remote sensing

MDPI

Article

Generative Adversarial Networks Based on Transformer Encoder and Convolution Block for Hyperspectral Image Classification

Jing Bai [1], Jiawei Lu [1], Zhu Xiao [2,*], Zheng Chen [1] and Licheng Jiao [1]

[1] School of Artificial Intelligence, Xidian University, Xi'an 710071, China; baijing@mail.xidian.edu.cn (J.B.); lujiawei@stu.xidian.edu.cn (J.L.); chenzheng@cmtt.chinamobile.com (Z.C.); lchjiao@mail.xidian.edu.cn (L.J.)
[2] College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China
* Correspondence: zhxiao@hnu.edu.cn

Abstract: Nowadays, HSI classification can reach a high classification accuracy when given sufficient labeled samples as training set. However, the performances of existing methods decrease sharply when trained on few labeled samples. Existing methods in few-shot problems usually require another dataset in order to improve the classification accuracy. However, the cross-domain problem exists in these methods because of the significant spectral shift between target domain and source domain. Considering above issues, we propose a new method without requiring external dataset through combining a Generative Adversarial Network, Transformer Encoder and convolution block in a unified framework. The proposed method has both a global receptive field provided by Transformer Encoder and a local receptive field provided by convolution block. Experiments conducted on Indian Pines, PaviaU and KSC datasets demonstrate that our method exceeds the results of existing deep learning methods for hyperspectral image classification in the few-shot learning problem.

Keywords: generative adversarial networks; transformers; few-shot learning

Citation: Bai, J.; Lu, J.; Xiao, Z.; Chen, Z.; Jiao, L. Generative Adversarial Networks Based on Transformer Encoder and Convolution Block for Hyperspectral Image Classification. *Remote Sens.* **2022**, *14*, 3426. https://doi.org/10.3390/rs14143426

Academic Editor: Lionel Bombrun

Received: 10 June 2022
Accepted: 13 July 2022
Published: 16 July 2022

1. Introduction

By analyzing hyperspectral images (HSIs), we can explore both abundant spatial information and rich spectral information [1–3]. Compared with RGB images, it can be applied in fields such as mineral detection, disaster prevention and precision agriculture by precisely classifying each pixel [4–6]. In environmental protection, HSI can detect gas [7], oil spills [8], water quality [9,10] and vegetation coverage [11,12].

There are hundreds of spectral bands in each pixel taken from a hyperspectral image and thus forms a three-dimensional data cube. Every spectral band in the cube can be seen as a 2D image. By analyzing the vast amount of information in this 3D cube, each pixel can be predicted with a unique label, and various classes are discriminated as accurately as possible. Through the rapid development of classification accuracy, HSIs have become the foundation of military, agriculture and astronomy.

In the early times, researchers mainly focused on traditional machine learning methods such as logistic regression [13], neural networks [14], principal component analysis (PCA) [15] and support vector machines (SVM) [16]. However, these methods cannot fully utilize the non-linear information in the high-dimensional hyperspectral data.

In the deep learning era, the convolutional neural network (CNN) has achieved satisfactory results with the invention of different models. The CNN can effectively capture features from raw pixels by exploiting the shape, layout and texture of ground objects which combines both the spatial information and spectral information. In [17], a 2D-CNN and a 1D-CNN are combined together to explore more useful features for classification from spatial information and spectral information. Since a 3D-CNN have more advantages in processing the 3D information, Li et al. [18] and Chen et al. [19] developed a classification framework consisting of 3D convolution blocks to process the cubes around each

pixel. In [20], Xu used the dual-channel model to combine a 3D-CNN and a 2D-CNN to learn useful spatial information and spectral information of HSIs. Then, this extracted information is merged and put into a classification block formed of fully connected layers to improve the accuracy. Recently, the SOTA in hyperspectral image classification has been able to reach 99% classification accuracy in the condition of sufficient labeled data.

However, these good results are obtained only under the condition of sufficiently labeled data. While a human can classify new classes by learning a few labeled samples, the performance of these methods decreases sharply when labeled samples are scarce. It is time-consuming and costly to label the data manually. If we only train the network until enough pixels have been labeled, it will be impossible to perform classification in real time. Learning how to obtain good results under the condition that there are only a few labeled classes has recently attracted more and more attention. The so-called few-shot classification means that each class is given K-labeled samples as training data to make predictions on the whole dataset. Usually, the value of K will be set to a small number here, which is 5, 10, 15, 20 and 25 in our experimental settings.

In order to solve a few-shot problem, the unlabeled data and outer dataset are considered to solve the problem [21]. Semi-supervised methods and active learning methods have been proposed based on the assumption that there is no severe shift between the two data distributions which are the target domain data and source domain data. VSCNN [22] uses active learning to select valuable samples from uncertain dataset to form training sample set and improve the small sample classification. However, affected by various environmental conditions such as light or atmosphere, even the pixels from the two different domains, which are the target domain and source domain, have the same labeled class, and the target and source domain usually have significant spectral shift. Domain-adaptation methods are proposed in order to solve this cross-domain problem.

DCFSL [23] is proposed by combining few shot learning and a domain adaptation strategy in the conditional adversarial manner together to address the issue that there may be different data distributions between the target domain and the source domain. MDL4OW [24] improves the classification accuracy by identifying unknown classes. MDL4OW uses the statistical mode EVT to estimate the unknown score and a new evaluation metric to evaluate the accuracy. These methods try to both solve the few-shot learning problem by applying a framework of utilizing other datasets. However, the fitness of outer dataset is still a burden of the few-shot problem.

In combination with metric learning, domain adaptation can solve the few-shot learning problem without involving an external dataset. Metric learning can learn a relationship between sample pairs by mapping the samples into a metric space. In this space, the distance between the samples of same classes will be as close as possible and the distance between the samples without the same classes is as large as possible. S-DMM [25] proposes a model based on metric learning and then learns the similarity between sample pairs using a Siamese network and an auto-encoder. S-DMM solves the cross-scene HSI classification by applying the deep learning method. However, the metric learning method has the defect of being very time-consuming.

Solving the few-shot problem while being time-efficient and without using other data is not a trivial task. Nevertheless, the above methods cannot meet the requirements. In summary, the few-shot problem in classifying HSI faces the following challenges:

How to reach a high accuracy in few-shot problem. Considering the cost of manually labeling every pixel in a hyperspectral image, reaching a satisfying accuracy result under the condition of giving a few training samples can bring great economic benefits. However, this is difficult because the network relies on learning the distribution of labeled samples to make predictions. If the amount of training data is not large enough, it will be very difficult for the network to achieve high accuracy.

How to solve the few-shot problem without involving an outer dataset. Because of the existence of severe shifts in the sample distribution between the source dataset and the target dataset, finding an appropriate outer dataset as the source dataset is a hard task. As

we want to solve the few-shot problem by applying different datasets successfully, it may be a better choice to achieve high accuracy without including an outer dataset. In this way, searching for useful source datasets for every target dataset will not be required.

How to solve the few-shot problem with a fast speed. The proposed methods usually have huge time requirements to solve the above problem because of the defects in the methods themselves, such as metric learning. In some cases, classifying an HSI as quick as possible is very important.

