



## Article

# Data Augmentation and Spectral Structure Features for Limited Samples Hyperspectral Classification

Wenning Wang<sup>1,2,3,4,\*</sup>, Xuebin Liu<sup>1,3</sup> and Xuanqin Mou<sup>2</sup>

- <sup>1</sup> Key Laboratory of Spectral Imaging Technology of CAS, Xi'an Institute of Optics and Precision Mechanics, CAS, Xi'an 710119, China; lxb@opt.ac.cn
- <sup>2</sup> Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China; xqmou@mail.xjtu.edu.cn
- <sup>3</sup> University of Chinese Academy of Sciences, No.19(A) Yuquan Road, Beijing 100049, China
- <sup>4</sup> School of Information Science and Engineering, Shandong Agricultural University, Tai'an, Shandong, China
- \* Correspondence: wwn812@stu.xjtu.edu.cn; Tel.: +86-0538-8246257

**Abstract:** For both traditional classification and current popular deep learning methods, the limited sample classification problem is very challenging, and the lack of samples is an important factor affecting the classification performance. Our work includes two aspects. First, the unsupervised data augmentation for all hyperspectral samples not only improves the classification accuracy greatly with the newly added training samples, but also further improves the classification accuracy of the classifier by optimizing the augmented test samples. Second, an effective spectral structure extraction method is designed, and the effective spectral structure features have a better classification accuracy than the true spectral features.

**Keywords:** hyperspectral classification; data augmentation; structural features; small sample classification



**Citation:** Wang, W.; Liu, X.; Mou, X. Data Augmentation and Spectral Structure Features for Limited Samples Hyperspectral Classification. *Remote Sens.* **2021**, *13*, 547. <https://doi.org/10.3390/rs13040547>

Academic Editor: Jaime Zabalza  
Received: 4 January 2021  
Accepted: 1 February 2021  
Published: 3 February 2021

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



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

## 1. Introduction

In many remote sensing applications, hyperspectral image (HSI) classification is one of the important applications [1–3]. The classification of HSIs with small samples is a very challenging research topic. HSI classification aims to train the classifier according to some labeled pixel samples and then predict the labels corresponding to other pixel samples in the image to obtain the spatial distribution of different objects in the image. In the practical application of HSI classification, the manual labeling of pixel samples is time-consuming, laborious, and expensive, resulting in few labeled pixel samples. A small number of labeled training samples cannot fully describe the data distribution, leading to the over fitting phenomenon of the method based on deep neural networks and greatly reduces the classification accuracy. In recent years, although few works have proposed solutions [4–6], the classification accuracy remains limited due to the failure to fully mine the structural information in unlabeled samples.

Machine learning needs a large number of labeled data. When the number of labels is insufficient, semi supervised classification and data augmentation are possible solutions. Many scholars have conducted much research on semi supervised classification. As a very effective solution, semi-supervised learning (SSL) has attracted great attention for HSI classification [7]. Relevant studies have shown the effectiveness of spectral-spatial information-based SSL methods [8–11].

Data augmentation is mainly for training samples. In recent years, significant advancements have been achieved on the design of data augmentations for natural language processing (NLP) [12], vision [13,14], and speech [15,16] in supervised settings. Despite the promising results, data augmentation provides a limited effectiveness.

More recently, [17–19] described HS data augmentation techniques where pixels are grouped in blocks, and different block pairs are used as input to a convolutional neural

network (CNN). In [20], the samples in the true dataset were shifted along its first principal component or based on the average value in each band. The augmentation based on randomly erasing parts of the input patches was also proven effective for HSI classification in [21]. Finally, generative adversarial networks have been proposed recently as a data augmentation technique to generate new samples mimicking the distribution of the true data [22–24].

The unsupervised data augmentation proposed by Google not only augments the data of training samples, but also augments the data of unlabeled samples and achieves good results in image recognition and text classification. However, in the aspect of HS data classification, due to people's insufficient knowledge of HS data, only a few studies have been conducted on the augmentation of HS data [21,25]. In this study, the unsupervised data augmentation of HS data is attempted to improve the classification of small training samples.

The structural feature of the spectrum is a new feature, which is different from the spectral intensity feature. The use of continuous wavelet analysis [26] to extract vegetation information has achieved good results in vegetation moisture inversion and wheat pest diagnosis. Mexican hat feature extraction is considered ideal for classification, and good structural features can improve its classification accuracy.

Recently, deep learning methods have achieved excellent performance in many fields, such as data dimensionality reduction [27]. In 2014, deep learning methods even surpassed the human-level face recognition performance [28]. Deep learning methods aim to learn the representative and discriminative features of data in a hierarchical manner. In [29], Zhang et al. provided a technical tutorial for the application of deep learning in remote sensing. In [30], Chen et al. employed a deep learning method to handle HSI classification for the first time, where a stacked autoencoder was adopted to extract the deep features in an HSI. On the basis of this work, some improved autoencoder-based methods have been proposed, including the stacked denoising autoencoder [31,32], the stacked sparse autoencoder [33], and the convolutional autoencoder [34]. Research indicates that the CNN could also provide effective deep features for HSI classification. Some promising works were reported in [35–39]. However, deep-learning-based methods must usually train networks with a complex structure, leading to a time-consuming training process. In [40], Chan et al. proposed a simplified deep learning baseline called the PCA network (PCANet). PCANet is a much simpler network than CNN. PCA filters are chosen as the convolution filter bank in each layer, a binary quantization is used as the nonlinear layer, and the feature pooling layer is replaced by block-wise histograms of the binary codes. Though simple, experiments indicate that PCANet is already quite on par with and often better than state-of-the-art deep learning based features in many image classification tasks [41]. In [41], Pan et al. developed a simplified deep learning model for HSI classification based on PCANet. A deep learning framework based on the rolling guidance filter and the vertex component analysis network can achieve higher classification accuracy than traditional deep learning-based methods.

The main contributions of this paper are as follows:

First, a reasonable data augmentation method is designed for HS data, which not only verifies the effectiveness of the augmented training samples for classification, but also studies the effectiveness of the unlabeled samples for classification.

Second, a structural feature extraction method is designed, and the experimental results show that the structural feature is effective for classification.

Third, the experimental results on two HSI datasets show that the combination of data augmentation and structural features can effectively improve the accuracy of small sample HSI classification.

The general flow chart is shown in Figure 1.

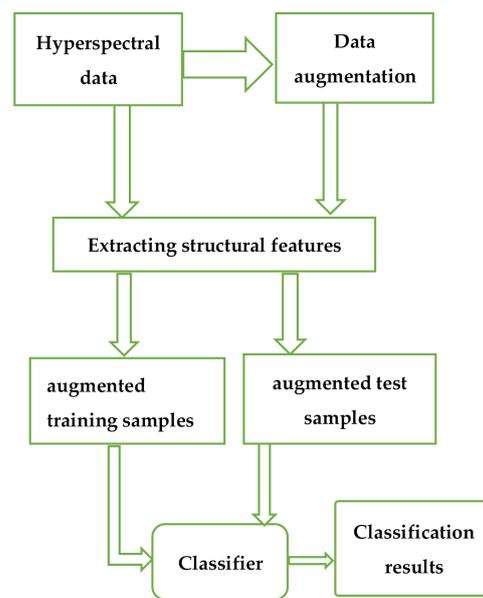


Figure 1. General flow chart.

## 2. The Proposed Data Augmentation

Data augmentation is an effective method of solving the problem of small samples. However, unlike image and text data, people’s cognition of HS data is not perfect, and the augmentation methods for image and text data are unsuitable for HS data. We hope that the added samples conform to the distribution model of the sample set, so the simple random disturbance and the augmented samples obtained by adding noise have little significance for classification.

