remote sensing

MDPI

Article

Supervised Contrastive Learning-Based Classification for Hyperspectral Image

Lingbo Huang [1], Yushi Chen [1,*], Xin He [1] and Pedram Ghamisi [2,3]

[1] School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China
[2] The Institute of Advanced Research in Artificial Intelligence (IARAI), 1030 Vienna, Austria
[3] Helmholtz-Zentrum Dresden-Rossendorf, Machine Learning Group, Helmholtz Institute Freiberg for Resource Technology, Chemnitzer Str. 40, 09599 Freiberg, Germany
* Correspondence: chenyushi@hit.edu.cn

Abstract: Recently, deep learning methods, especially convolutional neural networks (CNNs), have achieved good performance for hyperspectral image (HSI) classification. However, due to limited training samples of HSIs and the high volume of trainable parameters in deep models, training deep CNN-based models is still a challenge. To address this issue, this study investigates contrastive learning (CL) as a pre-training strategy for HSI classification. Specifically, a supervised contrastive learning (SCL) framework, which pre-trains a feature encoder using an arbitrary number of positive and negative samples in a pair-wise optimization perspective, is proposed. Additionally, three techniques for better generalization in the case of limited training samples are explored in the proposed SCL framework. First, a spatial–spectral HSI data augmentation method, which is composed of multiscale and 3D random occlusion, is designed to generate diverse views for each HSI sample. Second, the features of the augmented views are stored in a queue during training, which enriches the positives and negatives in a mini-batch and thus leads to better convergence. Third, a multi-level similarity regularization method (MSR) combined with SCL (SCL–MSR) is proposed to regularize the similarities of the data pairs. After pre-training, a fully connected layer is combined with the pre-trained encoder to form a new network, which is then fine-tuned for final classification. The proposed methods (SCL and SCL–MSR) are evaluated on four widely used hyperspectral datasets: Indian Pines, Pavia University, Houston, and Chikusei. The experiment results show that the proposed SCL-based methods provide competitive classification accuracy compared to the state-of-the-art methods.

Keywords: contrastive learning; convolutional neural network (CNN); hyperspectral image (HSI); supervised classification

check for updates

Citation: Huang, L.; Chen, Y.; He, X.; Ghamisi, P. Supervised Contrastive Learning-Based Classification for Hyperspectral Image. *Remote Sens.* **2022**, *14*, 5530. https://doi.org/10.3390/rs14215530

Academic Editors: Junjun Jiang and Jiayi Ma

Received: 22 September 2022
Accepted: 25 October 2022
Published: 2 November 2022

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

Copyright: © 2022 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 sensors obtain the spectral and spatial information of observed targets simultaneously. The abundant information obtained from the observed target makes the hyperspectral image (HSI) useful for various applications [1]. Many techniques have been proposed for the proper processing of HSI data [2].

Classification is one of the fundamental tasks in HSI processing. Given a set of training samples consisting of a set of images and corresponding labels, the purpose of the classification is to train a model that will be able to assign a proper label to an unknown image. As a pixel-based classification task [3,4], the HSI classification task aims to categorize the content of each pixel in the scene [5].

HSI classification is the basis of many applications, such as land usage, agriculture recognition, mineralogy, surveillance, healthcare, and environmental sciences [6,7]. Due to its importance, HSI classification has been widely investigated, and a wide diversity of HSI classification methods have been proposed in the past three decades [8–10] and have mainly focused on spectral or spatial–spectral information. For spectral information-based classification, the study in [11] designed a vegetation index for HSI classification

which was based on a short wavelength infrared spectrum. Spectral unmixing techniques have also been explored for HSI classification [12]. For example, an enhanced bilinear mixing model was proposed in [13] for subpixel classification and gained competitive performance. Besides spectral information, the HSI obtains abundant spatial information with the development of imaging technology. In order to fully explore the spatial features of the HSI, many approaches have been proposed, such as extended MPs (EMPs) [14], the extended multi-attribute profile (EMAP) [15], and extinction profiles (EPs) [16].

Due to their powerful capability to extract discriminant and robust features, deep learning models have been widely investigated in many fields, including classification and regression tasks that involve image [17,18], language [19], and speech [20]. Recently, deep learning-based methods have been used for HSI feature extraction and classification. Many deep models, such as the stacked auto-encoder [21], the deep believe network [22], and the convolutional neural network (CNN), have been used for HSI classification [23,24]. Among the many deep learning models, the CNN-based methods have achieved state-of-the-art performance. In recent years, the research interests of CNN-based HSI classification methods have focused mainly on modifying CNNs [25,26] and combining CNNs with existing machine learning frameworks [27,28].

Recently, many works have improved the architecture of the CNN so that it can adapt to the characteristics of the HSI and perform better for HSI classification. In [29], the typical residual unit architecture was combined with a pyramidal structure to enhance the performance of the CNN. The authors in [30] proposed an end-to-end spectral–spatial 3D residual CNN to explore the spatial–spectral features of the HSI. The attention mechanism is another popular way to improve the performance of CNNs. In [31], a spatial–spectral dense CNN framework with a feedback attention mechanism was presented. In [32], a double-branch dual-attention mechanism was used to refine and optimize the extracted feature maps for the benefit of both channel and spatial attention.

Other machine learning or image processing techniques can be combined with a CNN for a better HSI classification performance, such as transfer learning [33], ensemble learning [34], and other spatial feature extraction methods [35]. An additional RGB dataset was utilized to pre-train 3D lightweight CNNs based on transfer learning [36]. In [37] and [38], morphological profiles followed by a CNN are used to fully extract the spatial features of the HSI. In addition, a transformer was explored together with a CNN to extract spectral–spatial features for HSI classification [39].

The above approaches obtained a superior performance if there was a sufficient number of labeled samples. However, it is costly and time-consuming to obtain high-quality labeled samples for HSIs. The limited number of training samples usually results in the problem of overfitting for deep learning-based classification methods and thus hinders researchers from improving classification accuracy [40]. To alleviate the problem, many deep learning-based methods for HSI classification have been explored [41], most of which can be summarized in the following three strategies: (1) Methods aided by the use of unlabeled samples. This strategy extracts useful information from the abundant unlabeled samples of HSIs to benefit the deep model's training. Semi-supervised learning is a typical and promising framework for this strategy. For example, self-training [42] and co-training [43] have been explored for semi-supervised classification of HSIs. In addition, the generative adversarial network (GAN) has also been widely used for HSI classification in a semi-supervised manner [44]. (2) Methods with data combination. This strategy enlarges the number of inputs through the combination of training samples. For instance, pairs of pixels were built in [45] to enlarge the number of training inputs. In [46], the Siamese neural network, whose input is a pair of training samples, was explored to extract more discriminative features. (3) Methods with data augmentation. This strategy increases the amount of data by slightly modifying the existing data or creating synthetic data based on some rules. For example, Haut et al. [47] randomly occluded the pixels of different rectangular spatial regions in the HSI to generate training images, reducing the risk of overfitting for deep models.

Based on the above analysis, supervised contrastive learning (SCL), which is an extension of self-supervised contrastive learning [48–50] for supervised setting, is explored in this paper for HSI classification with limited training samples. The proposed SCL-based methods measure the similarities between different sample pairs and then use the designed supervised contrastive loss to increase similarities of positive pairs belonging to the same class in the latent space, while decrease similarities are for distinct classes. Through pairwise comparison, the learnt features can be more discriminative. Obviously, both data augmentation and data combination are adopted in SCL, which provides a strong ability to deal with the overfitting problem in HSI classification with limited training samples.

Specifically, on one hand, the proposed SCL designs a data augmentation method composed of multiscale and 3D random occlusion to increase the amount of training samples. Traditional augmentation methods are usually unilateral, only focusing on the scale, the spectral, or the spatial information of the HSIs. The designed data augmentation method in this paper not only learns more complex structural information via different input scales, it also encourages the CNN to utilize spatial–spectral information from the entire HSI, rather than relying on a small subset of spatial or spectral features. Therefore, more diverse training inputs can be generated when compared with other augmentation methods.

On the other hand, the SCL pairs training samples to enlarge the number of training inputs and pre-trains a feature encoder by optimizing the intra- and inter-class similarities using supervised contrastive loss. The low diversity of input pairs in a training mini-batch usually limits the classification performance in traditional data combination-based methods. Therefore, a queue is deigned in the proposed SCL to store the features of the augmented samples, which further enriches the positives and negatives in a mini-batch and thus improves the classification accuracy. In addition, a momentum-based moving average strategy is utilized in order to prevent the training procedure from fitting train samples quickly, which may easily lead to overfitting problem. Furthermore, a regularization method is also designed for SCL to prevent the positive/negative pairs from being too close with/far away from each other on the training set, which may cause the overfitting problem.

The contributions of this paper are listed as follows:

(1) A supervised contrastive learning method for HSI classification is designed. In SCL, the labeled data are paired to pre-train a CNN-based feature encoder by the proposed supervised contrastive loss. To increase the diversity of the data pairs in a mini-batch and thus benefit the training procedure, SCL maintains a label queue and a feature queue, which are updated by a momentum-based moving average encoder.
(2) A data augmentation method composed of multiscale and 3D random occlusion is proposed for HSI supervised contrastive learning. Multiscale augmentation randomly generates input samples with different window sizes, resulting in more complex spatial and structural information. Three-dimensional random occlusion disturbs the spatial–spectral content of the input, leading to better generalization by the model. The combination of these two data augmentation methods helps to improve the classification accuracy.
(3) A regularization method, MSR, is combined with SCL (SCL–MSR) to further improve the generalization performance and classification accuracy of the supervised contrastive learning network.

The rest of the paper is organized as follows. Section 2 presents the proposed supervised contrastive learning framework for HSI classification. Section 3 describes the data classification results and the analysis of the comprehensive experiments. In Section 4, the paper's conclusion is briefly summarized.

2. Methodology

Motivated by recent works on contrastive learning algorithms in computer vision and hyperspectral image classification, this paper designs a CL-based supervised pre-training framework for the HSI that learns representations by maximizing agreement between augmented views of samples belonging to the same class (positive pairs) in the latent space,

while maximizing the difference between two distinct classes (negative pairs). This can be achieved via the proposed supervised contrastive loss. As illustrated in Figure 1, the pre-training framework comprises the following five major components.

(1) A stochastic data augmentation module that generates two correlated views of any given HSI training sample $(\mathbf{x}, y)$:

$$\mathbf{x}_q, \mathbf{x}_k = Aug(\mathbf{x}), \tag{1}$$

where $Aug(\cdot)$ represents the augmentation mapping function.

(2) A CNN-based encoder f_q that maps augmented training input $\mathbf{x}_q$ to a representation feature vector $\mathbf{q}$:

$$\mathbf{q} = f_q(\mathbf{x}_q). \tag{2}$$

(3) A momentum-based moving average encoder f_k that shares the same architecture with f_q. Its weights are progressively updated from f_q, and the representation feature is obtained by:

$$\mathbf{k} = f_k(\mathbf{x}_k). \tag{3}$$

(4) A feature queue $\mathbf{K} = \{\mathbf{k}_i\}_{i=1}^{H}$ and its corresponding label queue $\mathbf{Y} = \{y_i\}_{i=1}^{H}$, where H denotes the length. The feature queue is used to restore the HSI features generated by f_k over the past few epochs, aiming to increase the positive and negative pairs in the training process.

(5) A contrastive learning-based loss function defined for the supervised pre-training task.

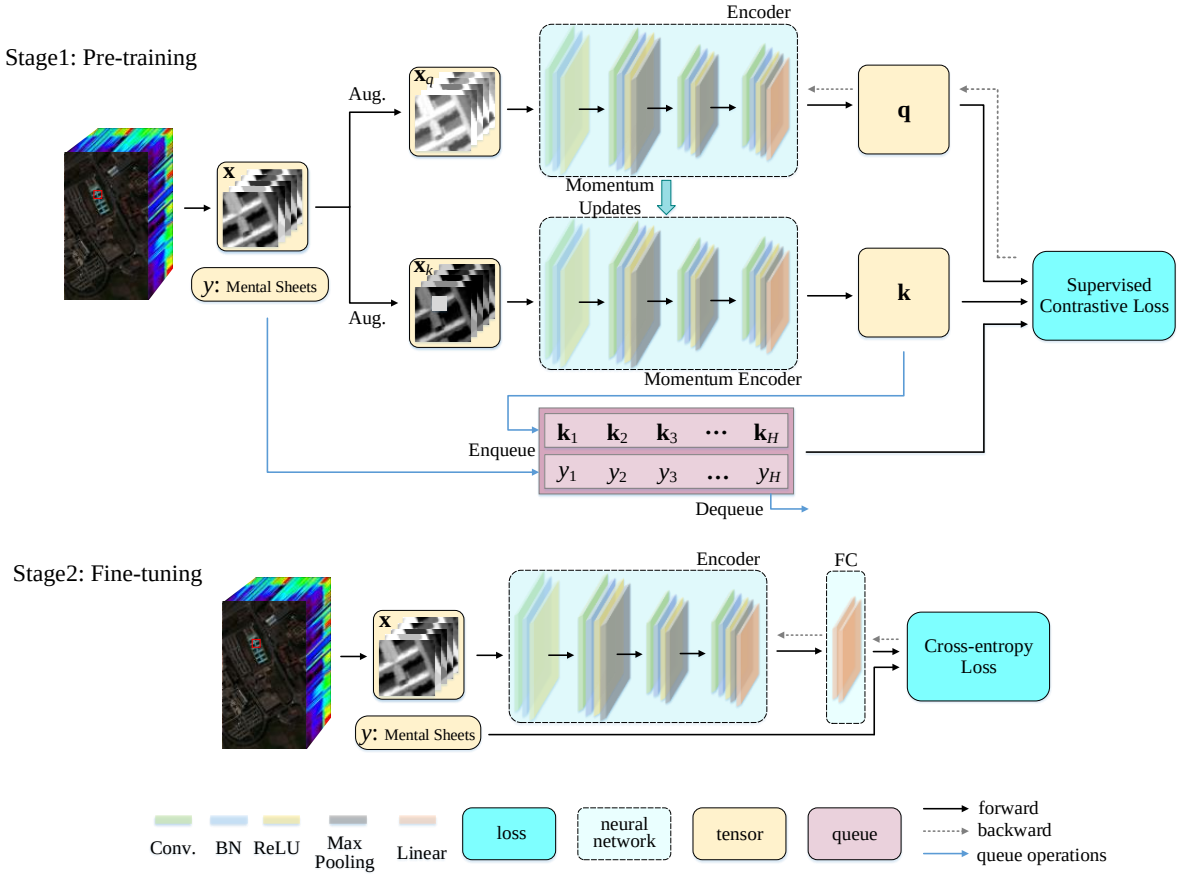


Figure 1. The proposed HSI classification method overview—training flow. For illustrative purposes, a single image flow instead of a batch is shown here. Stage 1: pre-training based on supervised contrastive learning. Stage 2: fine-tuning for the final classification task.

The proposed methods will be described in detail in the following subsections.

2.1. Data Augmentation

Data augmentation is mainly used to create more training samples and improve the generalization performance of the deep model for HSI classification. From the perspective of scale and the spatial–spectral content for the HSI samples, the two data augmentation methods, multiscale and 3D random occlusion, are introduced, respectively.

(1) Multiscale augmentation (MA): When using CNN to deal with HSI classification, it is often necessary to construct input samples by using neighborhood windows. Hoping that the CNN can learn more complex spatial structural information via different window sizes, multiscale augmentation is used here. Specifically, for a given HSI pixel, the MA randomly selects a spatial size (e.g., 23 × 23) from several candidate sizes (e.g., 27 × 27, 25 × 25, 23 × 23) and forms the corresponding cube. Then, the HSI cube is resized to the desired spatial size (e.g., 27 × 27) for the CNN's input, using bilinear interpolation. Figure 2 illustrates the procedure of MA.