Considering the above problems, we propose a new method employing all the benefits of convolution blocks and Transformer Encoders to solve few-shot learning in this paper. Convolution blocks have the benefits of shared weight, spatial subsampling and local receptive fields, and Transformer Encoders have the advantages of dynamic attention, better generalization and global context fusion. Combined with a generative adversarial network, this method can ensure the similarity between generated and original samples. We do not use any other dataset or unlabeled data in this paper to solve the few-shot learning problem. The main contributions of our paper are as follows:

(1) For the first time, a convolution block, Transformer Encoder and Generative Adversarial Network are combined together to realize the few-shot classification of HSIs. Through this model, we can learn the data distribution by only using a few samples and can reach a high accuracy on different datasets. With this efficient model, we also achieve the aim of not using outer datasets.
(2) We solve the few-shot problem with better time efficiency. Considering the time consumption of training Transformers, we speed up the training time by combining the Transformer Encoder with convolution blocks.
(3) The method proposed in the paper achieved good classification results on the Indian Pines, PaviaU and KSC datasets compared with other few-shot learning methods.

2. Related Work

2.1. Transformer Combined with Convolution

Convolution blocks can capture local features efficiently by using local receptive fields. While self-attention-based architectures, such as Transformers [26], have the advantages that convolution-based architectures do not have, they can capture global information by dynamic attention and global context fusion. LSTM and CNN are combined by replacing the feature fusion block with LSTM in Wang et al. [27]. SENet [28] uses the squeeze operation and excitation operation to obtain the relationship between channels. Moreover, SENet is improved by CBAM [29] through adding spatial attention. The Split-Attention block is introduced in ResNeSt [30] to extract the attention between multi-layer feature-maps. Swin-Transformer and UperNet are combined for segmentation in hyperspectral image classification in Xu et al. [31]. TRS [32] combines the ResNet with the Transformer by replacing the convolutions in the ResNet with a Multi-Head Self-Attention layer. SATNet [33] improves the self-attention mechanism by introducing a spectral attention mechanism to extract the spectral-spatial features. HSI-BERT [34] first tries self-attention-based architecture and then proposes BERT as the framework to classify HSIs. HSI-BERT can obtain good results under the condition of sufficient samples. Recently, ViT [35] has been proposed to try using a Transformer on the image classification and obtain state-of-the-art performance by training and testing on the ImageNet dataset. The Transformer comes from the nature processing language. Its main idea is to split images into patches, treat these patches as tokens and then input them into standard Transformer layers repeatedly. CVT [36] combines both the convolution block and the Transformer Encoder and has the advantage of learning local and global relations efficiently.

2.2. Generator Combined with Self-Attention

The generative adversarial network [37] is an efficient method for solving few-shot learning problems by generating more samples to train a discriminator to achieve the best classification result. However, the GAN is hard to train because the information cannot flow

efficiently across the generator and discriminator, which is the essential point to generate samples having distributions similar to real samples. At first, the GAN is included to solve small-sampled problems which take the training set on a small percent considering the whole dataset. CA-GAN [38] uses collaborative and competitive training and uses joint spatial–spectral hard attention modules to solve small-sampled problems by suppressing less useful features and emphasizing more discriminative ones. SaGAAN [39] adds the cross-domain loss term to generate high-quality generated samples and includes the self-attention mechanism to reduce unintentional noises. While the few-shot problem requires taking a lower and fixed number of samples in every class as the trainset, these methods deteriorate greatly in this condition.

3. Methodology

In order to learn the data distribution with only a few samples, the neighbors around a pixel are taken as a whole, which represent this pixel label and input the network as training sample. This cube around a pixel has a window size of $W \times W$, which has W in width and W in height. Considering the spectral bands have N channels, the whole cube has a size of $W \times W \times N$. In our network, this network will use a dual-channel block and fusion block to make a classification. The dual-channel block will learn the spatial information and spectral information around the label pixel and compress the cube to an appropriate size. The output of the dual-channel block is the input of the fusion block, and the fusion block will perform the final classification through learning the input. A generator is used to generate a same-sized cube to promote the classification accuracy. In the ablation experiments, classification results are improved by the generator. The whole network is shown in Figure 1.

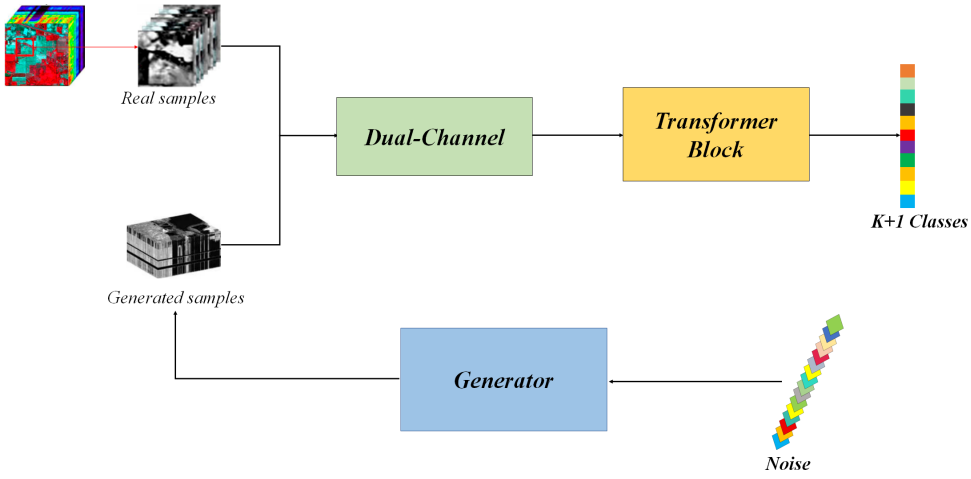

Figure 1. The overall framework of the proposed method for few-shot classification. As can be seen, the framework is a classic Generative Adversarial Network framework.

3.1. Transformer Encoder

Transformer Encoder, as a self-attention-based architecture, has achieved great success in natural language processing (NLP) by training on a large text corpus with over 100B parameters. The main idea of Transformer Encoder is the use of a self-attention mechanism, which learns three matrices representing query, key and value and then obtains the long-distance relationship using Equation (1):

$$A(x_1) = \sum_{i=1}^{n} \left(\frac{Q_1 {K_i}^T}{\sqrt{d}}\right) V_i, \tag{1}$$

$$\begin{aligned} &Attention(x_1, x_2, ..., x_n) \\ &\quad = soft\max(A(x_1), A(x_2), ..., A(x_n)), \end{aligned} \tag{2}$$

where V, K and Q are the abbreviations of value, key and query, respectively. d is the dimensions of Q and K, and n defines the sequence length of x. By learning the attention matrix of each node, the Transformer Encoder can obtain the global relationship between nodes. The architecture of the Transformer Encoder used in this paper is shown in Figure 2.