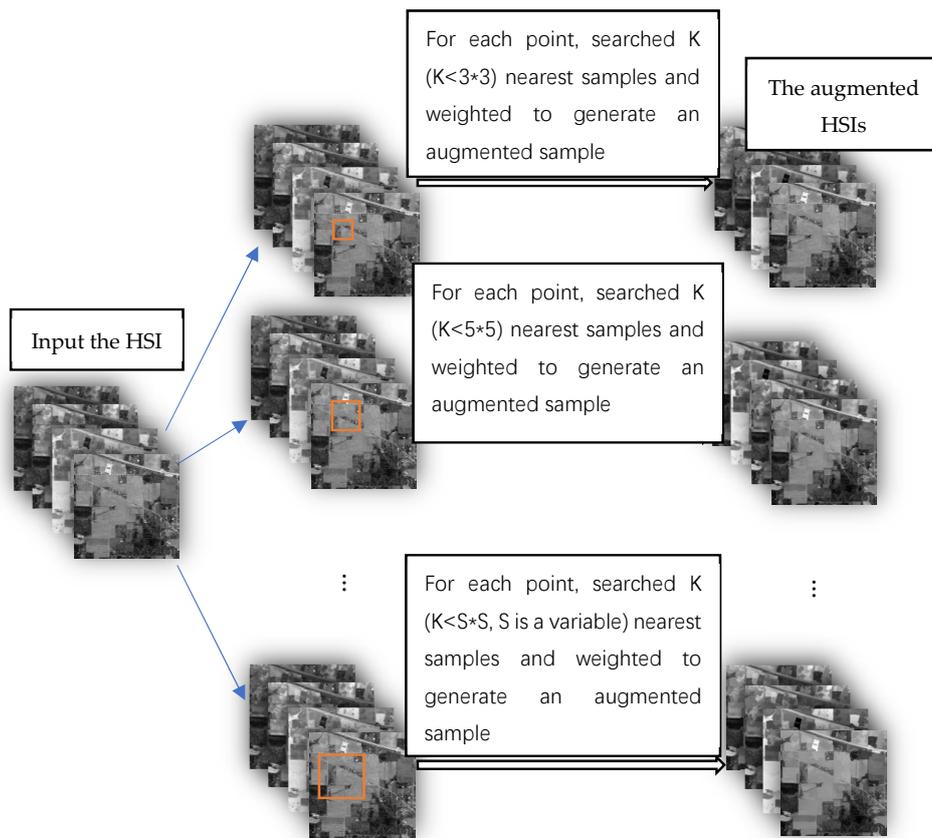
HSIs are often obtained from earth observation, and the distribution of ground objects has certain distribution rules. Our method is described as follows. For the current samples, we search for similar samples in the neighborhood of different window sizes to generate similar sample sets of different window sizes. The cluster center of the sample set moves to the real cluster center as the scale of the sample set increases. These clustering centers are in accordance with the distribution of the sample set, so we take different clustering centers as the augmented samples of each current sample.

### 2.1. Data Augmentation Based on Searching Similar Samples of Different Window Sizes

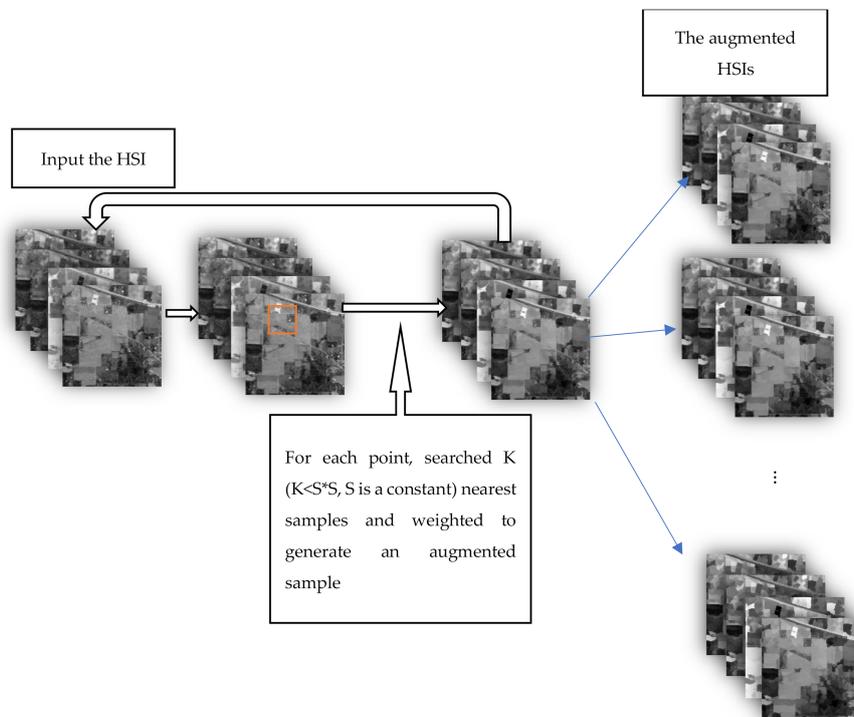
HSI is a natural image with certain distribution characteristics and has the aggregation distribution characteristics of a natural image. The neighborhood of each sample always has similar samples. The key to determining the quality of new samples is to accurately search the same sample set under different window sizes. In natural images, the same kind of samples may gather together in any region shape and may be adjacent to any kind of samples. Thus, determining whether the samples are of the same kind with a fixed threshold is impossible.

We use double conditions to search similar samples, that is, we select similar samples from quantity and similarity. In the neighborhood of rule samples, we first select the most likely samples. Then, according to the similarity between the candidate and current samples, the contribution weight of the sample to the cluster center is calculated. The samples with a large similarity have a greater contribution to the cluster center than those with small similarity.

According to the above principles, we design two data augmentation algorithms. The flow chart of the two data augmentation algorithms is shown in Figure 2.



(a)



(b)

**Figure 2.** The flow charts of the two data augmentation algorithms: (a) the flow chart of the data augmentation algorithm 1; and (b) the flow chart of the data augmentation algorithm 2.

Figure 2a is the flow chart of the data augmentation algorithm 1; Figure 2b is the flow chart of the data augmentation algorithm 2.

### 2.1.1. Data Augmentation Algorithm 1

Data augmentation algorithm 1 is as follows:

1. In the HSI, the current sample is the center point, and the neighborhood regions of different window sizes are defined. The size of the region is  $W = S \times S$ ,  $S = 3, 5, 7, 9, 11, \dots$

2. In the neighborhood region, the Euclidean distance between the neighborhood and current samples is calculated, and the samples in the neighborhood are sorted according to the distance. We select the nearest neighbor  $K$  ( $K < S \times S$ ) samples as candidate samples. The connectivity of the candidate samples is detected, and the sample points whose spatial positions are not adjacent are eliminated.

3. The contribution of the candidate sample to the augmentation sample is defined as weight. Weight  $\omega_{ij}$  is calculated using Equation (1). Weight  $\omega_{ij}$  is set for each candidate sample based on the similarity between each candidate sample and the current sample. If the similarity is large, then the weight is small. In Equation (1),  $X_j$  is the candidate sample, and  $X_i$  is the current sample.

$$\omega_{ij} = e^{\left(\frac{X_i \cdot X_j^T}{\sqrt{|X_i| \cdot |X_j|}} - 1\right)} \quad (1)$$

4. An augmentation sample is calculated using Equation (2), where  $X_{aug}$  is the augmentation sample of the current point.

$$X_{aug} = \left(\sum_j \omega_{ij} \cdot X_j\right) / \sum_j \omega_{ij} \quad (2)$$

5. Different-sized neighborhood windows are searched to obtain various candidate sample sets. Different candidate sample sets generate diverse augmentation sample sets of the current point. 6. The augmented samples of each pixel in the hyperspectral image are calculated to generate the augmented hyperspectral images.

### 2.1.2. Data Augmentation Algorithm 2

Data augmentation algorithm 2 is as follows:

1. In the HSI (after iteration, the HSI is replaced by the HSI augmentation data), the current sample is the center point. The size of window  $W$  is constant, that is,  $W = S \times S$ ,  $S = 7$

2. In the neighborhood region, the Euclidean distance between the neighborhood and current samples is calculated, and the samples in the neighborhood are sorted according to the distance. We select the nearest neighbor  $K$  ( $K < S \times S$ ) samples as candidate samples. The connectivity of the candidate samples is detected, and the sample points whose spatial positions are not adjacent are eliminated.

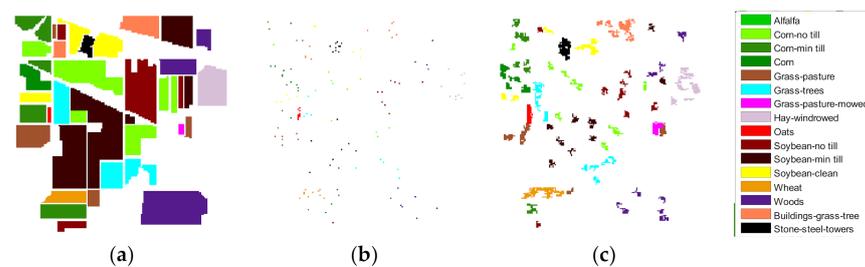
3. The contribution of the candidate sample to the augmentation sample is defined as weight  $\omega_{ij}$ . Weight  $\omega_{ij}$  is calculated using Equation (1). Weight  $\omega_{ij}$  is set for each candidate sample based on the similarity between each candidate sample and the current sample. If the similarity is large, then the weight is small. In Equation (1),  $X_j$  is the candidate sample, and  $X_i$  is the current sample.

4. An augmentation sample is calculated using Equation (2), where  $X_{aug}$  is the augmentation sample of the current point.