(2) Three-dimensional random occlusion augmentation (ROA): In remote sensing, data occlusion usually occurs when some areas of Earth's surface are not visible from the remote sensor due to an external factor (e.g., clouds). Motivated by the previous work in [47], the 3D ROA is designed for data augmentation that also takes the spectral bands of the HSI into account.

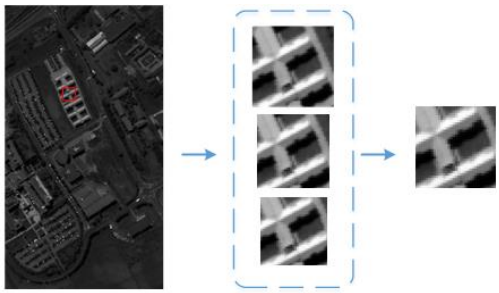
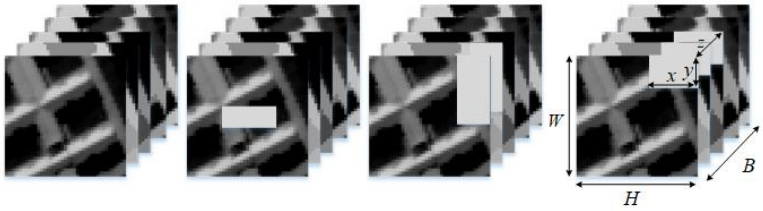


Figure 2. Multiscale data augmentation. Red box represents the region of an HSI sample.

The 3D ROA randomly selects a cuboid region $\mathbf{x}_e \in \mathbb{R}^{x \times y \times z}$ from a given input HSI cube $\mathbf{x} \in \mathbb{R}^{W \times W \times B}$ and covers its pixels with a random value. To obtain $\mathbf{x}_e$, its volume (denoted by V_e, which has a value which is between a minimum and a maximum threshold) is first calculated: $V_e = rand(v_{min} \cdot V, v_{max} \cdot V)$, where V represents the volume of $\mathbf{x}$. The next step is to obtain x, y, and z as follows:

$$x = \sqrt[3]{\frac{V_e l_e}{{r_e}^2}}, y = \sqrt[3]{V_e l_e r_e}, z = \sqrt[3]{\frac{V_e r_e}{{l_e}^2}}, \tag{4}$$

where r_e and l_e are also randomly selected values between the minimum and maximum threshold values, $l_e = rand(l_{min}, l_{max})$, $r_e = rand(r_{min}, r_{max})$, which control the shape of the cuboid $\mathbf{x}_e$. Finally, the location of $\mathbf{x}_e$ is randomly selected, and the cuboid is then filled with a predetermined value, e.g., 0.5, for simplicity. Figure 3 illustrates some examples of 3D ROA.



(**a**) (**b**) (**c**) (**d**)

Figure 3. Examples of 3D random occlusion augmentation: (**a**) original (not occluded) inputs, (**b**–**d**) occluded inputs with occluded zones shown in grey.

Algorithm 1 demonstrates the procedure of the data augmentation methods in this paper.

Algorithm 1. Pseudocode of data augmentation for HSI.

Input: HSI cube $\mathbf{x} \in \mathbb{R}^{W \times W \times B}$, minimum scale S, occlusion probability p, occlusion shape parameters $v_{min}, v_{max}, l_{min}, l_{max}, r_{min}, l_{max}$
Output: augmented image $\mathbf{x}_a$
$s = RandSelect(W, W-2, \cdots, S)$
$\mathbf{x}_c = CenterCrop(\mathbf{x}, [s, s, B])$
$\mathbf{x}_a = Resize(\mathbf{x}_c, [W,\ W,\ B])$
$p_1 = Rand(0, 1)$
if $p_1 > p$ **then**
  return $\mathbf{x}_a$
else
  $V = W \times W \times B$
  $V_e = Rand(v_{min} \cdot V, v_{max} \cdot V)$
  $l_e = Rand(l_{min},\ l_{max})$
  $r_e = Rand(r_{min},\ r_{max})$
  Get x, y, and z using Equation (4)
  $x_1 = Randint(0, W - x)$
  $y_1 = Randint(0, W - y)$
  $z_1 = Randint(0,\ B - z)$
  $\mathbf{x}_a[x_1 : x_1 + x, y_1 : y_1 + y, z_1 : z_1 + z] = 0.5$
return $\mathbf{x}_a$
RandSelect: select one element randomly; *CenterCrop*: crop the given image at the center; *Resize*: resize the given image to the desired size; *Rand*: select random float numbers from given range, following the "uniform distribution"; *Randint*: select random integers from given range, following the "discrete uniform distribution".

2.2. Supervised Contrastive Learning for HSI Classification

This section introduces the proposed algorithm in detail. As illustrated by Figure 2, the training process for HSI classification entails two stages: pre-training and fine-tuning.

In the first stage, the CNN-based encoder is pre-trained using labeled training samples in a contrastive manner. The pre-training process comprises the following steps.

First, the data augmentation is performed on input HSI cube $\mathbf{x}$ to obtain two correlated views, $\mathbf{x}_q$ and $\mathbf{x}_k$. Second, the feature vectors $\mathbf{q}$ and $\mathbf{k}$ are computed from the two augmented views via the CNN-based encoder f_q and the momentum encoder f_k, respectively. The positive pairs are then obtained from $\mathbf{q}$, $\mathbf{k}$, and the features in the queue that share the same label with $\mathbf{q}$. The negative pairs are obtained from $\mathbf{q}$ and the features in the queue whose labels are different from $\mathbf{q}$. Meanwhile, $\mathbf{k}$ and its corresponding label y are enqueued to the queue. Next, the positive and negative pairs are fed to the supervised contrastive loss for updating the encoder's weights. Finally, the weights of momentum encoder f_k are progressively updated from f_q.

In the second stage, a fully connected layer is added following the pre-trained encoder to form a new network for HSI classification. This new network will then be fine-tuned by using cross-entropy loss to accomplish the final classification task.

Three important components of SCL, queue, momentum update, and supervised contrastive loss, are described in detail as follows.

Queue details: Due to the limited number of training samples, a queue scheme composed of a feature queue and a label queue is adopted for HSI classification. As mentioned above, the feature queue is used to restore the features generated in the past few epochs and can decouple the number of positive and negative pairs from the mini-batch size. Therefore, the queue size K can be treated as a hyperparameter and set to be much larger than the mini-batch size to form more feature pairs in the current training epoch. In addition, a label queue is maintained and updated along with the feature queue. This label queue contains the corresponding labels for the features in the queue.

The queue adopts the strategy of "first-in, first-out". Specifically, the current mini-batch is enqueued to the queue, and the oldest mini-batch, which is the least consistent with the newest HSI samples, is removed from the queue. In this way, the queue always represents a sampled subset of all training samples.

A positive/negative pair is the pair of the two samples' features which belong to the same class/two different classes. For a given HSI $\mathbf{x}$ and its corresponding representation feature $\mathbf{q}$, we compare its label with the label queue and obtain positive pairs labeled 1 if $\mathbf{q}$ has the same class label as the features in the queue; otherwise, the negative pairs labeled 0 are obtained. This procedure can be formulated as:

$$l(\mathbf{q}, \mathbf{k}_i) = \begin{cases} 1, & y = y_i \\ 0, & y \neq y_i \end{cases}, \tag{5}$$

where $l(\cdot)$ denotes the label function for the pairs.

As is shown in Figure 4, the label of current input image is *mental sheets*, and there are features of the samples corresponding to the different classes in the feature queue. These features and labels in the previous epochs are stored. Then, we compare the label *mental sheets* with the label queue one by one. If they are different (i.e., *Meadows lies first in the queue*), a negative pair of corresponding features is generated. Using the queue module, lots of positive and negative pairs can be generated for training in the current epoch.

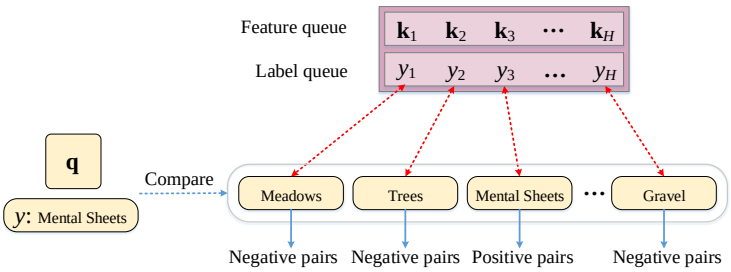


Figure 4. The generation process of pairs in SCL.

Momentum update details: The feature queue contains both the current mini-batch features and the older features, and thus, the gradient cannot propagate to all the features in the queue. Therefore, updating the weights in the encoder related to the queue needs to be considered. An intuitive idea is for the queue encoder f_k to share the same weights with the other encoder f_q, ignoring the gradient. By doing this, the oldest mini-batch of features may be very different from the newest ones. Even the features belonging to the same image may be dissimilar enough due to the rapidly changing encoder, which may lead to poor generalization. Therefore, a momentum-based moving average strategy is utilized to address this issue.

Let θ_k denote the weights of f_k, and θ_q denote the weights of f_q. θ_k is updated by:

$$\theta_k = m\theta_k + (1-m)\theta_q, \tag{6}$$

where $m \in [0,1]$ is a momentum coefficient. This coefficient is used to control the update speed of the queue encoder. During training, only the parameters θ_q are updated by back-propagation. From Equation (6), it can be seen that θ_k is the moving average of θ_q. The larger the value of m, the more slowly the weights update. As a result, though the features in the queue are generated by f_q in different training epochs, the differences among these features will be less.

Supervised contrastive loss details: Let $sim(\mathbf{a},\mathbf{b}) = \mathbf{a}^T\mathbf{b}/(\|\mathbf{a}\|\|\mathbf{b}\|)$ represent the inner product between $\updownarrow_2$ normalized $\mathbf{a}$ and $\mathbf{b}$. Then, the supervised contrastive loss function is defined as

$$\mathcal{L}^{SCL} = -\log \frac{\sum_{i=1}^{H} l_i e^{s_i/T}}{\sum_{i=1}^{H} l_i e^{s_i/T} + \sum_{i=1}^{H}(1-l_i)e^{s_i/T}}, \tag{7}$$

where $l_i = l(\mathbf{q}, \mathbf{k}_i)$ and $s_i = sim(\mathbf{q}, \mathbf{k}_i)$. T is a temperature parameter, playing a role in controlling the strength of the penalties on the hard negative samples. The term in the numerator represents the sum of the positive pairs' similarity scores, while the second term in the denominator represents the sum of the negative pairs' similarity scores. Equation (7) can be simplified as

$$\mathcal{L}^{SCL} = \log\left(1 + \sum_{i=1}^{H}(1-l_i)e^{\frac{s_i}{T}} \cdot \sum_{i=1}^{H} l_i e^{-\frac{s_i}{T}}\right). \tag{8}$$

From Equation (8), it can be seen that the supervised contrastive loss seeks to reduce the negative scores and increase the positive scores.

Considering the feature $\mathbf{k}$ generated by the queue encoder in the current epoch, Equation (8) can be modified as

$$\mathcal{L}^{SCL} = \log\left(1 + \sum_{i=0}^{H}(1-l_i)e^{\frac{s_i}{T}} \cdot \sum_{i=0}^{H} l_i e^{-\frac{s_i}{T}}\right), \tag{9}$$

where we define $\mathbf{k}_0 = \mathbf{k}$ and $l_0 = l(\mathbf{q}, \mathbf{k}_0) = 1$. In this way, positive pairs are obtained in two ways: (1) as features of the augmented images from the same image ($\mathbf{q}$, $\mathbf{k}$) and (2) as features in the queue $\{\mathbf{k}_i\}_{i=1}^{H}$ with the same label as the current feature $\mathbf{q}$.

2.3. Multi-Level Similarity Regularization for SCL

Although data augmentation and data combination are performed for HSI, there is still the risk of overfitting in SCL; that is, positive pairs can be pulled too closely together, while the negative pairs can be pushed too far apart in the feature space. In order to further alleviate the overfitting problem caused by limited training samples for HSI classification, multi-level similarity regularization (MSR) is introduced here.

The MSR predefines a set of levels and forces the similarities of the sample pairs to move towards the levels. As illustrated by Figure 5, in the SCL's training procedure, the supervised pre-training contrastive loss pulls the features belonging to the same class towards each other and pushes different classes' features away in the embedding space. Apart from the push/pull effect caused by SCL, the MSR also forces the similarities to align with a set of predefined levels (denoted by dashed lines in Figure 5), preventing the positive/negative pairs from being too close to/far away from each other.

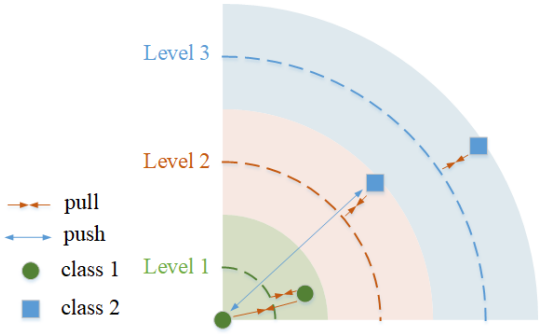


Figure 5. Multi-level similarity regularization for SCL model.

Let $\mathbf{L} = \{L_n\}_{n=1}^{A}$ denote a set of pre-defined similarity levels. The function $r(s,\ L_i, \mathbf{L})$ is an assignment function that indicates whether the given similarity s is the closest to the given level L_i; it is defined as

$$r(s,\ L_i, \mathbf{L}) = \begin{cases} 1, & if \underset{L_n \in \mathbf{L}}{\operatorname{argmin}} |s - L_n| \text{ is } L_i, \\ 0, & otherwise. \end{cases} \tag{10}$$

MSR minimizes the difference between a given pairwise similarity and the corresponding closest level, which can be achieved by minimizing the following loss:

$$\mathcal{L}^{MSR} = \frac{1}{H+1} \sum_{i=0}^{H} \sum_{m=1}^{N} r(s_i,\ L_m, \mathbf{L}) \cdot |s_i - L_m|. \tag{11}$$

Note that the levels are initialized with given values, while they can be updated to optimally regularize the pairwise similarity during the training process.

When using MSR, the total loss function $\mathcal{L}^{SCL-MSR}$ is defined as the sum of $\mathcal{L}^{CLSP}$ and $\mathcal{L}^{MSR}$:

$$\mathcal{L}^{SCL-MSR} = \mathcal{L}^{SCL} + \mathcal{L}^{MSR}. \tag{12}$$

Algorithm 2 demonstrates the pre-training procedure for SCL or SCL–MSR.

Algorithm 2. Pseudocode of SCL-based methods for HSI.

Input: input training samples loader, temperature T, momentum m.
Initialization: encoder networks f_q and f_k with $\theta_q = \theta_k$
feature queue $\mathbf{K}$ of K elements
label queue $\mathbf{Y}$ of K elements
Output: pre-trained encoder network f_q
for $\mathbf{x}$, y **in** loader:
$\mathbf{x}_q = Aug(\mathbf{x})$ using Algorithm 1
$\mathbf{x}_k = Aug(\mathbf{x})$ using Algorithm 1
$\mathbf{q} = f_q(\mathbf{x}_q)$
$\mathbf{k} = f_k(\mathbf{x}_k)$
$Detach(\mathbf{k})$
get positive and negative pairs using Equation (5)
compute the SCL-based loss using Equation (9) or Equation (12)
back-propagation and update the encoder network f_q
update the momentum encoder network f_k using Equation (6)
update the label queue $\mathbf{K}$ and the feature queue $\mathbf{Y}$
return f_q
Detach: block the gradient of the given tensor.

