

Article

Masked Graph Convolutional Network for Small Sample Classification of Hyperspectral Images

Wenkai Liu ¹, Bing Liu ² , Peipei He ^{1,*}, Qingfeng Hu ¹, Kuiliang Gao ² , and Hui Li ¹

¹ College of Surveying and Geo-Informatics, North China University of Water Resources and Electric Power, Zhengzhou 450046, China

² Institute of Geospatial Information, PLA Strategic Support Force Information Engineering University, Zhengzhou 450002, China

* Correspondence: hepei@ncwu.edu.cn

Abstract: The deep learning method has achieved great success in hyperspectral image classification, but the lack of labeled training samples still restricts the development and application of deep learning methods. In order to deal with the problem of small samples in hyperspectral image classification, a novel small sample classification method based on rotation-invariant uniform local binary pattern (RULBP) features and a graph-based masked autoencoder is proposed in this paper. Firstly, the RULBP features of hyperspectral images are extracted, and then the k-nearest neighbor method is utilized to construct the graph. Furthermore, self-supervised learning is conducted on the constructed graph so that the model can learn to extract features more suitable for small sample classification. Since the self-supervised training mainly adopts the masked autoencoder method, only unlabeled samples are needed to complete the training. After training, only a small number of samples are used to fine-tune the graph convolutional network, so as to complete the classification of all nodes in the graph. A large number of classification experiments on three commonly used hyperspectral image datasets show that the proposed method could achieve higher classification accuracy with fewer labeled samples.



Citation: Liu, W.; Liu, B.; He, P.; Hu, Q.; Gao, K.; Li, H. Masked Graph Convolutional Network for Small Sample Classification of Hyperspectral Images. *Remote Sens.* **2023**, *15*, 1869. <https://doi.org/10.3390/rs15071869>

Academic Editor: Salah Bourennane

Received: 20 February 2023

Revised: 23 March 2023

Accepted: 27 March 2023

Published: 31 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Hyperspectral remote sensing images containing numerous bands could provide rich spectral and spatial information for ground objects, which makes it possible to distinguish ground objects more precisely [1,2]. Using the rich information provided by hyperspectral images (HSIs) to achieve ground object classification has been one of the research focuses in the field of hyperspectral remote sensing. Up to now, hyperspectral remote sensing technology has been widely used in precision agriculture, disaster investigation, resource management, geological exploration, archaeological analysis, urban planning, military and national defense, and many other fields [3–6]. However, the application and development of HSI classification technology has been restricted by the high-dimensional, nonlinear data characteristics of HSIs and the lack of labeled samples [7–9].

The early explorations on HSI classification are mainly carried out from the aspect of extracting manually designed features, which can be divided into three categories: spectral feature extraction, spatial feature extraction, and spatial-spectral feature extraction. Existing studies show that utilizing the spectral and spatial information in HSIs can effectively improve the accuracy of target classification and recognition. However, most of the early explorations are based on hand-designed features, which actually causes a lack of robustness and stability.

In recent years, the deep learning method could automatically carry out end-to-end feature learning from data, which makes it better adapted to HSI classification tasks [10,11].

The deep learning models, such as one-dimensional convolutional neural network (CNN) [12], stack autoencoder [13,14], and deep belief network [15], which can use one-dimensional spectral features, are first introduced into HSI classification tasks. Then, in order to use spatial information, 2D-CNNs treat different bands of HSIs as feature maps for end-to-end training [16–19]. Furthermore, in order to make good use of the three-dimensional spatial-spectral structure, the 3D-CNN directly takes the hyperspectral data cube as a whole and then uses the three-dimensional convolution operation to mine its three-dimensional spatial-spectral features [20–23]. The exploration of the above deep learning methods has promoted the development of HSI classification methods, but these methods often require a certain number of labeled training samples to ensure the classification effect, and the sparsity of labeled samples in HSIs leads to the problem of small samples [24,25]. To deal with the problem of small samples, researchers first explored the semi-supervised learning method to use a large number of unlabeled samples to improve the classification accuracy in the case of small samples [26]. For example, Liu et al. design a semi-supervised convolutional neural network for HSI classification using ladder networks [27], and Zhan et al. design a semi-supervised classification model for HSIs based on the generative adversarial network [28]. To further reduce the dependence of training on labeled samples, the latest machine learning methods, such as self-supervised learning [29] and meta-learning [30], are used to train the deep learning models' learning to extract features more suitable for small samples. At the same time, some researchers convert HSIs into graph structures—that is, pixel points and their feature descriptions are regarded as nodes and attributes of graphs—and then use graph models to complete classification. For example, Zuo et al. propose using the graph inductive learning method for sample classification of HSIs [31]. Hong et al. design a graph convolutional network for HSI classification [32]. Zhao et al. design a graph transformer network with the graph attention mechanism for HSI classification [33]. This work is mainly inspired by the graph convolution network and the latest generative self-supervised method. First, the k-nearest neighbor method is used to convert the entire HSI into a graph, and then the graph convolution network is used to conduct generative self-supervised learning on the constructed graph so that the model can automatically learn the features more suitable for subsequent small sample classification tasks. Finally, a small number of labeled samples are used to fine-tune the graph convolution network to complete the final classification. In summary, the contributions of this paper can be summarized as follows:

- A graph-based self-supervised learning method with the rotation-invariant uniform local binary pattern (RULBP) features is proposed to deal with the small sample problem of HSI classification. The goal of training is to reconstruct the features with masking. Therefore, the training process does not need to use any label information.
- The RULBP features of the HSI are used to calculate the k-nearest neighbor when constructing the graph so that the deep model could better use the spatial-spectral information of HSIs.
- A large number of classification experiments on three hyperspectral datasets verify the effectiveness of the proposed method for small sample classification.

The remainder of this paper is organized as follows. In Section 3, the proposed method is described in detail. In Section 4, extensive classification experiments are conducted. In Section 5, hyperparameter analysis and discussion are given. The conclusion is drawn in Section 6.

2. Related Work

2.1. Early Research

Aiming at the problem of HSI classification, early researchers focused on spectral features of HSIs and designed many spectral feature extraction methods, such as principal component analysis (PCA) [34,35], manifold learning [36], spectral shape feature [37,38], etc. The spectral feature extraction method combined with support vector machine (SVM), random forest (RF), and other classifiers could improve the classification accuracy to some extent. However, the wide existence of the phenomenon of the same object and different

spectrum in remote sensing images makes it very limited to classify HSIs only by using spectral features [39,40]. Adjacent samples in HSIs have a high probability of belonging to the same category, so researchers have conducted a lot of exploration to improve the classification accuracy of HSIs using spatial information. There are many ways to use the spatial information of HSIs. For example, spatial neighborhood information can be comprehensively considered in the feature extraction stage. Such representative methods include extended morphological profiles [41], Gabor texture features [42], local binary pattern texture features [43], etc. In the classification stage, the spatial neighborhood information of samples can also be comprehensively considered. This kind of method usually takes the spatial neighborhood information of samples as a constraint item of classifier optimization so as to improve the classification accuracy by using spatial information [44,45]. Of course, after classification, spatial neighborhood information can also be used to post-process the classification results. For example, guided filtering can be used to post-process the classification results to remove isolated classification noise so as to improve the classification accuracy of HSIs [46,47]. The above spatial–spectral classification methods could significantly improve the classification effect of HSIs, but the disadvantage is that the classification effect needs strong expert experience, that is, the feature extraction rules or post-processing strategies need to be designed manually, which undoubtedly reduces the practicability of HSI classification technology [48].

2.2. Deep Learning-Based HSI Classification

Recently, HSI classification using deep learning models has become one of the research hotspots in the field of remote sensing. From the perspective of processing the data structure, the existing methods can be roughly divided into three types: raster-based methods, sequence-based methods, and graph structure-based methods. The 2D-CNN and 3D-CNN models are two representative methods of the first class. The 2D-CNN model can directly perform convolution operation on HSIs with raster structure and extract rich spatial information from each spectral band. For example, Zhang et al. design a novel diverse region-based 2D-CNN method for HSI classification. By aggregating spatial information in diverse neighborhoods, the performance of HSI classification can be effectively improved [49]. Gao et al. train the designed 2D-CNN model based on episode, further improving the accuracy and robustness of classification results [50]. In contrast, The 3D-CNN model regards the input HSI data as a complete cube and directly applies 3D convolution to extract the spatial–spectral features. For example, Xu et al. design a novel multiscale octave 3D-CNN model and Zhong et al. design a deep residual 3D-CNN model, achieving better classification results [51,52]. At the same time, some researchers try to treat HSIs along the spectral dimension as sequence data and then use the recurrent neural network model to complete classification [18,53,54]. To further improve the classification accuracy of HSIs, channel attention, spatial attention, and self-attention mechanisms are gradually widely used in the design of HSI deep learning classification models [55–57]. More recently, with the breakthrough performance of graph convolutional network (GCN) in computer vision, HSI classification methods based on GCN have begun to appear and receive a lot of attention. For example, Hong et al. take the lead in exploring the performance of GCN in HSI classification [32]. Liu et al. design a semi-supervised learning method based on the GCN model [58]. Zhao et al. introduce an attention mechanism to further improve the classification performance [33]. However, the above GCN-based methods all utilize labeled samples for supervised learning and conduct auxiliary semi-supervised training based on the graph relation structure, and there is a lack of research on the application of GCN-based self-supervised learning to HSI classification. At the same time, the classification performance still has great room for improvement. Among the existing research, the study closest to the proposed method is the work of Zuo et al. [59]. However, compared with this work, the proposed method has three advantages. First of all, Zuo et al. follow the method of supervised learning for model training, and the proposed method can utilize a large number of unlabeled samples for self-supervised

training and make accurate classification and prediction based on the extracted features. Second, an advanced generative self-supervised method, namely a partial masked strategy, is applied in the proposed method, and its related hyperparameters are fully discussed. Thirdly, the proposed method can achieve better classification performance.

