remote sensing

Article

Unlocking the Potential of Data Augmentation in Contrastive Learning for Hyperspectral Image Classification

Jinhui Li, Xiaorun Li and Yunfeng Yan *

College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China; 12010068@zju.edu.cn (J.L.); lxyly@zju.edu.cn (X.L.)
* Correspondence: 21210004@zju.edu.cn

Abstract: Despite the rapid development of deep learning in hyperspectral image classification (HSIC), most models require a large amount of labeled data, which are both time-consuming and laborious to obtain. However, contrastive learning can extract spatial–spectral features from samples without labels, which helps to solve the above problem. Our focus is on optimizing the contrastive learning process and improving feature extraction from all samples. In this study, we propose the Unlocking-the-Potential-of-Data-Augmentation (UPDA) strategy, which involves adding superior data augmentation methods to enhance the representation of features extracted by contrastive learning. Specifically, we introduce three augmentation methods—band erasure, gradient mask, and random occlusion—to the Bootstrap-Your-Own-Latent (BYOL) structure. Our experimental results demonstrate that our method can effectively improve feature representation and thus improve classification accuracy. Additionally, we conduct ablation experiments to explore the effectiveness of different data augmentation methods.

Keywords: data augmentation; band erasure; gradient mask; random occlusion; Bootstrap-Your-Own-Latent; hyperspectral image; spatial–spectral feature

Citation: Li, J.; Li, X.; Yan, Y. Unlocking the Potential of Data Augmentation in Contrastive Learning for Hyperspectral Image Classification. *Remote Sens.* **2023**, *15*, 3123. https://doi.org/10.3390/rs15123123

Academic Editor: Salah Bourennane

Received: 14 May 2023
Revised: 7 June 2023
Accepted: 8 June 2023
Published: 15 June 2023

1. Introduction

Hyperspectral images (HSI) have gained widespread use due to their ability to provide extensive spectral and spatial information [1]. With hundreds of bands, HSIs can distinguish surface materials based on their unique spectral characteristics with exceptional spectral resolution. This feature makes them highly valuable for various applications, including vegetation surveys, atmospheric research, military detection, environmental monitoring [2] and landcover classification [3]. HSIC is a key research area within the hyperspectral field and involves the classification of individual pixels based on the rich spectral information they contain. As hardware technology continues to improve, the spatial resolution of hyperspectral sensors also increases, allowing for the incorporation of spatial information from surrounding pixels in classification efforts. Currently, the combination of spectral and spatial features is the primary approach in the HSIC field [4].

The abundance of bands in HSI presents a significant challenge in classification. Processing such large amounts of data directly without reduction would require a network of immense scale and huge computational memory. Furthermore, high spectral resolution creates spectral redundancy, which can be addressed through dimensionality reduction techniques that preserve critical information while reducing data size. Feature extraction is a widely used method for reducing data dimensions in HSI by extracting or sorting effective features for subsequent use. Common methods for feature extraction include principal component analysis (PCA) [5], independent component analysis (ICA) [6], linear discriminant analysis (LDA) [7], multidimensional scaling (MDS) [8], etc. These algorithms are still widely used as preprocessing methods due to their simplicity and effectiveness. With the increasing maturity of deep learning algorithms, more sophisticated algorithms are being developed to extract features from HSIs. Presently, the prevalent classification method involves using supervised or unsupervised feature extraction algorithms to extract spectral or spatial–spectral features, followed by classifier training using the extracted features.

The earlier developed feature extraction algorithms were based on supervised deep learning. In supervised learning, convolutional neural networks (CNNs) play an crucial role, evolving from one-dimensional CNNs [9] that only extract spectral features to two-dimensional and three-dimensional CNNs [10] that extract both spatial and spectral information. Roy et al. proposed the HybridSN network, which combines 2D and 3D convolutions to further enhance classification accuracy [11]. Zhong et al. introduced the classic residual network into the hyperspectral domain and designed the SSRN network [12]. Zhong et al. combined attention mechanism and CNNs [13]. Apart from CNNs, deep recurrent neural networks (DRNN) [14], deep feed-forward networks (DFFN), and other networks have also achieved promising results in HSIC.

However, supervised learning often heavily relies on labeled data, necessitating a sufficient number of labeled samples to achieve optimal training results. In the case of HSIs, both data collection and labeling involve significant human and time costs. Consequently, in recent years, the focus of feature extraction algorithms has gradually shifted towards unsupervised deep learning. The fundamental difference between unsupervised learning and supervised learning lies in the fact that the training data of unsupervised learning are unlabeled, and samples are classified based on their similarities, reducing the distance between data of the same class and increasing the distance between data of different classes. Without the constraints of labels, unsupervised learning can unleash the potential of models, enabling them to autonomously discover and explore data, learn more latent features, and ultimately result in models with better robustness and generalization. Unsupervised learning can be divided into generative learning and discriminative learning. Generative models learn to model the underlying probability distribution of input data. They are trained on large amounts of data and leverage this information to synthesize new samples that resemble the original data. The most basic generative deep learning algorithms include autoencoders (AE) [15] and generative adversarial networks (GAN) [16]. Variants of AEs, such as the adversarial autoencoder (AAE) [17], variational autoencoder (VAE) [18], and masked autoencoder (MAE) [19], have been widely used for feature extraction in hyperspectral image analysis. GANs optimized with algorithms such as deep convolutional GAN (DCGAN) [20], information maximizing GAN (InfoGAN) [21], and multitask GAN [22] have also achieved remarkable results in HSIC.

Discriminative learning models the conditional probability and learns the optimal boundary between different classes. Contrastive learning is a typical discriminative learning algorithm in deep learning, which aims to acquire representations by contrasting positive and negative pairs in the latent space. Positive pairs are spatially close but spectrally similar patches, whereas negative pairs are either spectrally dissimilar or spatially distant patches. By minimizing the distance between positive pairs and maximizing the distance between negative pairs, the model learns to encode both spatial and spectral information in the latent space.

Contrastive learning has made rapid progress in recent years, and many variants such as Moco [23], SimCLR [24], BYOL [25], SwAV [26], and SimSiam [27] have been proposed and gradually applied in the field of hyperspectral data analysis [28]. These methods differ in their choice of contrastive loss, encoder architecture, and training strategy. However, they share a common goal of learning representations that capture the underlying structure of hyperspectral data. Furthermore, the key to contrastive learning is to prevent model collapse, which means that all data converge to the same constant solution after feature representation.

For contrastive learning, we can integrate additional optimization techniques to encourage the model to learn more representative features while ensuring that it does not collapse. In handling the spatial–spectral features of HSIs, spatial and spectral information are often combined into the same sample. Although this approach is simple and compensates for the lack of spectral information, directly inputting the entire sample cube into the model results in a significant amount of redundant information interfering with feature extraction. Some studies have separated spatial and spectral information into different samples and used cross-domain contrastive learning to extract them separately [28–30]. This approach can reduce a lot of redundant information but may also lead to the loss of valuable sub-key information. Coordinating the extraction of spatial and spectral information, preserving useful information as much as possible, reducing the interference of useless information, and increasing the model's attention to key information are essential for improving the efficiency of contrastive learning.

In this study, we incorporated multiple optimization strategies into contrastive learning to improve feature extraction. These strategies include band erasure (BE), random occlusion (RO) [31], and gradient mask (GM). The motivation of our study is to greatly improve the feature extraction effect and unlock the potential of the model by adding new data augmentation methods while the model remains unchanged. Experimental results have shown that each of these strategies can significantly improve feature performance and classification accuracy. Furthermore, when combined, they can lead to unexpected improvements.

In the following sections, we will first introduce the relevant algorithms' background knowledge (Section 2). Then, we will provide a detailed description of our proposed method's overall framework and each module (Section 3). Next, we will describe and analyze our comparative and ablation experiments (Section 4). Finally, we will present our conclusions (Section 5). The main contributions of this study are as follows:

(1) We propose the band erasure strategy to improve spectral features extracted by contrastive learning.
(2) We propose the gradient masking strategy to enhance the model's attention to key areas, reduce attention to edge positions, and minimize the interference of useless information on features.
(3) We propose the Unlocking-the-Potential-of-Data-Augmentation (UPDA) strategy, which involves adding superior data augmentation methods to improve the features extracted by contrastive learning. We used UPDA in the BYOL structure and improved the classification accuracy to a new level.