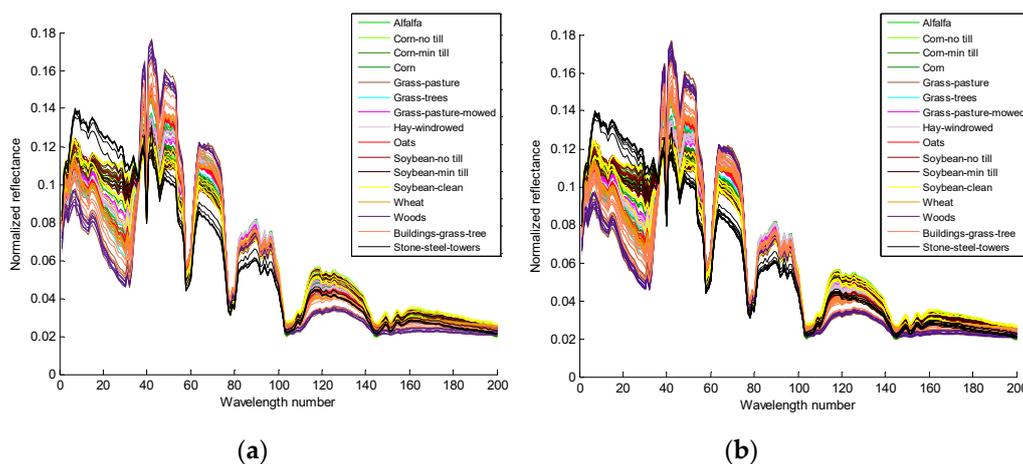
5. The augmented sample of each pixel in the hyperspectral image are calculated to generate the augmented hyperspectral image. Return to 1, the true hyperspectral image is replaced by the augmented hyperspectral image.

6. Every iteration, an augmented hyperspectral image generates. Assuming  $N$  iterations,  $N$  augmented hyperspectral images generate

Figure 3c shows that by selecting a limited quantity of candidate samples, the same class of samples can be selected from any shape area. The fewer the number of candidate samples that are selected, the purer is the sample set of the same selected class. However, if the quantity is extremely small, the quality of the augmented samples will be affected. Figure 4 presents the Indian Pines data, Figure 4a is the training samples graph composed of 80 true training samples (16 classes, 5 training samples of each class), and Figure 4b has 3 times the number of augmented training samples.



**Figure 3.** The Indian Pines data: (a) labeled image of Indian Pines data, (b) randomly selected samples, and (c) candidate samples of the same class.

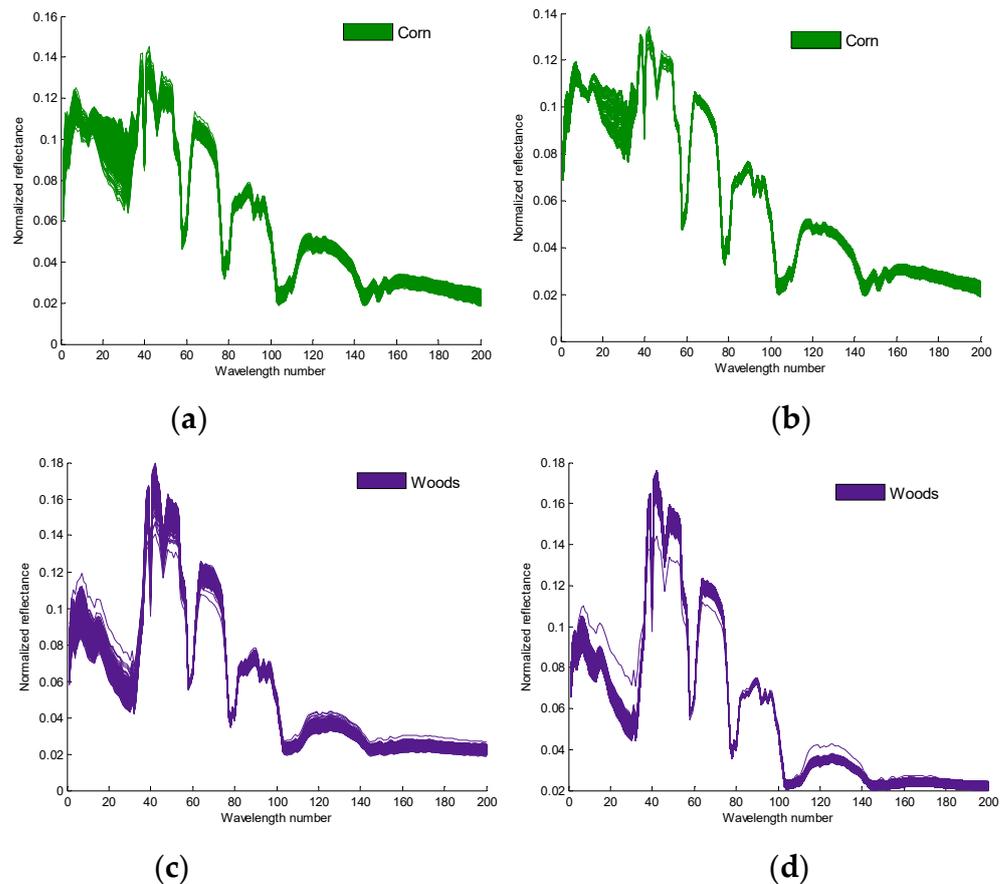


**Figure 4.** The spectral reflectance of the training samples in the Indian Pines data: (a) the training samples (the training samples include 16 classes, each class randomly takes 5 training samples, and each color represents a class), and (b) the 3 times augmented training samples.

## 2.2. Optimal Selection of Test Samples after Data Augmentation

Our data augmentation algorithm not only increases the training samples, but also generates new test samples. With the expansion of the region or the increase of the number of iterations, the new augmented samples tend to move to the cluster center, that is, gradually near the real cluster center of the sample set, and the samples near the cluster center tend to have higher classification accuracy.

For comparison, we draw the true and augmented samples on the same figure. Figure 4 shows that the clustering degree of the new samples is better than that of the true test sample. Figure 5a,c are the 4th and the 14th true spectral reflectance of the Indian Pines data, respectively; Figure 5b,d are the augmented spectral reflectance of the Indian Pines data by the third iteration of algorithm 2, respectively.



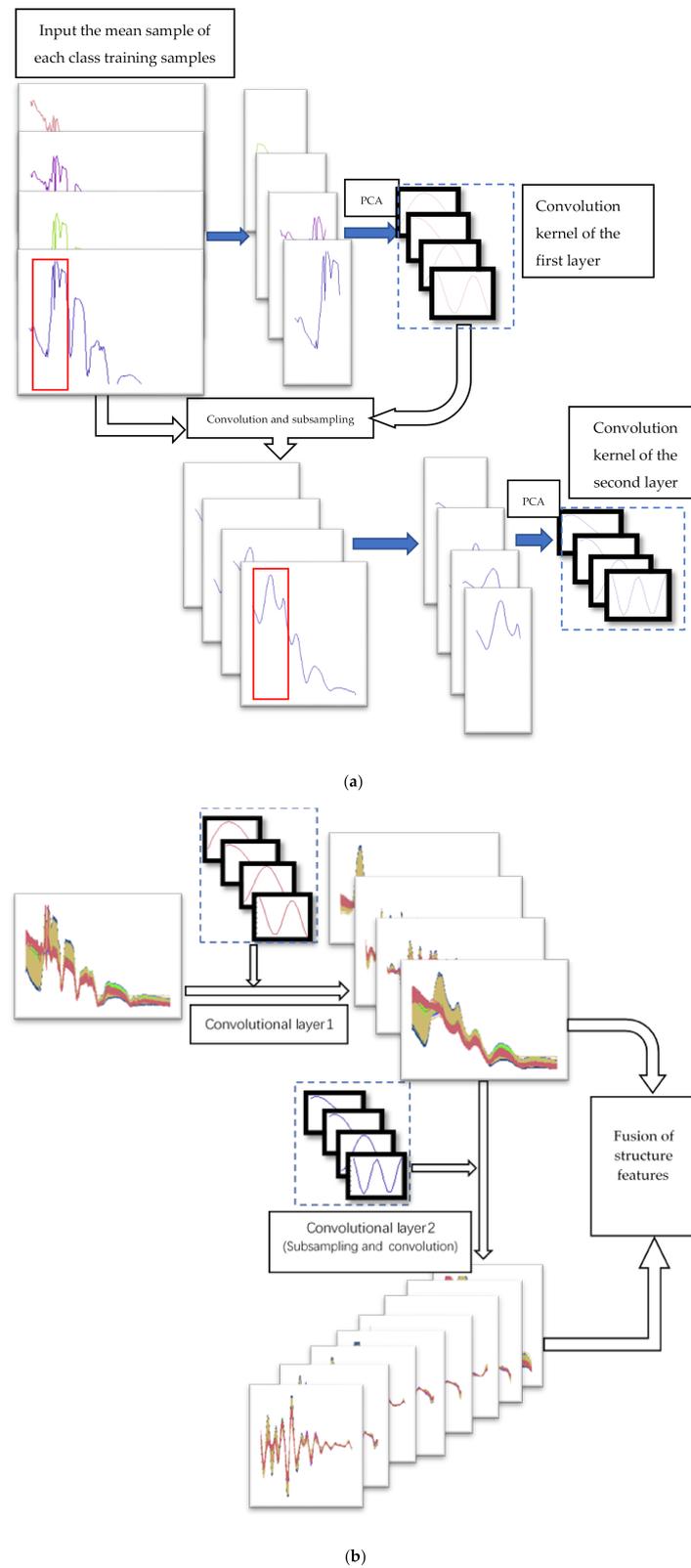
**Figure 5.** The true spectral reflectance in the Indian Pines data and the augmented spectral reflectance on the Indian Pines data by the third iteration of algorithm 2: (a) the spectral reflectance of the 4th true samples, (b) the spectral reflectance of the 4th augmented samples, (c) the spectral reflectance of the 14th true samples, and (d) the spectral reflectance of the 14th augmented samples.