3. Methodology

In this section, the workflow of the proposed method is first described, and then the proposed method is introduced in detail in three parts: RULBP feature extraction, graph construction, and self-supervised learning.

3.1. Workflow

To further improve the accuracy of small sample classification of HSIs, a novel graph-based self-supervised learning method is proposed in this paper. Figure 1 shows the workflow of the proposed method, which includes two stages: self-supervised training and inference. For a given HSI, the RULBP features are first extracted to make full use of the spatial and spectral information in the image and effectively reduce band redundancy. Then, the K-neighbors algorithm is used to construct the graph based on the extracted RULBP features, and the original HSI with grid structure is converted into the graph structure data. Next, a certain proportion of nodes are masked randomly, and the whole masked graph is fed to the encoder to learn the encoded node representations. During the representations' decoding, the randomly selected nodes are remasked, and the decoder learns to reconstruct the original node features using the masked representations. The loss is calculated on the original node features and the reconstructed node features. Actually, the proposed method is a generative self-supervised learning method, which can achieve efficient feature learning without any labeled samples. During the inference stage, only the encoder is used for fine-tuning and prediction. Specifically, all parameters of the encoder are frozen and the encoder is only responsible for generating the learned representations. Then, a simple linear classifier is established and trained on a small number of labeled samples without a mask. Finally, the trained classifier is used for inference and evaluation.

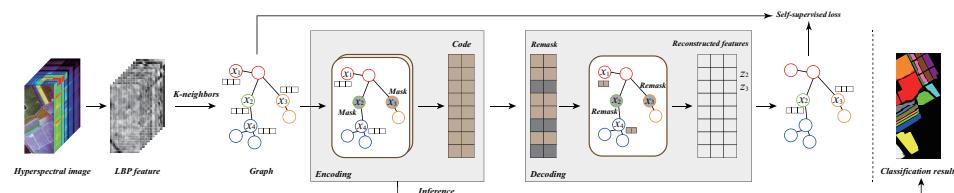


Figure 1. Workflow of the proposed method.

Briefly speaking, the proposed method has three advantages. First, extracting the RULBP features can make full use of the spatial and spectral information in the HSI. Second, constructing the graph structure can effectively utilize the relation information between pixels and improve the data efficiency. Third, the self-supervised learning mode combined with the dual-mask strategy can perform semantic feature learning without labeled samples, significantly reducing the dependence of the deep model on supervised annotation information.

3.2. RULBP Feature Extraction

To further improve the accuracy of the HSI small sample classification, the RULBP features are adopted in the proposed method. As an algorithm used to describe the local texture features of input images, the LBP has the advantages of simple calculation, rotation

invariance, and gray invariance. Specifically, the rotation-invariant local binary mode (RULBP) code of each pixel in the input HSI is calculated as follows.

$$RULBP_{P,R}(x,y) = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c), & \text{if } U \leq 2 \\ P + 1, & \text{otherwise} \end{cases} \quad (1)$$

In Equation (1), R denotes the neighborhood radius of the center pixel, and P denotes the number of pixels sampled in the neighborhood. g_c denotes the gray value of the center pixel, while g_0, \dots, g_p denote the gray value of the pixels in the neighborhood. The symbol U represents the number of hops from 0 to 1 or from 1 to 0 in the binary code. Referring to the relevant research [31], the values of R and P are set to 1 and 8, respectively. Therefore, according to Equation (1), there are 10 RULBP codes for each pixel, which can effectively improve the stability of LBP feature extraction.

Figure 2 gives the schematic of generating RULBP features. For the input HSI, the PCA features are first generated to preserve the main spectral features while reducing band redundancy. Next, each principal component is treated as a gray image and used to generate the RULBP codes. The RULBP feature of each pixel is determined by the frequency statistics of the RULBP codes in its neighborhood. As mentioned above, each pixel has 10 RULBP codes, so after RULBP feature extraction, each principal component can actually generate a feature map with 10 dimensions. Finally, the RULBP features of all principal components are connected with the original principal components to form the final feature matrix for graph construction. For example, an input HSI with 30 principal components will eventually generate a 330-dimension ($330 = 30 \times 10 + 30$) RULBP feature.

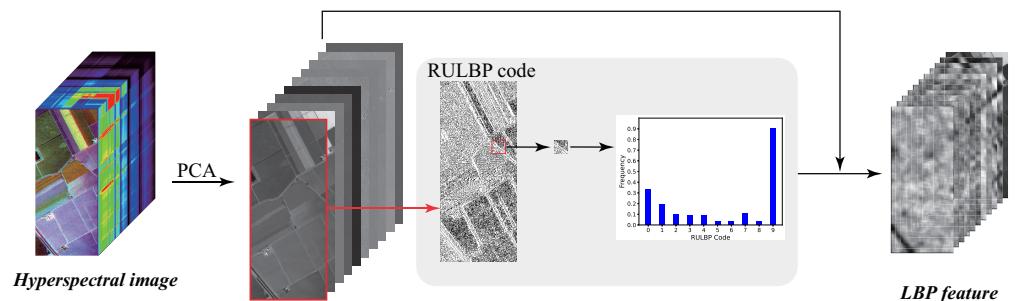


Figure 2. Schematic of generating the RULBP features.

3.3. Graph Construction by K-Neighbors

Undoubtedly, constructing the graph structure based on the extracted RULBP features is a very important step in the proposed method, because the effect of the constructed graph has a direct impact on the expression quality of the relations between pixels. Figure 3 shows the adopted graph construction method in our work. After RULBP feature extraction, the original HSI is organized into a matrix $X \in \mathbb{R}^{N \times C}$. Specifically, the symbol N denotes the number of samples and C denotes the dimension of vectors. In this matrix, the distance between samples can be denoted as

$$d_{ij} = \left(\sum_{c=1}^C |x_{ic} - x_{jc}|^p \right)^{\frac{1}{p}}, \quad (2)$$

where p is an optional parameter, and x_i and x_j are samples at different locations. Furthermore, a symmetric distance matrix can be generated by calculating the distance values of all sample pairs, which can be denoted as $D_m = [d_{ij}] \in \mathbb{R}^{N \times N}$. In this matrix, the distance between the i th pixel and the j th pixel can be denoted as d_{ij} at row i and column j . For each pixel, the K-Neighbors algorithm selects the K pixels closest to it as the neighbor nodes. After establishing the neighbor nodes of all pixels, the graph construction process is

completed. Consequently, the input HSI with the grid structure is converted to the graph where each pixel has K neighbor node.

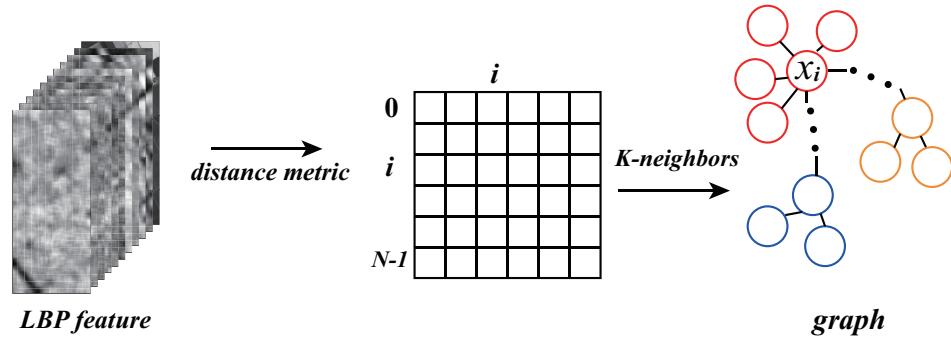


Figure 3. Schematic of graph construction by K-Neighbors.

3.4. Graph-Based Self-Supervised Learning

The proposed method, actually a generative self-supervised learning method, can utilize a large number of unlabeled samples for deep feature learning. The whole process of graph-based self-supervised training is shown in Figure 1. For clarity, several mathematical symbols and corresponding meanings are given first. Given a graph constructed on the extracted RULBP features, it can be denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{X})$, where \mathcal{V} is the node set, $\mathcal{X} \in \mathbb{R}^{N \times d}$ is the input node feature matrix. $\mathcal{A} \in \{0, 1\}^{N \times N}$ is the adjacency matrix and $N = |\mathcal{V}|$ is the number of nodes. In addition, the graph encoder and decoder can be denoted as f_E and f_D , and the encoded node representations can be denoted as \mathcal{H} . Therefore, the whole workflow of graph-based self-supervised learning can be denoted as

$$\mathcal{H} = f_E(\mathcal{A}, \mathcal{X}), \mathcal{G}' = f_D(\mathcal{A}, \mathcal{H}), \quad (3)$$

where \mathcal{G}' denotes the reconstructed graph. In the proposed method, it is actually the reconstructed features.

To perform the self-supervised training with masking, a subset of nodes $\tilde{\mathcal{V}} \subset \mathcal{V}$ is first sampled randomly. Then, the features of these sampled nodes are masked with a learnable vector $x_{[M]} \in \mathbb{R}^d$. Therefore, the node feature \tilde{x}_i in the masked feature matrix $\tilde{\mathcal{X}}$ can be denoted as follows.

$$\tilde{x}_i = \begin{cases} x_{[M]}, & v_i \in \tilde{\mathcal{V}} \\ x_i, & v_i \notin \tilde{\mathcal{V}} \end{cases} \quad (4)$$

It is known to all that in a graph, the connection relations are the important information to enhance the features of the nodes. Therefore, to avoid the occurrence of some extreme cases—for example, all the neighbors of a node being masked—a uniform random sampling strategy is adopted in the proposed method. In addition, referring to existing research, setting a relatively large mask ratio is better to reduce redundancy information and effectively improve the robustness of the deep model.