2. Related Work

2.1. Contrastive Learning

Contrastive learning is a type of self-supervised learning that involves constructing pairs of similar and dissimilar examples to learn a representation learning model. The goal is to learn a model that projects similar samples close together in a projection space, whereas dissimilar samples are projected far apart. Essential factors in contrastive learning include how to construct similar and dissimilar samples, how to design a representation learning model that adheres to the above principles, and how to prevent model collapse, which occurs when all data converge to a single constant solution after feature representation. Currently, there are many contrastive learning methods available, and they can be roughly categorized into those based on negative samples [24], contrastive clustering [26], asymmetric network structures [25,27], and redundancy elimination loss functions, depending on the approach used to prevent model collapse.

Bootstrap-Your-Own-Latent (BYOL) is a typical asymmetric structure-based approach, where the online network has an extra predictor compared to the target network. Furthermore, the two branches are connected by an asymmetric similarity loss. BYOL extracts features of samples by training the ability of online network to predict the output of target network, thereby learning the potential connections between positive sample pairs. Unlike other contrastive learning methods, BYOL only requires positive sample pairs, not negative ones, which makes the BYOL model more robust and generalizable.

He et al. proposed Simsiam [27] and analyzed the necessary factors to make the network not collapse. Its structure is similar to BYOL, retaining the predictor of the online network, but without EMA (exponential moving average). Simsiam proves that EMA is not necessary to prevent collapse but removing it will sacrifice part of the accuracy [32].

In this work, in order to achieve better classification accuracy, we continue to use the BYOL structure.

2.2. Data Augmentation

2.2.1. Normal Data Augmentation

Data augmentation is a widely used technique in contrastive learning that increases the diversity of the training data by creating new samples that are variations of the original data. This technique helps to reduce overfitting and improve the model's ability to generalize. In contrastive learning, data augmentation is typically applied to both the anchor and positive samples to create new pairs of samples for training.

Normal augmentation methods, such as random cropping, flipping, rotation, color jittering, and Gaussian noise injection, are often used in contrastive learning [33]. The augmented samples are paired with their corresponding original samples to form positive pairs for training, thereby increasing the number of positive pairs and enhancing the diversity of the training data. This can improve the performance of the contrastive learning model.

Although they can be used to process hyperspectral data, these normal augmentation methods were originally designed for RGB or grayscale images and do not take into account the unique characteristics of HSIs.

In [34], the author pointed out that in the existing contrastive learning, it is not ideal to use various data augmentation methods to map the original data to the same space and then perform various downstream tasks. Blindly using data augmentation methods may be harmful to the learned features. We believe that the choice of data augmentation method should be based on the specific downstream tasks and the shape of the data. To fully leverage the potential of contrastive learning in HSIC, it is necessary to develop new data augmentation techniques that are tailored to hyperspectral data.

2.2.2. Random Occlusion

The random occlusion (RO) technique is a data augmentation method that involves randomly masking or occluding a portion of the input data during training. This technique simulates missing or incomplete information in the input data and forces the model to learn robust features that can still accurately classify the data, even when certain regions are missing. Random occlusion can be applied to various types of input data, such as images, text, and audio. In image classification tasks, random occlusion can be applied by randomly masking a portion of the image with a black rectangle or by replacing a portion of the image with random noise. The size and location of the occluded region can also be randomized to increase the diversity of the training data. By using random occlusion as a data augmentation technique, the model can learn to be more robust to incomplete or missing data, which can improve its performance on real-world situations where the input data may be noisy or incomplete.

When we were thinking about how to extract more representative features, we tried to explain why traditional augmentation methods make sense in feature extraction. We suspected that when using color distortion, we can encourage models to pay more attention to features except color, and when using Gaussian noise, we can tell our models to not be obsessed with small, discrete morphological features. We guessed that if we occlude an uncritical area, the model will pay more attention to the rest. So, as long as we do not occlude the features at key positions, the model can extract better space features.

RO was first used in HSIC in [31], where Haut et al. applied this technique to a CNN network and demonstrated its effectiveness. In our previous work [35], we integrated random occlusion into the BYOL structure for HSIC and achieved significant improvements in classification accuracy across various datasets.

Based on previous experiments, we have proved that proper occlusion strategies can improve results effectively when applied to HSIC. For our experiments, when the occlusion value is set to 1, and the area set to 10% of the patch area, the network works best.

3. Method

In this section, we provide a detailed introduction to the UPDA method and its application in the BYOL structure. Specifically, we first present the overall framework of the proposed method, followed by a detailed explanation of the newly proposed data augmentation methods, band erasure and gradient mask. Finally, we describe in detail the process of contrastive learning training and training of a classifier using the extracted features.

3.1. Framework of the Method

Figure 1 illustrates the framework of the proposed method, which mainly consists of the UPDA process and the contrastive learning feature extraction process, with the former including data preprocessing.

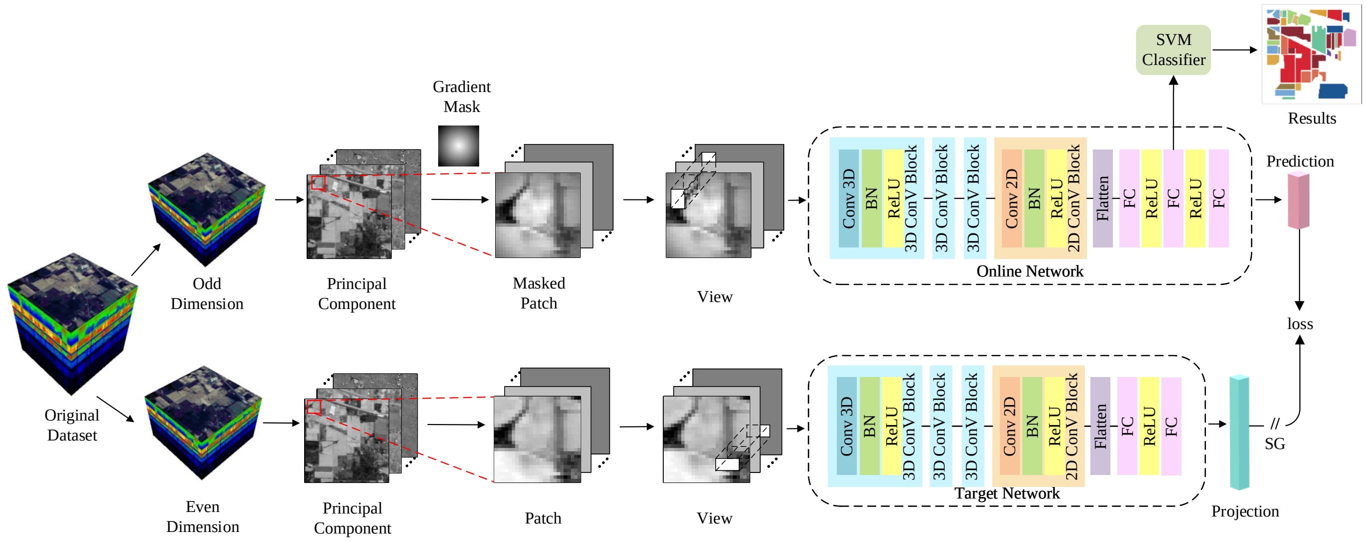


Figure 1. The framework of our method (UPDA + BYOL). The original dataset is band erased to obtain the data of the two branches. After PCA and sliding window segmentation, two series of patch groups are obtained. A pair of patches is used as an example in the figure. Gradient mask is applied to the patch of upper branch, and two views are obtained. Two views are regarded as a pair of positive samples and put into the backbone network of BYOL. SG denotes the stop gradient. After training, the features are saved to train the SVM classifier.

Let $\mathbf{X} \in \mathbb{R}^{h \times w \times c}$ be the original input hyperspectral image, where h and w are the height and width of the image, and c is the number of bands. First, the data are subjected to band erasure, where odd and even layers are erased separately to obtain two parts, $\mathbf{X}_0$ and $\mathbf{X}_1$, forming two branches.

PCA is then performed on each branch to reduce dimensionality, and the resulting samples are segmented into patches: $\mathbf{P}_0$ and $\mathbf{P}_1$. The patches in the upper branch are weighted using a gradient mask, whereas the patches in the lower branch are not processed. Then, both of them are randomly masked to obtain the views $\mathbf{V}_0$ and $\mathbf{V}_1$, which are input into the BYOL structure.

The upper branch of the BYOL structure is an online network, while the lower branch is a target network. The former has an additional linear layer, i.e., the predictor. The loss value is calculated based on the outputs of the upper and lower branches and used to update the network parameters.