In the fifth part of the experimental analysis, with the same classifier, the classification accuracy of the augmented test samples is significantly higher than that of the true HS test samples.

### 3. The Proposed Structural Feature Extraction

Structural features have many applications in the field of recognition. In image classification, handwritten numeral classification, and other fields, structural features are very important features. The deep learning developed in recent years also finds structural features conducive to classification at different scales. In the field of HS, some low-level structural feature extraction appeared in the early stage, such as the differential structure of the spectrum and structural feature extraction based on continuous wavelets [26], which prove the effectiveness of structural features.

PCANet is a concise deep learning method that generates convolution kernels to extract structural features from different layers. PCANet extracts the last layer of features for classification. However, we think that the features that are beneficial to classification may be the structural features from different scales. We improved PCANet and designed an effective structural feature extraction method, which we call the multi structure feature fusion (MSFF) method. The flow chart of the structure extraction is shown in Figure 6.



**Figure 6.** Flow chart of feature extraction of multi-layer structure: (a) the flow chart of generating the convolution kernel for structure feature extraction, and (b) the flow chart of structural feature extraction of hyperspectral data.

Our structural feature extraction algorithm is as follows:

1. Input the training samples. Calculate the mean value of each class of training samples to obtain the mean spectrum set of all classes,  $M_N = \{M_i\}$ ,  $i$  denotes the  $i^{\text{th}}$  class;  $M$  denotes the mean sample set of the  $N$  classes), and the size of the mean sample of each class is  $L$ . ( $L$  is the number of bands in the mean sample)

2. For a given  $M_i$ ,  $m_{ij}$  denote a patch of  $M_i$ ; collect all the patches (assuming size  $1 \times K$ , step is 1) around each band,  $\widehat{M}_i$  denotes a matrix of small patches,  $\widehat{M}_i$  can be expressed by Equation (3)

$$\widehat{M}_i = \begin{bmatrix} m_{i1} \\ \vdots \\ m_{i(L-K+1)} \end{bmatrix} \quad (3)$$

where each mean sample can obtain  $(L - K + 1)$  patches.

3. There are  $N$  mean samples available, a total of  $N \times (L - K + 1)$  vectored patches are available,  $\widehat{M}$  denotes a matrix of patches of all classes, and  $\widehat{M}$  can be expressed by Equation (4)

$$\widehat{M} = \begin{bmatrix} \widehat{M}_1 \\ \vdots \\ \widehat{M}_N \end{bmatrix} \quad (4)$$

where  $\widehat{M}$  is a matrix with  $N \times (L - K + 1)$  rows and  $K$  columns. The PCA principal components of the  $\widehat{M}$  matrix are calculated. Some eigenvectors with large eigenvalues are the convolution kernels of the first layer.

4. Use the convolution kernel of the first layer to extract the first volume structural features of the mean samples, and then subsample the structural features as the input of the second layer.

5. Return to the second step, and input the first volume structural features after subsampling. The size of the patch remains unchanged and the convolution kernels of the second layer are generated.

6. Use the two-layer convolution kernels to extract the structural features of the spectrum.

7. Fuse the two-tier structural features and dimension reduction into the classifier.

#### 4. MLR/SVM Classification

To test the advantages and universality of the proposed method for classification, two classifiers: multiple linear regression (MLR) classification and support vector machines (SVM) classification are used to classify the samples processed by our method. Experiments show that MLR and SVM are superior in classification with limited samples.

##### 4.1. The Classification of the Augmentation Hyperspectral Data (HS\_AUG)

The classification steps of the HS\_AUG data are as follows:

1. According to the augmentation algorithm, each pixel of the hyperspectral image can generate  $N$  (assuming  $N$  times of augmentation) independent augmentation samples. If the true sample is added, then each pixel has  $(N + 1)$  independent samples. Each of the  $(N + 1)$  samples independently represents the current pixel. If  $P$  pixels are taken as the training pixels, then we will have  $(N + 1) \times P$  training samples to train the classifier, and the number of training samples will be increased by  $(P + 1)$  times.

2. We use  $(N + 1) \times P$  training samples to train the classifier.

3. The final classification decision can be realized in two methods: First, each test pixel also has  $(N + 1)$  independent test samples, which are sent to the classifier for independent judgment, and the current pixel prediction result is determined using the voting method. Second, the last augmented test sample of the augmentation algorithm is selected as the test sample of the current pixel (the last augmented sample is closer to the cluster center) and then sent to the above classifier for prediction.

#### 4.2. The Classification of the Augmentation and Msff Structural Feature Hyperspectral Data (HS\_AUG\_MSFF)

The classification steps of the HS\_AUG\_MSFF data are as follows:

1. All hyperspectral data are augmented.
2. MSFF is used to extract the structural features of all data as the features of the hyperspectral data.
3. The number of training pixels is  $P$ , and the number of training samples is  $(N + 1) \times P$ , which are used to train the classifier.
4. The final classification decision can be realized in two methods: First, each test pixel also has  $(N + 1)$  independent test samples, which are sent to the classifier for independent judgment, and the current pixel prediction result is determined using the voting method. Second, the last augmented test sample of the augmentation algorithm is selected as the test sample of the current pixel (the last augmented sample is closer to the cluster center) and then sent to the above classifier for prediction.

### 5. Experiments and Discussion

In this section, we divide the experiment into three main parts: the data augmentation algorithm experiment, the experiment on the effectiveness of the structural features, and the comprehensive experiment on two datasets. The first two parts are the theoretical verification of the algorithm. We use MLR and SVM classifiers to verify the algorithm on the Indian Pines dataset. In the third and the fourth parts, we apply data augmentation and structural features to different data sets to calculate the classification sensitivity by MLR classification (in many experiments, the MLR classifier is better than the SVM classifier).

In the following figures and tables, each experimental data is the average of 30 experiments. All experiments are conducted in MATLAB running on an 8-GHz Intel(R) Core(TM) i5-2450M processor.

The Indian Pines and Pavia University hyperspectral data sets are adopted to demonstrate the effectiveness of the proposed method.

(1) Indian Pines dataset: This dataset is a hyperspectral image of an agricultural area collected by an airborne visible infrared imaging spectrometer (AVIRIS). The image size is  $145 \times 145$  pixels and the spatial resolution is 20 m. It contains 220 bands (wavelength range of 0.4–2.5  $\mu\text{m}$ ), in which 20 bands with severe water absorption are removed and 200-band hyperspectral data are retained. The data has a corresponding markup template. The dataset contains 10,366 samples with 16 known features. The high similarity of the feature spectra of the dataset and the uneven number of various sample types pose a great challenge for the high-precision classification of objects. Accordingly, this high similarity problem has been widely used in the performance testing of hyperspectral image classification algorithms.

(2) The Pavia University data set was acquired by the Reflective Optics System Imaging Spectrometer sensor during a flight campaign over Pavia, northern Italy. This data set consists of  $610 \times 340$  pixels, with a spatial resolution of 1.3 m per pixel. After 12 noisy spectral bands are removed, 103 spectral bands are retained in the experiments. The ground truth contains nine representative urban classes.