To further prompt the encoder to learn robust and compressed features, the generated representations \mathcal{H} are also masked, as shown in Figure 1. Specifically, the representations of the masked node are remasked again with another mask which can be denoted as $h_{[M]} \in \mathbb{R}^{d_h}$. Therefore, the masked representations \tilde{h}_i can be denoted as follows.

$$\tilde{h}_i = \begin{cases} h_{[M]}, & v_i \in \tilde{\mathcal{V}} \\ h_i, & v_i \notin \tilde{\mathcal{V}} \end{cases} \quad (5)$$

According to Equation (5), the graph decoder has to utilize the features of neighboring unmasked nodes to reconstruct the input feature of masked nodes. Actually, the above two mask strategies promote the model to utilize the connection relations in graph structure to

achieve reconstruction from the two levels of nodes and deep representation, respectively, which can effectively enhance the learning of the semantic relations between pixels.

Finally, the loss between the original graph and the constructed graph is calculated as

$$\mathcal{L}_{SCE} = \frac{1}{|\tilde{\mathcal{V}}|} \sum_{v_i \in \tilde{\mathcal{V}}} \left(1 - \frac{x_i^T z_i}{\|x_i\| \cdot \|z_i\|} \right)^\gamma, \quad \gamma \geq 1 \quad (6)$$

which is actually the scaled cosine error (SCE). The scaling factor γ is used to reweight the contribution of samples to parameters updating. When $\gamma > 1$, the corresponding losses of the predictions with high confidence are usually smaller than 1 and decay faster to zero.

4. Experimental Results

4.1. HSI Datasets

To compare the classification performance of different methods, three widely used HSIs are used for experiments. The three HSIs contain University of Pavia (UP), Indian Pines (IP), and Salinas (SA), and the details are listed in Table 1. As we can see, the three HSIs possess completely different resolutions, spectral ranges, and observation regions, providing different data environments to comprehensively evaluate the classification performance of different models. In the process of self-supervised learning, all samples are regarded as unlabeled samples for graph construction and feature learning, and in the process of fine-tuning, only a few labeled samples are used for retraining the model. Specifically, in the experiments, 20 samples per class are used for retraining (fine-tuning), another 20 samples per class are used as the validation set, and all the remaining samples are used as the test set to evaluate the classification performance of the proposed method.

Table 1. HSI datasets. Ground Sample Distance (GSD) (m), Spatial size (pixel), Spectral range (nm), Airborne Visible Infrared Imaging Spectrometer (AVIRIS), Reflective Optics System Imaging Spectrometer (ROSIS).

	UP	IP	SA
Spatial size	610 × 340	145 × 145	512 × 217
Spectral range	430–860	400–2500	400–2500
No. of bands	103	200	204
GSD	1.3	20	3.7
Sensor type	ROSIS	AVIRIS	AVIRIS
Areas	Pavia	Indiana	California
No. of classes	9	16	16
Labeled samples	42776	10249	54129

4.2. Experimental Settings

The hardware environment includes a computer equipped with an AMD Ryzen 9 5900HX CPU and an Nvidia RTX 3080 GPU. The software environment is developed on Python and libraries such as Pytorch, sklearn, and numpy. The hyperparameters are described in the order of the workflow. For the UP, IP, and SA dataset, the 15, 30, and 30 principal components are first extracted, which can be used to generate the RULBP features with the dimensions of 165, 330, and 330, respectively. During graph construction, the number of neighbors of each nodes is set to 200. The standard GAT model [60] is used as the encoder network and the decoder network, which means the model has a symmetrical structure. Specifically, the network layer is set to 2 and the number of hidden units is set to 512. The first layer consists of eight attention heads followed by an exponential linear unit, while the second layer consists of a single attention head followed by a softmax activation and is used for classification. During self-supervised training, the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1 \times 10^{-8}$ is used for network updating. The initial learning rate is set to 0.0001 and the cosine learning rate decay is adopted. The scaling factor γ is 3. The maximum number of epochs of self-supervised training is set to 1500, and the early

stop strategy is adopted. Specifically, the validation set is used to evaluate performance, and if there is no further improvement after 20 epochs, the self-supervised training process will be stopped. The epochs of fine-tuning are set to 300, and the batchsize is set to 32, ensuring that the model can get full training. In addition, the mask ratio is set to 50%. More details and analysis about the hyperparameter settings are given in Section 5, and the overall accuracy (OA), average accuracy (AA), and kappa coefficient are used as evaluation criteria. All experiments will run 10 times and the average \pm deviation will be taken as results to display.

4.3. Results and Comparison

In this section, the effectiveness and performance of the proposed method will be comprehensively verified by comparing with several influential and advanced hyperspectral image classification methods. Since the goal of the proposed method is to improve the small sample classification performance of hyperspectral images, we focus on selecting specialized small sample classification methods for comparison. First, three influential few-shot learning methods are selected, including Deep Few-Shot Learning (DFSL) [30], Relation Network of Few-Shot learning (RN-FSC) [25], and Graph Information Aggregation Cross-Domain Few-Shot Learning (Gia-CFSL) [61]. Among them, Gia-CFSL is one of the latest and state-of-the-art few-shot learning methods. In addition, a transfer learning method, Cross-Domain Pretrained Model (CDPM) [62], is also selected. The four methods above all conduct the model's pre-training based on supervised learning on source data with sufficient labels and concentrate on improving the classification performance on the target data with few labels. Second, because the proposed method belongs to the self-supervised learning method and requires no labels in the procedure of feature learning, two similar unsupervised learning methods are further selected for comparison, including Unsupervised Meta Learning With Multi-view Constraints (UM²L) [10] and Unsupervised Spectrum Motion Feature Learning (SMF-UL) [63]. The settings of all comparative methods refer to their original literature, including the combination of source data, and the analysis results of accuracy with different methods on three hyperspectral datasets (the test set illustrated in Section 4.1) are shown in Tables 2–4.

Table 2. The analysis results of accuracy with different methods on the UP data (%) (20 training samples in each class).

	Supervised				Unsupervised		
	DFSL	RN-FSC	CDPM	Gia-CFSL	UM ² L	SMF-UL	RULBP-MGCN
1	90.79 \pm 3.97	93.7 \pm 2.41	70.74 \pm 8.92	95.27 \pm 2.24	88.87 \pm 2.99	96.36 \pm 0.98	93.19 \pm 1.03
2	97.13 \pm 1.65	94.09 \pm 2.22	84.71 \pm 5.42	96.79 \pm 1.45	93.67 \pm 1.14	97.89 \pm 0.75	99.75 \pm 0.34
3	71.83 \pm 7.14	69.89 \pm 5.64	80.38 \pm 3.25	75.86 \pm 5.48	88.09 \pm 2.95	88.25 \pm 2.12	90.54 \pm 2.09
4	90.08 \pm 10.94	91.39 \pm 8.10	90.44 \pm 5.62	79.84 \pm 4.31	88.24 \pm 2.36	68.62 \pm 5.67	73.79 \pm 3.12
5	99.53 \pm 0.22	99.20 \pm 0.42	98.61 \pm 0.78	99.97 \pm 0.03	97.81 \pm 1.87	96.38 \pm 1.07	100.00 \pm 0.00
6	66.63 \pm 14.14	59.74 \pm 10.59	70.81 \pm 8.97	64.25 \pm 7.12	94.81 \pm 1.24	91.37 \pm 1.42	89.04 \pm 1.15
7	84.72 \pm 3.90	69.58 \pm 7.92	90.81 \pm 6.67	64.37 \pm 7.41	94.77 \pm 1.98	85.87 \pm 2.69	82.57 \pm 2.38
8	75.91 \pm 6.28	78.72 \pm 7.42	77.24 \pm 5.47	87.75 \pm 4.34	89.09 \pm 4.67	87.51 \pm 2.42	90.49 \pm 2.97
9	96.83 \pm 1.28	95.97 \pm 2.10	98.49 \pm 1.12	99.51 \pm 0.37	93.16 \pm 3.92	99.79 \pm 0.17	73.55 \pm 0.86
OA	85.77 \pm 4.62	84.24 \pm 3.93	82.83 \pm 2.48	86.10 \pm 1.92	90.41 \pm 1.67	91.84 \pm 1.39	92.29 \pm 1.27
AA	85.94 \pm 2.58	83.59 \pm 3.62	84.47 \pm 2.97	84.84 \pm 1.88	91.06 \pm 1.45	91.22 \pm 1.58	88.10 \pm 1.13
Kappa	81.77 \pm 5.51	79.66 \pm 4.81	76.76 \pm 3.69	82.17 \pm 1.99	88.41 \pm 1.62	89.43 \pm 2.19	90.01 \pm 1.86

Table 3. The analysis results of accuracy with different methods on the IP data (%) (20 training samples in each class).