After pretraining, the results of the projector are saved as features and input into the SVM classifier for training to complete classification or other downstream tasks. In this way, features are easily migrated between different machines, so that we can extract features on machines with high configuration and complete downstream tasks on any machine. For more detailed information, please refer to the following two subsections.

3.2. Unlocking the Potential of Data Augmentation

3.2.1. Band Erasure

The essence of band erasure is to divide the original HSI into two datasets with non-overlapping bands. In this process, odd and even layers are extracted at intervals, resulting in two datasets that are obtained by erasing half of the bands from the original data and have non-overlapping bands. Band erasure can improve the representativeness of subsequent feature extraction. On the one hand, because hyperspectral data have high spectral resolution and spectral information has a large redundancy, removing half of the bands will not lose critical spectral information and can reduce the interference of redundant information in each branch on feature extraction. On the other hand, the data in the two branches after band erasure are complementary. During the training process of contrastive learning, in order to make the loss function converge, the model tends to focus on shared information and reduce attention to unimportant details. This shared information is often the essential features. We have tried different erasing strategies, such as removing a string of continuous bands or leaving only 1/4 or 1/3 of the total number of bands; however, they are not as effective as the current erasure strategy.

The c spectral bands are divided into c_0 and c_1 bands, where $c = c_0 + c_1$, resulting in $\mathbf{X}_0 \in \mathbb{R}^{h \times w \times c_0}$ and $\mathbf{X}_1 \in \mathbb{R}^{h \times w \times c_1}$. PCA is then performed on each branch, and the first d principal component maps are selected. The value of d varies depending on the dataset, with 30 for the IP dataset and 15 for the PU and SA datasets. The principal component maps are then segmented into patches using a sliding window of size $s \times s$ with a stride of 1, resulting in $s \times s \times d$ cubes. Here, s is set to 25, and the edges are padded with zeros. Two sets of patches, $\mathbf{P}_0 = \{\mathbf{p}_{0,0}, \mathbf{p}_{0,1}, \cdots, \mathbf{p}_{0,N}\}(\mathbf{p}_{0,i} \in \mathbb{R}^{s \times s \times d})$ and $\mathbf{P}_1 = \{\mathbf{p}_{1,0}, \mathbf{p}_{1,1}, \cdots, \mathbf{p}_{1,N}\}(\mathbf{p}_{1,i} \in \mathbb{R}^{s \times s \times d})$, are obtained, where N is the number of valid pixels, and the corresponding labels are $\mathbf{Y} = \{y_0, y_1, \cdots, y_N\}(y_i \in N)$. The patches contain both spectral and spatial information, with the spectral information compressed by PCA. The class of the center pixel represents the class of the entire patch, so it can be said that the spatial and spectral information closer to the center pixel is more important.

3.2.2. Gradient Mask

The essence of a gradient mask is a weight matrix with the highest value at the center and decreasing values as they move away from the center. The size of the matrix is $s \times s$, which is the same as the patch size. The label of the patch is determined by the label of the center pixel, so we can infer that the closer to the center of the patch, the more critical the information. The purpose of setting a gradient mask is to reduce the model's attention to spatial and spectral information at the patch edges and to increase its attention to the key information at the center. It can be said that this is a forced attention transfer method. Specifically, we set the weight of the center to 1 and the weight of the four vertices to 0 and interpolate the other values linearly based on their distance from the center. The weight calculation formula is as Equation (1).

$$\mathbf{mask}_{i,j} = 1 - \sqrt{\frac{(i - center)^2 + (j - center)^2}{2center^2}} \tag{1}$$

where (i, j) is the location of element in the gradient mask, $(center, center)$ is the location of central element, and $center = (s+1)/2$. **mask** is the resulting gradient mask.

The resulting mask is displayed as a grayscale image in Figure 2. The gradient mask is multiplied element-wise with patches $\mathbf{P}_0$ to obtain $\mathbf{P}_0{}'$, whereas P1 is not processed with the gradient mask. The specific processing process is multiplication of the mask with the corresponding position element of each layer of the patch to obtain a weighted new patch.

This ensures that the input samples of the upper and lower branches have small differences at the center and large differences at the edges. During the training process of contrastive learning, in order to achieve the goal of reducing the loss function of the upper and lower branches, the model gradually reduces its attention to the edges of the patch and focuses more on the key information at the center.

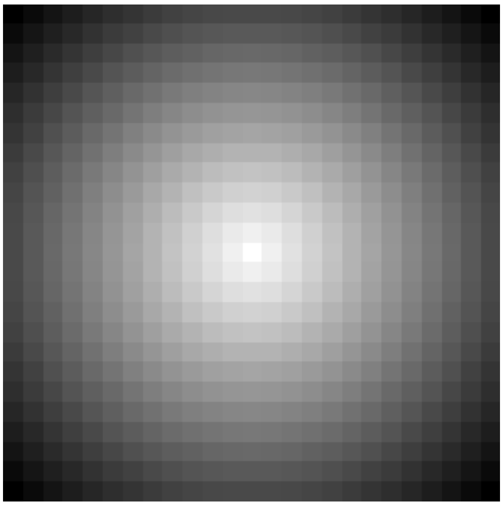


Figure 2. Visualization of the Gradient Mask.

3.3. Contrastive Learning

We put $(\mathbf{v}_i, \mathbf{v}_i{}')$ into the BYOL structure, where v_i is input to the online network, which is then encoded to obtain the representation $\mathbf{u}_i = f_{online}(\mathbf{v}_i)$. The representation $\mathbf{z}_i = g_{online}(\mathbf{u}_i)$ is then obtained through a projector, and the predicted result $\mathbf{k}_i = q_{online}(\mathbf{z}_i)$ is obtained through a predictor. On the other hand, $\mathbf{v}_i{}'$ is input to the target network, which is then encoded to obtain $\mathbf{u}_i{}' = f_{target}(\mathbf{v}_i{}')$, and the representation $\mathbf{z}_i{}' = g_{target}(\mathbf{u}_i{}')$ is obtained through a projector. Specifically, the encoder consists of three 3D convolutional layers, a convolutional layer, and a fully connected layer. After convolution, the output is flattened and then input to the fully connected layer. Each layer is followed by batch normalization and a ReLU activation function. The projector consists of a fully connected layer, batch normalization, and a ReLU activation function. The predictor is a narrower fully connected layer, also followed by batch normalization and a ReLU activation function.

The computation process of the online and target networks can be represented by Equations (2) and (3). For BYOL, its optimization objective is for the positive examples of the online network to approach the positive examples of the target network in the representation space. Therefore, we update the parameters of the online network using a loss function and update the parameters of the target network using an exponential moving average based on the parameters of the online network, with the update step controlled by the hyperparameter τ. The loss $\mathcal{L}$ is calculated based on the outputs of the two branches. First, we perform L2 normalization on $\mathbf{k}_i$ and $\mathbf{z}_i{}'$, $\overline{\mathbf{k}}_i = \mathbf{k}_i / \|\mathbf{k}_i\|_2$, $\overline{\mathbf{z}}' = \mathbf{z}_i{}' / \|\mathbf{z}_i{}'\|_2$ and then take the L2-normalization of their difference, as shown in Equation (4).

$$\mathbf{k}_i = q_{online}(g_{online}(f_{online}(\mathbf{v}_i))) \tag{2}$$

$$\mathbf{z}_i = g_{target}(f_{target}(\mathbf{v}_i{}')) \tag{3}$$

$$\mathcal{L} = \left\| \overline{\mathbf{k}}_i - \overline{\mathbf{z}}' \right\|_2^2 = 2 - 2 \cdot \frac{\langle \mathbf{k}_i, \mathbf{z}' \rangle}{\|\mathbf{k}_i\| \cdot \|\mathbf{z}'\|} \tag{4}$$

As the BYOL structure is asymmetric, in order to fully utilize the data, we exchange the input patches of the two branches, i.e., inputting $\mathbf{v}_i$ to the target network and $\mathbf{v}_i{}'$ to the online network, and calculate the loss function $\mathcal{L}'$. It can also greatly increase the efficiency of optimization. The loss function of BYOL is shown as Equation (5):

$$\mathcal{L}^{BYOL} = \mathcal{L} + \mathcal{L}' \tag{5}$$

$\mathcal{L}^{BYOL}$ is the symmetric loss and we update the parameters according to it. The process of parameter update can be represented as Equations (6) and (7):

$$W_{online} \leftarrow optimizer(W_{online}, \nabla_{W_{online}} \mathcal{L}^{BYOL}, \eta) \tag{6}$$

$$W_{target} \leftarrow \tau W_{target} + (1 - \tau) W_{online} \tag{7}$$

where W_{online} is the parameter of the online network, while W_{target} is the parameter of the target network. η is the learning rate and τ is the weight of parameter update. ← means assignment.