3. Results

3.1. Datasets Description

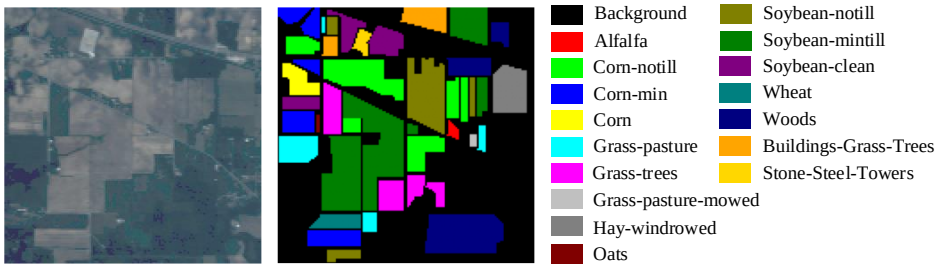
In the experiments, four widely used hyperspectral datasets, Indian Pines, Pavia University, Houston, and Chikusei, are employed to evaluate the performances of the proposed methods.

(1) Indian Pines: This dataset mainly describes the scene of multiple agricultural fields in Northwestern Indiana, USA, acquired by the Airborne Visible/Infrared Imaging Spectrometer sensor in June 1992. The dataset contains 145 × 145 pixels with a spatial resolution of 20 m × 20 m. There are 220 spectral bands with wavelengths ranging from 400 nm to 2500 nm recorded in this dataset. In the experiments, 20 low signal-to-noise ratio (SNR) bands were removed due to water absorption, and the remaining 200 bands were used for evaluation. A total of 10,249 labeled samples, belonging to 16 different land cover types, are labeled in this dataset. Figure 6 illustrates the false color composite images and the corresponding ground truth map of the Indian Pines dataset. The numbers of training and test samples per class are listed in Table 1.

(2) Pavia University: This dataset mainly covered an urban area with some manmade buildings and plants, acquired by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor over the University of Pavia, Italy. The dataset contains 610 × 340 pixels with a resolution of around 1.3 m ×1.3 m. After removing the noisy and water absorption bands, 103 bands were reserved with wavelengths ranging from 430 nm to 860 nm. The dataset contains 42,776 labeled pixels from nine different land cover types. Figure 7 shows the false color composite images and corresponding ground truth map of the Pavia University dataset. The numbers of the training and test samples per class are listed in Table 2.

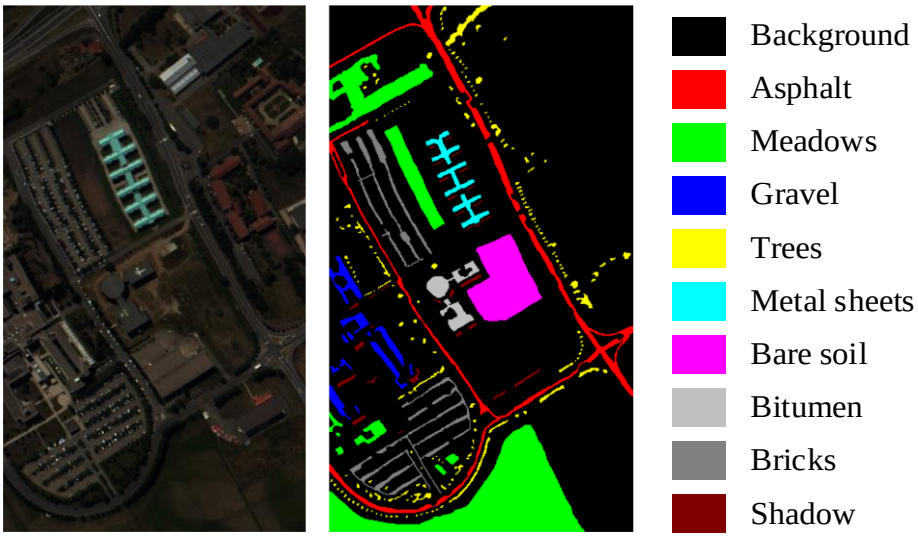
(3) Houston: This dataset was acquired over an urban area surrounding the University of Houston campus by the National Center for Airborne Laser Mapping. It had been released in the 2013 IEEE GRSS Data Fusion Contest [51]. The dataset contains 349 × 1905 pixels with a spatial resolution of 2.5 m × 2.5 m. It consists of 144 bands with wavelengths ranging from 380 nm to 1050 nm. A total of 15,029 labeled pixels corresponding to 15 different land cover types are collected in the dataset. Figure 8 illustrates the false color composite images and corresponding ground truth maps. Table 3 lists the numbers of each class and their corresponding training and test samples.

(4) Chikusei: The Chikusei dataset is an aerial hyperspectral dataset which was captured by the Headwall Hyperspec-VNIR-C sensor in Chikusei, Japan on July 29, 2014 [52]. This dataset contains 128 spectral bands with wavelengths ranging from 343 to 1018nm. The spatial size is 2517 × 2335, and the spatial resolution is 2.5 m. There are a total of 19 land types, including urban and rural areas. Figure 9 illustrates the false color composite images and the corresponding ground truth map of the Chikusei dataset. Table 4 lists the numbers of each class and their corresponding training and test samples.



(**a**) (**b**)

Figure 6. Indian Pines dataset: (**a**) false color map, (**b**) ground truth.



(**a**) (**b**)

Figure 7. Pavia University dataset: (**a**) false color map, (**b**) ground truth.

Table 1. Land cover classes and numbers of samples in the Indian Pines dataset.

No.	Class Name	Training Samples	Test Samples	Total Samples
1	Alfalfa	20	26	46
2	Corn-notill	20	1408	1428
3	Corn-mintill	20	810	830
4	Corn	20	217	237
5	Grass-pasture	20	463	483
6	Grass-trees	20	710	730
7	Grass-pasture-mowed	20	8	28
8	Hay-windrowed	20	458	478
9	Oats	15	5	20
10	Soybean-notill	20	952	972
11	Soybean-mintill	20	2435	2455
12	Soybean-clean	20	573	593
13	Wheat	20	185	205
14	Woods	20	1245	1265
15	Buildings-Grass-Trees	20	366	386
16	Stone-Steel-Towers	20	73	93
	Total	315	9934	10,249

Table 2. Land cover classes and numbers of samples in the Pavia University dataset.

No.	Class Name	Training Samples	Test Samples	Total Samples
1	Asphalt	20	6611	6631
2	Meadows	20	18,629	18,649
3	Gravel	20	2079	2099
4	Trees	20	3044	3064
5	Mental sheets	20	1325	1345
6	Bare soil	20	5009	5029
7	Bitumen	20	1310	1330
8	Bricks	20	3662	3682
9	Shadow	20	927	947
	Total	180	42,596	42,776

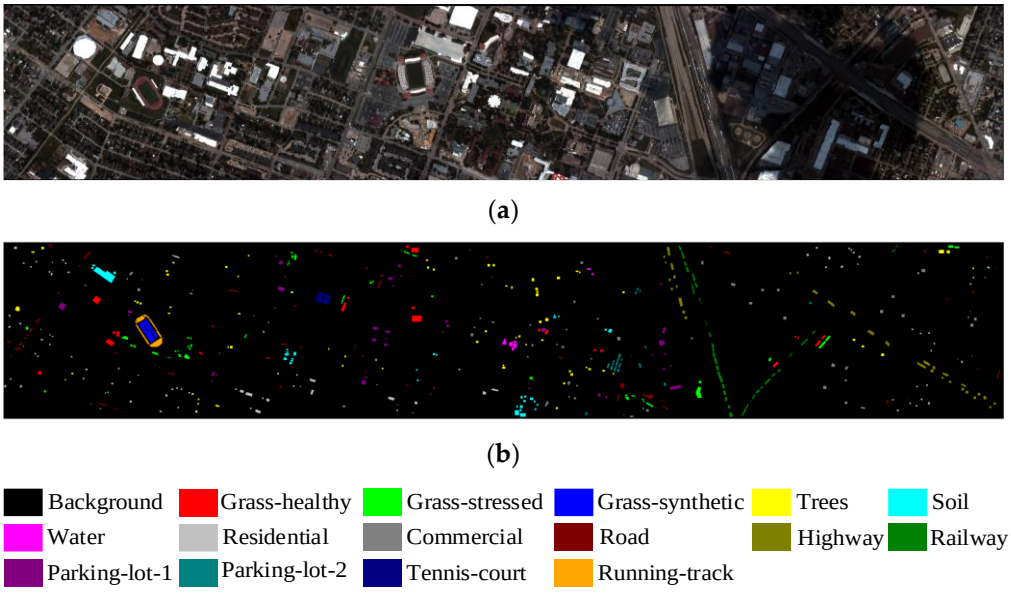


Figure 8. Houston dataset: (**a**) false color map, (**b**) ground truth.

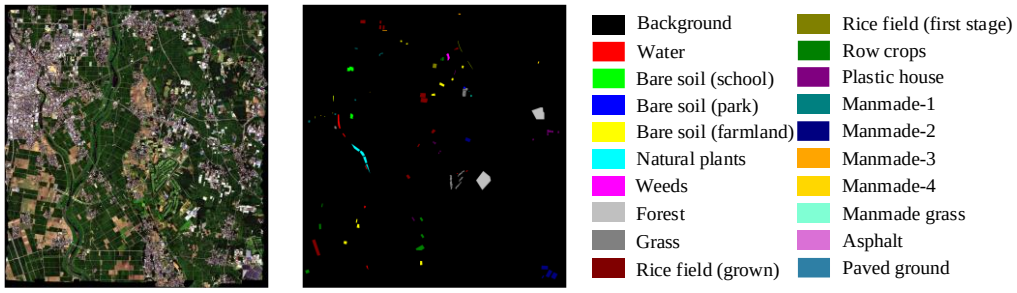


Figure 9. Chikusei dataset: (**a**) false color map, (**b**) ground truth.

Table 3. Land cover classes and numbers of samples in the Houston dataset.

No.	Class Name	Training Samples	Test Samples	Total Samples
1	Grass-healthy	20	1231	1251
2	Grass-stressed	20	1234	1254
3	Grass-synthetic	20	677	697
4	Tree	20	1224	1244
5	Soil	20	1222	1242
6	Water	20	305	325
7	Residential	20	1248	1268
8	Commercial	20	1224	1244
9	Road	20	1232	1252
10	Highway	20	1207	1227
11	Railway	20	1215	1235
12	Parking-lot-1	20	1213	1233
13	Parking-lot-2	20	449	469
14	Tennis-court	20	4008	428
15	Running-track	20	640	660
	Total	300	14,729	15,029

Table 4. Land cover classes and numbers of samples in the Chikusei dataset.

No.	Class Name	Training Samples	Test Samples	Total Samples
1	Water	5	2840	2845
2	Bare soil (school)	5	2854	2859
3	Bare soil (park)	5	281	286
4	Bare soil (farmland)	5	4847	4852
5	Natural plants	5	4292	4297
6	Weeds	5	1103	1108
7	Forest	5	20,511	20,516
8	Grass	5	6510	6515
9	Rice field (grown)	5	13,364	13,369
10	Rice field (first stage)	5	1263	1268
11	Row crops	5	5956	5961
12	Plastic house	5	2188	2193
13	Manmade-1	5	1215	1220
14	Manmade-2	5	7659	7664
15	Manmade-3	5	426	431
16	Manmade-4	5	217	222
17	Manmade grass	5	1035	1040
18	Asphalt	5	796	801
19	Paved ground	5	140	145
	Total	95	77,497	77,592

3.2. Experimental Setup

For the four datasets, the samples were divided into two subsets, which contained the training and testing samples, respectively.

The proposed SCL-based methods are evaluated on the four datasets and compared with several existing methods, including SVM with extended morphological profiles (EMP–SVM), CNN, Siamese CNN pre-trained by supervised contrastive loss (SiamSCL), SSRN [30], DBMA [53], DBDA [32], and FDSSC [54].

Specifically, for EMP–SVM, a grid search strategy is utilized together with five-fold cross-validation to find the proper C and γ $\left(C = 10^{-4}, 10^{-3}, \ldots, 10^{3}, \gamma = 10^{-4}, 10^{-3}, \ldots, 10^{3}\right)$. The first four principal components (PCs) are used when calculating EMP. For each PC, three openings and closings by reconstruction are conducted with a circular structuring element whose initial size is four and whose step size increment is two.

For SSRN, DBMA, DBDA, and FDSSC, the experimental settings are the same as those in [32].

For CNN, SiamSCL, SCL, and SCL–MSR, we use the same backbone for feature extraction, the detailed architecture of which is shown in Table 5. It contains five blocks. Each of the first four blocks consists of a convolutional layer, a BN layer, and an ReLU operation. The 1×1 convolution in the first block is mainly used for dimension reduction and less overfitting. Each of the second, third, and fourth blocks includes a 2×2 max-pooling layer. The fifth block contains only a linear layer, outputting a 256-dimension feature for each HSI cube. For the final classification task, a fully connected layer is added to the backbone to form the overall classification model. In order to use spatial information, the input images with a spatial size of 27×27 ($W = 27$) are fed to the 2D CNN.

Table 5. Architecture of CNN.

No.	Convolution	ReLU	Pooling	Padding	Stride	BN	Linear
1	$1 \times 1 \times 32$	YES	No	NO	1	YES	-
2	$4 \times 4 \times 32$	YES	2×2	NO	1	YES	-
3	$3 \times 3 \times 64$	YES	2×2	NO	1	YES	-
4	$4 \times 4 \times 128$	YES	2 2	NO	1	YES	-
5	-	NO	NO	NO	-	NO	128×256

We used the Adam optimizer to train the SCL-based methods for a total of 300 epochs, including SiamSCL, SCL, and SCL–MSR. The cosine learning rate scheduler with an initial learning rate of 0.001 was adopted in the experiments.

When training the CNN or fine-tuning the SCL-based methods, the multi-step learning rate scheduler is utilized. The initial learning rate is set to be 0.001, and it is divided by 10 after 80 and 160 epochs. In the training procedure, the total number of epochs is set to 180 for all four datasets. In addition, the mini-batch method is adopted, where the batch size is set to 512.

For the data augmentation used in SCL and SCL–MSR, the minimum scale S and the occlusion probability p are treated as hyperparameters to be analyzed. The occlusion shape parameters are $v_{min} = 0$, and v_{max} =0.00625, which derives from 1/4 of the height, 1/4 of the width, and 1/10 of the number of bands. Furthermore, l_{min} =0.2, $l_{max} = 1/l_{min}$, r_{min} =0.2, and $r_{max} = 1/r_{min}$. The occlusion value is set to be 0.5, as suggested by [47].

In the experiments, we set the value of queue length H to be h times the number of total training samples N, namely $H = hN$. The temperature T, queue length ratio h, and momentum coefficient m are analyzed in the experiments.

In this portion of the experiments, the classification performance is mainly evaluated using overall accuracy (OA), average accuracy (AA), and the kappa coefficient (K). The experiments are repeated 10 times, and the training samples are randomly chosen from all the labeled samples each time.

3.3. Classification Results and Analysis

The classification results of the different methods over four datasets are reported in Tables 6–9. All the reported results are the average values of 10 runs with respect to different random initializations.