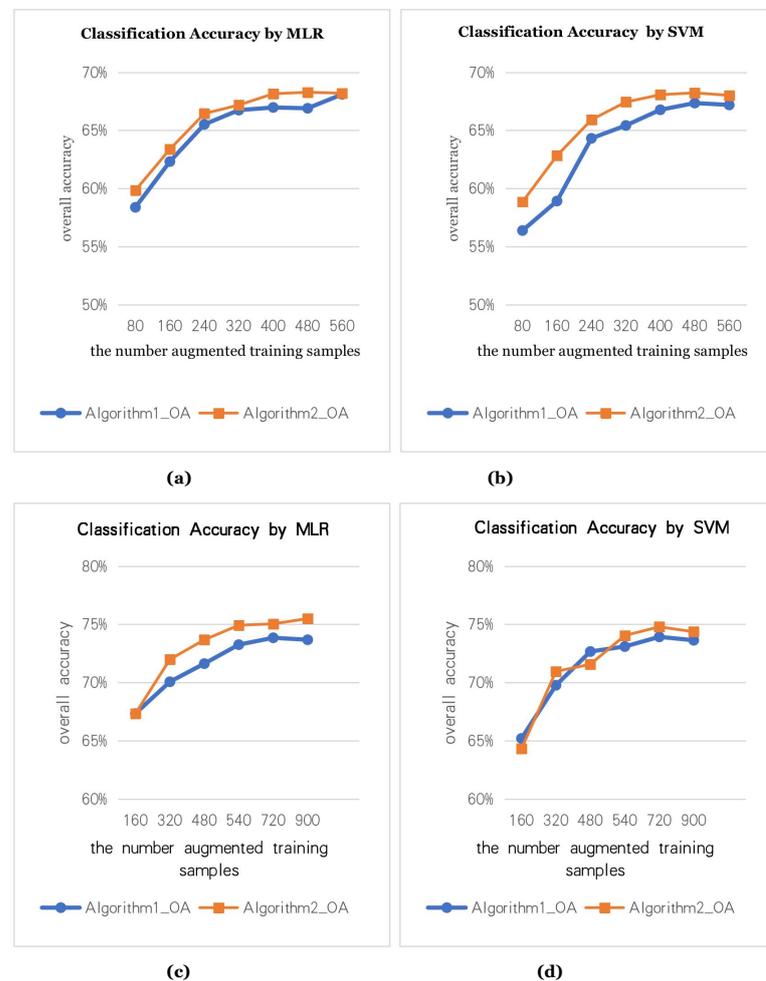
#### 5.1. Data Augmentation Experiment

In this part of the experiment, we verify two problems: the effectiveness of the augmented training samples and the optimal use of the augmented data.

##### 5.1.1. Effectiveness of Augmentation

We use the data augmentation algorithm to generate the augmented training samples and calculate the classification accuracy. In the true HS data, we select different numbers of true training samples, namely, 80 true training samples (5 for each class), 160 true training samples (10 for each class). Figure 7a,b are the classification accuracy graphs with 80 true training samples (5 for each class). Figure 7c,d are the classification accuracy graphs with

160 true training samples (10 for each class). Two data augmentation methods are used to augment the training samples to  $N$  times, where the X-axis is the number of the augmented samples. To test the effectiveness of the data augmentation algorithm, we use two classifiers (MLR, SVM) for the experiments.



**Figure 7.** Comparison of classification accuracy of two kinds of augmented training samples on the Indian Pines data: (a) the multiple linear regression (MLR) classification accuracy of the augmented training samples with 80 true training samples, (b) the support vector machine (SVM) classification accuracy of the augmented training samples with 80 true training samples, (c) the MLR classification accuracy of the augmented training samples with 160 true training samples, and (d) the SVM classification accuracy of the augmented training samples with 160 true training samples.

Figure 7 shows that the augmented training samples can effectively improve the classification accuracy. However, the augmentation of samples is not endless. After 3 to 4 augmentations, the classification accuracy stops improving.

In augmentation algorithm 1, the size of the neighborhood region is (3, 5, 7, 9, 11), that is, when generating augmented samples 4 to 5 times, the classification accuracy is good.

In augmentation algorithm 2, we set the size of the neighborhood region to 7. Then, the augmented samples generated by iterating 3 to 4 times have better classification accuracy.

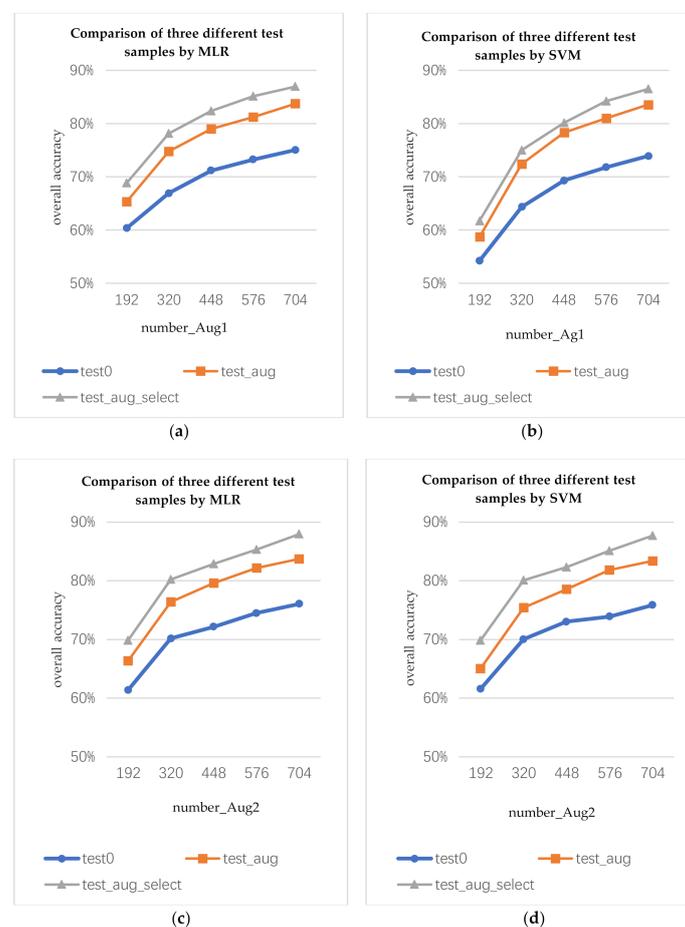
### 5.1.2. Optimal Utilization of Augmented Data

The data augmentation algorithm can generate not only augmented training samples, but also augmented test samples. In this section, we discuss how to use augmented test samples. We use the augmented training samples to train the classifier and classify the true, augmented, and selected augmented test data. The three test samples are marked as

test0, test\_aug, and test\_aug\_select, respectively. For the test\_aug, each pixel will obtain multiple prediction results. Here, we use the voting method to determine the prediction results of the current pixel. For test\_aug\_select, the prediction result of each pixel depends on the optimal augmented data.

The number of the augmented training samples are marked as the following representative variables: number\_Aug1 and number\_Aug2 (number\_Aug1 is the number of the augmented samples by data augmentation algorithm 1; number\_Aug2 is the number of the augmented samples by data augmentation algorithm 2). number\_Aug1 and number\_Aug2 are 4 times the number of the true training samples.

As shown in Figure 8, by using the same augmented training samples to train the classifiers, different test samples will produce different classification results. The classification accuracy of the augmented test samples is higher than that of the true test samples. The classification accuracy of the selected augmented test sample is the best.



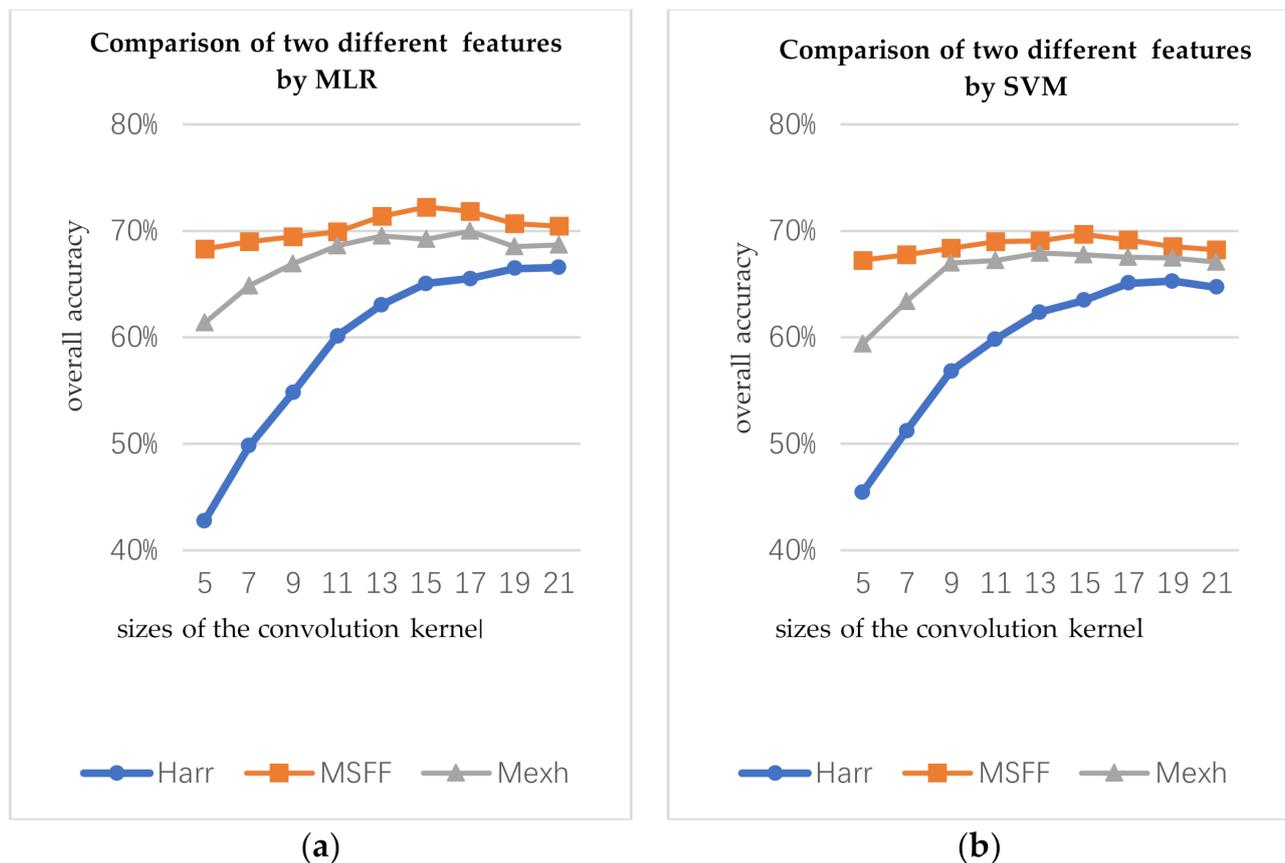
**Figure 8.** Comparison of classification results of three test samples on the Indian Pines data: (a) prediction results of MLR classifier trained by augmented training samples by data augmentation algorithm 1 for three test samples, (b) prediction results of SVM classifier trained by augmented training sample 1 for three test samples, (c) prediction results of MLR classifier trained by augmented training sample 2 for three test samples, and (d) prediction results of SVM classifier trained by augmented training sample 2 for three test samples.