		Supervised			Unsupervised		RULBP-MGCN
	DFSL	RN-FSC	CDPM	Gia-CFSL	UM ² L	SMF-UL	
1	60.02 ± 13.70	58.67 ± 14.12	98.26 ± 0.75	88.95 ± 3.74	90.01 ± 3.91	71.44 ± 10.90	90.20 ± 2.85
2	66.82 ± 6.76	63.75 ± 5.62	55.08 ± 8.64	72.80 ± 6.62	70.99 ± 6.36	74.43 ± 5.09	89.85 ± 4.87
3	66.71 ± 4.46	47.38 ± 9.72	62.14 ± 5.24	98.38 ± 0.77	79.10 ± 5.01	72.46 ± 6.05	71.08 ± 2.63
4	61.68 ± 6.48	46.69 ± 8.86	72.01 ± 4.92	97.45 ± 1.69	89.69 ± 4.56	73.23 ± 4.37	90.03 ± 3.99
5	90.83 ± 6.02	86.82 ± 5.95	84.91 ± 5.51	92.79 ± 2.51	89.34 ± 3.63	84.28 ± 6.32	94.17 ± 4.96
6	98.33 ± 0.84	93.32 ± 3.33	91.68 ± 2.98	95.81 ± 1.42	96.14 ± 1.24	96.89 ± 2.14	95.89 ± 2.17
7	47.73 ± 9.86	35.83 ± 10.61	98.92 ± 0.99	76.85 ± 8.83	90.01 ± 3.07	54.80 ± 8.58	82.35 ± 5.69
8	100.00 ± 0.0	98.86 ± 0.74	92.37 ± 1.95	100.00 ± 0.0	99.23 ± 0.45	99.89 ± 0.07	100.00 ± 0.00
9	29.19 ± 7.60	30.62 ± 8.97	100.00 ± 0.0	47.76 ± 9.97	100.00 ± 0.0	62.05 ± 9.45	32.26 ± 9.78
10	59.38 ± 8.59	56.14 ± 7.62	74.67 ± 6.66	69.09 ± 4.91	74.02 ± 5.62	79.94 ± 4.32	79.40 ± 3.42
11	83.47 ± 3.99	70.08 ± 8.82	55.50 ± 9.92	87.46 ± 3.84	83.42 ± 4.21	91.21 ± 2.19	97.10 ± 1.63
12	85.22 ± 5.54	36.36 ± 10.72	54.92 ± 9.21	54.58 ± 7.73	78.70 ± 4.06	62.54 ± 6.06	96.89 ± 1.20
13	95.26 ± 2.86	82.91 ± 6.63	98.82 ± 0.82	98.97 ± 0.79	99.51 ± 0.32	92.43 ± 1.48	91.89 ± 2.12
14	97.89 ± 2.10	96.92 ± 2.95	83.60 ± 6.74	96.71 ± 1.41	95.53 ± 1.14	99.21 ± 0.64	99.60 ± 0.38
15	70.78 ± 4.53	65.92 ± 7.62	67.45 ± 8.54	79.98 ± 6.79	95.80 ± 1.77	94.88 ± 1.71	97.89 ± 0.98
16	69.55 ± 8.94	50.49 ± 11.64	99.89 ± 0.49	90.05 ± 3.42	98.92 ± 0.45	96.48 ± 0.98	86.79 ± 0.37
OA	75.72 ± 2.08	66.32 ± 3.91	69.89 ± 2.94	80.35 ± 2.04	83.78 ± 2.49	81.54 ± 2.56	90.75 ± 2.04
AA	71.87 ± 1.75	63.80 ± 3.45	78.52 ± 2.78	80.11 ± 2.43	81.77 ± 2.73	79.72 ± 2.29	87.21 ± 1.98
Kappa	72.74 ± 2.29	62.24 ± 3.21	65.49 ± 2.59	77.89 ± 3.75	80.30 ± 2.61	80.11 ± 2.83	89.53 ± 2.33

Table 4. The analysis results of accuracy with different methods on the SA data (%) (20 training samples in each class).

		Supervised			Unsupervised		RULBP-MGCN
	DFSL	RN-FSC	CDPM	Gia-CFSL	UM ² L	SMF-UL	
1	100.00 ± 0.0	98.14 ± 1.64	97.49 ± 1.12	99.02 ± 0.12	99.99 ± 0.01	98.24 ± 0.70	99.85 ± 0.12
2	98.90 ± 1.10	99.44 ± 0.02	99.57 ± 0.12	99.67 ± 0.23	98.60 ± 0.24	98.97 ± 0.88	100.00 ± 0.00
3	94.17 ± 3.29	92.91 ± 2.63	98.09 ± 0.66	98.92 ± 0.45	96.59 ± 0.92	95.04 ± 1.63	91.46 ± 1.53
4	97.77 ± 1.98	97.95 ± 1.62	98.73 ± 1.42	91.87 ± 2.42	98.05 ± 0.93	96.24 ± 1.21	84.19 ± 4.86
5	97.81 ± 0.89	99.87 ± 0.04	98.82 ± 1.01	99.99 ± 0.01	98.67 ± 1.01	98.93 ± 0.62	100.00 ± 0.00
6	99.52 ± 0.42	99.23 ± 0.42	98.86 ± 0.61	99.99 ± 0.01	98.72 ± 0.62	98.14 ± 0.58	100.00 ± 0.00
7	99.68 ± 0.64	96.52 ± 1.45	98.61 ± 0.78	99.40 ± 0.26	99.04 ± 0.27	97.96 ± 1.07	100.00 ± 0.00
8	85.93 ± 3.32	81.82 ± 2.39	68.06 ± 5.94	87.22 ± 1.45	87.36 ± 2.38	93.59 ± 1.24	95.20 ± 0.79
9	99.57 ± 3.30	99.55 ± 0.04	98.58 ± 1.14	99.51 ± 0.23	98.55 ± 0.93	98.55 ± 0.69	99.56 ± 0.74
10	85.72 ± 0.25	88.04 ± 3.42	92.41 ± 2.12	96.05 ± 1.14	95.58 ± 1.12	94.22 ± 1.01	99.48 ± 0.89
11	69.79 ± 3.63	63.09 ± 5.63	95.28 ± 0.42	92.36 ± 1.74	96.13 ± 1.45	94.22 ± 1.62	85.05 ± 1.12
12	88.41 ± 10.13	93.75 ± 1.24	99.93 ± 0.02	99.74 ± 0.22	99.15 ± 0.21	94.68 ± 0.95	95.76 ± 1.03
13	94.76 ± 3.93	86.80 ± 3.42	99.65 ± 0.12	95.26 ± 1.45	99.15 ± 0.23	87.81 ± 2.18	81.78 ± 1.21
14	88.07 ± 3.77	95.17 ± 2.14	96.23 ± 1.14	98.19 ± 0.24	96.63 ± 1.45	82.26 ± 3.17	95.17 ± 2.06
15	71.88 ± 6.27	65.48 ± 3.45	71.59 ± 1.42	72.96 ± 2.45	92.36 ± 2.44	97.45 ± 0.46	99.59 ± 0.24
16	99.97 ± 0.04	95.01 ± 1.04	97.22 ± 2.45	98.27 ± 0.45	99.42 ± 0.07	99.15 ± 0.12	99.76 ± 0.06
OA	89.93 ± 0.54	87.88 ± 2.34	89.15 ± 1.45	92.30 ± 1.96	94.35 ± 1.01	95.84 ± 0.42	97.06 ± 0.38
AA	92.01 ± 0.74	90.67 ± 1.04	93.37 ± 1.15	95.36 ± 2.40	95.06 ± 1.23	95.27 ± 0.79	95.41 ± 0.66
Kappa	88.83 ± 0.59	86.57 ± 1.45	88.27 ± 2.40	91.44 ± 2.03	94.94 ± 1.12	95.37 ± 0.88	96.73 ± 0.74

As we can observe, although supervised learning methods have a strong supervision for the pre-training with the sufficient labels in the feature learning, their performance usually cannot compete with unsupervised learning methods. This is mainly because unsupervised learning methods get rid of the restriction of labels by constructing specific learning tasks, so that the models can acquire robust feature extraction capabilities by being pre-trained on numerous unlabeled samples. Among supervised learning methods,

the latest advanced method, Gia-CFSL, obtains the highest accuracy results, which mainly benefits from the joint training with abundant labels in the source data and few labels in the target data. By contrast, unsupervised learning methods do not require any labels in the pre-training of models and merely require few labels of target data to fine-tune the final classifier. Therefore, unsupervised learning methods usually can obtain more robust feature extraction capabilities in the pre-training, which is important for the superior classification performance. The performance of several unsupervised learning methods naturally surpasses the supervised learning methods. Among them, UM²L trains the deep relation network by constructing the unsupervised meta-learning tasks with multiview constraints, SMF-UL trains the spectrum motion network by constructing the unsupervised spectrum motion feature learning task, and both can better learn the discriminative feature contained in the hyperspectral data for small sample classification. Different from them, the proposed method, RULBP-MGCN, belongs to the generative self-supervised learning and pre-trains the model by reconstructing the graph-based features with masking, which can lead the graph convolutional network to better capture long-distance and large-range information relevancy by operating convolution on graph nodes. Furthermore, the RULBP features of hyperspectral data are used to calculate the k-nearest neighbor when constructing the graph so that the deep model can better use the spatial–spectral information. Therefore, the experimental results show that the RULBP-MGCN can obtain the highest classification accuracy in three HSI datasets among all comparative methods, which effectively verifies the superior performance of our method, RULBP-MGCN, in small samples classification.