(1) Classification Results: Table 6 demonstrates the classification accuracies for the Indian Pines dataset. It can be observed that the proposed SCL and SCL–MSR are superior to the other methods with 20 training samples per class. In particular, SCL outperforms DBDA by 0.69 percentage points, 11.39 percentage points, and 0.00074 in terms of OA, AA, and K. Compared to the original CNN, SCN improves the classification accuracy by 1.33 percentage points, 0.54 percentage points, and 0.0149 in terms of OA, AA, and K. In addition, SiamSCL works better than the original CNN due to the use of contrastive loss for supervised pre-training. SCL–MSR also achieves better results than SCL, which demonstrates the effectiveness of the multi-level similarity regularization. Table 7 demonstrates the classification accuracies for the Pavia University dataset. It is apparent that the proposed SCL-based methods still achieve better classification results for the Pavia University dataset when compared with the other methods. Specifically, SCL outperforms FDSSC by 1.06 percentage points, 2.53 percentage points, and 0.00138 in terms of OA, AA, and K, respectively. Both SCL and SCL–MSR work better with respect to the classification accuracy for each class. From Table 8, it can be seen that SCL and SCL–MSR obtain higher classification results for the Houston dataset, similar to those of the Pavia University and Indian Pines datasets. Table 9 demonstrates the classification accuracies for the Chikusei dataset. It can be seen that the proposed SCL and SCL–MSR achieves higher classification accuracies than the comparison methods.

(2) Analysis of the reason why the algorithms perform perfectly for some classes while not for the others: As far as we are concerned, there are three major factors that could affect the classification accuracies.

Table 6. Testing data classification results (mean ± standard deviation) on the Indian Pines dataset.

Class	EMP–SVM	CNN	SiamSCL	SSRN	DBMA	DBDA	FDSSC	SCL	SCL–MSR
	96.54 ± 2.07	100.0 ± 0.00	100.0 ± 0.00	86.04 ± 9.73	66.41 ± 13.17	79.35 ± 12.80	89.04 ± 10.60	100.0 ± 0.00	100.0 ± 0.00
	63.74 ± 7.45	78.96 ± 6.22	79.51 ± 4.00	86.85 ± 6.00	84.14 ± 7.00	86.89 ± 10.74	92.35 ± 6.46	81.46 ± 2.96	83.32 ± 3.12
	76.56 ± 4.40	87.75 ± 5.96	88.96 ± 6.73	86.54 ± 5.65	84.42 ± 8.07	86.88 ± 8.79	87.91 ± 11.64	89.00 ± 4.76	90.06 ± 3.03
	81.94 ± 5.14	98.53 ± 2.56	98.29 ± 2.05	73.29 ± 13.10	82.27 ± 10.20	81.64 ± 11.47	80.76 ± 14.64	98.39 ± 2.19	98.34 ± 2.13
	86.57 ± 3.85	92.18 ± 3.21	91.71 ± 2.57	98.00 ± 1.96	93.89 ± 5.25	97.02 ± 3.74	98.69 ± 1.36	93.00 ± 2.49	93.13 ± 1.82
	92.85 ± 4.63	95.41 ± 2.53	94.93 ± 3.73	97.79 ± 1.40	98.14 ± 1.32	96.54 ± 1.25	97.95 ± 1.56	94.62 ± 3.97	91.87 ± 5.09
	92.50 ± 6.12	100.0 ± 0.00	100.0 ± 0.00	73.78 ± 18.70	39.00 ± 26.53	58.08 ± 28.22	66.78 ± 30.24	100.0 ± 0.00	100.0 ± 0.00
	94.10 ± 3.25	99.80 ± 0.59	99.85 ± 0.39	99.46 ± 0.83	99.36 ± 1.21	99.62 ± 0.58	99.93 ± 0.10	99.83 ± 0.52	99.87 ± 0.46
	98.00 ± 6.00	100.0 ± 0.00	100.0 ± 0.00	44.13 ± 20.56	15.44 ± 8.84	32.65 ± 11.22	32.03 ± 21.44	100.0 ± 0.00	100.0 ± 0.00
	68.28 ± 7.20	88.93 ± 3.10	89.09 ± 4.48	79.53 ± 6.45	78.91 ± 5.13	81.08 ± 8.08	79.27 ± 12.02	89.29 ± 4.41	90.42 ± 4.30
	59.22 ± 4.69	87.41 ± 5.47	88.21 ± 3.66	92.94 ± 3.50	92.90 ± 3.57	95.43 ± 4.19	92.20 ± 6.64	89.66 ± 2.93	89.25 ± 3.98
	66.61 ± 7.68	83.84 ± 6.23	84.56 ± 6.21	82.78 ± 10.73	81.50 ± 11.12	90.57 ± 13.45	87.94 ± 14.32	85.46 ± 6.03	88.38 ± 3.25
	97.08 ± 1.81	99.57 ± 0.89	99.41 ± 0.89	95.49 ± 3.82	97.35 ± 2.96	91.42 ± 5.35	92.61 ± 4.98	99.57 ± 0.76	99.46 ± 0.76
	86.39 ± 5.68	97.39 ± 1.71	97.45 ± 1.80	97.87 ± 1.14	97.70 ± 1.42	98.43 ± 1.05	98.57 ± 8.41	97.82 ± 1.52	97.27 ± 1.44
	71.48 ± 8.55	97.19 ± 2.60	97.62 ± 2.22	87.30 ± 7.75	80.60 ± 5.53	88.99 ± 5.46	83.25 ± 12.18	97.57 ± 2.64	98.17 ± 1.65
	95.48 ± 4.71	99.18 ± 0.91	99.32 ± 0.92	79.51 ± 4.48	78.45 ± 8.91	68.09 ± 12.75	80.42 ± 6.41	99.18 ± 0.91	99.17 ± 0.67
OA (%)	73.32 ± 2.25	89.62 ± 1.72	90.15 ± 1.27	89.44 ± 1.38	87.96 ± 1.24	90.26 ± 3.06	89.71 ± 2.72	90.95 ± 1.35	91.23 ± 1.40
AA (%)	82.96 ± 1.31	94.14 ± 0.72	94.31 ± 0.72	85.08 ± 2.08	79.41 ± 1.59	83.29 ± 2.47	84.98 ± 3.97	94.68 ± 0.80	94.92 ± 0.67
K × 100	69.93 ± 2.50	88.20 ± 1.93	88.80 ± 1.42	88.00 ± 1.55	86.32 ± 1.34	88.95 ± 3.43	88.30 ± 3.08	89.69 ± 1.52	90.02 ± 1.57

Table 7. Testing data classification results (mean ± standard deviation) on the Pavia University dataset.

Class	EMP–SVM	CNN	SiamSCL	SSRN	DBMA	DBDA	FDSSC	SCL	SCL–MSR
	81.27 ± 6.60	90.85 ± 4.53	92.02 ± 5.09	97.84 ± 1.71	98.47 ± 0.70	98.74 ± 1.20	98.59 ± 1.61	94.58 ± 3.78	93.42 ± 3.45
	83.13 ± 3.26	93.83 ± 3.49	95.32 ± 4.20	97.72 ± 0.81	98.08 ± 1.35	99.51 ± 0.36	99.19 ± 0.38	96.62 ± 3.96	97.62 ± 2.78
	81.60 ± 4.51	98.12 ± 1.27	98.88 ± 0.86	83.71 ± 8.17	78.83 ± 10.58	90.81 ± 12.10	92.84 ± 5.72	98.86 ± 0.96	98.56 ± 1.90
	95.29 ± 2.44	96.29 ± 1.39	96.18 ± 2.05	97.70 ± 2.01	88.43 ± 4.01	92.74 ± 7.92	94.75 ± 5.75	96.89 ± 0.99	96.73 ± 1.17
	99.26 ± 0.26	99.33 ± 0.52	99.49 ± 0.45	99.86 ± 0.27	96.67 ± 4.57	99.53 ± 0.63	99.88 ± 0.12	99.58 ± 0.34	99.40 ± 0.54
	80.27 ± 6.31	99.47 ± 0.63	99.86 ± 0.27	91.98 ± 3.69	86.60 ± 8.54	90.96 ± 5.45	95.65 ± 2.21	99.27 ± 1.36	99.86 ± 0.32
	93.11 ± 1.56	99.39 ± 0.68	99.52 ± 0.43	88.49 ± 12.03	95.13 ± 8.12	93.80 ± 8.83	96.36 ± 2.67	99.71 ± 0.36	99.63 ± 0.44
	83.86 ± 3.96	98.94 ± 0.80	99.02 ± 0.78	84.79 ± 7.36	85.56 ± 8.62	89.83 ± 6.47	83.63 ± 11.84	98.99 ± 0.85	99.08 ± 0.89
	99.85 ± 0.12	96.66 ± 1.33	96.76 ± 1.55	99.41 ± 0.94	92.34 ± 3.50	96.82 ± 1.69	97.35 ± 3.44	96.54 ± 1.40	96.43 ± 2.02
OA (%)	84.53 ± 2.22	95.26 ± 1.74	96.18 ± 2.15	94.72 ± 1.17	92.93 ± 1.75	95.87 ± 1.85	96.07 ± 1.89	97.13 ± 1.82	97.43 ± 1.49
AA (%)	88.63 ± 1.57	96.99 ± 0.74	97.45 ± 0.82	93.50 ± 2.07	91.13 ± 1.54	94.75 ± 2.36	95.36 ± 1.64	97.89 ± 0.70	97.86 ± 0.69
K × 100	80.00 ± 2.76	93.80 ± 2.23	95.00 ± 2.78	93.02 ± 1.53	90.75 ± 2.23	94.58 ± 2.41	94.84 ± 2.42	96.22 ± 2.35	96.62 ± 1.94

Table 8. Testing data classification results (mean ± standard deviation) on the Houston dataset.

Class	EMP–SVM	CNN	SiamSCL	SSRN	DBMA	DBDA	FDSSC	SCL	SCL–MSR
	92.99 ± 4.30	92.59 ± 4.56	93.23 ± 4.93	96.25 ± 2.94	94.79 ± 3.34	93.48 ± 5.59	96.41 ± 2.30	94.25 ± 5.17	93.11 ± 4.73
	93.06 ± 5.72	97.00 ± 2.21	96.54 ± 2.35	97.65 ± 2.48	92.77 ± 4.62	95.10 ± 3.78	97.64 ± 2.11	97.46 ± 1.94	97.29 ± 1.79
	98.97 ± 1.10	98.66 ± 1.41	99.03 ± 1.33	99.93 ± 0.22	99.76 ± 0.51	100.0 ± 0.00	100.0 ± 0.00	98.98 ± 1.20	98.41 ± 1.85
	94.75 ± 2.94	97.66 ± 1.81	98.04 ± 1.51	95.98 ± 4.13	94.93 ± 3.23	97.13 ± 2.17	95.63 ± 3.79	98.46 ± 1.45	97.31 ± 2.02
	96.51 ± 4.52	97.47 ± 5.11	98.05 ± 5.12	95.41 ± 2.36	96.65 ± 2.67	97.66 ± 2.42	97.60 ± 2.48	98.75 ± 3.03	98.58 ± 3.34
	94.72 ± 3.42	95.38 ± 3.52	95.01 ± 3.91	97.31 ± 7.83	96.88 ± 3.61	97.37 ± 2.23	99.80 ± 0.34	94.89 ± 3.70	95.51 ± 3.79
	85.54 ± 4.67	90.54 ± 2.41	91.44 ± 1.99	92.10 ± 2.47	86.21 ± 4.33	91.93 ± 3.62	92.56 ± 4.64	93.04 ± 2.74	91.79 ± 2.42
	69.36 ± 4.90	78.48 ± 6.64	79.56 ± 3.12	93.23 ± 3.49	90.28 ± 4.81	94.88 ± 3.19	92.68 ± 3.66	80.74 ± 4.68	89.27 ± 3.89
	75.81 ± 6.81	90.60 ± 3.94	92.00 ± 2.80	89.87 ± 3.83	86.17 ± 4.01	88.57 ± 2.27	90.82 ± 3.19	91.83 ± 5.15	91.49 ± 3.93
	87.63 ± 4.01	96.06 ± 4.12	97.12 ± 3.88	86.49 ± 6.78	91.35 ± 2.74	89.76 ± 3.83	89.12 ± 4.70	97.75 ± 2.87	99.34 ± 1.31
	85.58 ± 7.72	91.52 ± 4.71	94.24 ± 3.87	90.45 ± 1.94	91.57 ± 4.51	95.44 ± 2.10	92.70 ± 3.56	94.79 ± 2.62	95.81 ± 1.68
	76.18 ± 6.14	91.81 ± 5.48	92.94 ± 5.34	89.91 ± 4.73	90.33 ± 5.84	93.15 ± 3.27	93.40 ± 3.86	92.48 ± 6.75	95.03 ± 3.85
	56.44 ± 5.90	96.08 ± 2.62	95.10 ± 3.78	93.52 ± 5.65	77.27 ± 7.71	82.75 ± 6.06	83.22 ± 9.14	95.03 ± 3.63	95.06 ± 2.66
	97.94 ± 2.53	99.93 ± 0.22	100.0 ± 0.00	97.67 ± 3.29	95.37 ± 6.72	98.13 ± 2.59	98.07 ± 2.72	100.0 ± 0.00	100.0 ± 0.00
	99.08 ± 0.46	99.59 ± 1.06	99.72 ± 0.70	96.93 ± 1.96	96.17 ± 2.20	95.05 ± 2.47	96.22 ± 3.18	99.58 ± 0.97	99.92 ± 0.19
OA (%)	86.56 ± 1.36	93.36 ± 0.92	94.03 ± 0.91	93.32 ± 1.05	91.58 ± 0.69	93.67 ± 0.92	93.97 ± 0.99	94.65 ± 0.73	94.84 ± 0.72
AA (%)	86.97 ± 1.26	94.23 ± 0.67	94.80 ± 0.73	94.18 ± 1.12	92.03 ± 0.71	94.03 ± 0.87	94.39 ± 1.00	95.20 ± 0.60	95.39 ± 0.60
K × 100	85.47 ± 1.47	92.82 ± 0.99	93.64 ± 0.98	92.78 ± 1.13	90.90 ± 0.75	93.16 ± 1.00	93.48 ± 1.07	94.21 ± 0.79	94.42 ± 0.77

Table 9. Testing data classification results (mean ± standard deviation) on the Chikusei dataset.