After reaching the specified number of iterations during training, we save the parameters of the encoder and projector and use the output of the projector as features inputted into the downstream task. In order to evaluate the ability of the contrastive learning model to extract features after introducing UPDA, we use the classical SVM classifier as the downstream task, with classification accuracy as the indicator to evaluate the performance of the features.

We split the sample patches into training and testing sets and use the features of the training set to train the classifier. The features of the test set are input to the trained classifier to obtain the classification result $\hat{\mathbf{Y}} = \{\hat{y}_0, \hat{y}_1, \cdots, \hat{y}_N\}(\hat{y}_i \in N)$. According to $\hat{\mathbf{Y}}$ and $\mathbf{Y}$, we calculate the accuracy of every class, the overall accuracy (OA), and the average accuracy (AA).

4. Experiments and Results

In this section, we report some classification experiments based on our method, and provide a detailed analysis of the experimental results.

4.1. Datasets

In the classification experiments, three publicly available datasets are used to show the superior performance of our method UPDA, including the Indian Pines (IP), Pavia University (PU), and Salina (SA) datasets.

The IP dataset was gathered by the AVIRIS sensor over the Indian Pines test site in northwestern Indiana. It includes 145 × 145 pixels and 224 spectral reflectance bands within the wavelength range of $0.4 \sim 2.5 \times 10^{-6}$ m. After eliminating bands covering water absorption regions, 200 bands remained, and the ground truth was composed of 16 classes. However, the dataset suffers from an uneven sample size issue, with the largest class having over 2000 samples and the smallest class having only 20 samples. Additionally, the dataset has a relatively low spatial resolution, which poses a challenge for classification. For more information on the IP dataset, please refer to Figure 3 and Table 1.

The PU dataset comprises of 610 × 340 pixels and 103 spectral bands within the wavelength range of $0.43 \sim 0.86 \times 10^{-6}$ m. The ground truth is composed of nine classes. This dataset is characterized by complex background pixels, small connected regions, and numerous discontinuous pixels. Moreover, its wavelength range is smaller than the other two datasets, resulting in less spectral information. For more information on the PU dataset, please refer to Figure 4 and Table 2.

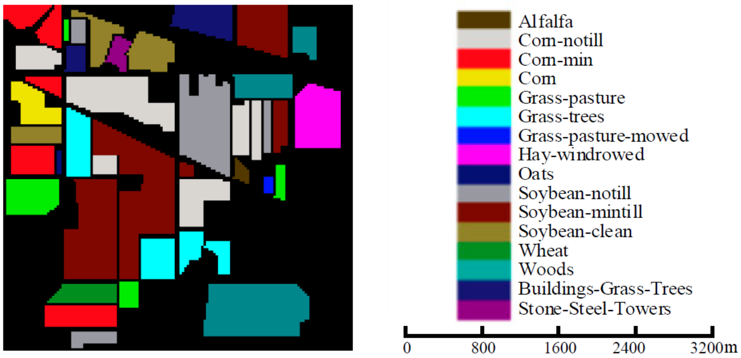


Figure 3. Groundtruth of the IP dataset.

Table 1. Details of the IP dataset.

IP Dataset			
Class Name	**No.**	**Number**	**Training**
Alfalfa	1	46	5
Corn-notill	2	1428	143
Corn-mintill	3	830	83
Corn	4	237	24
Grass-pasture	5	483	48
Grass-trees	6	730	2
Grass-pasture-mowed	7	28	3
Hay-windrowed	8	478	48
Oats	9	20	2
Soybean-notill	10	972	97
Soybean-mintill	11	2455	246
Soybean-clean	12	593	59
Wheat	13	205	21
Woods	14	1265	127
Buildings-Grass-Trees-Drives	15	386	39
Stone-Steel-Towers	16	93	9
Total		10,249	

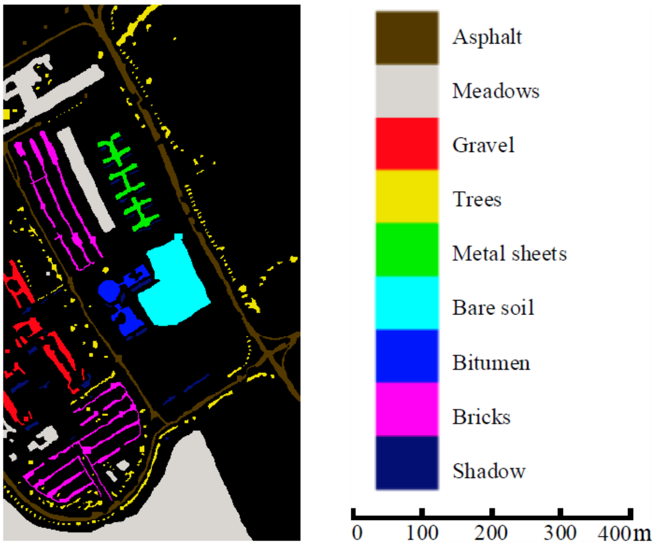


Figure 4. Groundtruth of the PU dataset.

Table 2. Details of the PU dataset.

PU Dataset			
Class Name	**No.**	**Number**	**Training**
Asphalt	1	6631	663
Meadows	2	18,649	1865
Gravel	3	2099	210
Trees	4	3064	306
Painted metal sheets	5	1345	135
Bare Soil	6	5029	503
Bitumen	7	1330	133
Self-Blocking Bricks	8	3682	368
Shadows	9	947	95
Total		42,776	

The SA dataset is an image with 512 × 217 pixels and 224 spectral bands within the wavelength range of 0.36 $\sim$ 2.5 $\times 10^{-6}$ m. After removing 20 water-absorbing bands, 204 spectral bands remain, and the ground truth is composed of 16 landcover classes. Compared to the other two datasets, the SA dataset is easier to classify due to its fewer obvious flaws such as uneven samples, low resolution, or complex background. For more information on the SA dataset, refer to Figure 5 and Table 3.

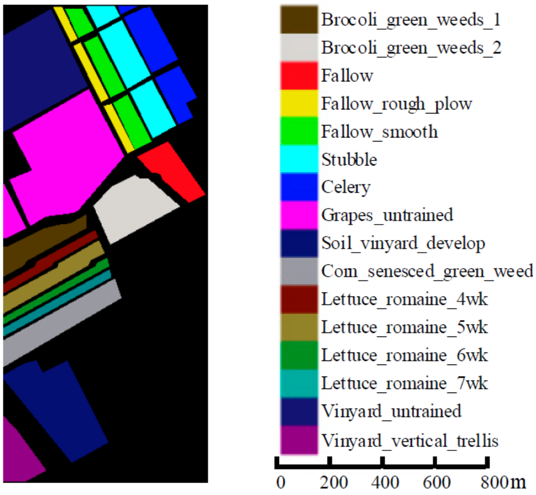


Figure 5. Groundtruth of the SA dataset.

4.2. Experimental Settings

We designed the backbone network for contrastive learning based on structure of HybridSN, which consists of 3D convolutional layers, and 2D convolutional layers, and the specific network shape of online network and target network are shown in Table 4.

We set the batch size to 128 and the size of the input patch s, to 25 × 25. Considering the low spatial resolution and uneven samples of the IP dataset, the principal component maps were taken as the first 30 for the IP dataset and the first 15 for the PU and SA datasets. In training the classifier, all available labeled pixels were divided into training and testing sets, 10% were randomly selected as training sets for the IP and PU datasets, and 5% for the SA dataset. The number of contrastive learning epochs was set to 50. The coefficient of EMA, τ, was set to 0.99. The selection of the above parameters is partly based on [11,25] and partly verified by experiments.

In order to verify whether the patch size s and the number of principal component graphs d are appropriate, we conducted some verification experiments on the IP and PU dataset. Indeed, when $s = 25$ and $d = 30$, the features extracted by BYOL make the classifier to achieve the highest classification accuracy on the IP dataset, while $d = 15$ on the PU dataset. The results are shown in Tables 5–7.