### 5.2. Comparison of Classification Effects of Structure Characteristics at Different Scales

In the first part, the types of convolution kernels include Mexh wavelet, Haar wavelet, and the MSFF proposed in this work, and the scale of each convolution kernel is 5, 7, 9, 11, 13, 15, 17, 19, and 21, respectively. Mexh-OA represents the OA of the Mexh wavelet structural features. Haar-OA represents the OA of the Haar wavelet structural features.

MSFF-OA represents the OA of the MSFF structural features spectrum. The number of the training for three structural features is same (10 per class).

Figure 9 presents the relationship between the classification accuracy and scales of the convolution kernels. By classifying the experiments according to the same conditions, the OA of the MSFF structural features are approximately 2% to 3% higher than that of Mexh wavelet structural features.



**Figure 9.** Classification accuracies for the different structure features of different scales on the Indian Pines data: (a) comparison of classification results of three features in the MLR classifier and (b) comparison of classification results of three features in the SVM classifier.

### 5.3. Comprehensive Experiment

In this section, we combine data augmentation and structural features to classify on two datasets (Indian Pines and paviaU). Given that the classification performance of the MLR classifier is better than that of the SVM, we only use the MLR classifier to perform classification experiments, calculate the classification sensitivity of the data, and draw the prediction image.

With the same number of true training samples, the classification results of the four kinds of data are compared: the true HS data; MSFF data; augmented HS data, comprehensive data augmentation and MSFF structural features data.

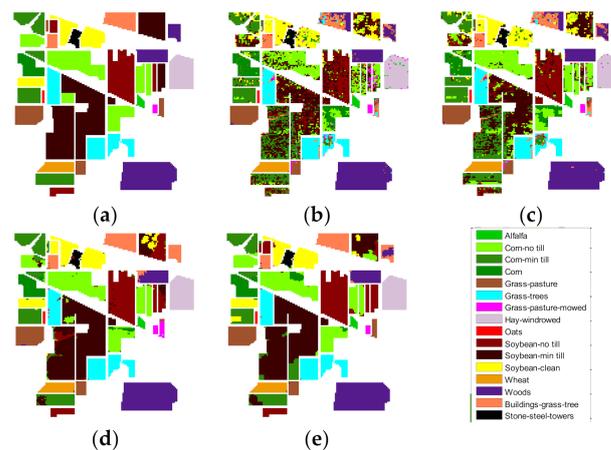
Table 1 shows the classification sensitivity of Indian Pines data, with 10 true training samples selected for each class. Table 2 shows the classification sensitivity of paviaU, with 10 true training samples selected for each class. Figures 10 and 11 are the classified prediction images.

**Table 1.** Classification accuracies per class (%) for different features, by MLR on the Indian Pines data.

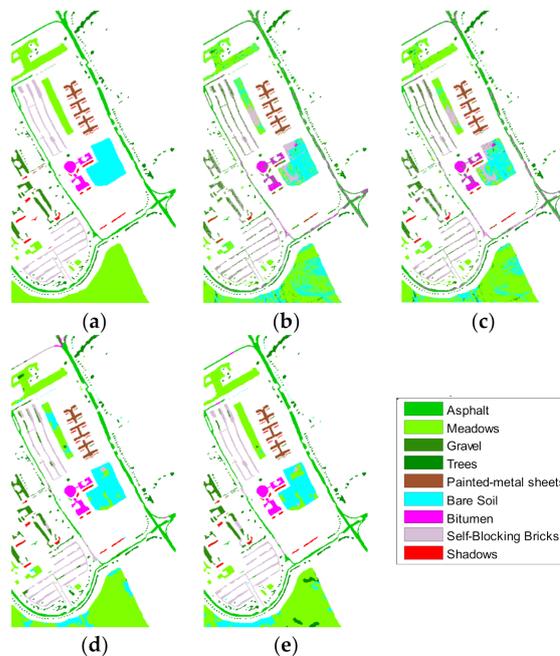
Labels	Class	HS	MSFF	HS_AUG	HS_AUG_MSFF
1	Alfalfa	77.27	82.87	98.38	96.03
2	Corn-no till	69.81	74.34	91.56	92.64
3	Corn-min till	51.12	55.12	80.44	82.40
4	Corn	40.97	42.48	72.66	74.95
5	Grass-pasture	89.07	87.52	96.47	96.61
6	Grass-trees	94.62	95.54	98.62	98.27
7	Grass-pasture-mowed	45.13	60.01	71.10	68.49
8	Hay-windrowed	98.79	99.14	99.82	99.85
9	Oats	39.26	46.88	67.19	76.34
10	Soybean-no till	53.69	59.60	74.05	76.58
11	Soybean-min till	75.33	80.38	94.39	94.65
12	Soybean-clean	63.92	66.28	85.30	89.37
13	Wheat	96.15	97.68	98.66	97.87
14	Woods	95.49	95.66	99.11	98.92
15	Buildings-grass-tree	62.90	62.81	79.88	79.13
16	Stone-steel-towers	88.36	94.26	85.99	88.54
	AA	71.37	75.03	85.71	86.12
	OA	71.78	74.78	86.42	87.76
	Kappa	68.33	71.69	85.03	86.20

**Table 2.** Classification accuracies per class (%) of different features by MLR on the PaviaU data.

Labels	Class	HS	MSFF	HS_AUG	HS_AUG_MSFF
1	Asphalt	85.61	87.31	97.36	98.40
2	Meadows	88.73	89.44	95.99	94.24
3	Gravel	49.16	52.19	76.38	88.85
4	Trees	69.63	70.14	92.71	93.66
5	Painted-metal sheets	99.19	99.29	99.11	99.63
6	Bare Soil	46.42	46.90	69.29	70.93
7	Bitumen	41.41	45.39	81.71	86.13
8	Self-Blocking Bricks	61.29	64.73	81.13	84.74
9	Shadows	99.36	99.33	99.68	99.89
	AA	71.20	72.75	85.55	86.69
	OA	70.53	72.20	84.72	85.85
	Kappa	60.67	64.77	81.95	83.99



**Figure 10.** Classification results of different features on Indian Pines data: (a) label data, (b) prediction image of true spectral data, (c) prediction image of MSFF data, (d) prediction image of augmented data, and (e) prediction image of data combining data augmentation and MSFF.



**Figure 11.** Classification results of different features on PaviaU data: (a) label data, (b) prediction image of true spectral data, (c) prediction image of multi structure feature fusion (MSFF) data, (d) prediction image of augmented data, and (e) prediction image of data combining data augmentation and MSFF.

#### 5.4. Comparative Experiments with Other Algorithms

We compare the proposed methods (HS\_AUG/HS\_AUG\_MSFF) with several methods, such as hyperspectral image classification using joint sparse model and discontinuity preserving relaxation (JSPR) [42], a new deep learning-based hyperspectral image classification method RVCANET) [41], spectral and spatial classification of hyperspectral images based on random multi-graphs (HC-RMG) [43], and spectral–spatial hyperspectral classification via structural–kernel collaborative representation (SKCR) [44].

The algorithms we use for comparison are comparable with the contents of this study. For example, RVCANET is a classification method based on a deep learning NN. The source codes of these methods have been provided by the true authors in literature.

During the experiment, the numbers of training samples used for all the codes are uniformly modified to 5 and 10 for each class. All methods are implemented 30 times, and

their average results are calculated. Tables 3 and 4 present the details of the experimental data. Figures 12 and 13 show the results of a single experiment conducted on all the methods, using 10 training samples per class. The experiments show the superiority of the OA of the proposed method over those of existing methods with limited samples. The running time of all experiments was measured by MATLAB on an 8-GHz Intel (R) core (TM) i5-2450m processor.