Class	EMP–SVM	CNN	SiamCLSP	SSRN	DBMA	DBDA	FDSSC	CLSP	CLSP–MSR
	83.55 ± 10.60	92.99 ± 4.40	91.74 ± 4.18	83.51 ± 12.94	84.50 ± 11.82	83.44 ± 13.8	86.47 ± 12.42	91.17 ± 4.62	93.42 ± 3.92
	93.83 ± 3.84	99.54 ± 0.53	99.59 ± 0.49	98.07 ± 2.02	99.82 ± 0.23	99.65 ± 0.51	98.55 ± 3.30	99.60 ± 0.53	99.45 ± 0.52
	98.01 ± 2.62	99.57 ± 0.98	99.78 ± 0.43	28.93 ± 10.75	23.02 ± 5.54	31.63 ± 15.77	29.06 ± 14.94	97.30 ± 5.46	97.08 ± 6.14
	50.19 ± 20.7	82.66 ± 16.10	82.82 ± 15.22	90.14 ± 11.38	89.34 ± 9.32	87.22 ± 10.73	84.33 ± 10.55	86.55 ± 1.78	86.07 ± 16.79
	96.70 ± 2.76	99.95 ± 0.02	99.99 ± 0.02	95.10 ± 3.32	97.64 ± 2.67	96.53 ± 3.24	94.59 ± 3.69	99.97 ± 5.36	99.98 ± 0.03
	87.28 ± 12.13	95.62 ± 3.64	95.26 ± 3.61	73.53 ± 22.89	71.42 ± 22.95	85.41 ± 24.14	81.26 ± 18.48	95.27 ± 3.86	95.27 ± 3.86
	82.13 ± 7.49	99.97 ± 0.05	99.97 ± 0.07	95.66 ± 3.70	94.69 ± 4.92	99.37 ± 0.87	98.10 ± 1.66	99.99 ± 0.02	99.98 ± 0.07
	91.93 ± 2.72	93.05 ± 2.99	94.42 ± 3.42	96.71 ± 4.96	99.06 ± 0.97	99.90 ± 0.27	98.81 ± 2.01	93.91 ± 3.02	95.23 ± 1.95
	79.34 ± 20.97	94.59 ± 10.58	98.22 ± 2.42	96.57 ± 3.77	95.30 ± 5.11	99.43 ± 0.46	96.95 ± 4.78	97.74 ± 3.03	98.69 ± 1.75
	99.26 ± 0.55	99.94 ± 0.17	99.92 ± 0.17	81.93 ± 9.86	80.55 ± 15.50	89.73 ± 5.23	82.11 ± 12.83	99.64 ± 0.96	99.98 ± 0.07
	66.40 ± 14.47	82.22 ± 10.90	79.41 ± 12.74	94.58 ± 11.37	93.09 ± 3.32	97.42 ± 3.29	94.36 ± 10.23	85.51 ± 8.7	85.19 ± 8.01
	69.20 ± 11.50	84.48 ± 9.13	85.50 ± 8.78	91.50 ± 6.20	92.15 ± 4.92	96.78 ± 4.48	89.21 ± 11.74	85.74 ± 8.34	85.53 ± 9.46
	95.09 ± 1.97	95.97 ± 1.48	96.15 ± 1.77	96.16 ± 7.62	92.84 ± 7.39	98.75 ± 2.27	92.87 ± 9.70	96.12 ± 1.72	96.18 ± 1.56
	86.85 ± 11.24	89.49 ± 10.80	90.76 ± 10.58	99.80 ± 0.33	98.09 ± 2.53	99.60 ± 7.82	99.75 ± 0.50	91.70 ± 8.5	92.60 ± 10.94
	91.01 ± 17.23	91.78 ± 8.43	91.19 ± 16.23	93.87 ± 9.69	92.99 ± 9.33	98.12 ± 5.20	96.65 ± 7.27	91.78 ± 10.4	91.78 ± 6.43
	93.73 ± 7.85	95.67 ± 6.04	95.66 ± 6.04	93.60 ± 7.32	94.38 ± 5.15	98.24 ± 3.48	92.51 ± 2.45	94.29 ± 6.89	96.04 ± 7.87
	93.39 ± 6.38	96.06 ± 8.39	94.97 ± 8.70	98.35 ± 1.65	96.53 ± 2.88	96.62 ± 2.32	96.83 ± 4.24	98.51 ± 1.61	98.98 ± 1.78
	88.52 ± 12.17	83.98 ± 11.2	85.30 ± 11.83	69.53 ± 13.85	64.40 ± 14.60	72.33 ± 13.82	59.54 ± 14.88	85.10 ± 12.3	83.79 ± 11.08
	88.07 ± 7.69	98.86 ± 3.43	98.85 ± 3.43	24.50 ± 16.27	14.22 ± 9.28	35.81 ± 35.81	61.44 ± 25.40	99.79 ± 0.64	100.0 ± 0.00
OA (%)	81.58 ± 4.64	93.87 ± 2.28	94.51 ± 1.66	91.46 ± 3.62	90.12 ± 4.35	94.39 ± 2.39	92.79 ± 3.20	95.20 ± 1.91	95.58 ± 2.02
AA (%)	86.02 ± 3.06	93.50 ± 1.40	93.66 ± 1.26	84.32 ± 2.98	82.84 ± 2.27	88.39 ± 2.24	85.97 ± 3.09	94.19 ± 1.29	94.49 ± 1.55
K × 100	78.97 ± 5.31	92.95 ± 2.60	93.68 ± 1.90	90.18 ± 4.13	88.65 ± 4.96	93.55 ± 2.73	91.72 ± 3.65	94.47 ± 2.18	94.91 ± 2.31

(i) The inherent cluster properties of different classes. In the feature space, samples from the same class should be close to each other, while samples from different classes should be far away from each other. In fact, the samples from some specific classes are in good agreement with the above requirements; that is, they have better clustering properties, and they are prone to be classified well compared to other classes. For example, Figure 10 shows the average spectrum reflectance curves of different classes on the Pavia University dataset. The average spectrum reflectance curves of the classes *mental sheets* and *meadows* are significantly different from other classes, which provides a potential to achieve high classification accuracies.

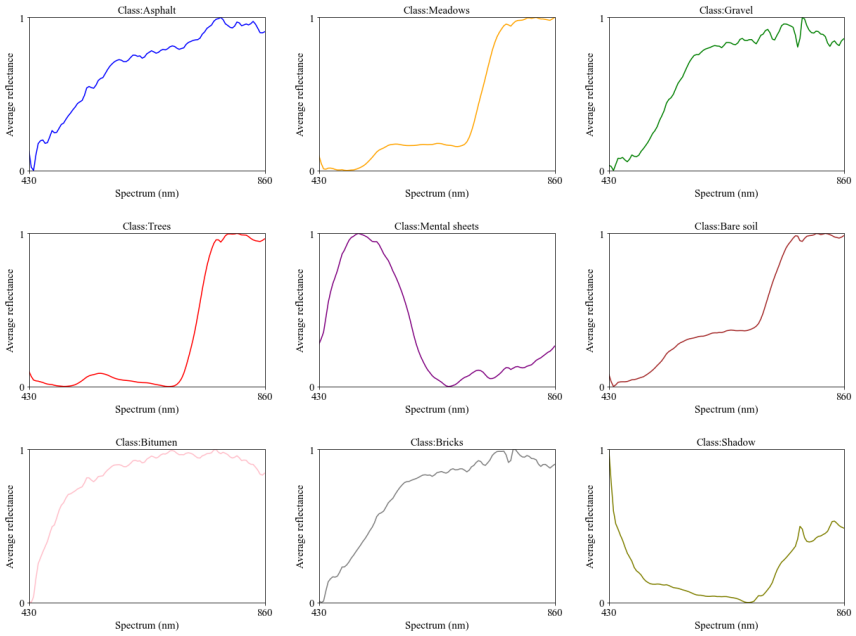


Figure 10. The average spectrum reflectance curves of different classes on the Pavia University dataset.

(ii) The distribution difference between the training set and the test set. If the training set and the test set share great similarity, a well-trained classifier on the training set will often perform well on the test set. Otherwise, it will perform poorly due to overfitting. In some cases, the samples of some classes are concentrated in a small area, and therefore, there is little difference among the samples. The randomly divided training set and test set share great similarity, which leads to a good test performance. For example, the *Oats*, *Grass-pasture-mowed,* and *Alfalfa* in the Indian Pines dataset have few labeled samples and cover small regions, and the CNN and SCL can achieve high accuracies. However, the DBDN, DBMA, and FDSSC obtain lower accuracies for the oats in the Indian Pines dataset compared with the CNN. The reason is that the DBDN, DBMA, and FDSSC all adopt an attention mechanism, which makes them focus on the pixels from other classes in some runs. This is related to the following third factor.

(iii) The properties of the given classifiers. The classification ability is also vital, and different classifiers have different classification performances. Generally speaking, the CNN-based methods usually achieve higher classification accuracies than the traditional methods such as EMP–SVM, due to its powerful feature extraction ability. They can learn implicit and complex patterns and have the potential to achieve higher accuracies.

(3) Analysis of the reason why the proposed SCL and SCL–MSR obtain lower accuracies in some classes compared to other methods: Generally speaking, the DBDA, DBMA, and FDSSC all adopt an attention mechanism, which makes the model pay more attention to the information contributed to the classification according to the training set. The CNN, SiamSCL, SCL, and SCL–MSR share the same backbone, which is a vanilla convolutional neural network.

To use spatial information, we use a neighborhood region surrounding the given pixel as the input for deep models. The attention mechanism makes DBDA and FDSSC focus on the most useful areas, especially when the input images contain complex land cover, such as boundaries. For example, some samples of the Grass-synthetic class in the Houston dataset are very close to the Running-track class, and some images may be similar to each other (e.g., samples in the red square in Figure 11.). In this case, the attention mechanism provides better discriminability than SCL and thus leads to better classification accuracies. This phenomenon can be found in other datasets.

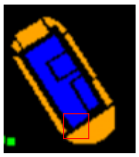


Figure 11. The ground truth of the Grass-synthetic class and the Running-track class in the Houston dataset. The red box represents the region where samples are likely to be ambiguous.

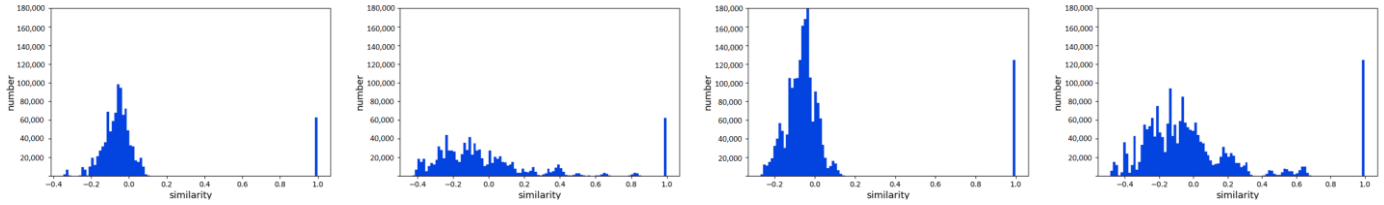
(4) Analysis of the reason why SCL–MSR does not obtain a better performance than SCL for some classes: SCL–MSR acts to alleviate the overfitting problem. It disturbs the training process to prevent the positive/negative pairs from being too close to/far away from each other. However, deep neural networks (e.g., CNN) can be class-biased: some classes ("easy" classes) are easy to learn and converge faster than other classes ("hard" classes) [55]. After adding MSR, for those classes that are easy to learn and even to be overfitting, MSR plays a positive role which prevents the model from being overfitting on the classes, and it thus achieves high accuracies, while for some classes on which the model is not trained well the MSR has a negative effect which makes the model underfit these classes and makes it finally obtain worse accuracies.

3.4. Ablation Experiments

(1) Experiments of temperature T and queue length ratio h: As mentioned before, T is a temperature parameter that controls the concentration level of the distribution. From Equation (9), one can see that a smaller value of T will make the distribution of the features

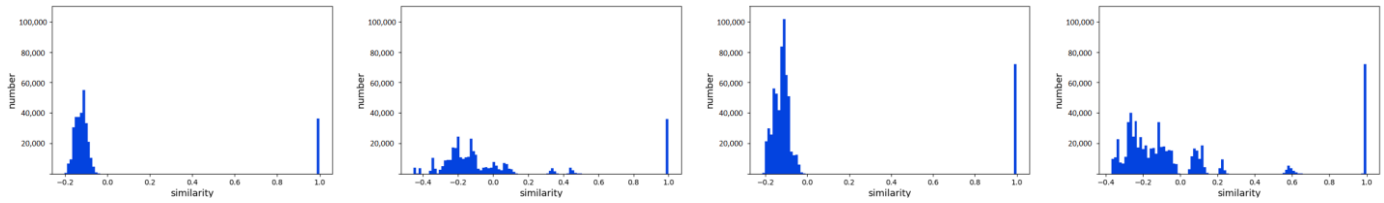
more concentrated, which means that the convergence will be faster but will carry more risk of overfitting. Parameter h controls the length of the queue in SCL. A larger value of h means a longer queue and a greater variety of features in the queue. A proper value of h will help SCL generalize better.

Figures 12–15 show the distribution of the similarity scores $\{\boldsymbol{q}\cdot\boldsymbol{k}_i\}_{i=1}^{hN}$ in a batch on the four datasets. It can be seen that the distributions shown in (a) and (c) are more concentrated than those in (b) and (d), due to the smaller T. Furthermore, the value of h has little influence on similarity. It can be seen that that the similarity distributions corresponding to the different values of h are similar when the value of T is fixed.



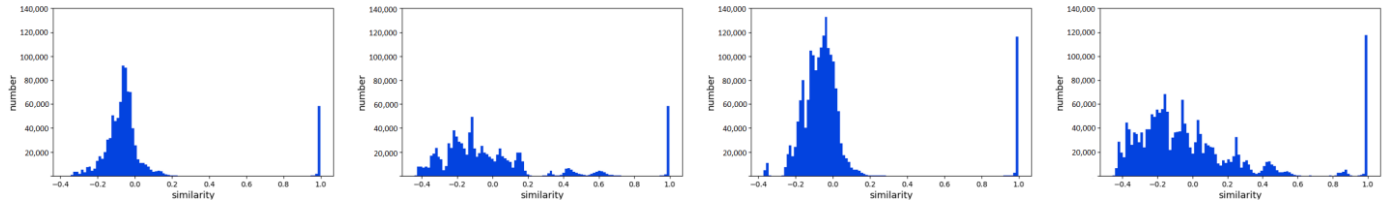
(**a**) (**b**) (**c**) (**d**)

Figure 12. Distribution of similarity over the Indian Pines dataset, when epoch = 300: (**a**) T = 0.25, h = 10; (**b**) T = 1.0, h = 10; (**c**) T = 0.25, h = 20; (**d**) T = 1.0, h = 20.



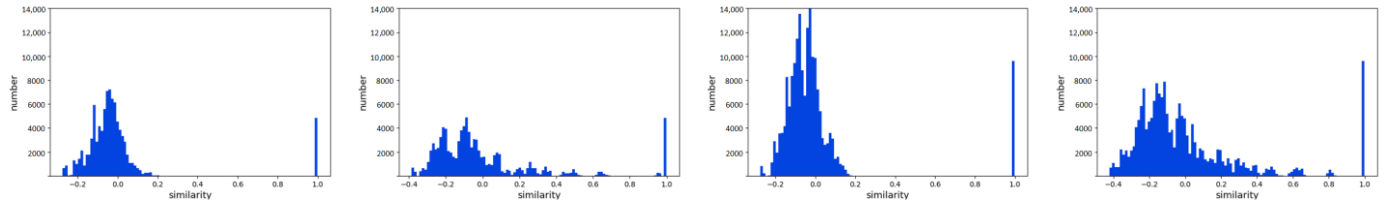
(**a**) (**b**) (**c**) (**d**)

Figure 13. Distribution of similarity over the Pavia University dataset, when epoch = 300: (**a**) T = 0.25, h = 10; (**b**) T = 1.0, h = 10; (**c**) T = 0.25, h = 20; (**d**) T = 1.0, h = 20.



(**a**) (**b**) (**c**) (**d**)

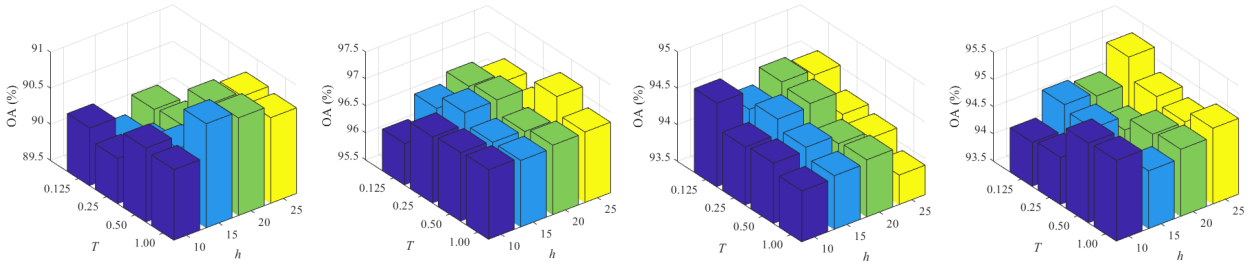
Figure 14. Distribution of similarity over the Houston dataset, when epoch = 300: (**a**) T = 0.25, h = 10; (**b**) T = 1.0, h = 10; (**c**) T = 0.25, k = 20; (**d**) T = 1.0, h = 20.