Table 3. Details of the SA dataset.

SA Dataset			
Class Name	**No.**	**Number**	**Training**
Brocoli_green_weeds_1	1	2009	100
Brocoli_green_weeds_2	2	3726	186
Fallow	3	1976	99
Fallow_rough_plow	4	1394	70
Fallow_smooth	5	2678	134
Stubble	6	3959	198
Celery	7	3579	179
Grapes_untrained	8	11,271	564
Soil_vinyard_develop	9	6203	310
Corn_senesced_green_weeds	10	3278	164
Lettuce_romaine_4wk	11	1068	53
Lettuce_romaine_5wk	12	1927	96
Lettuce_romaine_6wk	13	916	46
Lettuce_romaine_7wk	14	1070	54
Vinyard_untrained	15	7268	363
Vinyard_vertical_trellis	16	1807	90
Total		**54,129**	

Table 4. Structure of backbone network. The output shape of each layer is defined in PyTorch style. Batch size in the shape array is denoted by −1.

Layer Type	Online Network	Target Network
input	[−1,1,15/30,25,25]	[−1,1,15/30,25,25]
Conv3d	[−1,8,7,23,23]	[−1,8,7,23,23]
BatchNorm3d	[−1,8,7,23,23]	[−1,8,7,23,23]
ReLU	[−1,8,7,23,23]	[−1,8,7,23,23]
Conv3d	[−1,16,5,21,21]	[−1,16,5,21,21]
BatchNorm3d	[−1,16,5,21,21]	[−1,16,5,21,21]
ReLU	[−1,16,5,21,21]	[−1,16,5,21,21]
Conv3d	[−1,32,3,19,19]	[−1,32,3,19,19]
BatchNorm3d	[−1,32,3,19,19]	[−1,32,3,19,19]
ReLU	[−1,32,3,19,19]	[−1,32,3,19,19]
Conv2d	[−1,64,17,17]	[−1,64,17,17]
BatchNorm2d	[−1,64,17,17]	[−1,64,17,17]
ReLU	[−1,64,17,17]	[−1,64,17,17]
Linear	[−1,1024]	[−1,1024]
ReLU	[−1,1024]	[−1,1024]
Linear	[−1,128]	[−1,128]
ReLU	[−1,16]	/
Linear	[−1,16]	/

Table 5. Comparison of different patch sizes s in the IP dataset, setting d as 30, introducing only RO.

s	21	23	25	27
OA (%)	93.86	95.03	96.61	96.21

Table 6. Comparison of different number of principal component graphs d in the IP dataset, setting s as 25, introducing only RO.

d	15	20	25	30
OA (%)	95.11	96.00	95.94	96.61

Table 7. Comparison of different number of principal component graphs d in the PU dataset, setting s as 25, introducing only RO.

d	13	15	17	20
OA (%)	97.98	98.40	97.46	97.35

As for the SVM classifier, we chose "rbf" kernels, the penalty coefficient was set to 10 or 100, and the gamma was set to 0.01 or 0.001. They were obtained through experimental traversal.

The hardware configuration for the experiment was as follows: NVIDIA GeForce GTX 1660 SUPER GPU, Intel Core i7-10700 CPU, and 16 GB DRAM. The software configuration included Windows 10, Python 3.8, Pytorch 1.8.0, and Scikit-learn 0.24.2.

Two well-known metrics were used to evaluate the performance of different methods: accuracy (OA) and average accuracy (AA). All experiments were repeated three times, and the mean values of the results were recorded.

4.3. Classification Performance

To prove the superiority of our method, seven other unsupervised feature extraction methods were adopted as a baseline for comparison: PCA, tensor principal component analysis (TPCA) [36], stacked sparse autoencoder (SSAE) [37], unsupervised deep feature extraction (EPLS) [38], 3D convolutional autoencoder (3DCAE) [39], ContrastNet [40], and basic BYOL [25].

The classification results in the IP dataset are shown in Table 8. Our method obtains the best OA and AA values, and performs the best in 10 classes, particularly in classes 3, 4, 10, 11, 14, and 15. In classes with very few samples such as Alfalfa, Oats and Stone-Steel-Towers, our method leaves some room for improvement, and in classes with small inter-class gaps, "UPDA + BYOL" is good at distinguishing them, such as class Corn-nottill, Corn-mintill, Corn, Grass-pasture, Grass-trees, Grass-pasture-mowed, Soybean-nottill, Soybean-mintill, and Soybean-clean, which belong to different stages of the same crop. The introduction of UPDA increases the OA of BYOL by 2.29%, and increases the AA by 2.84%, which is a giant leap.

The classification results in the PU dataset are shown in Table 9. The OA and AA of our method are the highest and it obtains the best accuracy in six of nine classes. Particularly in classes 1, 2, 3, 6, 7, and 8, "UPDA + BYOL" performs much better than the other methods. Regrettably, in the class shadows "UPDA + BYOL" is not as good as the baselines. We guess that the ground cover of the class shadows is not single and our method tends to divide different material in the same class. In addition, our method outperforms the original BYOL in every class, OA, and AA, which demonstrates the effect of introducing UPDA. Furthermore, the introduction of UPDA increases the OA of BYOL by 1.51% and increases the AA by 3.46%.

The classification results in the SA dataset are shown in Table 10. "UPDA + BYOL" performs the best in OA, AA, and all classes except classes 1, 6, 11, and 12. After introducing UPDA, the OA, AA, and accuracy values in each class are all close to 100%, and the introduction of UPDA increases the OA of BYOL by 0.47% and increases the AA by 0.46%, which demonstrates that our method has excellent performance on ideal datasets.

The time consumption of our method and ContrastNet on three datasets are shown in Table 11. Time consumption of our method is significantly shorter than ContrastNet.

Overall, our method achieves the best results on the three datasets and greatly outperforms other comparative methods.

Table 8. Comparison in the Indian Pines dataset. 10% labeled samples were used for training, and the rest were used for testing. The highest accuracy values in every row are in bold. Some of the data in the table are quoted from [39].

No.	Class	PCA	TPCA	SSAE	EPLS	3DCAE	ContrastNet	BYOL	UPDA + BYOL
1	Alfalfa	39.02	60.97	56.25	58.72	90.48	85.37	**92.68**	90.24
2	Corn-notill	72.30	87.00	69.58	59.91	92.49	**97.15**	94.42	95.33
3	Corn-mintill	72.02	94.51	75.36	71.34	90.37	97.95	94.42	**98.26**
4	Corn	55.87	79.34	64.58	74.31	86.90	95.62	92.64	**97.34**
5	Grass-pasture	93.09	93.08	88.81	97.95	94.25	96.09	97.01	**98.85**
6	Grass-trees	94.67	96.34	87.00	96.44	97.07	96.80	98.22	**98.78**
7	Grass-pasture-mowed	80.00	76.00	90.00	54.02	91.26	70.67	98.67	**100.00**
8	Hay-windrowed	98.37	**99.76**	89.72	88.99	97.79	98.68	99.53	99.30
9	Oats	88.89	**100.00**	**100.00**	58.89	75.91	70.37	75.93	83.33
10	Soybean-nottill	74.49	79.51	77.19	73.10	87.34	97.45	94.36	**98.06**
11	Soybean-mintill	69.58	85.42	77.58	70.78	90.24	98.40	97.47	**99.17**
12	Soybean-clean	65.29	84.24	72.00	57.51	95.76	93.57	88.01	**96.25**
13	Wheat	98.37	98.91	87.80	**99.25**	97.49	95.32	97.30	96.22
14	Woods	91.39	98.06	93.48	95.07	96.03	98.51	98.39	**99.71**
15	Buildings-Grass-Trees-Drives	48.99	87.31	72.36	91.26	90.48	96.73	94.52	**98.66**
16	Stone-Steel-Towers	87.95	96.38	97.22	91.27	**98.82**	79.76	75.40	84.92
	AA (%)	76.89	89.31	81.18	77.43	92.04	91.78	93.06	**95.90**
	OA (%)	76.88	88.55	79.78	77.18	92.35	97.08	95.71	**98.00**

Table 9. Comparison in the University of Pavia dataset. A total of 10% labeled samples were used for training, and the rest were used for testing. The highest accuracy in every row are in bold. Some of the data in the table are quoted from [39].