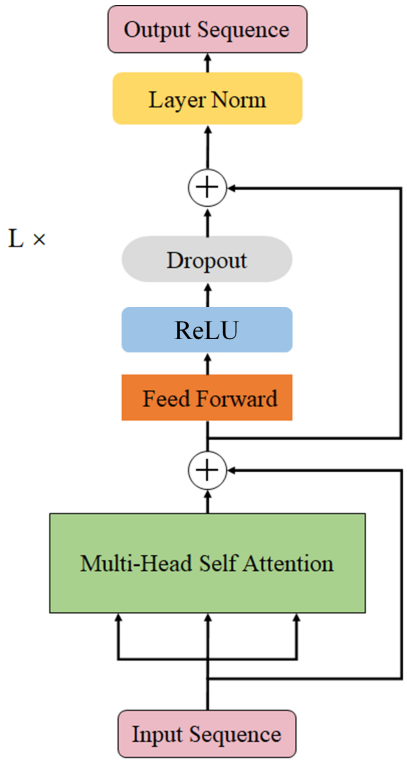

Figure 2. The Transformer Encoder module in this paper.

3.2. Spatial Feature Extraction

At first, the spectral information is compressed by applying 1×1 convolution kernels while not decreasing original spatial relationships. The spectral bands are decreased from N bands to 3 bands to make this channel focus on the spatial correlations. After spectral feature extraction, each pixel in a patch can provide more useful spatial information for few-shot classification. The compressed cube is passed through three successive hybrid convolution and Transformer Encoder layers. This channel splits the cube into M patches, and every patch has p in width and p in height, where $M = (W/p)^2$. The relationship between each patch will be learned by the Transformer through dot-product attention, and convolution manipulation is applied after every Transformer Encoder in order to obtain the local attention while also attaining the global attention. A ReLU activation function is applied after each convolution manipulation, and a batch normalization layer is applied after each ReLU activation function. The detailed structure of Spatial Feature Extraction is shown in Table 1.

Table 1. The detailed structure of Spatial Feature Extraction.

Stage	Output	Spatial Feature Extraction
S1	$15 \times 15 \times 200$	(1×1,3,stride 1)
S2	$15 \times 15 \times 3$	Attention (3×3,3,stride 1)
S3	$15 \times 15 \times 3$	Attention (3×3,3,stride 1)
S4	$15 \times 15 \times 3$	Attention (3×3,3,stride 1)

3.3. Spectral Feature Extraction

Two-dimensional convolution and self-attention blocks are used in this channel to compress N bands gradually in order to learn spectral information. The input cube with size of $W \times W \times N$ (width and height both have W pixels, and spectral bands are N) is split into M groups. Each group's width and each group's height both have p pixels, where $M = (W/p)^2$. The bands in every pixel will learn the relationship with other bands in the same group through group-divided attention. The group-divided self-attention is shown in Figure 3. This group-divided self-attention will make the remaining spectral information focuses on the pixels in the original $p \times p$ group and learns the spectral correlation between the pixels in the original $p \times p$ group. Before every self-attention block, the cube is convoluted with a kernel of 3 in size and 1 in stride in order to introduce the spatial subsampling, joint weighting and local receptive fields of the convolution block into this channel. This step will also realize suitable spectral information. A ReLU activation function is applied after each convolution manipulation, and a batch normalization layer is applied after each ReLU activation function.

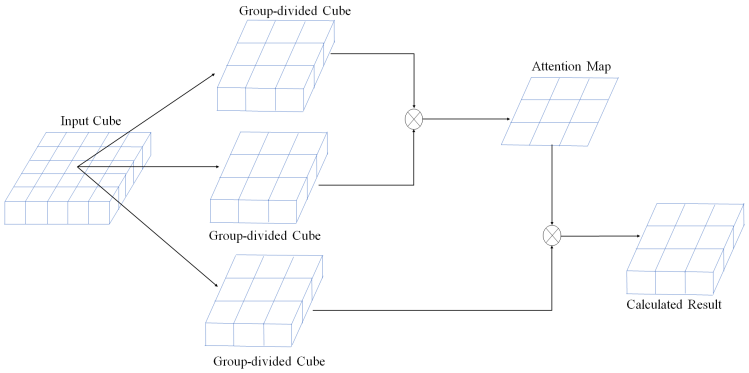

Figure 3. The Group-divided Self-attention.

The dual-channel block is shown in Figure 4. In order to illustrate it clearly, the network sample of a 3D cube in Indian Pines is used as an example. This cube has a size of $15 \times 15 \times 200$ (width and height both have 15 pixels, and the number od spectral bands is 200). The detailed structure of Spectral Feature Extraction is shown in Table 2.

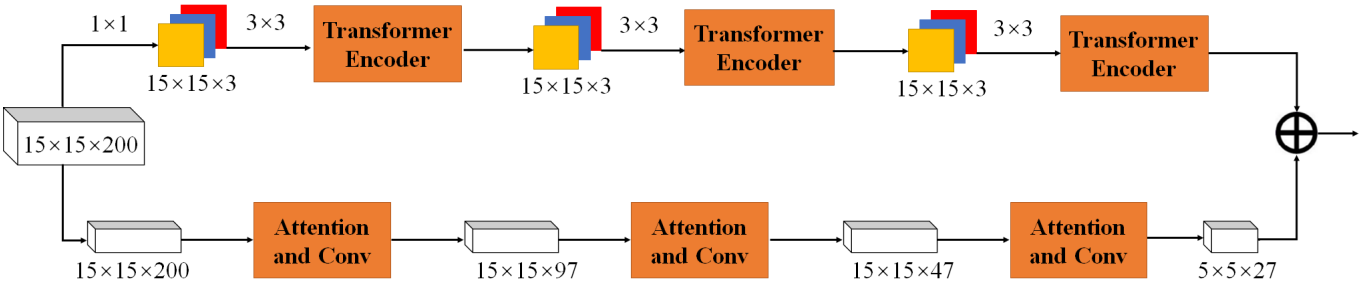

Figure 4. The dual-channel module that extracts spatial features and spectral features.

Table 2. The detailed structure of Spectral Feature Extraction.

Stage	Output	Spectral Feature Extraction
S1	$15 \times 15 \times 200$	$(1 \times 1,97$,stride 1)
S2	$15 \times 15 \times 97$	Reshape Linear Reshape

Table 2. *Cont.*

Stage	Output	Spectral Feature Extraction
S3	$25 \times 27 \times 3 \times 97$	Group-divided Self-attention Reshape
S4	$15 \times 15 \times 97$	(3×3,47,stride 1)
S5	$15 \times 15 \times 47$	Reshape Linear Reshape
S6	$25 \times 27 \times 3 \times 47$	Group-divided Self-attention Reshape
S7	$15 \times 15 \times 47$	(3×3,27,stride 3)
S8	$5 \times 5 \times 27$	Reshape Linear Reshape
S9	$25 \times 3 \times 27$	Group-divided Self-attention Reshape

3.4. Fusion Block

The spectral channel obtains the output size of $p \times p \times f$, where $f = (W/p)^2 \times 3$. The spatial channel obtains the output size of $W \times W \times 3$. Then, we reshape the output of the spatial channel into a cube of size $p \times p \times f$ and concatenate it with spectral channel output at the last dimension. By using these two channels, we can abandon the position embedding, which is used in ViT [35] to retain positional information. In order to fuse the information extracted from two channels represented by spatial and spectral, a fully connected layer is applied at the last dimension to obtain the output size $p \times p \times f$. After that, the $p \times p \times f$ cube and an extra learnable embedding are passed to three successive convolution and Transformer Encoder hybrid layers. This extra learnable embedding is used as a classification token to obtain prediction results by going through an MLP. The Fusion Block is shown in Figure 5.