**Table 3.** Comparison of classification accuracies (%) of different algorithms for the Indian Pines data.

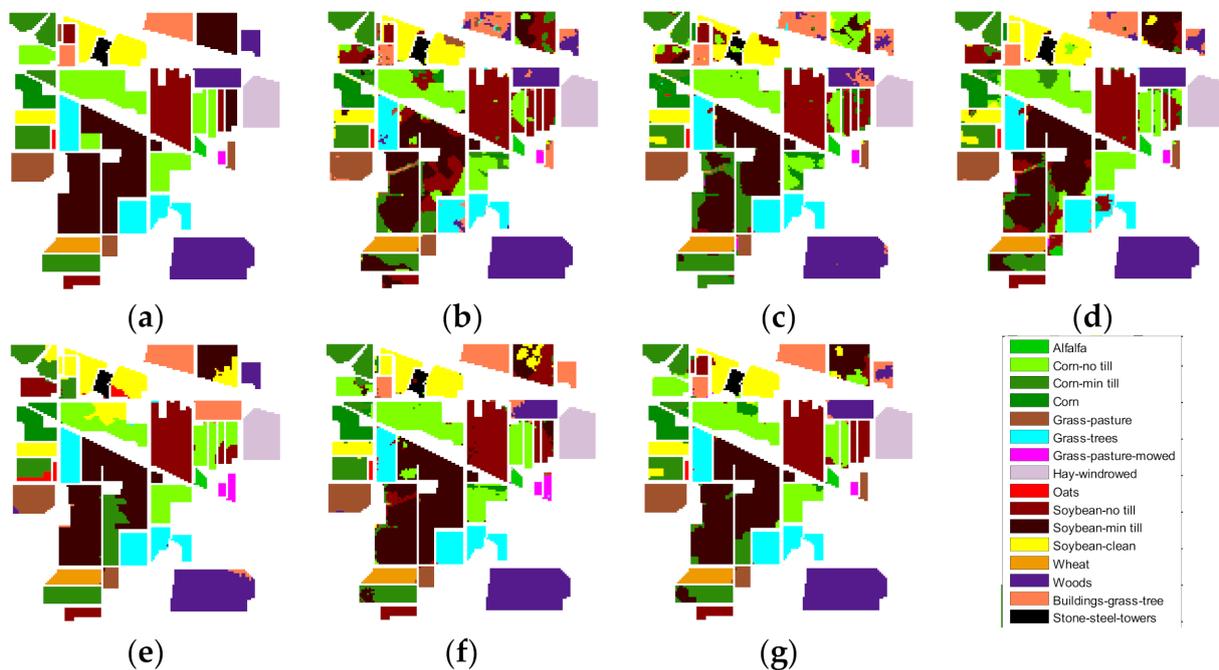
Algorithm	Evaluation Parameters	5	10
JSPR	AA	82.57	87.48
	OA	72.52	76.31
	Kappa	69.40	74.55
	T(s)	60.13	151.38
RVCANET	AA	81.56	87.46
	OA	69.38	78.65
	Kappa	65.76	76.52
	T(s)	3400.45	3420.06
HC-RMG	AA	72.57	82.48
	OA	70.52	83.31
	Kappa	69.40	82.55
	T(s)	10.13	10.38
SKCR	AA	80.82	86.25
	OA	72.89	84.24
	Kappa	69.76	82.25
	T(s)	1.73	2.52
HS_AUG	AA	77.69	85.71
	OA	79.15	86.42
	Kappa	76.87	85.03
	T(s)	56.36	58.65
HS_AUG_MSFF	AA	78.95	86.12
	OA	80.36	87.76
	Kappa	77.68	86.20
	T(s)	57.41	58.23

**Table 4.** Comparison of classification accuracies (%) of different algorithms for the PaviaU data.

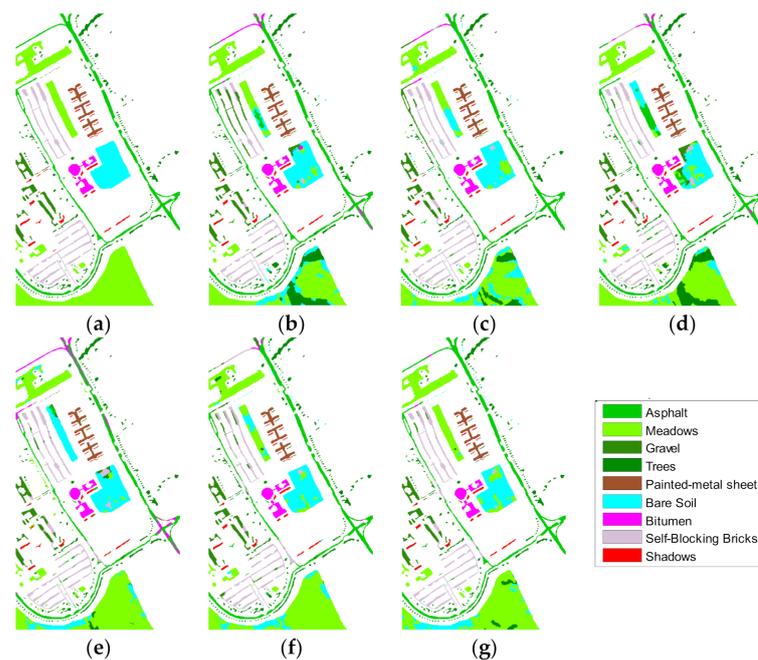
Algorithm	Evaluation Parameters	5	10
JSPR	AA	80.65	85.75
	OA	73.52	77.25
	Kappa	67.55	76.25
	T(s)	150.25	221.96
RVCANET	AA	82.45	86.52
	OA	68.38	77.69
	Kappa	64.85	76.02
	T(s)	14,404.5	15,420.46

Table 4. Cont.

Algorithm	Evaluation Parameters	5	10
HC-RMG	AA	73.65	83.26
	OA	71.36	81.57
	Kappa	69.58	81.95
	T(s)	23.32	25.64
SKCR	AA	80.96	87.25
	OA	73.68	83.24
	Kappa	68.38	79.25
	T(s)	3.34	3.75
HS_AUG	AA	79.08	85.55
	OA	76.04	84.72
	Kappa	70.69	81.95
	T(s)	425.46	426.32
HS_AUG_MSFF	AA	80.63	86.69
	OA	78.25	85.85
	Kappa	77.18	83.99
	T(s)	427.26	429.45



**Figure 12.** Classification results of different algorithms for the Indian Pines data: (a) label, (b) joint sparse model and discontinuity preserving relaxation (JSPP), (c) RVCANET, (d) hyperspectral images based on random multi-graphs (HC-RMG), (e) spectral-spatial hyperspectral classification via structural-kernel collaborative representation (SKCR) (f) HS\_AUG, and (g) HS\_AUG\_MSFF.



**Figure 13.** Classification results of different algorithms for Indian Pines data: (a) label (b) JSJR (c) RVCANET (d) HC-RMG (e) SKCR (f) HS\_AGU and (g) HS\_AUG\_MSFF.

## 6. Conclusions

Our work is divided into two aspects. First, an HS data augmentation algorithm was designed. Second, the method of extracting spectral structural features was studied.

The designed data augmentation method is different from the traditional augmentation method, which adds new samples by adding noise, rotating, and performing other operations. The method does not perform data augmentation of the image but of the spectrum. Essentially, our method uses the image attributes of the HSI to reasonably create some new samples from the neighborhood of the spectrum. The new spectrum conforms to the distribution of the spectrum, which is a meaningful data augmentation. Experiments show that when the training samples are limited, the data augmentation can effectively improve the classification accuracy. We also verified that the augmentation of unlabeled samples can greatly improve the final classification results.

Our structural feature method also proves that the structural feature of the hyperspectral data is different from the intensity feature of the spectrum, and the structural feature of hyperspectral data can even achieve better classification accuracy.

We did not use the spatial features of HSIs for classification. We only used the spatial relationship to perform spectral augmentation, and we only made some preliminary attempts on the augmentation algorithm of HS data. Thus, the method of HS data augmentation needs further development. In addition, we only designed a simple structural feature extraction method, but we look forward to better structural feature extraction methods. Effectively fusing the various structural features of the hyperspectral data will also be an important research direction.

**Author Contributions:** Conceptualization, W.W.; Methodology, W.W.; Project administration, X.L. and X.M.; Writing—original draft, W.W.; Writing—review & editing, W.W., X.L. and X.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Natural Science Foundation of China, grant number. 61501456 and 11701337, and The APC was funded by National Natural Science Foundation of China, grant number.61501456 and 11701337.

**Acknowledgments:** The authors would like to thank Purdue University for providing the AVIRIS image of Indian Pines and to thank University of Pavia for providing the ROSIS data set. We would

like to take this opportunity to thank the editors and the reviewers for their detailed comments and suggestions, which greatly helped us to improve the clarity and presentation of our manuscript.