(**a**) (**b**) (**c**) (**d**)

Figure 15. Distribution of similarity over the Chikusei dataset, when epoch = 300: (**a**) T = 0.25, h = 10; (**b**) T = 1.0, h = 10; (**c**) T = 0.25, k = 20; (**d**) T = 1.0, h = 20.

Figure 16 illustrates the overall accuracies obtained by using different values of temperature T and queue length ratio h. In this experiment, the grid search strategy is adopted to validate the proper T and h $\left(T = 2^{-3}, 2^{-2}, 2^{-1}, 2^{0}; k = 10, 15, 20, 25\right)$. It can be seen that different datasets require different values of T and h for proper training. For example, the Indian Pines dataset usually prefers a larger value of T. However, a smaller value of T is more appropriate for the Houston dataset. It indicates that the choice of the T value is related to the complexity of the HSI dataset used.



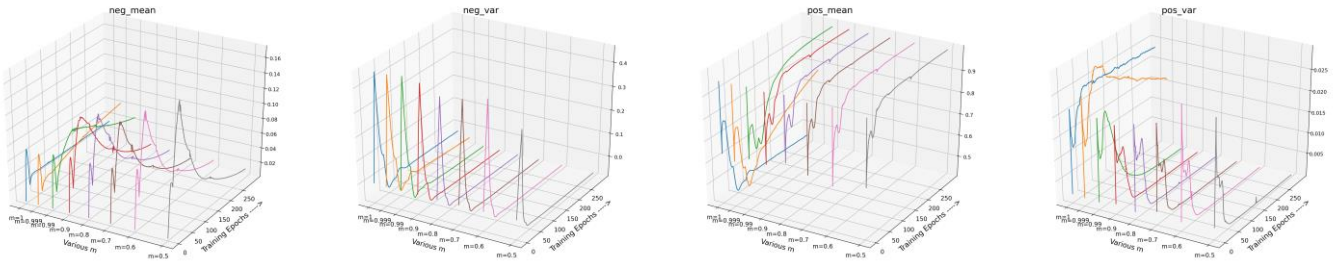
(**a**) (**b**) (**c**) (**d**)

Figure 16. Classification accuracies over different values of T and h: (**a**) the Indian Pines dataset; (**b**) the Pavia University dataset; (**c**) the Houston dataset; (**d**) the Chikusei dataset.

As illustrated by Figure 16, T and h are set to the optimal value for different datasets. Specifically, the values of T and h are 1.0 and 15 for the Indian Pines dataset, respectively, while T is 0.5 and h are 25 for the Pavia University dataset, and T is set to 0.125 and h is set to 10 for the Houston dataset. For the Chikusei dataset, the best values of T and h are 0.125 and 25.

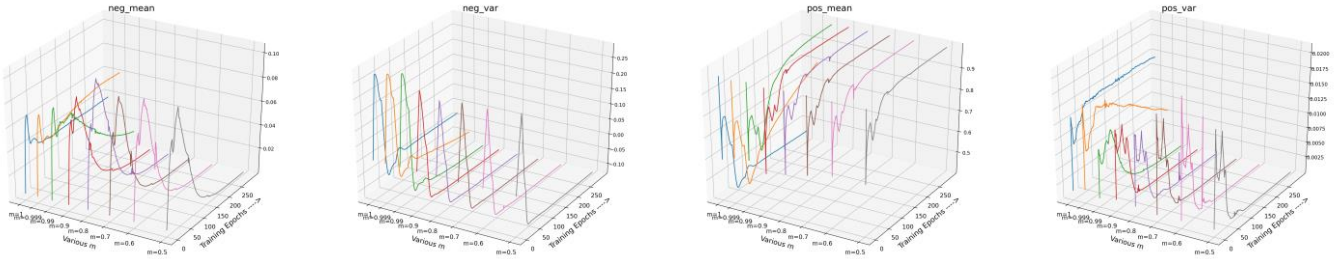
(2) Experiments of Momentum Coefficient *m:* Figures 17–20 show four types of similarity statistics employed when using various values of m, including mean of negative similarity, variance of negative similarity, mean of positive similarity, and variance of positive similarity. It is worth noting that in the early training process, the values of similarity statistics are not as expected due to the queue initialization.

From Figures 17–20, one can see that as the training process progresses, the mean of negative similarity, the variance of negative similarity, and the variance of positive similarity will decrease, but the mean of positive similarity will increase. Generally speaking, a smaller value of m means the faster update of the momentum encoder. For m = 1.0, the means of both negative and positive similarity are nearly constants, and the variances are large. In this case, the momentum encoder does not update its parameters, which makes the SCL model collapse. For m = 0.999, the SCL model is also hard to converge. As the value of m decreases, the mean of negative/positive similarity will increase/decrease more quickly. Specifically, for the mean of the negative similarity scores, the end value of m = 0.6 (pink) and m = 1.0 (grey) is much lower than m = 0.99 (green). However, a smaller value of m makes the model more prone to the risk of overfitting, which does harm to the generalization.



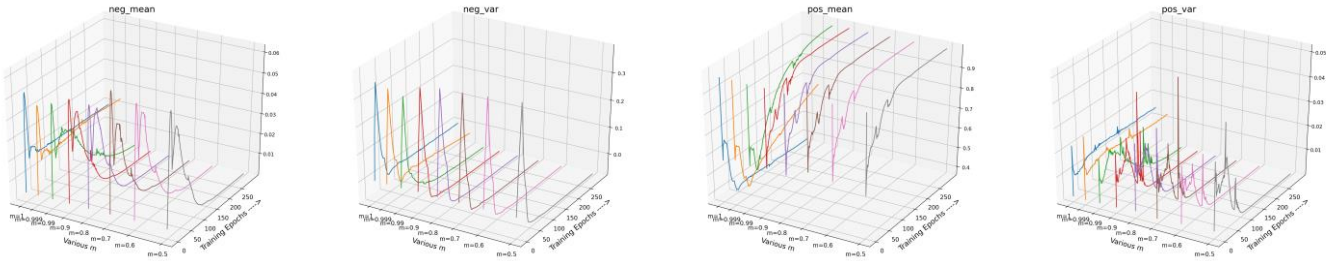
(**a**) (**b**) (**c**) (**d**)

Figure 17. Similarity statistics of various m in SCL training on the Indian Pines dataset: (**a**) mean of negative similarity; (**b**) variance of negative similarity; (**c**) mean of positive similarity; (**d**) variance of positive similarity. Different colors corresponds to different values of m.



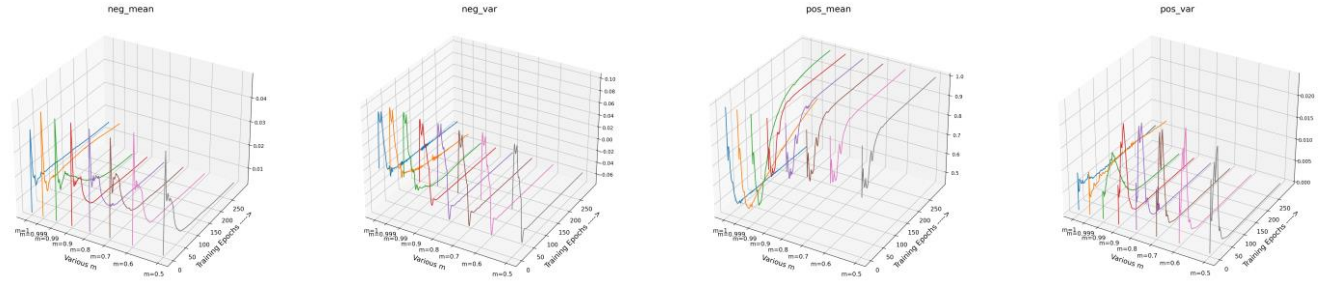
(**a**) (**b**) (**c**) (**d**)

Figure 18. Similarity statistics of various m in SCL training on the Pavia University dataset: (**a**) mean of negative similarity; (**b**) variance of negative similarity; (**c**) mean of positive similarity; (**d**) variance of positive similarity. Different colors corresponds to different values of m.



(**a**) (**b**) (**c**) (**d**)

Figure 19. Similarity statistics of various m in SCL training on the Houston dataset: (**a**) mean of negative similarity; (**b**) variance of negative similarity; (**c**) mean of positive similarity; (**d**) variance of positive similarity. Different colors corresponds to different values of m.



(**a**) (**b**) (**c**) (**d**)

Figure 20. Similarity statistics of various m in SCL training on the Chikusei dataset: (**a**) mean of negative similarity; (**b**) variance of negative similarity; (**c**) mean of positive similarity; (**d**) variance of positive similarity. Different colors corresponds to different values of m.

Figure 21 shows the overall accuracy of SCL with different values of momentum coefficient m. The results indicate that a larger value of m or a smaller one is not suitable for SCL. We find that m = 0.99 is proper for most cases via experiments. For simplicity and universality, we choose 0.99 as the value of m and control the training process by changing the values of temperature T and queue length ratio h.

(3) Experiments of Augmentation Techniques: Figure 22 shows the classification results of SCL using different data augmentation techniques over the four datasets. From Figure 22, it can be seen that the use of both multiscale and random occlusion augmentations makes the SCL perform better than when only one or no augmentation technique is used, which demonstrates the effectiveness of the introduced data augmentation methods.

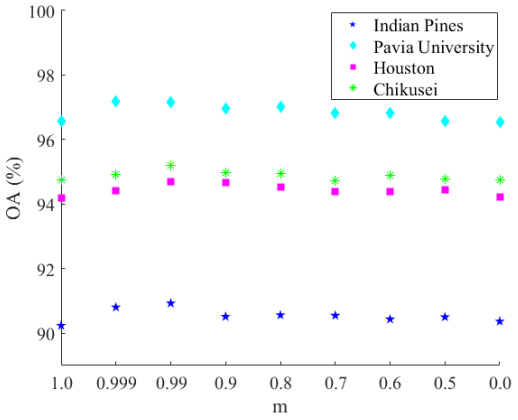


Figure 21. The overall accuracy of SCL with different values of momentum coefficient m.

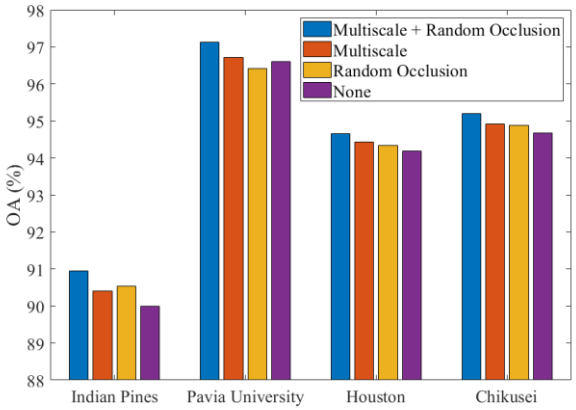


Figure 22. The overall accuracy of SCL using different data augmentation techniques.

Figure 23 shows the SCL classification overall accuracies on the four datasets over different values of p and S. From (a), one can see that a smaller value of S is more likely to gain a better OA, and the best values of p and S for the Indian Pines dataset are 0.6 and 19, respectively. For the Pavia University dataset, the best classification performance is obtained when $p = 0.2$ and $S = 19$. A smaller value of S (e.g., 19) seems to yield higher classification accuracy when the value of p is small (e.g., 0.2, 0.4, and 0.6), whereas a smaller S is more suitable if the value of p is set to be higher, e.g., 0.8. It can be seen that the best values of p and S for the Houston dataset are 0.6 and 19, respectively, and the best values of p and S for the Chikusei dataset are 0.8 and 23.

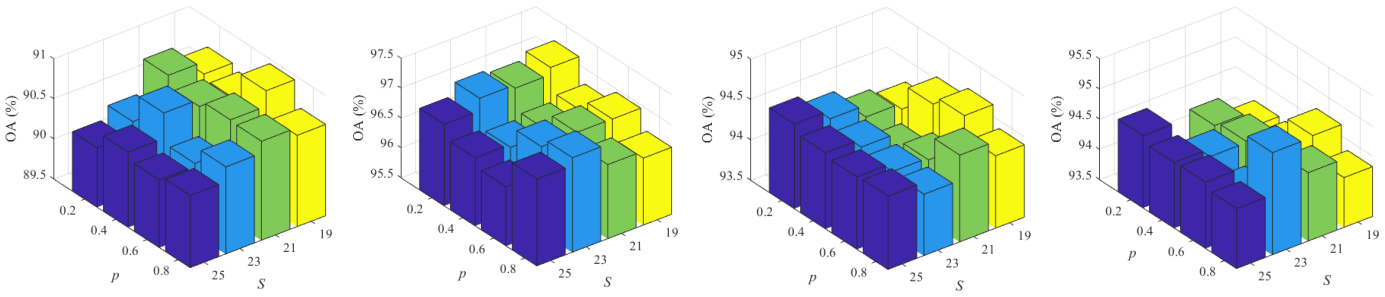


Figure 23. Classification accuracies over different values of p and S: (**a**) Indian Pines dataset; (**b**) Pavia University dataset; (**c**) Houston dataset; (**d**) Chikusei dataset.

(4) Experiments of MSR Loss: As mentioned above, the classification accuracy gains obtained by using multi-level similarity regularization can be seen in Tables 6–9.

By observing the distribution of the similarity scores for SCL, we find that most of the similarity scores are concentrated between $[-0.3, 0.3]$; so, we set the initial regularization levels as $\{-0.2, 0, 0.2\}$. Figure 24 illustrates the difference between SCL and SCL–MSR on the four datasets, with respect to the distribution of the similarity scores after training. The distribution of the similarity scores is reshaped by the regulation levels. It is worth noting that the levels can be learned by SCL–MSR to find proper values, just as (b) shows.

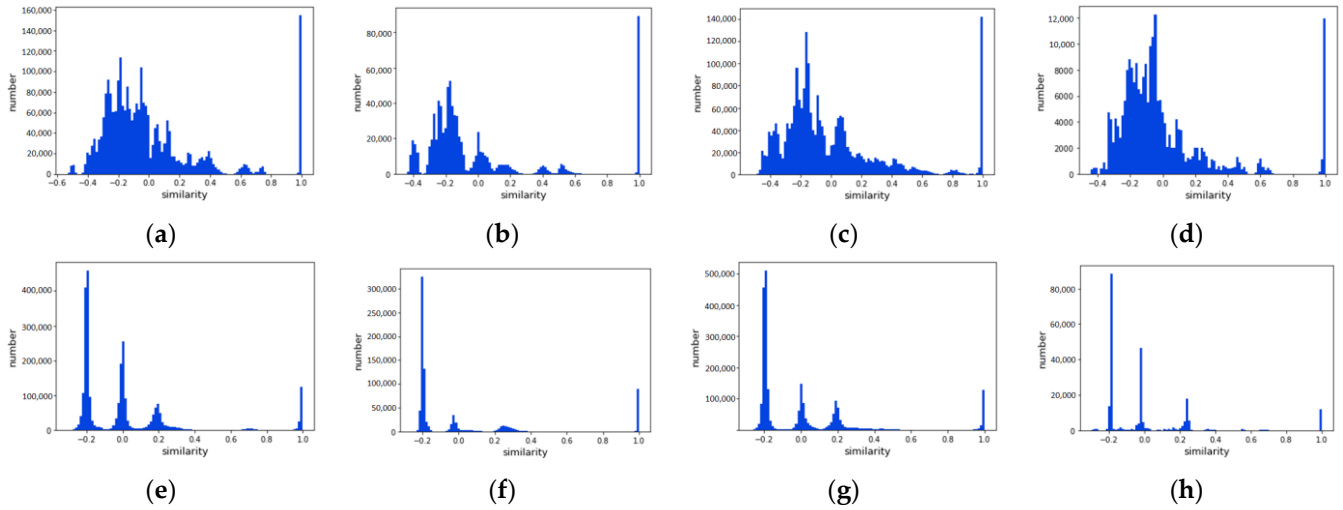


Figure 24. Distribution of similarity when $T = 1.0$, $h = 25$. (**a**–**d**) SCL over the Indian Pines dataset, the Pavia University dataset, the Houston dataset, and the Chikusei dataset; (**e**–**h**) SCL–MSR over the Indian Pines dataset, the Pavia University dataset, the Houston dataset, and the Chikusei dataset.