No.	Class	PCA	TPCA	SSAE	EPLS	3DCAE	ContrastNet	BYOL	UPDA + BYOL
1	Asphalt	90.89	96.17	95.72	95.95	95.21	99.49	98.21	**99.94**
2	Meadows	93.27	97.95	94.13	95.91	96.06	99.98	99.62	**99.99**
3	Gravel	82.60	86.50	87.47	94.33	91.32	99.06	95.15	**99.17**
4	Trees	92.41	94.84	96.91	99.28	98.28	97.75	96.41	**99.27**
5	Painted metal sheets	98.98	**100.00**	99.76	99.92	95.55	99.81	98.27	99.78
6	Bare Soil	92.00	94.76	95.76	93.57	95.30	99.90	99.98	**100.00**
7	Bitumen	85.83	91.89	91.18	98.17	95.14	99.83	96.18	**99.89**
8	Self-Blocking Bricks	82.96	89.04	82.47	91.23	91.38	**98.79**	95.12	98.75
9	Shadows	**100.00**	98.94	**100.00**	99.78	99.96	94.84	80.79	94.09
	AA (%)	90.99	95.64	93.71	96.33	95.36	98.83	95.53	**98.99**
	OA (%)	91.37	94.45	93.51	95.13	95.39	99.46	98.04	**99.65**

4.4. Feature Visualization

To further assess the effectiveness of UPDA, we utilized the t-distributed stochastic neighbor embedding (t-SNE) [41] method to visualize the learned representation of spatial–spectral features. For comparison, we also visualized the feature extracted by the BYOL without UPDA. The results are presented in Figures 6–8.

In the IP dataset, subfigure (b) displays a more concentrated distribution of color blocks belonging to the same class, with fewer fine points, and the shape is more towards blocks than bars. Classes with few samples such as classes 1, 7, and 9 become easier to separate from other classes.

In the PU dataset, the classes 1, 4, 7, 8, and 9 are partly mixed in subfigure (a), but there are fewer overlapping parts of the color blocks in (b), which means that there exists a larger gap between classes. Furthermore, it is obvious that blocks of class 2 and 6 become more tightly connected after introducing UPDA.

Table 10. Comparison in the Salinas dataset. Five percent of the labeled samples were used for training, and the rest were used for testing. The highest accuracy value in every row is in bold. Some of the data in the table are quoted from [39].

No.	Class	PCA	TPCA	SSAE	EPLS	3DCAE	ContrastNet	BYOL	UPDA + BYOL
0	Brocoli_green_weeds_1	97.48	99.88	**100.00**	99.99	**100.00**	99.93	98.36	99.95
1	Brocoli_green_weeds_2	99.52	99.49	99.52	99.92	99.29	99.80	99.99	**100.00**
2	Fallow	99.41	99.04	94.24	98.75	97.13	99.95	99.95	**100.00**
3	Fallow_rough_plow	99.77	99.84	99.17	98.52	97.91	98.01	99.19	**99.82**
4	Fallow_smooth	98.70	98.96	98.82	98.33	98.26	99.48	98.99	**99.92**
5	Stubble	99.65	99.80	**100.00**	99.92	99.98	99.94	99.99	99.97
6	Celery	99.94	99.84	99.94	97.69	99.64	99.79	99.61	**99.99**
7	Grapes_untrained	83.90	84.11	80.73	78.86	91.58	99.53	99.32	**99.93**
8	Soil_vinyard_develop	99.97	99.60	99.47	99.54	99.28	99.71	99.85	**100.00**
9	Corn_senesced_green_weeds	96.89	95.76	92.12	95.98	96.65	99.80	99.95	**100.00**
10	Lettuce_romaine_4wk	96.84	96.14	96.62	98.60	97.74	**99.80**	99.15	99.67
11	Lettuce_romaine_5wk	99.95	99.07	97.75	99.44	98.84	**99.98**	99.53	99.95
12	Lettuce_romaine_6w	99.54	**100.00**	95.81	98.85	99.26	98.20	**100.00**	**100.00**
13	Lettuce_romaine_7wk	97.24	95.74	96.65	98.56	97.49	98.62	98.72	**99.74**
14	Vinyard_untrained	76.68	79.54	79.73	83.13	87.85	99.53	98.98	**99.90**
15	Vinyard_vertical_trellis	97.90	98.40	99.12	99.50	98.34	99.57	**100.00**	**100.00**
	AA (%)	96.46	93.24	95.61	96.55	97.45	99.48	99.47	**99.93**
	OA (%)	92.87	96.57	92.11	92.35	95.81	99.60	99.48	**99.95**

Table 11. Comparison of different methods' time consumption.

Method	Time Consumption(s)		
	IP	PU	SA
ContrastNet	2048.92	3778.73	4651.70
UPDA + BYOL	1545.26	2544.02	3214.00

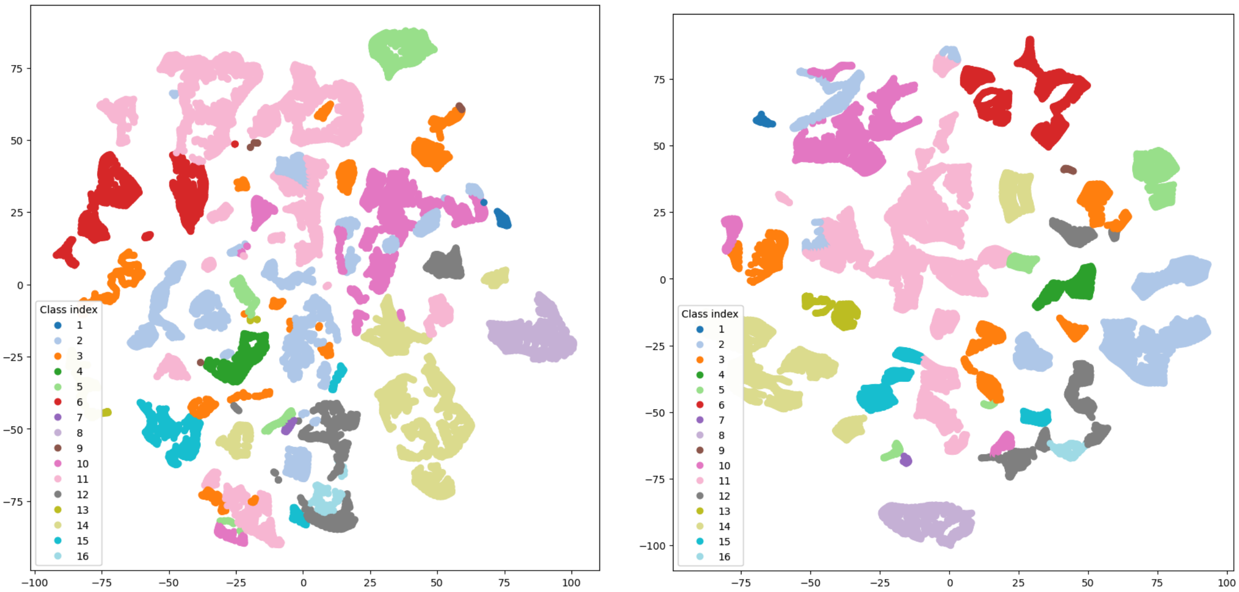
In the SA dataset, the pores in the color block of (b) are smaller, and the fine blocks are almost gone. In addition, the contact points between different color blocks are reduced. After introducing UPDA, the features of classes 13 and 14 are glued together, but the features of classes 1, 10, and 15 are obviously more isolated.

By comparing the visualization of features, we can conclude that UPDA can help BYOL to reduce the intra-class gap, increase the inter-class gap, and improve the representation of features.