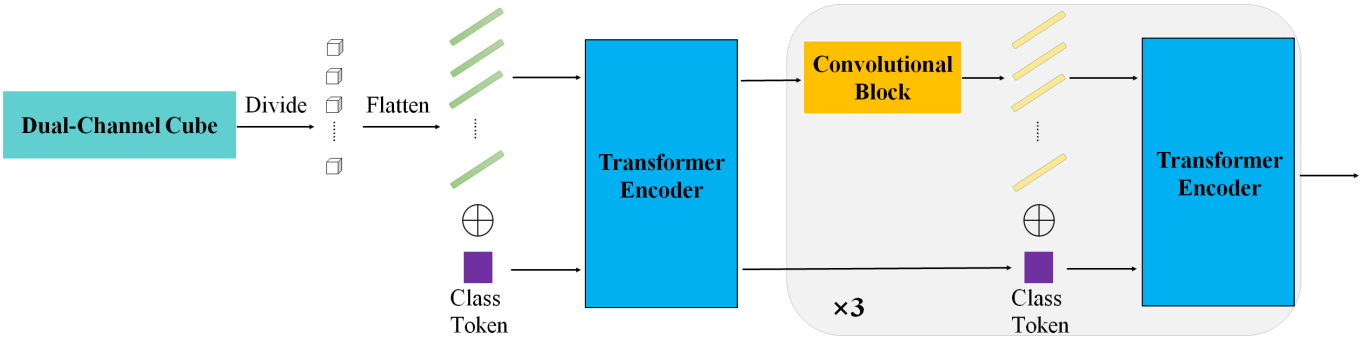

Figure 5. The Fusion Block that learns the class token to obtain the classification result.

3.5. Generator

In the NLP domain, the Transformer takes words as a sequence and computes the importance between each word. However, a hyperspectral image cannot be divided pixel-by-pixel, as this would result in a sequence that is too long to process in terms of both computational cost and efficiency. Inspired by some GANs which generate images layer-by-layer, we iteratively upscale the image size to reduce the computation cost and enhance network generation ability. We use shufflepixel and Transformer Encoder iteratively to make the resolution four times larger at each iteration. The Generator is shown in Figure 6. The detailed structure of the Generator is shown in Table 3. The framework of our system is shown in Algorithm 1.

Table 3. The detailed structure of the Generator.

Stage	Output	Generator
S1	256 × 4 × 4	Attention Deconvolution
S2	64 × 8 × 8	Attention Deconvolution
S3	16 × 16 × 16	Attention Deconvolution
S4	15 × 32 × 32	Attention Deconvolution
S5	15 × 15 × 200	Attention Deconvolution

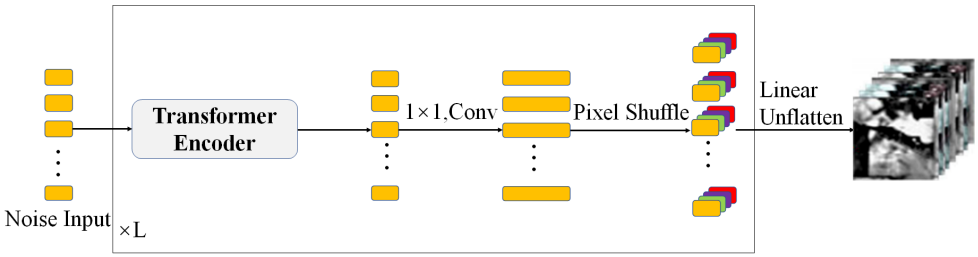

Figure 6. The Generator.

Algorithm 1 The framework of our system

Input:

The training data selected from K classes. The class labels of training samples. The test data from K classes.

Output:

1: Extracting the spatial information through the Spatial Feature Extract.
2: Extracting the spectral information through the Spectral Feature Extract.
3: Fusing the two channels' information through the Fusion Block.
4: Generating random noises from uniform distribution.
5: Generating samples from the generator by using the random noises.
6: Training the network through the generated samples.
RETURN: The labels of test data;

4. Experiments

In this section, three leading HSI data sets are selected to conduct HSI classification experiments. The experiments are implemented on the pytorch open source software framework using the NVIDIA 3080Ti graphics card.

The Indian Pines dataset was gathered in 1992 and has 224 bands. The AVIRIS stands for airborne visible infrared imaging spectromet, and its data were gathered in northwestern Indiana. The band's visible and infrared spectra range from 400 to 2500 nm, and 200 spectral bands are used in this paper because of the atmospheric absorption compared with the original 224 bands. The size of the dataset is 145 × 145, and some of these pixels are labeled to 16 classes. Table 4 shows the data division of the Indian Pines dataset for this experiment.

The Pavia University dataset was gathered by the ROSIS sensor in 2002 and has 115 bands. The ROSIS stands for Reflective Optics System Imaging Spectrometer, and its data were gathered over the University of Pavia campus during a flight campaign. The band's visible and infrared spectra range from 430 to 860 nm, and the ground resolution of this dataset is 1.3 m. Affected by noise and water absorption, some bands were abandoned, and 103 spectral bands are used in this paper. The size is 610 × 340 in this dataset, and some of these pixels are labeled to 9 classes. Table 5 shows the training and testing data division of the Pavia University dataset.

Table 4. The land cover category and data division of the Indian Pines Dataset.

Class No.	Class Name	Training	Test
1	Alfalfa	15	31
2	Corn-notill	15	1413
3	Corn-mintill	15	815
4	Corn	15	222
5	Grass-pasture	15	468
6	Grass-trees	15	715
7	Grass-pasture-mowed	15	13
8	Hay-windrowed	15	463
9	Oats	15	5
10	Soybean-notil	15	957
11	Soybean-mintill	15	2440
12	Soybean-clean	15	578
13	Wheat	15	190
14	Woods	15	1250
15	Buildings-Grass-Trees-Drives	15	371
16	Stone-Steel-Towers	15	78
Total		240	10,009

Table 5. The land cover category and data division on the Pavia University dataset.

Class No.	Class Name	Training	Test
1	Asphalt	15	6616
2	Meadows	15	18,634
3	Gravel	15	2084
4	Trees	15	3049
5	Painted metal sheets	15	1330
6	Bare Soil	15	5014
7	Bitumen	15	1315
8	Self-Blocking Bricks	15	3667
9	Shadows	15	932
Total		135	42,641

The Kennedy Space Center dataset was gathered by NASA AVIRIS in 1996 and has 224 bands. The band's visible and infrared spectra range from 400 to 2500 nm, and the ground resolution of this dataset is 18 m. Because of the existence of water absorption, some affected and low SNR bands were abandoned, and 176 spectral bands were used in this paper. The size of the dataset is 512 × 614, and some of these pixels are labeled to 13 classes. Table 7 shows the training and testing data division of the KSC dataset.

Table 6. The land cover category and data division on the Kennedy Space Center dataset.

