported by the National Key Research and Development Program of China (No. 2017YFB1301104 and 2017YFB1001900), the National Natural Science Foundation of China (No. 91648204 and 61803375), and the National Science and Technology Major Project.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets involved in this paper are all public datasets.

Acknowledgments: The authors acknowledge the State Key Laboratory of High-Performance Computing, College of Computer, National University of Defense Technology, China.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

HSI	Hyperspectral image;
SSC	Sparse subspace clustering;
CNN	Convolutional neural networks;
MLP	Multilayer perceptron.

References

1. Zhao, C.; Wang, Y.; Qi, B.; Wang, J. Global and local real-time anomaly detectors for hyperspectral remote sensing imagery. *Remote Sens.* **2015**, *7*, 3966–3985. [\[CrossRef\]](#)
2. Awad, M.; Jomaa, I.; Arab, F. Improved capability in stone pine forest mapping and management in Lebanon using hyperspectral CHRIS-Proba data relative to Landsat ETM+. *Photogramm. Eng. Remote Sens.* **2014**, *80*, 725–731. [\[CrossRef\]](#)
3. Ibrahim, A.; Franz, B.; Ahmad, Z.; Healy, R.; Knobelspiesse, K.; Gao, B.C.; Proctor, C.; Zhai, P.W. Atmospheric correction for hyperspectral ocean color retrieval with application to the Hyperspectral Imager for the Coastal Ocean (HICO). *Remote Sens. Environ.* **2018**, *204*, 60–75. [\[CrossRef\]](#)
4. Rodriguez, A.; Laio, A. Clustering by fast search and find of density peaks. *Science* **2014**, *344*, 1492–1496. [\[CrossRef\]](#)
5. Hartigan, J.A.; Wong, M.A. Algorithm AS 136: A k-means clustering algorithm. *J. R. Stat. Soc. Ser. (Appl. Stat.)* **1979**, *28*, 100–108. [\[CrossRef\]](#)
6. Maggioni, M.; Murphy, J.M. Learning by Unsupervised Nonlinear Diffusion. *J. Mach. Learn. Res.* **2019**, *20*, 1–56.
7. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD, Portland, OR, USA, 2–4 August 1996; Volume 96, pp. 226–231.
8. Roy, S.; Bhattacharyya, D.K. An approach to find embedded clusters using density based techniques. In Proceedings of the International Conference on Distributed Computing and Internet Technology, Bhubaneswar, India, 22–24 December 2005; pp. 523–535.
9. Cariou, C.; Le Moan, S.; Chehdi, K. Improving k-nearest neighbor approaches for density-based pixel clustering in hyperspectral remote sensing images. *Remote Sens.* **2020**, *12*, 3745. [\[CrossRef\]](#)
10. Chen, S.; Zhang, D. Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2004**, *34*, 1907–1916. [\[CrossRef\]](#) [\[PubMed\]](#)
11. Lin, J.; He, C.; Wang, Z.J.; Li, S. Structure preserving transfer learning for unsupervised hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1656–1660. [\[CrossRef\]](#)