4.5. Analysis of Different Strategies

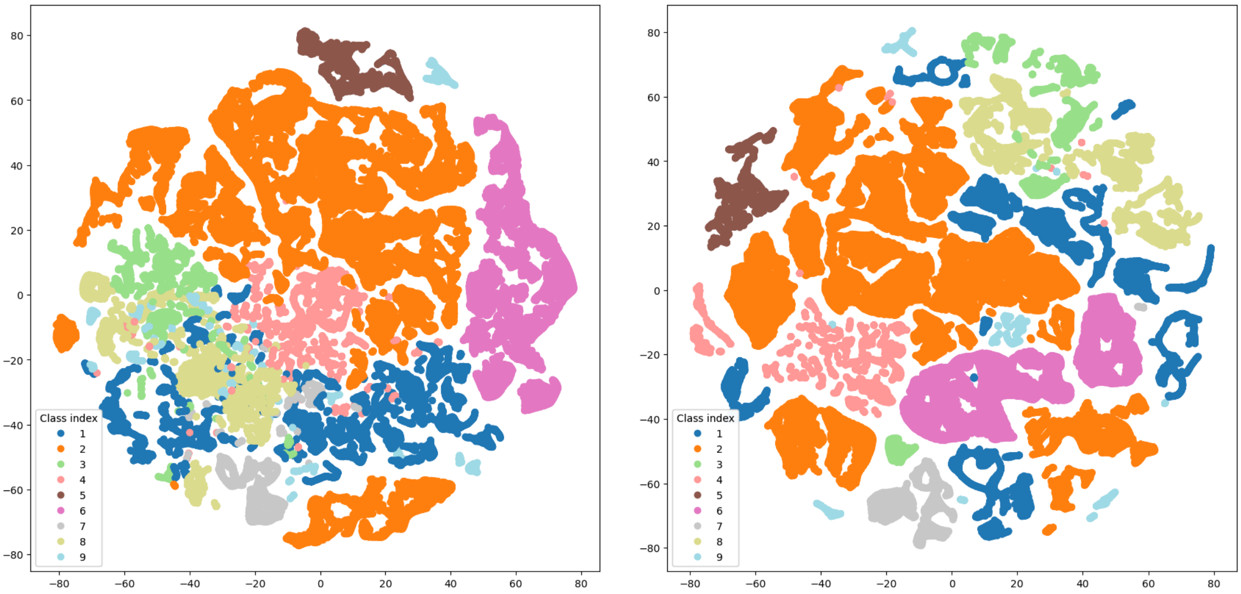
In order to investigate the impact of different data augmentation methods in UPDA on classification performance, we conducted ablation experiments. In addition to the blank group and the complete group, we created the "BE", "RO", "GM", "BE + RO", "BE + GM", and "RO + GM" groups, and conducted experiments on three datasets. We repeated each experiment three times and recorded the average OA.

Table 12 shows the results. The "BE + GM + RO" group achieved the best OA in all three datasets, which shows that the combination of the three strategies is most effective.



(**a**) BYOL (**b**) UPDA + BYOL

Figure 6. Visualization of extracted features in the Indian Pines dataset.

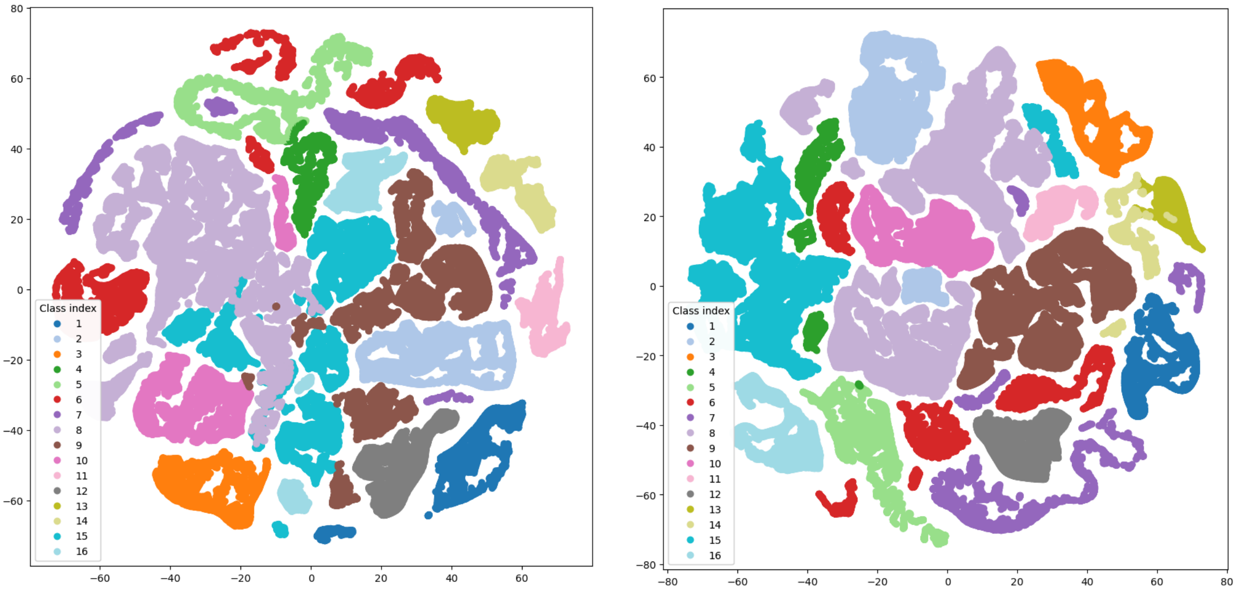


(**a**) BYOL (**b**) UPDA + BYOL

Figure 7. Visualization of extracted features in the University of Pavia dataset.

The "BE + GM" group achieved second place in the IP and SA dataset and third place in the PU dataset. The "GM + RO" group achieved second place in the PU dataset and third place in the SA dataset. In general, when the strategies are grouped in pairs, the effect varies according to different datasets.

When each strategy is joined individually, RO performs better than GM and BE, and BE performs the worst overall. There is no doubt that the introduction of each strategy improves the results, and they achieve the best classification performance when working together.



(**a**) BYOL (**b**) UPDA + BYOL

Figure 8. Visualization of extracted features in the Salinas dataset.

Table 12. The results of the ablation study. Ten percent of the labeled samples were used for training in the IP and PU datasets, and 5% were used in the SA dataset. In every dataset, the three highest OAs are in bold and marked by rank in the lower-right corner.

Group	**Strategy**			**OA(%)**		
-	**BE**	**RO**	**GM**	**IP**	**PU**	**SA**
(a)				95.71	98.04	99.48
(b)	✓			96.81	98.77	99.50
(c)		✓		97.09	99.29	99.86
(d)			✓	96.14	98.91	99.64
(e)	✓	✓		$\mathbf{97.95}_2$	$\mathbf{99.60}_3$	$\mathbf{99.90}_2$
(f)	✓		✓	96.96	98.90	99.69
(g)		✓	✓	$\mathbf{97.38}_3$	$\mathbf{99.64}_2$	$\mathbf{99.87}_3$
(h)	✓	✓	✓	$\mathbf{98.00}_1$	$\mathbf{99.65}_1$	$\mathbf{99.95}_1$

When considering the characteristics of the dataset, we can also analyze the strengths and weaknesses of each strategy. For the PU datasets with a small wavelength range and complex background, the effect of RO is the most significant and the effect of BE is the weakest when compared with other datasets. We can infer that RO can help to more effectively extract spatial features from environments where spatial features are not obvious, and BE can work better in datasets rich in spectral information. The performances of GM in different datasets are quite varied, and it is less pronounced when combined with other strategies.

5. Conclusions

This letter introduces a novel and efficacious approach, UPDA, which aims to bolster the performance of contrastive learning in HSIC. The approach is designed to enhance the representation of spatial–spectral features through a series of predominant data augmentation strategies designed for HSIs. By utilizing our method to extract features from a vast amount of data in an unsupervised manner and subsequently training a classifier with a small number of labels, exceptional classification accuracy can be achieved. The experimental results evidence the superiority of our method, and the ablation study underscores the effectiveness and distinctions of each strategy. Our method is highly adaptable, allowing for the incorporation of new strategies, and has tremendous potential to enhance the ability of contrastive learning, leading to better results with fewer labels. Furthermore, UPDA is not only applicable to BYOL but can also unlock the potential of other contrastive learning methods. We anticipate that our work will inspire novel ideas in fellow researchers and facilitate the more effective application of contrastive learning in the field of HSI.

Author Contributions: Methodology, J.L.; Software, J.L.; Formal analysis, J.L. and X.L.; Resources, X.L. and Y.Y.; Writing—original draft, J.L.; Writing—review & editing, J.L.; Visualization, J.L.; Funding acquisition, X.L. and Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: The research was funded by the National Nature Science Foundation of China under Grant No. 62171404, and in part by the Joint Fund Project of Chinese Ministry of Education under Grant No. 8091B022118.

Data Availability Statement: The data presented in this study are available on http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes.