An interesting and noteworthy result is that the proposed method improves the classification accuracy of the IP dataset much more than the other two datasets. Specifically, for the UP, SA, and IP datasets, the overall classification accuracy of the proposed method exceeds that of the second place by 0.45%, 1.22%, and 6.97%, respectively. As can be seen from Table 1, the IP dataset has a larger ground sampling distance (GSD), which means that the obtained HSI has a lower ground resolution, resulting in the increase in mixed pixels. In this case, the extraction and utilization of spatial information is particularly important, especially for the long-distance semantic information, since the discriminability of the spectral characteristic curve decreases with the decrease of ground resolution [4,5,49]. As mentioned above, the application of the masked strategy can improve the perception of the designed model of the semantic information in the long-distance space of whole HSIs, and the connection between nodes in the graph structure contains the relation between the long-distance pixels in non-Euclidean space. Therefore, compared with other methods, the proposed method has a superior capability of capturing the long-distance semantic information, which we believe should be the main reason why the proposed method performs particularly well on the IP datasets with a greater GSD.

Furthermore, to distinctly display the classification effect, we produce the classification maps of different methods for qualitative comparison, as shown in Figures 4–6. As we can observe, compared with the large area misclassification and noises caused by the relatively low accuracy of other methods, our method can effectively reduce the noises and produce a classification map more close to the ground truth under the condition of a small sample. Simultaneously, the graph convolutional network's capacity of capturing long-distance and large-range information relevancy enables the classification maps to have better homogeneity and continuity in the area of the identical ground objects, for instance, the Bare Soil area in UP data, the Oats area in IP data, the Grape area in SA data, etc.

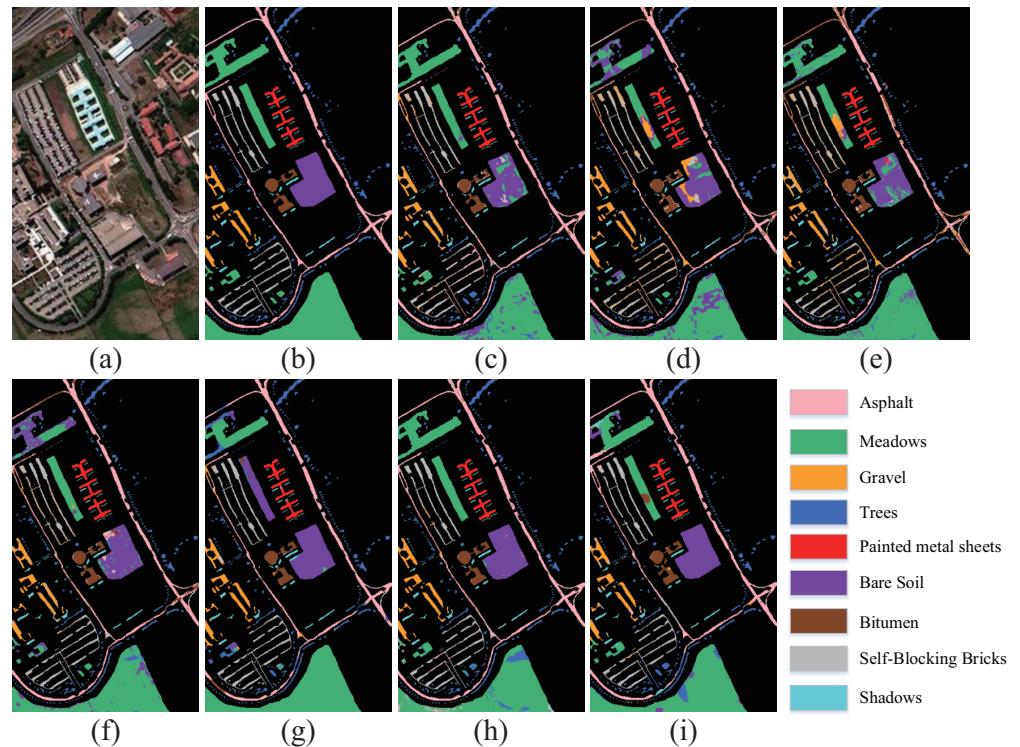


Figure 4. The classification maps of different methods on the UP data: (a) False-color image. (b) Ground-truth map. (c) DFSL. (d) RN-FSC. (e) CDPM. (f) Gia-CFSL. (g) UM²L. (h) SMF-UL. (i) RULBP-MGCN.

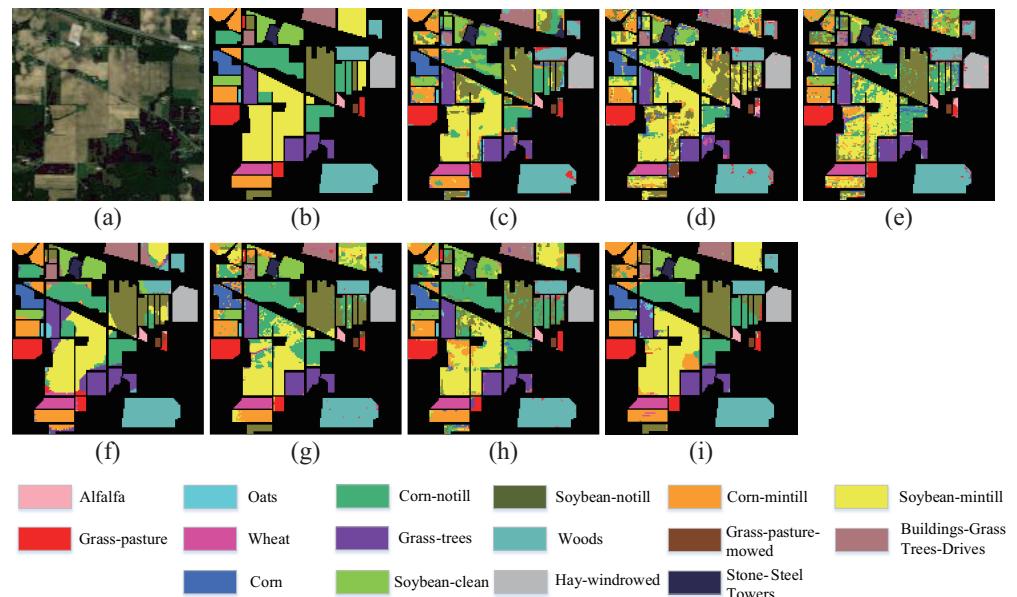


Figure 5. The classification maps of different methods on the IP data: (a) False-color image. (b) Ground-truth map. (c) DFSL. (d) RN-FSC. (e) CDPM. (f) Gia-CFSL. (g) UM²L. (h) SMF-UL. (i) RULBP-MGCN.

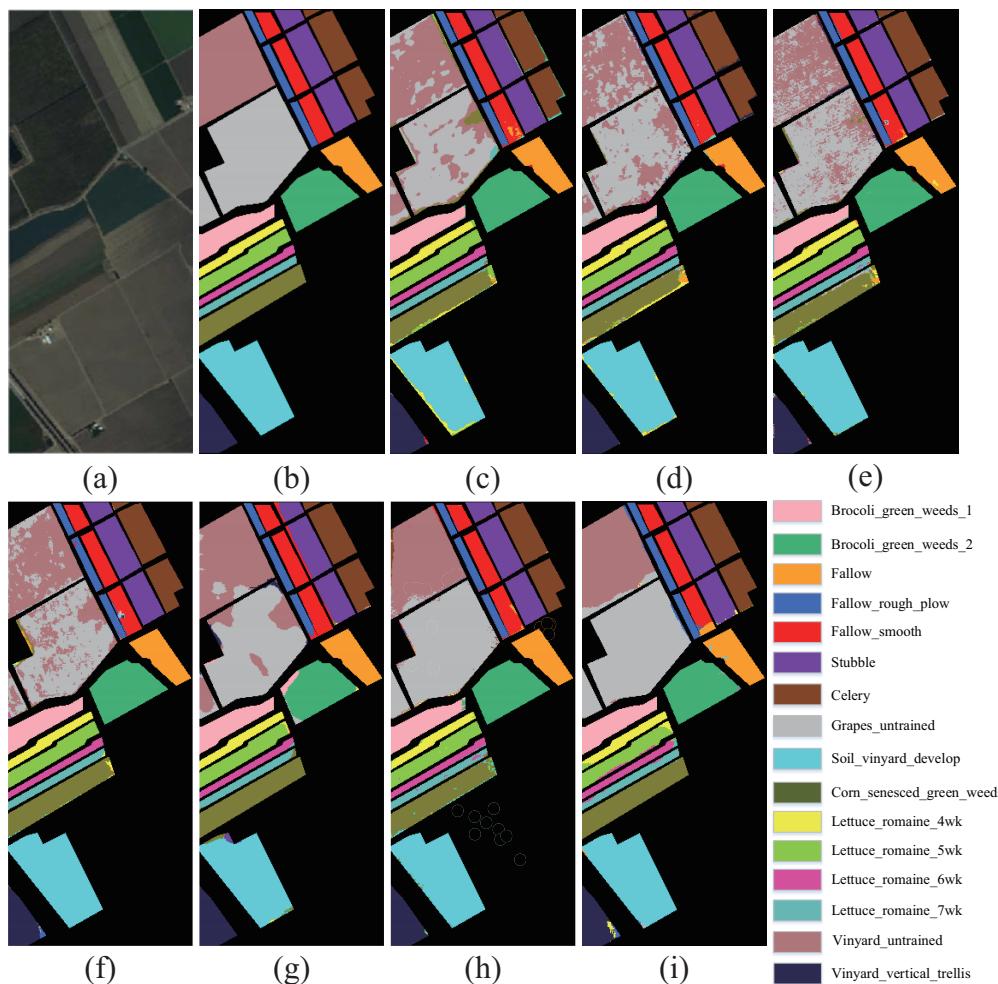


Figure 6. The classification maps of different methods on the SA data: (a) False-color image. (b) Ground-truth map. (c) DFSL. (d) RN-FSC. (e) CDPM. (f) Gia-CFSL. (g) UM²L. (h) SMF-UL. (i) RULBP-MGCN.