Class No.	Class Name	Training	Test
1	Scrub	15	746
2	Willow swamp	15	228
3	Cabbage palm hammock	15	241
4	Cabbage palm/oak hammock	15	237
5	Slash pine	15	146
6	Oak/broadleaf hammock	15	214
7	Hardwood swamp	15	90
8	Graminoid marsh	15	416
9	Spartina marsh	15	505
10	Cattail marsh	15	389
11	Salt marsh	15	404

Table 7. *Cont.*

Class No.	Class Name	Training	Test
12	Mud flats	15	488
13	Water	15	912
Total		195	5016

To demonstrate how our method performs, we compared our method with eight different methods, including SVM [40], 2D-CNN [41], 3D-CNN [18], HSI-BERT [34], CA-GAN [38], DCFSL [23], VSCNN [22] and S-DMM [25]. DCFSL, VSCNN and S-DMM are the few-shot learning methods in hyperspectral image classification and obtain good results. DCFSL can utilize other datasets by combining few-shot learning and a domain adaptation strategy in the conditional adversarial manner. VSCNN uses active learning to select valuable samples from uncertain dataset to form training sample set and improve the few-shot learning ability. S-DMM can learn more features by learning the similarity between sample pairs using a Siamese network and an auto-encoder based on metric-learning.

For the fairness of the experiments, all the methods use their optimal parameters. The experiment is divided into five groups for IP and PU by the number of training samples, and the training samples of every class in each group have the numbers of 5, 10, 15, 20 and 25, respectively. In addition, the experiment is divided into three groups for Kennedy Space Center by the number of training samples, and the training samples of every class in each group have the numbers of 15, 20 and 25, respectively. Taking five per class for example, five samples are randomly selected from every class as the samples for training and the left samples are used as the testing set. We adopt the overall accuracy (OA) as the evaluation metric to measure the classification performance. All experiments are averaged on 10 times independent training results.

The above experiments are shown in Tables 8–10. From the tables, we can find that as more samples are labeled, the accuracy reaches a higher score. Our proposed method outperforms in all conducted experiments, which demonstrates the ability of our method regardless of the change in the number of labeled samples. When other methods can obtain a good result on single dataset but cannot fit to the others, it means they do not have good adaptation ability, which is essential in few-shot learning problems. Because we cannot predict what dataset we will encounter, we need to have good results on different datasets.

Given 15 labeled samples as training samples per class, the corresponding classification maps of all the selected methods in IP dataset are shown in Figure 7. In addition, the corresponding detailed maps of PU and KSC are shown in Figures 8 and 9, respectively. It can obviously be seen that our classification map matches best with the image labeled with ground truth in all the images, which means that other methods assigned more incorrect labels to the pixels compared to our method. Moreover, Tables 11–13 show the detailed accuracy of every class classification with 15 labeled samples as training samples on different datasets.

Our method achieves better results on most land classes. In particular, on the Indian Pines dataset, our method obtains the highest classification results on 13 classes out of 16 classes. For the classes “Corn-notill”, “Soybean-mintill” and “Woods”, where the proportion between the number of testing sets and the number of training sets is huge, our method obtains classification results of 78.77%, 81.39% and 97.28%, respectively. Our method is greatly improved compared with other methods in category 3 and 11.

On the Pavia University dataset, our method obtains the highest classification results on four classes out of nine classes. For the class “Meadows” and “Bare Soil”, where the proportion between the number of testing set and the number of training sets is huge, our method obtains classification results of 97.57% and 100.0%, respectively. Our method is greatly improved compared with other methods in category 2.

On the KSC dataset, our method obtains the highest classification results on 6 out of 13 classes. For the classes “Scrub” and “Water”, where the proportion between the number

of testing sets and the number of training sets is huge, our method obtains classification results of 99.87% and 100.0%, respectively. Our method is greatly improved compared with other methods in category 2.

It can be seen that our method can make full use of a small number of training samples to extract effective features. From the perspective of AA, our method reaches the highest on the Indian Pines and KSC dataset. From the perspective of kappa, our method reaches the highest perfromance on three dataset. It can be seen that the classification results of each category are relatively balanced in the case of unbalanced proportion of training samples. Ablation experiments are shown in Tables 14–16. By introducing the generative adversarial network, the OA can be improved by around 2%. As can be seen, the classification results after adding the generator improve greatly.

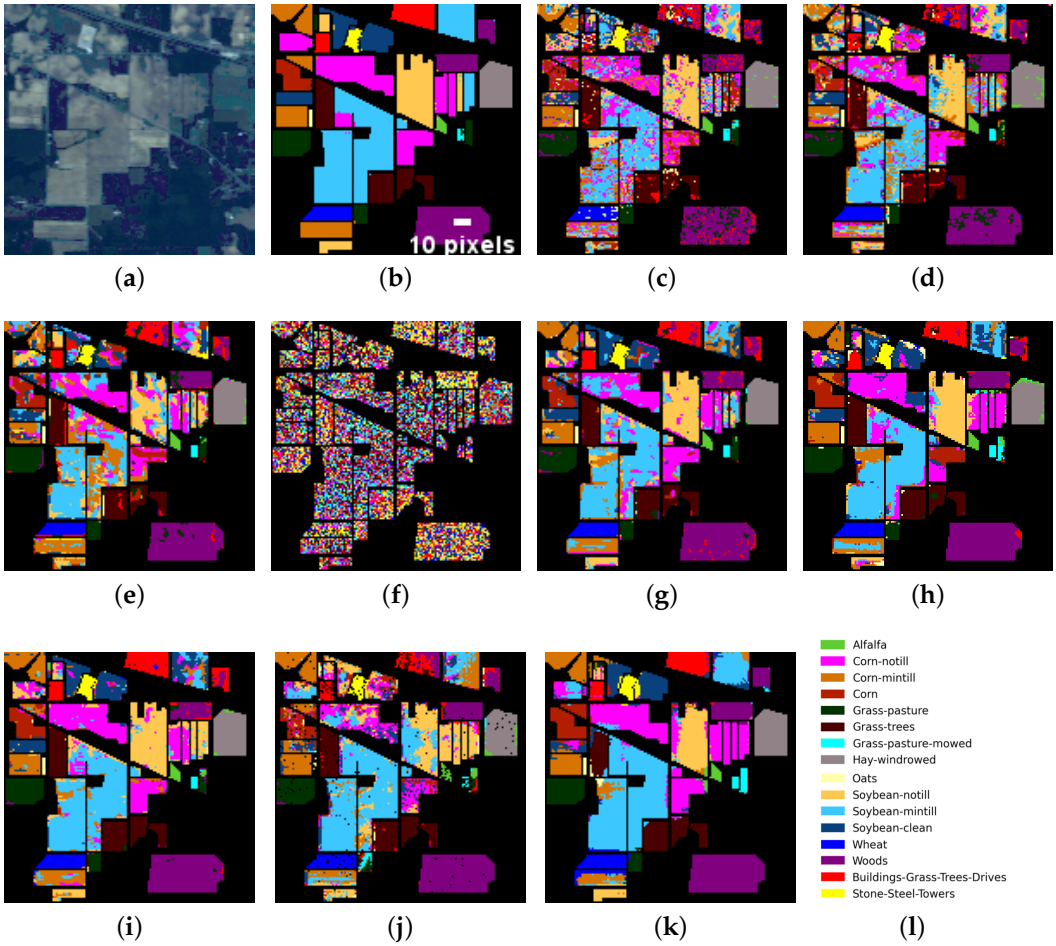

Figure 7. The classification maps for the IP with compared methods. (**a**) Source Image. (**b**) Ground Truth. (**c**) SVM. (**d**) 2D-CNN. (**e**) 3D-CNN. (**f**) HSI-BERT. (**g**) CA-GAN. (**h**) DCFSL. (**i**) VSCNN. (**j**) S-DMM. (**k**) Our proposed method. (**l**) Legend.