(5) Experiments of spectral unmixing and resolution: As this study is based on pixel-based classification, the effect of spectral unmixing and resolution is analyzed here. As is shown in [11–13], the pixel-based classification relies on the representation ability of the pixels, and it is necessary to take into account the spectrum mixing and resolution, which has also been demonstrated by the following designed experiments, including classification after spectral unmixing and classification when the spatial resolution is poor.

(i) Classification after spectral unmixing. We treat the Indian Pines, Pavia University, Houston, and Chikusei datasets as spectrally mixed data and use the spectral unmixing technique to process these datasets, following [56]. Then, the processed datasets are classified using different methods for evaluation.

The spectral unmixing decomposes each HSI spectrum as a mixture of endmembers with their proportions. The unmixing method in [56] considers a generalized spectral unmixing model that is a combination of a linear mixing model and a nonlinear model, given by:

$$\mathbf{x}_i = \alpha \mathbf{M}\mathbf{a}_i + (1-\alpha)\Phi(\mathbf{M}, \mathbf{a}_i) + \mathbf{n}, \tag{13}$$

where $\mathbf{x}_i$ is the i-th pixel sample containing B bands, and $\mathbf{M}$ denotes the endmember matrix. The abundance vector associated with $\mathbf{x}_i$ is denoted as $\mathbf{a}_i$. Φ is a nonlinear function that characterizes the interactions between the endmembers, and λ is a hyper-parameter balancing the weights of the linear part and the nonlinear part. An encoder–decoder architecture is designed based on Equation (13) to simulate the mixing procedure, and it is trained to estimate the abundance representations from the HSI.

Table 10 shows the classification results after spectral unmixing. From Table 10, it can be seen that the unmixing is helpful for classification. The overall accuracies of the proposed method and the other state-of-the-art methods are all higher than those using the original datasets. For example, SCL gains 0.49 percentage points, 0.57 percentage

points, and 0.0058 in terms of OA, AA, and K on the Indian Pines dataset after using spectral unmixing. In addition, the proposed methods still achieve better classification performances when compared with the other methods.

Table 10. Classification results after spectral unmixing.

		CNN	SiamSCL	SSRN	DBDA	FDSSC	SCL	SCL–MSR
Indian Pines	OA (%)	90.52 ± 1.25	90.92 ± 1.39	90.24 ± 1.56	90.92 ± 2.16	90.68 ± 1.89	91.44 ± 1.66	91.93 ± 1.24
	AA (%)	94.47 ± 0.72	94.78 ± 1.02	85.64 ± 1.28	84.08 ± 2.12	85.69 ± 2.54	95.25 ± 0.79	95.51 ± 0.65
	K × 100	89.22 ± 1.41	89.66 ± 1.53	88.74 ± 1.72	89.62 ± 2.68	89.44 ± 2.46	90.27 ± 1.84	90.82 ± 1.41
Pavia University	OA (%)	96.04 ± 1.97	96.72 ± 2.21	95.41 ± 1.26	96.61 ± 1.96	96.82 ± 1.94	97.51 ± 2.00	97.88 ± 1.61
	AA (%)	96.98 ± 1.27	97.96 ± 1.25	93.96 ± 2.26	95.39 ± 2.16	96.24 ± 1.56	97.67 ± 1.21	97.86 ± 0.91
	K × 100	94.80 ± 2.57	95.61 ± 2.41	93.42 ± 1.94	95.21 ± 2.55	95.63 ± 2.43	96.72 ± 2.62	97.20 ± 2.11
Houston	OA (%)	94.00 ± 1.30	94.56 ± 1.12	93.89 ± 1.05	94.31 ± 1.23	94.56 ± 1.15	94.90 ± 0.85	95.19 ± 0.83
	AA (%)	94.67 ± 1.11	95.15 ± 0.98	94.62 ± 1.25	94.83 ± 0.95	95.03 ± 0.98	95.42 ± 0.78	95.63 ± 0.64
	K × 100	93.52 ± 1.41	94.01 ± 1.26	93.28 ± 1.45	93.75 ± 1.12	94.01 ± 1.01	94.52 ± 0.92	94.75 ± 0.82
Chikusei	OA (%)	94.37 ± 2.02	94.86 ± 1.78	92.30 ± 3.26	94.89 ± 2.12	93.82 ± 2.86	95.54 ± 1.85	95.95 ± 1.78
	AA (%)	94.02 ± 1.60	94.02 ± 1.32	85.07 ± 2.73	88.92 ± 1.83	86.68 ± 2.52	94.58 ± 1.46	94.98 ± 1.68
	K × 100	93.50 ± 2.81	94.12 ± 1.98	90.86 ± 3.65	94.05 ± 2.49	92.43 ± 3.15	94.86 ± 2.14	95.16 ± 1.93

(ii) Classification when spatial resolution is poor. To obtain datasets whose spatial resolutions are poor, we downsample the original image (i.e., remove all odd rows and columns) and then resize them to the original sizes using bilinear interpolation. The obtained resolution of the hyperspectral images will be half of the original. Figure 25 shows the false color maps of the different resolutions for the Indian Pines and the Pavia University datasets.



(a) **(b)** **(c)** **(d)**

Figure 25. False color maps of the Indian Pines and Pavia University datasets. (**a**,**c**): original datasets; (**b**,**d**): downsampled datasets.

Table 11 shows the classification results on the four HSI datasets when the spatial resolution is poor. From Table 3, it can be seen that in the Pavia, Houston, and Chikusei datasets, the accuracies decrease when we change the resolution. For example, the OA of SCL when the spatial resolution is poor is lower than original resolution by 1.82 percentage points, and the Houston dataset suffers the most among these datasets. However, better classification performance is obtained on the Indian Pines dataset. We infer that it is because the Indian Pines contains large areas of homogeneity, and the spatial structure is not complex. The process of downsampling plays a role in image smoothing. In addition, the Indian Pines dataset does not lose much spatial information but removes some spatial noise. However, the other three datasets contain rich spatial information, and the spatial resolution is vital for classification.

Table 11. Classification results when the spatial resolution is poor.

		CNN	SiamSCL	SSRN	DBDA	FDSSC	SCL	SCL–MSR
Indian Pines	OA (%)	90.00 ± 1.66	91.28 ± 1.62	90.12 ± 1.26	91.35 ± 1.52	91.24 ± 1.45	91.86 ± 1.53	92.19 ± 1.49
	AA (%)	94.23 ± 0.81	95.43 ± 0.83	86.26 ± 1.72	81.89 ± 3.35	81.96 ± 3.21	95.69 ± 0.89	95.89 ± 0.95
	K × 100	88.62 ± 1.86	89.65 ± 1.78	88.85 ± 1.43	90.51 ± 1.78	90.42 ± 1.65	91.25 ± 1.68	91.46 ± 1.54
Pavia University	OA (%)	94.13 ± 2.48	94.54 ± 2.15	90.00 ± 1.79	95.06 ± 1.58	93.18 ± 1.54	95.69 ± 1.86	96.01 ± 1.56
	AA (%)	95.83 ± 1.26	96.01 ± 1.12	86.97 ± 1.74	93.49 ± 1.58	90.37 ± 1.74	97.26 ± 0.95	97.54 ± 0.89
	K × 100	92.35 ± 3.16	92.65 ± 2.56	86.91 ± 0.02	93.52 ± 2.03	91.07 ± 1.96	95.19 ± 2.14	95.44 ± 1.94
Houston	OA (%)	86.58 ± 1.06	87.13 ± 1.27	80.62 ± 1.38	85.58 ± 0.96	85.33 ± 1.22	87.69 ± 1.12	87.95 ± 1.08
	AA (%)	88.42 ± 0.90	88.96 ± 1.15	81.97 ± 1.39	86.45 ± 0.98	86.26 ± 1.36	89.56 ± 1.02	89.84 ± 1.13
	K × 100	85.49 ± 1.14	85.89 ± 1.29	79.05 ± 1.50	84.42 ± 1.03	84.16 ± 1.31	86.72 ± 1.25	87.02 ± 1.18
Chikusei	OA (%)	93.06 ± 3.34	93.52 ± 2.58	89.43 ± 3.65	93.59 ± 2.91	91.26 ± 3.42	94.22 ± 3.25	94.66 ± 3.45
	AA (%)	93.61 ± 1.60	94.03 ± 1.23	82.15 ± 2.31	86.30 ± 3.51	84.29 ± 3.25	94.86 ± 1.78	95.02 ± 1.91
	K × 100	92.03 ± 3.83	92.60 ± 3.05	88.56 ± 4.02	92.64 ± 3.31	90.83 ± 3.89	93.34 ± 3.77	93.68 ± 3.89

It can be seen that the proposed methods still achieve better classification performance when compared with the other methods.

To sum up, from the unmixing experiment it can be seen that the unmixing technique is helpful for pixel-based hyperspectral image classification, and from the resolution experiment, it can be seen that the resolution has an important influence on the HSI classification performance. This indicates that it is better to develop unmixing and super-resolution techniques for pixel-based HSI classification to obtain better performance.

3.5. Algorithm Complexity

Table 12 shows the algorithm complexity for different classification methods. FLOPs is the abbreviation of floating point operations, which means the number of floating point operations needed for a given model. In addition, it is understood as the amount of calculation. Param. Means the number of parameters to be trained in a given model. The FLOPs and Param. Can be used to measure the complexity of a model.

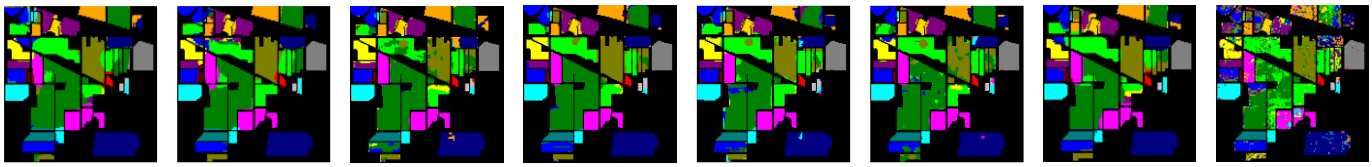
Table 12. The number of FLOPs and parameters.

	CNN	SiamSCL	SSRN	DBMA	DBDA	FDSSC	SCL	SCL–MSR
FLOPs	32.32M	96.96M	158.38M	245.59M	161.30M	265.0M	96.96M	96.96M
Param.	0.44M	0.44M	0.36M	0.61M	0.38M	1.227M	0.88M	0.88M

The proposed SCL and SCL–MSR are pre-training frameworks which use a vanilla CNN as a backbone. Their numbers of FLOPs and parameters are related to the CNN. Specifically, the SiamSCL, SCL, and SCL–MSR all adopt Siamese architecture and need fine-tuning so that their FLOPs are three times that of the CNN. However, SCL and SCL–MSR have twice as many parameters as CNN due to the momentum update module. It can be seen that the proposed SCL-based methods have fewer FLOPs when compared with SSRN, DBMA, DBDA, and FDSSC, and the SCL and SCL–MSR have fewer parameters than FDSSC. Taking the algorithm complexity and accuracy into consideration, the proposed methods are thought to be competitive.

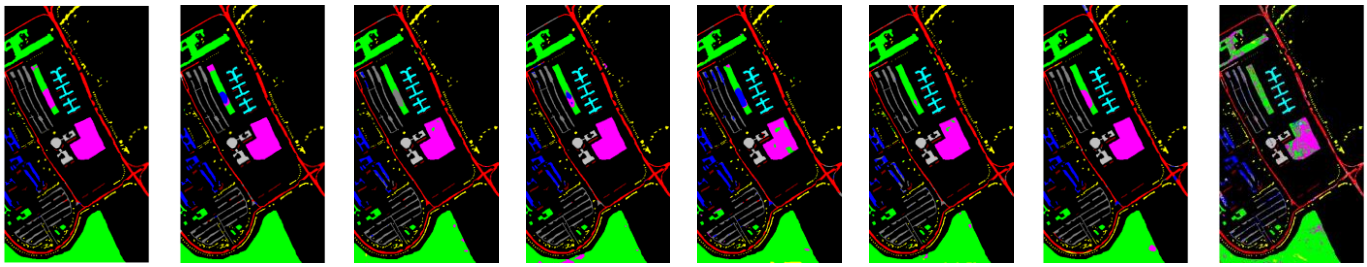
3.6. Classification Maps of Different Classification Methods

Figures 26–29 show the classification maps of the different methods for the four datasets. The classification performance obtained by the proposed methods is better than with other methods, which can be clearly seen from the classification maps.



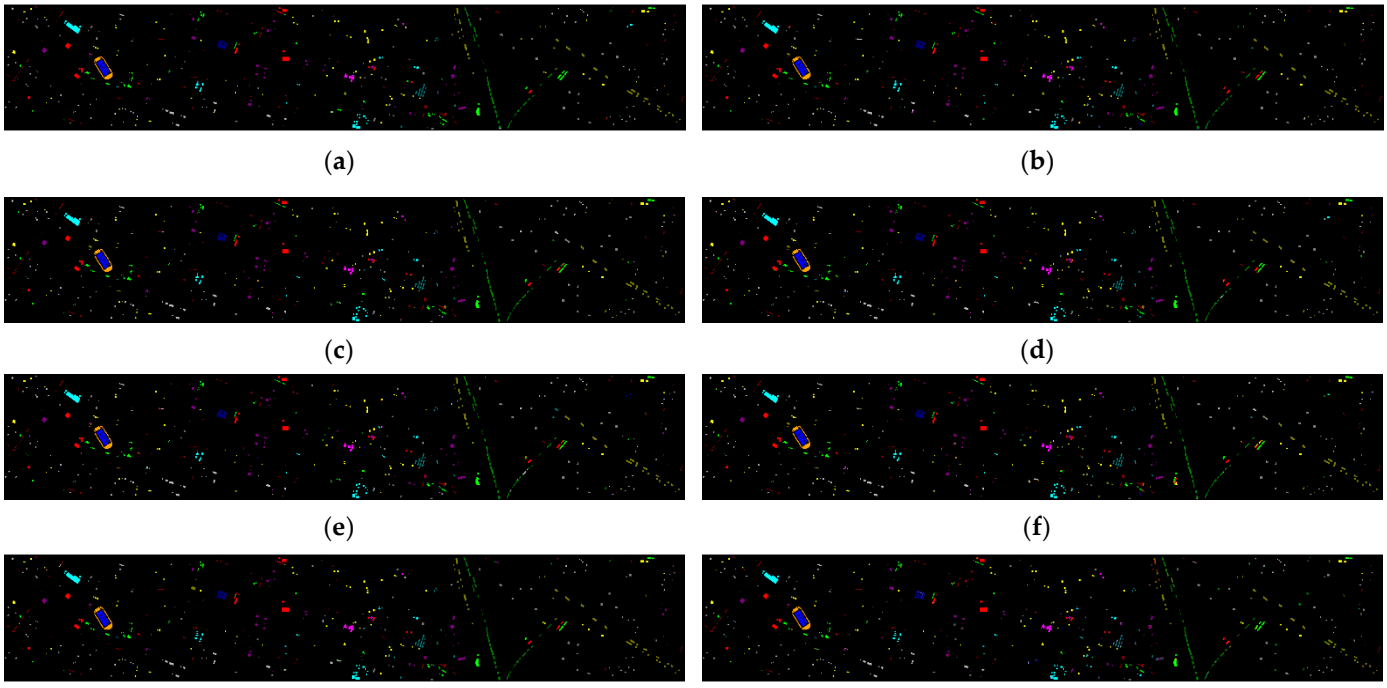
(**a**) (**b**) (**c**) (**d**) (**e**) (**f**) (**g**) (**h**)

Figure 26. Classification maps using different methods on the Indian Pines dataset: (**a**) SCL–MSR; (**b**) SCL; (**c**) FDSCC; (**d**) DBDA; (**e**) DBMA; (**f**) SSRN; (**g**) SiamSCL; (**h**) EMP–SVM.