4.4. Influence of Labeled Training Samples

Because the pre-training of RULBP-MGCN does not require any labels, the fine-tuning of the linear classifier by labeled samples will determine the final classification performance of RULBP-MGCN. The comparison experiments of the last section have effectively demonstrated the superior performance of the proposed method under the condition of small samples. This section will further explore the performance of RULBP-MGCN varying with the number of labeled training samples, so we expand the size of training samples step by step when fine-tuning the liner classifier, and the results are shown in Figure 7. As we can observe, RULBP-MGCN can not only achieve excellent performance in small sample classification, but also obtain the stable improvement of performance in three hyperspectral datasets with the size of training samples expanding. This also shows its wide applicability in practice.

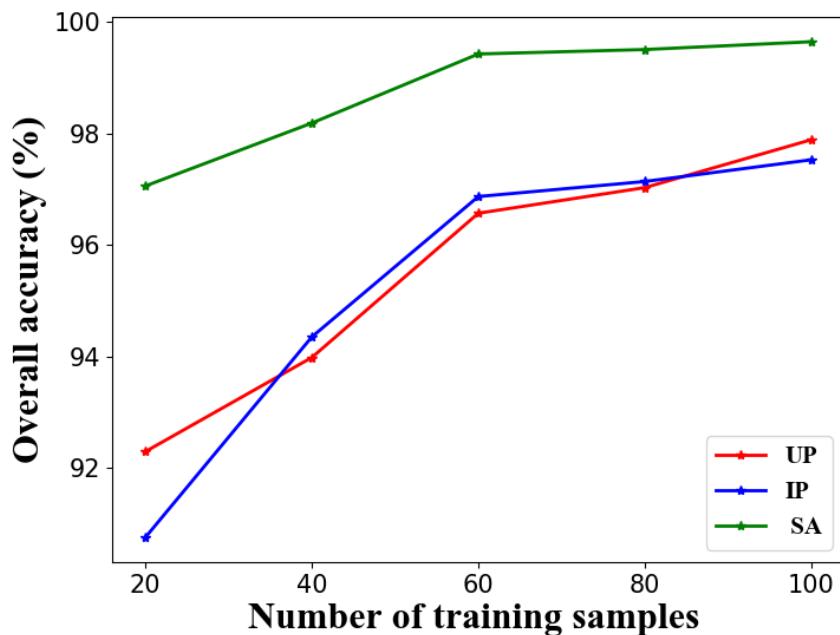


Figure 7. The performance of RULBP-MGCN varying with the number of labeled training samples.

5. Analysis and Discussion

In this section, the influence of several important hyperparameters on the classification results of different HSI datasets is explored and analyzed in detail. In addition, the execution time of the proposed method is compared and analyzed with other methods.

5.1. Influence of Different Feature Extraction Methods

In the proposed method, the feature extraction of HSIs is the first key step. Obviously, the effect of feature extraction determines the quality and level of information contained in the subsequent constructed graph. Therefore, this section explores the influence of different feature extraction techniques on the classification accuracy of the proposed method. Table 5 lists the results. Specifically, the PCA method retains the first 30 principal components, and the feature extraction processes of EMP and LBP are carried out according to Section 3.2. It can be seen that the classification performance with the PCA method, which retains the main spectral features as the basic information of graph construction, is significantly worse than that of other methods. In addition, the extraction of EMP and LBP features in HSIs for graph construction can significantly improve the classification accuracy, since the spatial structure information has been retained and utilized to a certain extent. Obviously, in the proposed method, conducting graph construction and self-supervised learning based on the RULBP features of HSIs can achieve the best classification performance.

Table 5. Influence of different feature extraction methods on classification accuracy.

	PCA	EMP	LBP	RULBP
UP	80.33	89.12	90.31	92.29
IP	81.36	87.69	89.06	90.75
SA	88.09	95.34	96.17	97.06

5.2. Influence of Dimension of RULBP Features

The introduction of RULBP features can effectively utilize the spatial texture information in HSIs. As shown in Figure 2, the RULBP features are extracted from each principal component, which means that the number of principal components directly determines the

dimension of RULBP features. To explore the influence of the dimension of RULBP features on classification results, the number of principal components is set as 1, 15, 30, and 45, which will generate the RULBP features with 11, 165, 330, and 495 dimensions, respectively. Table 6 shows the statistics results on the three HSI datasets. It can be seen that with the increase in dimension, the classification accuracy of the three datasets presents a trend of first rising and then decreasing, indicating that too large or too small a dimension value will lead to the decline in classification performance. For the UP dataset, the optimal value of dimension is 165, while the optimal value corresponding to the other two datasets is 330. As listed in Table 1, the number of bands in the SA and IP datasets is about twice that in the UP dataset, which can explain the above results from the perspective of the number of bands.

Table 6. Influence of the dimension of RULBP features on classification accuracy.

	11	165	330	495
UP	74.47	92.29	89.60	87.11
IP	72.97	84.95	90.75	90.87
SA	77.90	94.70	97.06	96.93

5.3. Influence of Number of Neighbors

The number of neighbors in the constructed graph will directly impact the performance and computational complexity of the graph convolutional network. Therefore, this section further explores the influence of the number of neighbors, so we adjust the number of neighbors when constructing the graph to pre-train the model, and the final classification performance is shown in Figure 8. As we can observe, the performance of RULBP-MGCN can be improved on the three hyperspectral datasets when increasing the number of neighbors from 50 to 200 step by step. Therefore, within a reasonable range, the increase in the number of neighbors enables the model to capture more subtle dependency between the graph nodes. However, an excess of number of neighbors will lead to unnecessary and redundant dependency and higher computational complexity, which causes a performance degradation of the model. Thus, we finally determine the optimal number of neighbors to be 200 for all hyperspectral data.

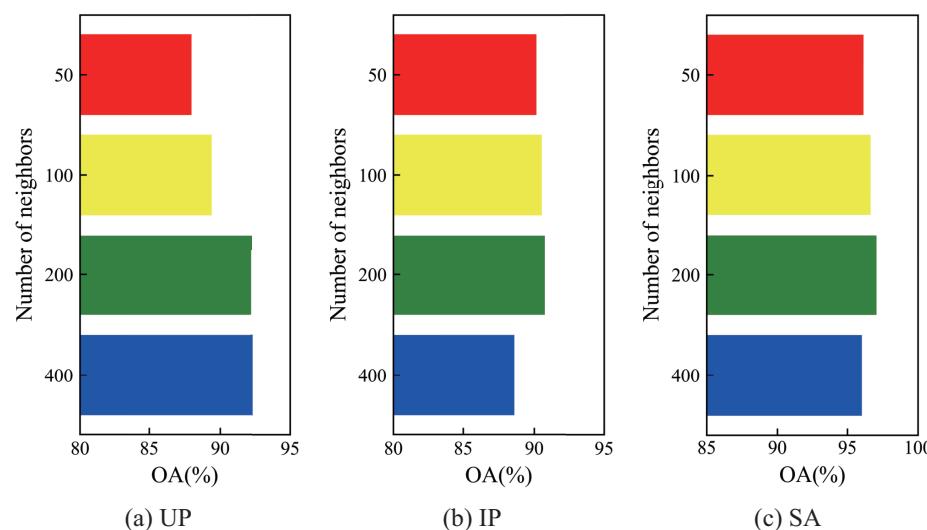


Figure 8. Influence of the number of neighbors on classification accuracy.

5.4. Influence of Mask Ratio

The mask ratio is one of the most important hyperparameters for generative self-supervised learning based on reconstruction. Currently, a lower mask ratio (<0.5) has been verified to be more effective in the NLP field [64,65]. Because language usually has

higher semantic content and is information-dense, a high mask ratio may lead the model to produce sophisticated language understanding when reconstructing the input. By contrast, images of nature always have high a redundancy of spatial information, which means that the model is capable of recovering the original spatial semantics even with fewer image patches. Thus, a higher mask ratio (>0.75) has been verified to be more effective in the CV field [66]. In this section, we adjust the mask ratio step by step to pre-train the model, and the final classification performance is shown in Figure 9. As we can observe, the optimal mask ratio is 0.5 for all three hyperspectral datasets for the RULBP-MGCN, and the higher and lower mask ratios both cause a slight performance degradation of the model. This is mainly because hyperspectral data not only have abundant spatial semantic information, but also contain complex spectral information, which means that a high mask ratio effective for the nature images will fail. This is also shows the unique characteristics of the graph-based generative self-supervised learning task for the hyperspectral data.

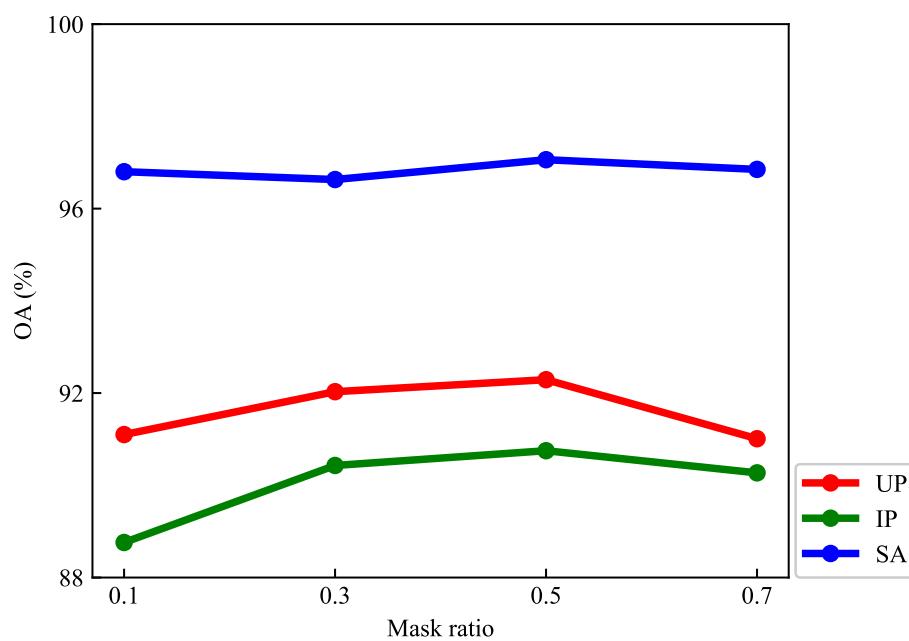


Figure 9. Influence of the mask ratio on classification accuracy.