Table 8. The classification results of our proposed and other leading methods on Indian Pines (%).

Number	SVM	2D-CNN	3D-CNN	HSI-BERT	CA-GAN	DCFSL	VSCNN	S-DMM	Ours
5	42.01 ± 1.77	39.81 ± 1.10	47.66 ± 1.83	40.08 ± 1.70	51.52 ± 1.81	67.23 ± 1.21	66.70 ± 1.56	60.70 ± 1.85	76.25 ± 0.92
10	51.34 ± 1.28	55.61 ± 1.89	54.46 ± 1.00	52.79 ± 0.77	70.77 ± 1.11	72.14 ± 0.66	80.06 ± 1.14	64.89 ± 0.94	86.28 ± 0.77
15	57.41 ± 1.40	57.72 ± 1.90	58.94 ± 1.27	58.50 ± 1.56	75.52 ± 1.28	77.45 ± 1.78	83.06 ± 1.04	67.04 ± 1.65	87.47 ± 1.45
20	64.32 ± 1.11	60.79 ± 1.53	65.89 ± 1.08	62.03 ± 1.31	80.54 ± 1.06	82.18 ± 1.02	86.13 ± 1.35	68.73 ± 1.24	89.26 ± 0.53
25	68.11 ± 0.63	64.39 ± 1.62	73.23 ± 1.30	66.87 ± 1.42	83.38 ± 0.86	83.20 ± 0.62	88.22 ± 1.40	69.05 ± 0.59	93.01 ± 1.14

Table 9. The classification results of our proposed and other leading methods on Pavia University (%).

Number	SVM	2D-CNN	3D-CNN	HSI-BERT	CA-GAN	DCFSL	VSCNN	S-DMM	Ours
5	62.53 ± 1.87	63.74 ± 1.30	62.92 ± 1.01	18.14 ± 1.59	64.63 ± 0.60	78.03 ± 1.39	71.95 ± 1.78	76.64 ± 0.62	85.95 ± 0.58
10	71.92 ± 1.97	66.37 ± 0.72	72.43 ± 1.64	58.12 ± 1.15	72.55 ± 1.40	85.86 ± 1.73	75.45 ± 1.09	83.26 ± 0.77	91.40 ± 1.39
15	78.67 ± 1.79	77.53 ± 1.50	75.24 ± 0.84	75.31 ± 1.59	76.81 ± 0.91	90.71 ± 0.56	81.63 ± 1.81	88.30 ± 1.03	93.20 ± 0.59
20	80.19 ± 1.60	79.51 ± 1.67	80.87 ± 1.08	76.10 ± 1.45	83.82 ± 0.85	93.68 ± 0.91	83.52 ± 1.36	92.26 ± 0.80	94.69 ± 0.76
25	81.72 ± 0.87	83.77 ± 1.70	82.97 ± 0.88	79.11 ± 1.62	84.99 ± 0.78	94.66 ± 0.83	87.19 ± 1.72	93.37 ± 1.97	96.38 ± 0.42

Table 10. The classification results of our proposed and other leading methods on the Kennedy Space Center dataset (%).

Number	SVM	2D-CNN	3D-CNN	HSI-BERT	CA-GAN	DCFSL	VSCNN	S-DMM	Ours
15	84.83 ± 1.51	80.53 ± 1.31	87.18 ± 1.00	82.93 ± 0.94	91.17 ± 1.54	97.59 ± 1.03	80.15 ± 0.62	95.83 ± 1.68	98.39 ± 0.63
20	86.49 ± 1.94	82.26 ± 0.84	90.53 ± 1.41	86.47 ± 0.67	94.34 ± 0.62	98.10 ± 1.32	85.44 ± 1.07	97.96 ± 1.32	99.54 ± 0.40
25	89.26 ± 0.54	86.23 ± 1.43	91.22 ± 1.00	90.16 ± 1.52	97.04 ± 1.87	99.16 ± 0.84	87.15 ± 1.92	98.08 ± 0.62	99.84 ± 0.15

Table 11. The classification results of our proposed and other leading methods on the Indian Pines dataset (%).

IP15	SVM	2D-CNN	3D-CNN	HSI-BERT	CA-GAN	DCFSL	VSCNN	S-DMM	Ours
class1	54.84 ± 0.58	70.97 ± 0.66	83.87 ± 1.02	93.55 ± 1.25	100.0 ± 0.0	100.0 ± 0.0	90.32 ± 1.08	91.67 ± 1.80	100.0 ± 0.0
class2	30.71 ± 1.85	34.89 ± 1.53	38.08 ± 0.85	42.60 ± 1.58	61.78 ± 0.65	60.79 ± 1.68	75.94 ± 1.78	47.18 ± 0.86	78.77 ± 1.03
class3	43.93 ± 0.85	48.47 ± 1.71	41.84 ± 1.32	48.22 ± 0.83	68.22 ± 0.77	78.77 ± 1.05	85.03 ± 1.61	44.88 ± 1.29	92.15 ± 1.42
class4	57.21 ± 1.08	58.56 ± 1.96	52.70 ± 0.56	81.98 ± 0.55	92.34 ± 1.58	94.59 ± 1.89	95.95 ± 0.95	33.04 ± 0.74	99.1 ± 0.9
class5	62.82 ± 0.56	81.41 ± 1.69	74.79 ± 1.96	86.54 ± 0.72	82.69 ± 1.86	85.68 ± 0.85	91.03 ± 1.78	78.44 ± 1.46	95.30 ± 0.66
class6	81.82 ± 0.50	91.75 ± 1.21	87.27 ± 1.84	87.83 ± 0.62	89.51 ± 1.55	96.64 ± 1.02	97.34 ± 1.38	92.50 ± 1.66	95.94 ± 0.73
class7	84.62 ± 1.05	92.31 ± 0.92	100.0 ± 0.0	92.31 ± 1.57	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
class8	88.12 ± 1.82	86.83 ± 0.56	94.38 ± 1.75	68.03 ± 0.84	99.78 ± 0.21	92.22 ± 1.93	97.84 ± 1.23	85.26 ± 1.94	100.0 ± 0.0
class9	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0