12. Murphy, J.M.; Maggioni, M. Unsupervised clustering and active learning of hyperspectral images with nonlinear diffusion. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 1829–1845. [\[CrossRef\]](#)
13. Elhamifar, E.; Vidal, R. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2765–2781. [\[CrossRef\]](#)
14. Liu, Y.; Dou, Y.; Jin, R.; Li, R.; Qiao, P. Hierarchical learning with backtracking algorithm based on the visual confusion label tree for large-scale image classification. *Vis. Comput.* **2021**, 1–21. [\[CrossRef\]](#)
15. Liu, Y.; Dou, Y.; Jin, R.; Qiao, P. Visual tree convolutional neural network in image classification. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 758–763.
16. Nagpal, C.; Dubey, S.R. A performance evaluation of convolutional neural networks for face anti spoofing. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.
17. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [\[CrossRef\]](#)
18. Zhong, Z.; Li, J.; Luo, Z.; Chapman, M. Spectral-spatial residual network for hyperspectral image classification: A 3-D deep learning framework. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 847–858. [\[CrossRef\]](#)
19. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
20. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
21. Wang, W.; Dou, S.; Jiang, Z.; Sun, L. A fast dense spectral-spatial convolution network framework for hyperspectral images classification. *Remote Sens.* **2018**, *10*, 1068. [\[CrossRef\]](#)
22. Ma, W.; Yang, Q.; Wu, Y.; Zhao, W.; Zhang, X. Double-branch multi-attention mechanism network for hyperspectral image classification. *Remote Sens.* **2019**, *11*, 1307. [\[CrossRef\]](#)
23. Li, R.; Zheng, S.; Duan, C.; Yang, Y.; Wang, X. Classification of hyperspectral image based on double-branch dual-attention mechanism network. *Remote Sens.* **2020**, *12*, 582. [\[CrossRef\]](#)
24. Zeng, M.; Cai, Y.; Liu, X.; Cai, Z.; Li, X. Spectral-spatial clustering of hyperspectral image based on Laplacian regularized deep subspace clustering. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 2694–2697.
25. Lei, J.; Li, X.; Peng, B.; Fang, L.; Ling, N.; Huang, Q. Deep spatial-spectral subspace clustering for hyperspectral image. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *31*, 2686–2697. [\[CrossRef\]](#)
26. Li, Y.; Hu, P.; Liu, Z.; Peng, D.; Zhou, J.T.; Peng, X. Contrastive clustering. In Proceedings of the 2021 AAAI Conference on Artificial Intelligence (AAAI), Vancouver, BC, Canada, 2–9 February 2021.
27. Paoli, A.; Melgani, F.; Pasolli, E. Clustering of hyperspectral images based on multiobjective particle swarm optimization. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 4175–4188. [\[CrossRef\]](#)
28. Zhong, Y.; Zhang, L.; Gong, W. Unsupervised remote sensing image classification using an artificial immune network. *Int. J. Remote Sens.* **2011**, *32*, 5461–5483. [\[CrossRef\]](#)
29. Zhai, H.; Zhang, H.; Zhang, L.; Li, P. Laplacian-regularized low-rank subspace clustering for hyperspectral image band selection. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 1723–1740. [\[CrossRef\]](#)
30. Tian, L.; Du, Q.; Kopriva, I.; Younan, N. Spatial-spectral Based Multi-view Low-rank Sparse Subspace Clustering for Hyperspectral Imagery. In Proceedings of the IGARSS 2018—2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 8488–8491.
31. Zhang, H.; Zhai, H.; Zhang, L.; Li, P. Spectral-spatial sparse subspace clustering for hyperspectral remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 3672–3684. [\[CrossRef\]](#)
32. Xie, J.; Girshick, R.; Farhadi, A. Unsupervised deep embedding for clustering analysis. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 9–14 June 2016; pp. 478–487.
33. Chang, J.; Wang, L.; Meng, G.; Xiang, S.; Pan, C. Deep adaptive image clustering. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5879–5887.
34. Fard, M.M.; Thonet, T.; Gaussier, E. Deep k-means: Jointly clustering with k-means and learning representations. *Pattern Recognit. Lett.* **2020**, *138*, 185–192. [\[CrossRef\]](#)
35. Barthakur, M.; Sarma, K.K. Semantic Segmentation using K-means Clustering and Deep Learning in Satellite Image. In Proceedings of the 2019 2nd International Conference on Innovations in Electronics, Signal Processing and Communication (IESC), Shillong, India, 1–2 March 2019; pp. 192–196.
36. Sodjinou, S.G.; Mohammadi, V.; Mahama, A.T.S.; Gouton, P. A deep semantic segmentation-based algorithm to segment crops and weeds in agronomic color images. *Inf. Process. Agric.* **2021**. [\[CrossRef\]](#)
37. Ji, P.; Zhang, T.; Li, H.; Salzmann, M.; Reid, I. Deep subspace clustering networks. *arXiv* **2017**, arXiv:1709.02508.
38. Mukherjee, S.; Asnani, H.; Lin, E.; Kannan, S. Clustergan: Latent space clustering in generative adversarial networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 4610–4617.

39. Chen, X.; Duan, Y.; Houthoofd, R.; Schulman, J.; Sutskever, I.; Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 2180–2188.
40. Egaña, Á.F.; Santibáñez-Leal, F.A.; Vidal, C.; Díaz, G.; Liberman, S.; Ehrenfeld, A. A Robust Stochastic Approach to Mineral Hyperspectral Analysis for Geometallurgy. *Minerals* **2020**, *10*, 1139. [[CrossRef](#)]
41. Xu, J.; Li, H.; Liu, P.; Xiao, L. A novel hyperspectral image clustering method with context-aware unsupervised discriminative extreme learning machine. *IEEE Access* **2018**, *6*, 16176–16188. [[CrossRef](#)]
42. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning, Online, 13–18 July 2020; pp. 1597–1607.
43. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 9729–9738.
44. Grill, J.B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.H.; Buchatskaya, E.; Doersch, C.; Pires, B.A.; Guo, Z.D.; Azar, M.G.; et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv* **2020**, arXiv:2006.07733.
45. You, C.; Li, C.G.; Robinson, D.P.; Vidal, R. Oracle based active set algorithm for scalable elastic net subspace clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3928–3937.
46. You, C.; Robinson, D.; Vidal, R. Scalable sparse subspace clustering by orthogonal matching pursuit. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3918–3927.
47. Yan, D.; Huang, L.; Jordan, M.I. Fast approximate spectral clustering. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 28 June–1 July 2009; pp. 907–916.
48. Mall, R.; Langone, R.; Suykens, J.A. Kernel Spectral Clustering for Big Data Networks. *Entropy* **2013**, *15*, 1567–1586. [[CrossRef](#)]