5.5. Execution Efficiency

Finally, the implementation efficiency of the proposed method is compared and analyzed, and the results are shown in Table 7. For the first three methods, the pre-training phase is the task-based meta-training. In this stage, a large number of source labeled samples are used for model training, so the required execution time is longer. In contrast, the proposed method can organize the HSIs to be classified into the graph structure and utilize the efficient graph convolution operation for faster model training. This advantage is also reflected in the classification phase. For the first three methods, the training phase refers to fine-tuning the deep models after pre-training using the target labeled samples, while in this stage, the proposed method utilizes the extracted features to train a simple linear network as a classifier. The DFSL+SVM method uses a support vector machine as a classifier, so the required fine-tuning time is the shortest, and the training time required by the proposed method is still acceptable in this phase. On the whole, the proposed method has the shortest execution time and better classification performance, indicating that its practical application space is larger.

Table 7. Efficiency analysis of different methods.

Phases	DFSL + SVM	RN-FSC	UM ² L	RULBP-MGCN
Pre-training	118.23 min	319.84 min	316.87 min	23.89 min
Training	9.15 s	80.47 s	363.32 s	113.74 s
Classification	1.88 s	19.12 s	141.83 s	9.73 s

6. Conclusions

To improve the classification accuracy of HSIs in the case of a small sample, a graph-based self-supervised learning method with the RULBP features is proposed in this paper. First, the RULBP features of HSIs are extracted from the principal components to make full use of the spatial and spectral information. Then, the simple and effective K-nearest neighbor method is used for graph construction to convert the HSIs with grid structure into graph structure. Next, the mask strategy is adopted to conduct the self-supervised training based on the constructed graph to learn the deep features more suitable for small sample classification. Extensive experiments have been conducted on three public datasets, and the results demonstrate that the proposed method can achieve better classification results than the existing advanced methods. Specifically, the classification accuracy of the UP and SA datasets is improved by 0.45% and 1.22%, respectively. For the IP datasets with lower ground resolution, the improved performance brought by the proposed method is better, and the classification accuracy is improved by at least 6.97%.

Although the accuracy of HSI small sample classification is improved effectively, the proposed method is still trained in a single domain. Therefore, in future work, the domain adaptation and meta learning techniques will be introduced to further improve the generalization ability of the model between different datasets.

Author Contributions: Methodology, B.L. and W.L.; investigation, P.H., Q.H. and K.G.; resources, B.L. and K.G.; writing—original draft preparation, B.L.; writing—review and editing, W.L.; visualization, Q.H., K.G. and H.L.; supervision, W.L., B.L. and H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Joint Funds of the National Natural Science Foundation of China (No. U21A20109), the National Natural Science Foundation of China (No. 41901285, 42277478), and the Natural Science Foundation of Henan Province (No. 222300420387).

Data Availability Statement: Publicly available datasets were analyzed in this study, which can be found here: https://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes (accessed on 8 January 2023).

Acknowledgments: The authors would like to thank all the professionals for kindly providing the codes associated with the experiments.

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

References

- Li, J.; Zhang, H.; Zhang, L.; Huang, X.; Zhang, L. Joint Collaborative Representation With Multitask Learning for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 5923–5936. [[CrossRef](#)]
- 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.* **2018**, *56*, 1909–1921. [[CrossRef](#)]
- Transon, J.; D’Andrimont, R.; Maugnard, A.; Defourny, P. Survey of Hyperspectral Earth Observation Applications from Space in the Sentinel-2 Context. *Remote Sens.* **2018**, *10*, 157. [[CrossRef](#)]
- Paoletti, M.; Haut, J.; Plaza, J.; Plaza, A. Deep learning classifiers for hyperspectral imaging: A review. *ISPRS J. Photogramm. Remote Sens.* **2019**, *158*, 279–317. [[CrossRef](#)]
- Zhang, L.; Zhang, L.; Du, B. Deep Learning for Remote Sensing Data: A Technical Tutorial on the State of the Art. *IEEE Geosci. Remote Sens. Mag.* **2016**, *4*, 22–40. [[CrossRef](#)]
- Guan, R.; Li, Z.; Li, T.; Li, X.; Yang, J.; Chen, W. Classification of Heterogeneous Mining Areas Based on ResCapsNet and Gaofen-5 Imagery. *Remote Sens.* **2022**, *14*, 3216. [[CrossRef](#)]
- Shi, C.; Sun, J.; Wang, T.; Wang, L. Hyperspectral Image Classification Based on a 3D Octave Convolution and 3D Multiscale Spatial Attention Network. *Remote Sens.* **2023**, *15*, 257. [[CrossRef](#)]

8. Zhao, L.; Tan, K.; Wang, X.; Ding, J.; Liu, Z.; Ma, H.; Han, B. Hyperspectral Feature Selection for SOM Prediction Using Deep Reinforcement Learning and Multiple Subset Evaluation Strategies. *Remote Sens.* **2023**, *15*, 127. [[CrossRef](#)]
9. Zhang, Y.; Zhao, T.; Xie, B.; Mei, S. Hyperspectral Image Super-Resolution Classification with a Small Training Set Using Spectral Variation Extended Endmember Library. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 3001–3004. [[CrossRef](#)]
10. Gao, K.; Liu, B.; Yu, X.; Yu, A. Unsupervised Meta Learning With Multiview Constraints for Hyperspectral Image Small Sample set Classification. *IEEE Trans. Image Process.* **2022**, *31*, 3449–3462. [[CrossRef](#)]
11. Tang, C.; Liu, X.; Zhu, E.; Wang, L.; Zomaya, A.Y. Hyperspectral Band Selection via Spatial-Spectral Weighted Region-wise Multiple Graph Fusion-Based Spectral Clustering. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Montreal, QC, Canada, 19–27 August 2021; pp. 3038–3044. [[CrossRef](#)]
12. Shafaey, M.A.; Melgani, F.; Salem, M.A.M.; Al-Berry, M.N.; Ebied, H.M.; El-Dahshan, E.S.A.; Tolba, M.F. Pixel-Wise Classification of Hyperspectral Images With 1D Convolutional SVM Networks. *IEEE Access* **2022**, *10*, 133174–133185. [[CrossRef](#)]
13. Özdemir, A.O.B.; Gedik, B.E.; Çetin, C.Y.Y. Hyperspectral classification using stacked autoencoders with deep learning. In Proceedings of the 2014 6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Lausanne, Switzerland, 24–27 June 2014; pp. 1–4. [[CrossRef](#)]
14. Li, J. Active learning for hyperspectral image classification with a stacked autoencoders based neural network. In Proceedings of the 2015 7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Tokyo, Japan, 2–5 June 2015; pp. 1–4. [[CrossRef](#)]
15. Zhong, P.; Gong, Z.; Li, S.; Schönlieb, C.B. Learning to Diversify Deep Belief Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3516–3530. [[CrossRef](#)]
16. Lee, H.; Kwon, H. Going Deeper With Contextual CNN for Hyperspectral Image Classification. *IEEE Trans. Image Process.* **2017**, *26*, 4843–4855. [[CrossRef](#)] [[PubMed](#)]
17. Zhang, S.; Xu, M.; Zhou, J.; Jia, S. Unsupervised Spatial-Spectral CNN-Based Feature Learning for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–17. [[CrossRef](#)]
18. Liang, L.; Zhang, S.; Li, J. Multiscale DenseNet Meets With Bi-RNN for Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 5401–5415. [[CrossRef](#)]
19. Huang, L.; Chen, Y. Dual-Path Siamese CNN for Hyperspectral Image Classification With Limited Training Samples. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 518–522. [[CrossRef](#)]
20. Ahmad, M.; Khan, A.M.; Mazzara, M.; Distefano, S.; Ali, M.; Sarfraz, M.S. A Fast and Compact 3-D CNN for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [[CrossRef](#)]
21. Ghaderizadeh, S.; Abbasi-Moghadam, D.; Sharifi, A.; Zhao, N.; Tariq, A. Hyperspectral Image Classification Using a Hybrid 3D-2D Convolutional Neural Networks. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 7570–7588. [[CrossRef](#)]
22. Roy, S.K.; Krishna, G.; Dubey, S.R.; Chaudhuri, B.B. HybridSN: Exploring 3-D-2-D CNN Feature Hierarchy for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 277–281. [[CrossRef](#)]
23. 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](#)]
24. Gao, K.; Yu, X.; Tan, X.; Liu, B.; Sun, Y. Small sample classification for hyperspectral imagery using temporal convolution and attention mechanism. *Remote Sens. Lett.* **2021**, *12*, 510–519. [[CrossRef](#)]
25. Gao, K.; Liu, B.; Yu, X.; Qin, J.; Zhang, P.; Tan, X. Deep Relation Network for Hyperspectral Image Few-Shot Classification. *Remote Sens.* **2020**, *12*, 923. [[CrossRef](#)]
26. Jia, S.; Jiang, S.; Lin, Z.; Li, N.; Xu, M.; Yu, S. A survey: Deep learning for hyperspectral image classification with few labeled samples. *Neurocomputing* **2021**, *448*, 179–204. [[CrossRef](#)]
27. Liu, B.; Yu, X.; Zhang, P.; Tan, X.; Yu, A.; Xue, Z. A semi-supervised convolutional neural network for hyperspectral image classification. *Remote Sens. Lett.* **2017**, *8*, 839–848. [[CrossRef](#)]
28. Zhan, Y.; Hu, D.; Wang, Y.; Yu, X. Semisupervised Hyperspectral Image Classification Based on Generative Adversarial Networks. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 212–216. [[CrossRef](#)]
29. Liu, B.; Yu, A.; Yu, X.; Wang, R.; Gao, K.; Guo, W. Deep Multiview Learning for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 7758–7772. [[CrossRef](#)]
30. Liu, B.; Yu, X.; Yu, A.; Zhang, P.; Wan, G.; Wang, R. Deep Few-Shot Learning for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 2290–2304. [[CrossRef](#)]
31. Zuo, X.; Yu, X.; Liu, B.; Zhang, P.; Tan, X.; Wei, X. Graph inductive learning method for small sample classification of hyperspectral remote sensing images. *Eur. J. Remote Sens.* **2020**, *53*, 349–357. [[CrossRef](#)]
32. Hong, D.; Gao, L.; Yao, J.; Zhang, B.; Plaza, A.; Chanussot, J. Graph Convolutional Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 5966–5978. [[CrossRef](#)]
33. Zhao, X.; Niu, J.; Liu, C.; Ding, Y.; Hong, D. Hyperspectral Image Classification Based on Graph Transformer Network and Graph Attention Mechanism. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [[CrossRef](#)]
34. Kang, X.; Xiang, X.; Li, S.; Benediktsson, J.A. PCA-Based Edge-Preserving Features for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 7140–7151. [[CrossRef](#)]