Table 11. *Cont.*

IP15	SVM	2D-CNN	3D-CNN	HSI-BERT	CA-GAN	DCFSL	VSCNN	S-DMM	Ours
class10	52.77 ± 1.08	63.64 ± 1.63	64.26 ± 0.53	59.98 ± 1.52	76.28 ± 0.51	71.89 ± 1.85	80.88 ± 1.51	66.74 ± 0.91	86.00 ± 0.32
class11	52.75 ± 0.66	48.69 ± 1.48	41.43 ± 0.54	46.68 ± 0.93	64.22 ± 1.60	65.66 ± 0.99	73.32 ± 1.08	70.39 ± 1.03	81.39 ± 1.18
class12	52.08 ± 0.92	47.58 ± 0.56	41.70 ± 1.74	39.45 ± 1.03	78.72 ± 0.66	73.18 ± 0.85	88.41 ± 0.53	40.82 ± 1.11	73.18 ± 1.29
class13	93.68 ± 0.76	97.89 ± 1.75	99.47 ± 0.53	86.32 ± 0.95	99.47 ± 0.53	100.0 ± 0.0	98.95 ± 1.04	99.49 ± 0.51	100.0 ± 0.0
class14	80.80 ± 0.78	58.80 ± 1.49	84.24 ± 1.76	70.48 ± 0.81	82.32 ± 1.23	93.28 ± 0.69	84.24 ± 1.50	81.35 ± 0.69	97.28 ± 1.12
class15	42.32 ± 1.11	57.68 ± 1.46	70.89 ± 1.45	62.53 ± 1.21	92.99 ± 1.68	87.87 ± 0.50	86.52 ± 0.58	68.35 ± 1.49	83.83 ± 0.81
class16	88.46 ± 1.57	94.87 ± 1.12	97.44 ± 1.60	84.62 ± 0.56	92.31 ± 1.29	100.0 ± 0.0	98.72 ± 1.24	98.80 ± 0.74	100.0 ± 0.0
OA	57.41 ± 1.40	57.72 ± 1.90	58.94 ± 1.27	58.50 ± 1.56	75.52 ± 1.28	77.45 ± 1.78	83.06 ± 1.04	67.04 ± 1.65	87.47 ± 1.45
AA	66.68 ± 1.67	70.90 ± 1.36	73.27 ± 1.30	71.94 ± 1.31	81.21 ± 0.84	87.54 ± 0.57	90.28 ± 0.58	74.93 ± 1.08	92.68 ± 0.47
kappa	52.29 ± 1.29	52.82 ± 1.09	54.06 ± 1.68	53.63 ± 1.05	72.69 ± 0.57	74.65 ± 0.72	80.89 ± 0.63	62.44 ± 1.63	85.78 ± 1.31

Table 12. The classification results of our proposed and other leading methods on the Pavia University dataset (%).

PU15	SVM	2D-CNN	3D-CNN	HSI-BERT	CA-GAN	DCFSL	VSCNN	S-DMM	Ours
class1	66.28 ± 0.50	40.10 ± 0.77	70.41 ± 1.25	68.91 ± 1.50	60.16 ± 0.67	74.55 ± 0.86	83.27 ± 1.63	96.97 ± 0.82	89.07 ± 1.13
class2	82.10 ± 0.87	93.15 ± 1.82	73.10 ± 1.44	87.44 ± 1.53	72.83 ± 0.92	97.20 ± 1.33	76.96 ± 0.54	81.15 ± 1.29	97.57 ± 1.46
class3	64.78 ± 1.19	83.01 ± 1.62	73.80 ± 0.61	33.59 ± 1.20	98.03 ± 0.73	80.57 ± 0.58	81.91 ± 0.91	92.69 ± 1.19	67.08 ± 0.69
class4	85.93 ± 1.12	90.03 ± 0.67	89.37 ± 1.07	69.86 ± 1.42	89.44 ± 0.83	94.62 ± 1.54	86.86 ± 0.54	97.50 ± 1.41	88.03 ± 0.69
class5	99.32 ± 0.68	98.5 ± 1.5	96.39 ± 1.97	92.18 ± 1.00	99.7 ± 0.29	100.0 ± 0.0	99.55 ± 0.45	100.0 ± 0.0	100.0 ± 0.0
class6	72.34 ± 1.61	52.11 ± 0.66	69.68 ± 0.64	48.54 ± 1.09	79.94 ± 1.35	90.37 ± 1.25	82.81 ± 1.44	84.73 ± 0.98	93.80 ± 0.78
class7	87.22 ± 1.52	68.44 ± 1.30	86.46 ± 1.50	66.92 ± 0.51	90.04 ± 0.69	92.47 ± 1.40	77.94 ± 1.52	97.71 ± 0.58	99.47 ± 0.53
class8	78.13 ± 1.82	76.85 ± 0.57	77.09 ± 1.83	83.50 ± 1.54	81.95 ± 1.65	81.62 ± 1.25	93.58 ± 1.32	93.23 ± 1.42	93.07 ± 1.13
class9	99.89 ± 0.10	100.0 ± 0.0	86.05 ± 0.94	88.73 ± 1.03	97.32 ± 1.72	100.0 ± 0.0	71.84 ± 1.55	99.89 ± 0.10	96.67 ± 0.57
OA	78.67 ± 1.79	77.53 ± 1.50	75.24 ± 0.84	75.31 ± 1.59	76.81 ± 0.91	90.71 ± 0.56	81.63 ± 1.81	88.30 ± 1.03	93.20 ± 0.59
AA	81.78 ± 1.06	78.02 ± 0.64	80.26 ± 1.41	71.08 ± 0.71	76.94 ± 0.76	90.20 ± 0.67	83.86 ± 0.55	93.76 ± 0.77	91.60 ± 0.55
kappa	72.32 ± 0.88	70.17 ± 0.60	68.43 ± 1.65	67.00 ± 1.33	71.02 ± 1.51	87.73 ± 1.67	76.46 ± 1.70	84.90 ± 1.65	91.00 ± 1.07

Table 13. The classification results of our proposed and other leading methods on the Kennedy Space Center (%).