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

Abbreviations

The following abbreviations are used in this manuscript:

HSI	Hyperspectral image
HSIC	Hyperspectral image classification
UPDA	Unlock the potential of data augmentation
BYOL	Bootstrap your own latent
PCA	Principal component analysis
ICA	Independent component analysis
LDA	Linear discriminant analysis
MDS	Multidimensional scaling
CNN	Convolutional neural network
DRNN	Deep recurrent neural network
DFFN	Deep feed-forward networks
AE	Autoencoder
GAN	Generative adversarial network
AAE	Adversarial autoencoder
VAE	Variational autoencoder
MAE	Masked autoencoder
DCGAN	Deep convolutional generative adversarial network
InfoGAN	Information-maximizing generative adversarial network
Moco	Momentum contrast
SimCLR	A simple framework for contrastive learning of visual representations
SimSiam	Simple Siamese
BE	Band erasure
RO	Random occlusion
GM	Gradient mask
OA	Overall accuracy
AA	Average accuracy
IP	Indian pines
PU	University of Pavia
SA	Salina
EMA	Exponential moving average
TPCA	Tensor principal component analysis
SSAE	Stacked sparse autoencoder
EPLS	Unsupervised deep feature extraction
3DCAE	3 dimensional convolutional autoencoder

References

1. Datta, D.; Mallick, P.K.; Bhoi, A.K.; Ijaz, M.F.; Shafi, J.; Choi, J. Hyperspectral image classification: Potentials, challenges, and future directions. *Comput. Intell. Neurosci.* **2022**, *2022*, 3854635. [CrossRef] [PubMed]
2. Stuart, M.B.; McGonigle, A.J.; Willmott, J.R. Hyperspectral imaging in environmental monitoring: A review of recent developments and technological advances in compact field deployable systems. *Sensors* **2019**, *19*, 3071. [CrossRef] [PubMed]
3. Tong, X.; Xie, H.; Weng, Q. Urban land cover classification with airborne hyperspectral data: What features to use? *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2013**, *7*, 3998–4009. [CrossRef]
4. Duan, P.; Ghamisi, P.; Kang, X.; Rasti, B.; Li, S.; Gloaguen, R. Fusion of Dual Spatial Information for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote. Sens.* **2021**, *59*, 7726–7738. [CrossRef]
5. Wold, S.; Esbensen, K.; Geladi, P. Principal component analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37–52. [CrossRef]
6. Villa, A.; Benediktsson, J.A.; Chanussot, J.; Jutten, C. Hyperspectral image classification with independent component discriminant analysis. *IEEE Trans. Geosci. Remote. Sens.* **2011**, *49*, 4865–4876. [CrossRef]
7. Du, Q. Modified Fisher's linear discriminant analysis for hyperspectral imagery. *IEEE Geosci. Remote. Sens. Lett.* **2007**, *4*, 503–507. [CrossRef]
8. Kruskal, J.B.; Wish, M.; Uslaner, E.M. Multidimensional scaling. In *Handbook of Perception and Cognition*; Academic Press: Cambridge, MA, USA, 1978.
9. Hu, W.; Huang, Y.; Wei, L.; Zhang, F.; Li, H. Deep convolutional neural networks for hyperspectral image classification. *J. Sensors* **2015**, *2015*, 258619. [CrossRef]
10. Chen, Y.; Jiang, H.; Li, C.; Jia, X.; Ghamisi, P. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Trans. Geosci. Remote. Sens.* **2016**, *54*, 6232–6251. [CrossRef]
11. Roy, S.K.; Krishna, G.; Dubey, S.R.; Chaudhuri, B.B. HybridSN: Exploring 3D-2D CNN Feature Hierarchy for Hyperspectral Image Classification. *IEEE Geosci. Remote. Sens. Lett.* **2020**, *17*, 277–281. [CrossRef]
12. 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]
13. Hang, R.; Li, Z.; Liu, Q.; Ghamisi, P.; Bhattacharyya, S.S. Hyperspectral image classification with attention-aided CNNs. *IEEE Trans. Geosci. Remote. Sens.* **2020**, *59*, 2281–2293. [CrossRef]
14. Zhang, X.; Sun, Y.; Jiang, K.; Li, C.; Jiao, L.; Zhou, H. Spatial sequential recurrent neural network for hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2018**, *11*, 4141–4155. [CrossRef]
15. Ballard, D.H. Modular Learning in Neural Networks. In Proceedings of the 6th National Conference on Artificial Intelligence (AAAI-87), Seattle, WA, USA, 13–17 July 1987; pp. 279–284.
16. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the 28th Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
17. Makhzani, A.; Shlens, J.; Jaitly, N.; Goodfellow, I.; Frey, B. Adversarial autoencoders. *arXiv* **2015**, arXiv:1511.05644.
18. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
19. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 16000–16009.
20. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434.
21. Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Proceedings of the 30th Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; Volume 29.
22. Hang, R.; Zhou, F.; Liu, Q.; Ghamisi, P. Classification of hyperspectral images via multitask generative adversarial networks. *IEEE Trans. Geosci. Remote. Sens.* **2020**, *59*, 1424–1436. [CrossRef]
23. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum Contrast for Unsupervised Visual Representation Learning. In Proceedings of the The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Online, 16–18 June 2020.
24. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the 37th International Conference on Machine Learning (PMLR), Vienna, Austria, 12–18 July 2020; pp. 1597–1607.
25. Grill, J.B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. Bootstrap your own latent-a new approach to self-supervised learning. In Proceedings of the 34th Conference on Neural Information Processing Systems, Online, 6–12 December 2020; Volume 33, pp. 21271–21284.
26. Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. In Proceedings of the 34th Conference on Neural Information Processing Systems, Online, 6–12 December 2020; Volume 33, pp. 9912–9924.
27. Chen, X.; He, K. Exploring simple siamese representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Online, 19–25 June 2021; pp. 15750–15758.
28. Hang, R.; Qian, X.; Liu, Q. Cross-Modality Contrastive Learning for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote. Sens.* **2022**, *60*, 5532812. [CrossRef]

29. Guan, P.; Lam, E.Y. Cross-domain contrastive learning for hyperspectral image classification. *IEEE Trans. Geosci. Remote. Sens.* **2022**, *60*, 5528913. [CrossRef]
30. Shu, Z.; Liu, Z.; Zhou, J.; Tang, S.; Yu, Z.; Wu, X.J. Spatial–Spectral Split Attention Residual Network for Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2022**, *16*, 419–430. [CrossRef]
31. 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]
32. Halvagal, M.S.; Laborieux, A.; Zenke, F. Predictor networks and stop-grads provide implicit variance regularization in BYOL/SimSiam. *arXiv* **2022**, arXiv:2212.04858.
33. Ding, K.; Xu, Z.; Tong, H.; Liu, H. Data augmentation for deep graph learning: A survey. *ACM SIGKDD Explor. Newsl.* **2022**, *24*, 61–77. [CrossRef]
34. Xiao, T.; Wang, X.; Efros, A.A.; Darrell, T. What should not be contrastive in contrastive learning. *arXiv* **2020**, arXiv:2008.05659.
35. Li, J.; Li, X.; Cao, Z.; Zhao, L. ROBYOL: Random-Occlusion-Based BYOL for Hyperspectral Image Classification. *IEEE Geosci. Remote. Sens. Lett.* **2022**, *19*, 6014405. [CrossRef]
36. Ren, Y.; Liao, L.; Maybank, S.J.; Zhang, Y.; Liu, X. Hyperspectral image spectral-spatial feature extraction via tensor principal component analysis. *IEEE Geosci. Remote. Sens. Lett.* **2017**, *14*, 1431–1435. [CrossRef]
37. Tao, C.; Pan, H.; Li, Y.; Zou, Z. Unsupervised spectral–spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification. *IEEE Geosci. Remote. Sens. Lett.* **2015**, *12*, 2438–2442.
38. Romero, A.; Gatta, C.; Camps-Valls, G. Unsupervised deep feature extraction for remote sensing image classification. *IEEE Trans. Geosci. Remote. Sens.* **2015**, *54*, 1349–1362. [CrossRef]
39. Mei, S.; Ji, J.; Geng, Y.; Zhang, Z.; Li, X.; Du, Q. Unsupervised Spatial–Spectral Feature Learning by 3D Convolutional Autoencoder for Hyperspectral Classification. *IEEE Trans. Geosci. Remote. Sens.* **2019**, *57*, 6808–6820. [CrossRef]
40. Cao, Z.; Li, X.; Feng, Y.; Chen, S.; Xia, C.; Zhao, L. ContrastNet: Unsupervised feature learning by autoencoder and prototypical contrastive learning for hyperspectral imagery classification. *Neurocomputing* **2021**, *460*, 71–83. [CrossRef]
41. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.