35. Zhou, L.; Ma, X.; Wang, X.; Hao, S.; Ye, Y.; Zhao, K. Shallow-to-Deep Spatial-Spectral Feature Enhancement for Hyperspectral Image Classification. *Remote Sens.* **2023**, *15*, 261. [[CrossRef](#)]
36. Li, F.; Xu, L.; Wong, A.; Clausi, D.A. Feature Extraction for Hyperspectral Imagery via Ensemble Localized Manifold Learning. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 2486–2490. [[CrossRef](#)]
37. Park, S.; Lee, H.J.; Ro, Y.M. Adversarially Robust Hyperspectral Image Classification via Random Spectral Sampling and Spectral Shape Encoding. *IEEE Access* **2021**, *9*, 66791–66804. [[CrossRef](#)]
38. Hu, X.; Wang, X.; Zhong, Y.; Zhao, J.; Luo, C.; Wei, L. SPNet: A Spectral Patching Network for End-To-End Hyperspectral Image Classification. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 963–966. [[CrossRef](#)]
39. Liu, B.; Gao, K.; Yu, A.; Ding, L.; Qiu, C.; Li, J. ES2FL: Ensemble Self-Supervised Feature Learning for Small Sample Classification of Hyperspectral Images. *Remote Sens.* **2022**, *14*, 4236. [[CrossRef](#)]
40. Sun, Y.; Liu, B.; Yu, X.; Yu, A.; Gao, K.; Ding, L. From Video to Hyperspectral: Hyperspectral Image-Level Feature Extraction with Transfer Learning. *Remote Sens.* **2022**, *14*, 5118. [[CrossRef](#)]
41. Fauvel, M.; Benediktsson, J.A.; Chanussot, J.; Sveinsson, J.R. Spectral and Spatial Classification of Hyperspectral Data Using SVMs and Morphological Profiles. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 3804–3814. [[CrossRef](#)]
42. Huang, K.K.; Ren, C.X.; Liu, H.; Lai, Z.R.; Yu, Y.F.; Dai, D.Q. Hyperspectral Image Classification via Discriminant Gabor Ensemble Filter. *IEEE Trans. Cybern.* **2022**, *52*, 8352–8365. [[CrossRef](#)]
43. Li, W.; Chen, C.; Su, H.; Du, Q. Local Binary Patterns and Extreme Learning Machine for Hyperspectral Imagery Classification. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 3681–3693. [[CrossRef](#)]
44. Khodadadzadeh, M.; Li, J.; Plaza, A.; Ghassemian, H.; Bioucas-Dias, J.M.; Li, X. Spectral–Spatial Classification of Hyperspectral Data Using Local and Global Probabilities for Mixed Pixel Characterization. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 6298–6314. [[CrossRef](#)]
45. Khodadadzadeh, M.; Rajabi, R.; Ghassemian, H. A novel approach for spectral-spatial classification of hyperspectral data based on SVM-MRF method. In Proceedings of the 2011 IEEE International Geoscience and Remote Sensing Symposium, Vancouver, BC, Canada, 24–29 July 2011; pp. 1890–1893. [[CrossRef](#)]
46. Zhang, W.; Du, P.; Lin, C.; Fu, P.; Wang, X.; Bai, X.; Zheng, H.; Xia, J.; Samat, A. An Improved Feature Set for Hyperspectral Image Classification: Harmonic Analysis Optimized by Multiscale Guided Filter. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 3903–3916. [[CrossRef](#)]
47. Dundar, T.; Ince, T. Sparse Representation-Based Hyperspectral Image Classification Using Multiscale Superpixels and Guided Filter. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 246–250. [[CrossRef](#)]
48. Liu, B.; Yu, X. Patch-Free Bilateral Network for Hyperspectral Image Classification Using Limited Samples. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 10794–10807. [[CrossRef](#)]
49. Zhang, M.; Li, W.; Du, Q. Diverse Region-Based CNN for Hyperspectral Image Classification. *IEEE Trans. Image Process.* **2018**, *27*, 2623–2634. [[CrossRef](#)] [[PubMed](#)]
50. Gao, K.; Liu, B.; Yu, X.; Zhang, P.; Tan, X.; Sun, Y. Small sample classification of hyperspectral image using model-agnostic meta-learning algorithm and convolutional neural network. *Int. J. Remote Sens.* **2021**, *42*, 3090–3122. [[CrossRef](#)]
51. Xu, Q.; Xiao, Y.; Wang, D.; Luo, B. CSA-MSO3DCNN: Multiscale Octave 3D CNN with Channel and Spatial Attention for Hyperspectral Image Classification. *Remote Sens.* **2020**, *12*, 188. [[CrossRef](#)]
52. 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.* **2018**, *56*, 847–858. [[CrossRef](#)]
53. Ma, A.; Filippi, A.M.; Wang, Z.; Yin, Z.; Huo, D.; Li, X.; Güneralp, B. Fast Sequential Feature Extraction for Recurrent Neural Network-Based Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 5920–5937. [[CrossRef](#)]
54. Hang, R.; Liu, Q.; Hong, D.; Ghamisi, P. Cascaded Recurrent Neural Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5384–5394. [[CrossRef](#)]
55. 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.* **2022**, *60*, 1–16. [[CrossRef](#)]
56. Haut, J.M.; Paoletti, M.E.; Plaza, J.; Plaza, A.; Li, J. Visual Attention-Driven Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 8065–8080. [[CrossRef](#)]
57. Bai, J.; Wen, Z.; Xiao, Z.; Ye, F.; Zhu, Y.; Alazab, M.; Jiao, L. Hyperspectral Image Classification Based on Multibranch Attention Transformer Networks. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–17. [[CrossRef](#)]
58. Tan, X.; Gao, K.; Liu, B.; Fu, Y.; Kang, L. Deep global-local transformer network combined with extended morphological profiles for hyperspectral image classification. *J. Appl. Remote Sens.* **2021**, *15*, 038509. [[CrossRef](#)]
59. Xibing, Z.U.O.; Bing, L.I.U.; Xuchu, Y.U.; Pengqiang, Z.H.A.N.G.; Kuiliang, G.A.O.; Enze, Z.H.U. Graph convolutional network method for small sample classification of hyperspectral images. *Acta Geod. Et Cartogr. Sin.* **2021**, *50*, 1358. [[CrossRef](#)]
60. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *CoRR* **2018**, *1050*, 10–48550.
61. Zhang, Y.; Li, W.; Zhang, M.; Wang, S.; Tao, R.; Du, Q. Graph Information Aggregation Cross-Domain Few-Shot Learning for Hyperspectral Image Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *1*–14. [[CrossRef](#)]
62. Lee, H.; Eum, S.; Kwon, H. Exploring Cross-Domain Pretrained Model for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–12. [[CrossRef](#)]

63. Sun, Y.; Liu, B.; Yu, X.; Yu, A.; Gao, K.; Ding, L. Perceiving Spectral Variation: Unsupervised Spectrum Motion Feature Learning for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–17. [[CrossRef](#)]
64. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; (Long and Short Papers). Burstein, J., Doran, C., Solorio, T., Eds.; Association for Computational Linguistics: Toronto, ON, Canada, 2019; Volume 1, pp. 4171–4186. [[CrossRef](#)]
65. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. In Proceedings of the Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, Virtual, 6–12 December 2020; Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; Lin, H., Eds.; NeurIPS: New Orleans, LA, USA, 2020.
66. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R.B. Masked Autoencoders Are Scalable Vision Learners. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, 18–24 June 2022; pp. 15979–15988. [[CrossRef](#)]

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.