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

## References

- Zhang, Y.; Du, B.; Zhang, L.; Liu, T. Joint Sparse Representation and Multitask Learning for Hyperspectral Target Detection. *IEEE Trans. Geosci. Remote Sens.* **2016**, *55*, 894–906. [[CrossRef](#)]
- Wang, C.; Zhang, L.; Wei, W.; Zhang, Y. When Low Rank Representation Based Hyperspectral Imagery Classification Meets Segmented Stacked Denoising Auto-Encoder Based Spatial-Spectral Feature. *Remote Sens.* **2018**, *10*, 284. [[CrossRef](#)]
- Pan, E.; Mei, X.; Wang, Q.; Ma, Y.; Ma, J. Spectral-spatial classification for hyperspectral image based on a single GRU. *Neurocomputing* **2020**, *387*, 150–160. [[CrossRef](#)]
- Samiappan, S.; Moorhead, R.J. Semi-Supervised Co-Training and Active Learning Framework for Hyperspectral Image Classification. In Proceedings of the 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Milan, Italy, 26–31 July 2015.
- Rasmus, A.; Valpola, H.; Honkala, M.; Berglund, M.; Raiko, T. Semi-Supervised Learning with Ladder Networks. *Computerence* **2015**, *9*, 3546–3554.
- Wei, W.; Zhang, L.; Li, Y.; Wang, C.; Zhang, Y. Intra-class Similarity Structure Representation based Hyperspectral Imagery Classification with Few Samples. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 1045–1054. [[CrossRef](#)]
- Li, J.; Gamba, P.; Plaza, A. A Novel Semi-Supervised Method for Obtaining Finer Resolution Urban Extents Exploiting Coarser Resolution Maps. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 4276–4287. [[CrossRef](#)]
- Andekah, Z.A.; Naderan, M.; Akbarizadeh, G. Semi-supervised Hyperspectral image classification using spatial-spectral features and superpixel-based sparse codes. In Proceedings of the 2017 Iranian Conference on Electrical Engineering (ICEE), Tehran, Iran, 2–4 May 2017.
- He, Z.; Liu, H.; Wang, Y.; Hu, J. Generative Adversarial Networks-Based Semi-Supervised Learning for Hyperspectral Image Classification. *Remote Sens.* **2017**, *9*, 1042. [[CrossRef](#)]
- Tan, K.; Li, E.; Du, Q.; Du, P. An efficient semi-supervised classification approach for hyperspectral imagery. *ISPRS J. Photogramm. Remote Sens.* **2014**, *97*, 36–45. [[CrossRef](#)]
- Wu, Y.; Mu, G.; Qin, C.; Miao, Q.; Ma, W.; Zhang, X. Semi-Supervised Hyperspectral Image Classification via Spatial-Regulated Self-Training. *Remote Sens.* **2020**, *12*, 159. [[CrossRef](#)]
- Yu, A.W.; Dohan, D.; Luong, M.T.; Zhao, R.; Chen, K.; Norouzi, M.; Le, Q.V. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. *arXiv* **2018**, arXiv:1804.09541.
- Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [[CrossRef](#)]
- Cubuk, E.D.; Zoph, B.; Mane, D.; Vasudevan, V.; Le, Q.V. Autoaugment: Learning Augmentation Policies from Data. *arXiv* **2018**, arXiv:1805.09501.
- Hannun, A.; Case, C.; Casper, J.; Catanzaro, B.; Diamos, G.; Elsen, E.; Ng, A.Y. Deep Speech: Scaling up End-to-End Speech Recognition. *arXiv* **2014**, arXiv:1412.5567.
- Park, D.S.; Chan, W.; Zhang, Y.; Chiu, C.C.; Zoph, B.; Cubuk, E.D.; Le, Q.V. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. *arXiv* **2019**, arXiv:1904.08779.
- Li, W.; Wu, G.; Zhang, F.; Du, Q. Hyperspectral image classification using deep pixel-pair features. *IEEE Trans. Geosci. Remote Sens.* **2016**, *55*, 844–853. [[CrossRef](#)]
- Li, W.; Chen, C.; Zhang, M.; Li, H.; Du, Q. Data augmentation for hyperspectral image classification with deep CNN. *IEEE Geosci. Remote Sens. Lett.* **2018**, *16*, 593–597. [[CrossRef](#)]
- Acción, Á.; Argüello, F.; Heras, D.B. Dual-Window Superpixel Data Augmentation for Hyperspectral Image Classification. *Remote Sens.* **2020**, *10*, 8833.
- Nalepa, J.; Myller, M.; Kawulok, M. Training- and Test-Time Data Augmentation for Hyperspectral Image Segmentation. *IEEE Geosci. Remote Sens. Lett.* **2019**, *17*, 292–296. [[CrossRef](#)]
- 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](#)]
- Alipourfard, T.; Arefi, H. Virtual Training Sample Generation by Generative Adversarial Networks for Hyperspectral Images Classification. *Int. Arch. Photogramm. Spat. Inf. Sci.* **2019**, *42*, 63–69. [[CrossRef](#)]
- Audebert, N.; Le Saux, B.; Lefèvre, S. Generative adversarial networks for realistic synthesis of hyperspectral samples. In Proceedings of the IGARSS 2018—2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 4359–4362.
- Zhu, L.; Chen, Y.; Ghamisi, P.; Benediktsson, J.A. Generative Adversarial Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5046–5063. [[CrossRef](#)]
- Xie, Q.; Dai, Z.; Hovy, E.; Luong, M.T.; Le, Q.V. Unsupervised Data Augmentation for Consistency Training. *arXiv* **2019**, arXiv:1904.12848.

26. Cheng, T.; Sánchez-Azofeifa, B.R.A. Spectroscopic determination of leaf water content using continuous wavelet analysis. *Remote Sens. Environ.* **2010**, *115*, 659–670. [[CrossRef](#)]
27. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [[CrossRef](#)] [[PubMed](#)]
28. Sun, Y.; Liang, D.; Wang, X.; Tang, X. Deepid3: Face Recognition with Very Deep Neural Networks. *arXiv* **2015**, arXiv:1502.00873.
29. 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](#)]
30. Chen, Y.; Lin, Z.; Zhao, X.; Wang, G.; Gu, Y. Deep learning-based classification of hyperspectral data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2094–2107. [[CrossRef](#)]
31. Ma, X.; Geng, J.; Wang, H. Hyperspectral image classification via contextual deep learning. *Eurasip J. Image Video Process.* **2015**, *2015*, 1–12. [[CrossRef](#)]
32. Liu, Y.; Cao, G.; Sun, Q. Hyperspectral classification via deep networks and superpixel segmentation. *Int. J. Remote Sens.* **2015**, *36*, 3459–3482. [[CrossRef](#)]
33. 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.
34. Zhao, W.; Guo, Z.; Yue, J.; Zhang, X.; Luo, L. On combining multiscale deep learning features for the classification of hyperspectral remote sensing imagery. *Int. J. Remote Sens.* **2015**, *36*, 3368–3379. [[CrossRef](#)]
35. Makantasis, K.; Karantzas, K.; Doulamis, A.; Doulamis, N. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In Proceedings of the 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Milan, Italy, 26–31 July 2015.
36. Wei, H.; Yangyu, H.; Li, W. Deep Convolutional Neural Networks for Hyperspectral Image Classification. *J. Sens.* **2015**, *2015*, 1–12.
37. He M, L.; Q, L. Hyperspectral Imagery Classification Using Sparse Representations of Convolutional Neural Network Features. *Remote Sens.* **2016**, *8*, 99.
38. Romero, A.; Gatta, C.; Camps-Valls, G. Unsupervised deep feature extraction for remote sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 1349–1362. [[CrossRef](#)]
39. 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](#)]
40. Chan T, H.; Jia, K.; Gao, S. PCANet: A Simple Deep Learning Baseline for Image Classification. *IEEE Trans. Image Process.* **2015**, *24*, 5017–5032. [[CrossRef](#)]
41. Pan, B.; Shi, Z.; Xu, X. R-VCANet: A New Deep-Learning-Based Hyperspectral Image Classification Method. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *99*, 1–12. [[CrossRef](#)]
42. Gao, Q.; Lim, S.; Jia, X. Hyperspectral Image Classification Using Joint Sparse Model and Discontinuity Preserving Relaxation. *IEEE Geosci. Remote Sens. Lett.* **2017**, *11*, 277–283.
43. Gao, F.; Wang, Q.; Dong, J.; Xu, Q. Spectral and Spatial Classification of Hyperspectral Images Based on Random Multi-Graphs. *Remote Sens.* **2018**, *10*, 1271. [[CrossRef](#)]
44. Tu, B.; Zhou, C.; Liao, X. Spectral-Spatial Hyperspectral Classification via Structural-Kernel Collaborative Representation. *IEEE Geosci. Remote Sens. Lett.* **2020**, *99*, 1–5.