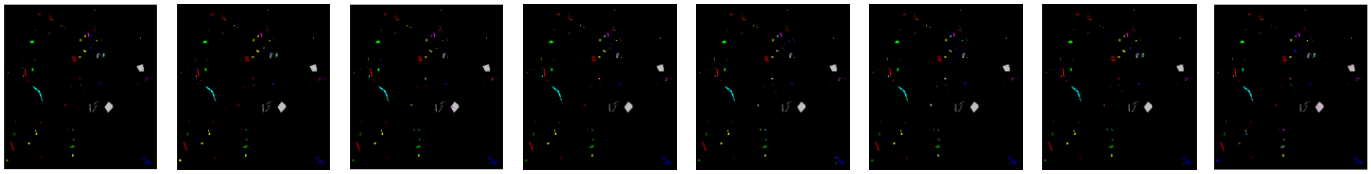
(**a**) (**b**) (**c**) (**d**) (**e**) (**f**) (**g**) (**h**)

Figure 27. Classification maps using different methods on the Pavia University dataset: (**a**) SCL–MSR; (**b**) SCL; (**c**) FDSCC; (**d**) DBDA; (**e**) DBMA; (**f**) SSRN; (**g**) SiamSCL; (**h**) EMP–SVM.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**) (**g**) (**h**)

Figure 28. Classification maps using different methods on the Houston dataset: (**a**) SCL–MSR; (**b**) SCL; (**c**) FDSCC; (**d**) DBDA; (**e**) DBMA; (**f**) SSRN; (**g**) SiamSCL; (**h**) EMP–SVM.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**) (**g**) (**h**)

Figure 29. Classification maps using different methods on the Chikusei dataset: (**a**) SCL–MSR; (**b**) SCL; (**c**) FDSCC; (**d**) DBDA; (**e**) DBMA; (**f**) SSRN; (**g**) SiamSCL; (**h**) EMP–SVM.

4. Conclusions

In this study, a contrastive learning-based supervised pre-training framework is proposed for hyperspectral image classification with limited training samples; it includes data augmentation methods for HSI, a queue, and a momentum update scheme for supervised pre-training. Additionally, the multilevel regularization method is combined with SCL for better performance. Verification experiments were conducted on the four widely used datasets (the Indian Pines, Pavia University, Houston datasets, and Chikusei), and the following conclusions can be drawn from the results:

(1) According to the comparative analysis of the classification results, the proposed methods outperform some existing state-of-the-art HSI classification methods in terms of OA, AA, and K.
(2) The combination of the two data augmentation methods, MA and ROA, can improve the classification performance of SCL for HSI classification. The experimental results show the effect of each method.
(3) The experimental results demonstrate that the queue and the momentum update scheme for SCL are effective for improving the classification accuracy.
(4) The use of MSR regularizes the training procedure of SCL and improves the generalization performance for HSI classification.

This research suggests areas of further exploration in the field of HSI classification. Future work will extend the supervised contrastive learning-based HSI classification to unsupervised and semi-supervised settings.

Author Contributions: Conceptualization: Y.C.; methodology: L.H. and Y.C.; writing—original draft preparation: L.H., Y.C., X.H., and P.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of China under the Grant 61971164 and the Grant U20B2041.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The Houston dataset is available at: https://hyperspectral.ee.uh.edu/ (accessed on 1 September 2020). The Indian Pines and Pavia University datasets are available at: http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes (accessed on 1 September 2020). The Chikusei dataset is available at: https://naotoyokoya.com/Download.html (accessed on 1 September 2020).

Acknowledgments: The authors would like to thank the Hyperspectral Image Analysis group and the NSF Funded Center for Airborne Laser Mapping (NCALM) at the University of Houston for providing the datasets used in this study and the IEEE GRSS Data Fusion Technical Committee for organizing the 2013 Data Fusion Contest. The authors gratefully acknowledge the Space Application Laboratory, Department of Advanced Interdisciplinary Studies, the University of Tokyo, for providing the Chikusei data.

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

References

1. Plaza, A.; Benediktsson, J.A.; Boardman, J.W.; Brazile, J.; Bruzzone, L.; Camps-Valls, G.; Chanussot, J.; Fauvel, M.; Gamba, P.; Gualtieri, A. Recent advances in techniques for hyperspectral image processing. *Remote Sens. Environ.* **2009**, *113*, S110–S122. [CrossRef]
2. Ghamisi, P.; Yokoya, N.; Li, J.; Liao, W.; Liu, S.; Plaza, J.; Rasti, B.; Plaza, A. Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 37–78. [CrossRef]
3. Li, H.; Li, Z.; Dong, W.; Cao, X.; Wen, Z.; Xiao, R.; Wei, Y.; Zeng, H.; Ma, X. An automatic approach for detecting seedlings per hill of machine-transplanted hybrid rice utilizing machine vision. *Comput. Electron. Agric.* **2021**, *185*, 106178. [CrossRef]
4. Lee, M.-K.; Golzarian, M.R.; Kim, I. A new color index for vegetation segmentation and classification. *Precis. Agric.* **2021**, *22*, 179–204. [CrossRef]
5. Benediktsson, J.A.; Ghamisi, P. *Spectral-Spatial Classification of Hyperspectral Remote Sensing Images*; Artech House: London, UK, 2015.
6. Chang, C.-I. *Hyperspectral Data Exploitation: Theory and Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2007.
7. Bioucas-Dias, J.M.; Plaza, A.; Camps-Valls, G.; Scheunders, P.; Nasrabadi, N.; Chanussot, J. Hyperspectral remote sensing data analysis and future challenges. *IEEE Geosci. Remote Sens. Mag.* **2013**, *1*, 6–36. [CrossRef]
8. Ghamisi, P.; Plaza, J.; Chen, Y.; Li, J.; Plaza, A.J. Advanced spectral classifiers for hyperspectral images: A review. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–32. [CrossRef]
9. Kuching, S. The performance of maximum likelihood, spectral angle mapper, neural network and decision tree classifiers in hyperspectral image analysis. *J. Comput. Sci.* **2007**, *3*, 419–423.
10. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [CrossRef]
11. Cimtay, Y.; Özbay, B.; Yilmaz, G.; Bozdemir, E. A new vegetation index in short-wave infrared region of electromagnetic spectrum. *IEEE Access* **2021**, *9*, 148535–148545. [CrossRef]
12. Heylen, R.; Parente, M.; Gader, P. A review of nonlinear hyperspectral unmixing methods. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 1844–1868. [CrossRef]
13. Çimtay, Y.; İlk, H.G. A novel bilinear unmixing approach for reconsideration of subpixel classification of land cover. *Comput. Electron. Agric.* **2018**, *152*, 126–140. [CrossRef]
14. Benediktsson, J.A.; Palmason, J.A.; Sveinsson, J.R. Classification of hyperspectral data from urban areas based on extended morphological profiles. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 480–491. [CrossRef]
15. Xia, J.; Dalla Mura, M.; Chanussot, J.; Du, P.; He, X. Random subspace ensembles for hyperspectral image classification with extended morphological attribute profiles. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 4768–4786. [CrossRef]
16. Fang, L.; He, N.; Li, S.; Ghamisi, P.; Benediktsson, J.A. Extinction profiles fusion for hyperspectral images classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 1803–1815. [CrossRef]
17. 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.
18. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [CrossRef]
19. Zhang, B.; Xiong, D.; Su, J. Neural machine translation with deep attention. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *42*, 154–163. [CrossRef]
20. Ma, X.; Wang, H.; Geng, J. Spectral–spatial classification of hyperspectral image based on deep auto-encoder. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 4073–4085. [CrossRef]
21. Tu, Y.-H.; Du, J.; Lee, C.-H. Speech enhancement based on teacher–student deep learning using improved speech presence probability for noise-robust speech recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2019**, *27*, 2080–2091. [CrossRef]
22. Chen, Y.; Zhao, X.; Jia, X. Spectral–spatial classification of hyperspectral data based on deep belief network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 2381–2392. [CrossRef]
23. He, N.; Paoletti, M.E.; Haut, J.M.; Fang, L.; Li, S.; Plaza, A.; Plaza, J. Feature extraction with multiscale covariance maps for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 755–769. [CrossRef]
24. Li, S.; Song, W.; Fang, L.; Chen, Y.; Ghamisi, P.; Benediktsson, J.A. Deep learning for hyperspectral image classification: An overview. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6690–6709. [CrossRef]
25. Yu, C.; Han, R.; Song, M.; Liu, C.; Chang, C.-I. A simplified 2D-3D CNN architecture for hyperspectral image classification based on spatial–spectral fusion. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 2485–2501. [CrossRef]
26. Li, X.; Ding, M.; Pižurica, A. Deep feature fusion via two-stream convolutional neural network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 2615–2629. [CrossRef]
27. Alam, F.I.; Zhou, J.; Liew, A.W.-C.; Jia, X.; Chanussot, J.; Gao, Y. Conditional random field and deep feature learning for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 1612–1628. [CrossRef]
28. Bhatti, U.A.; Yu, Z.; Chanussot, J.; Zeeshan, Z.; Yuan, L.; Luo, W.; Nawaz, S.A.; Bhatti, M.A.; Ain, Q.U.; Mehmood, A. Local Similarity-Based Spatial–Spectral Fusion Hyperspectral Image Classification with Deep CNN and Gabor Filtering. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5514215. [CrossRef]

29. Paoletti, M.E.; Haut, J.M.; Fernandez-Beltran, R.; Plaza, J.; Plaza, A.J.; Pla, F. Deep pyramidal residual networks for spectral–spatial hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 740–754. [CrossRef]
30. 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]
31. 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]
32. Yu, C.; Han, R.; Song, M.; Liu, C.; Chang, C.-I. Feedback attention-based dense CNN for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5501916. [CrossRef]
33. Jiang, Y.; Li, Y.; Zhang, H. Hyperspectral image classification based on 3-D separable ResNet and transfer learning. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1949–1953. [CrossRef]
34. Lv, Q.; Feng, W.; Quan, Y.; Dauphin, G.; Gao, L.; Xing, M. Enhanced-random-feature-subspace-based ensemble CNN for the imbalanced hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 3988–3999. [CrossRef]
35. Chen, Y.; Zhu, L.; Ghamisi, P.; Jia, X.; Li, G.; Tang, L. Hyperspectral images classification with Gabor filtering and convolutional neural network. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2355–2359. [CrossRef]
36. Zhang, H.; Li, Y.; Jiang, Y.; Wang, P.; Shen, Q.; Shen, C. Hyperspectral classification based on lightweight 3-D-CNN with transfer learning. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5813–5828. [CrossRef]
37. Aptoula, E.; Ozdemir, M.C.; Yanikoglu, B. Deep learning with attribute profiles for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 1970–1974. [CrossRef]
38. Roy, S.K.; Mondal, R.; Paoletti, M.E.; Haut, J.M.; Plaza, A. Morphological convolutional neural networks for hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 8689–8702. [CrossRef]
39. He, X.; Chen, Y.; Lin, Z. Spatial-spectral transformer for hyperspectral image classification. *Remote Sens.* **2021**, *13*, 498. [CrossRef]
40. Rao, M.; Tang, P.; Zhang, Z. Spatial–spectral relation network for hyperspectral image classification with limited training samples. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 5086–5100. [CrossRef]
41. Yue, J.; Zhu, D.; Fang, L.; Ghamisi, P.; Wang, Y. Adaptive spatial pyramid constraint for hyperspectral image classification with limited training samples. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5512914. [CrossRef]
42. Fang, L.; Zhao, W.; He, N.; Zhu, J. Multiscale CNNs ensemble based self-learning for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 1593–1597. [CrossRef]
43. Zhou, S.; Xue, Z.; Du, P. Semisupervised stacked autoencoder with cotraining for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 3813–3826. [CrossRef]
44. Zhan, Y.; Hu, D.; Wang, Y.; Yu, X. Semisupervised hyperspectral image classification based on generative adversarial networks. *IEEE Geosci. Remote Sens. Lett.* **2017**, *15*, 212–216. [CrossRef]
45. Li, W.; Wu, G.; Zhang, F.; Du, Q. Hyperspectral image classification using deep pixel-pair features. *IEEE Trans. Geosci. Remote Sens.* **2016**, *55*, 844–853. [CrossRef]
46. Liu, B.; Yu, X.; Zhang, P.; Yu, A.; Fu, Q.; Wei, X. Supervised deep feature extraction for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 1909–1921. [CrossRef]
47. Haut, J.M.; Paoletti, M.E.; Plaza, J.; Plaza, A.; Li, J. Hyperspectral image classification using random occlusion data augmentation. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1751–1755. [CrossRef]
48. Zhao, L.; Luo, W.; Liao, Q.; Chen, S.; Wu, J. Hyperspectral Image Classification with Contrastive Self-Supervised Learning Under Limited Labeled Samples. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 6008205. [CrossRef]
49. Hou, S.; Shi, H.; Cao, X.; Zhang, X.; Jiao, L. Hyperspectral Imagery Classification Based on Contrastive Learning. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5521213. [CrossRef]
50. Zhu, M.; Fan, J.; Yang, Q.; Chen, T. SC-EADNet: A Self-Supervised Contrastive Efficient Asymmetric Dilated Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5519517. [CrossRef]
51. Debes, C.; Merentitis, A.; Heremans, R.; Hahn, J.; Frangiadakis, N.; van Kasteren, T.; Liao, W.; Bellens, R.; Pižurica, A.; Gautama, S. Hyperspectral and LiDAR data fusion: Outcome of the 2013 GRSS data fusion contest. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2405–2418. [CrossRef]
52. Yokoya, N.; Iwasaki, A. *Airborne Hyperspectral Data over Chikusei*; SAL-2016-05-27; Space Application Laboratory, University of Tokyo: Tokyo, Japan, 2016; p. 5.
53. 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]
54. 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]
55. Wang, Y.; Ma, X.; Chen, Z.; Luo, Y.; Yi, J.; Bailey, J. Symmetric cross entropy for robust learning with noisy labels. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 322–330.
56. Guo, A.J.; Zhu, F. Improving deep hyperspectral image classification performance with spectral unmixing. *Signal Process.* **2021**, *183*, 107949. [CrossRef]