KSC15	SVM	2D-CNN	3D-CNN	HSI-BERT	CA-GAN	DCFSL	VSCNN	S-DMM	Ours
class1	70.38 ± 0.85	86.23 ± 1.60	89.41 ± 1.92	85.66 ± 0.71	88.20 ± 0.71	96.92 ± 1.68	97.15 ± 0.61	96.01 ± 1.49	99.87 ± 0.12
class2	81.14 ± 1.70	74.44 ± 1.09	86.40 ± 1.50	90.35 ± 1.03	85.53 ± 1.04	86.40 ± 0.90	91.28 ± 1.48	88.84 ± 1.03	100.0 ± 0.0
class3	94.19 ± 1.75	72.46 ± 1.98	85.06 ± 1.21	49.79 ± 0.78	95.02 ± 0.76	98.76 ± 0.94	80.09 ± 0.92	99.19 ± 0.81	96.68 ± 0.44
class4	43.04 ± 1.95	76.29 ± 0.76	54.01 ± 0.66	51.90 ± 0.89	90.72 ± 0.59	82.28 ± 1.34	42.29 ± 1.21	54.96 ± 1.08	86.08 ± 1.18
class5	73.97 ± 1.05	42.55 ± 1.29	83.56 ± 0.56	58.90 ± 1.54	90.41 ± 1.19	91.78 ± 1.66	58.09 ± 0.61	80.79 ± 0.86	93.84 ± 1.35
class6	66.36 ± 1.24	46.89 ± 1.03	76.64 ± 0.89	89.72 ± 1.21	94.39 ± 0.53	97.66 ± 0.77	70.59 ± 0.73	96.35 ± 1.07	100.0 ± 0.0
class7	94.44 ± 0.99	78.82 ± 0.65	100.0 ± 0.0	94.44 ± 1.13	100.0 ± 0.0	100.0 ± 0.0	70.00 ± 1.34	100.0 ± 0.0	97.78 ± 0.40
class8	91.11 ± 1.40	76.16 ± 0.80	92.55 ± 1.86	76.68 ± 1.71	86.78 ± 0.75	100.0 ± 0.0	62.81 ± 1.30	99.29 ± 0.50	96.63 ± 0.92
class9	87.52 ± 1.55	84.80 ± 1.65	60.59 ± 1.97	76.44 ± 0.90	86.73 ± 1.39	100.0 ± 0.0	74.55 ± 0.88	100.0 ± 0.0	99.60 ± 0.40
class10	87.92 ± 0.91	75.26 ± 1.22	93.32 ± 0.97	96.92 ± 1.01	86.12 ± 0.69	99.74 ± 0.26	61.48 ± 1.49	100.0 ± 0.0	99.49 ± 0.51
class11	98.27 ± 1.20	97.24 ± 1.90	93.07 ± 1.15	96.78 ± 1.80	92.57 ± 0.74	100.0 ± 0.0	78.68 ± 1.24	100.0 ± 0.0	100.0 ± 0.0
class12	87.91 ± 1.98	74.33 ± 1.98	93.85 ± 0.59	71.31 ± 1.77	88.52 ± 1.08	99.18 ± 0.81	78.24 ± 1.31	98.99 ± 0.87	97.95 ± 1.26
class13	97.81 ± 1.96	92.17 ± 1.30	100.0 ± 0.0	97.37 ± 0.56	100.0 ± 0.0	100.0 ± 0.0	99.89 ± 0.10	100.0 ± 0.0	100.0 ± 0.0
OA	84.83 ± 1.51	80.53 ± 1.31	87.18 ± 1.00	82.93 ± 0.94	91.17 ± 1.54	97.59 ± 1.03	80.15 ± 0.62	95.83 ± 1.68	98.39 ± 0.63
AA	82.62 ± 0.74	75.20 ± 0.85	85.27 ± 1.16	79.71 ± 0.81	84.64 ± 0.84	96.36 ± 1.74	74.24 ± 1.72	93.42 ± 0.72	97.53 ± 1.11
kappa	83.17 ± 1.90	78.23 ± 0.91	85.73 ± 0.60	80.99 ± 1.97	90.20 ± 1.67	97.31 ± 1.83	77.81 ± 1.96	95.35 ± 0.53	98.20 ± 1.20

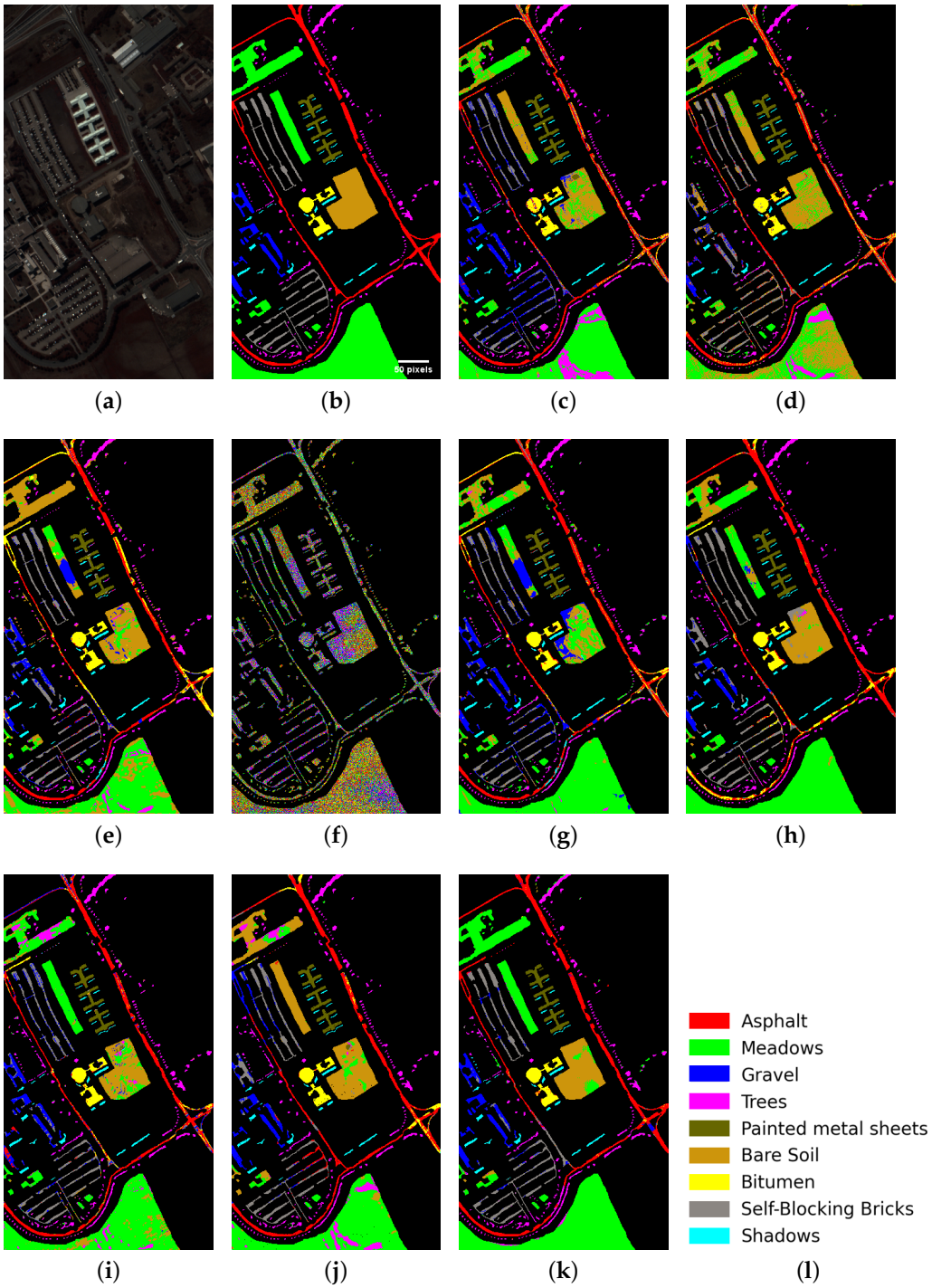

Figure 8. The classification maps for the PU with compared methods. (**a**) Source Image. (**b**) Ground Truth. (**c**) SVM. (**d**) 2D-CNN. (**e**) 3D-CNN. (**f**) HSI-BERT. (**g**) CA-GAN. (**h**) DCFSL. (**i**) VSCNN. (**j**) S-DMM. (**k**) Our proposed method. (**l**) Legend.

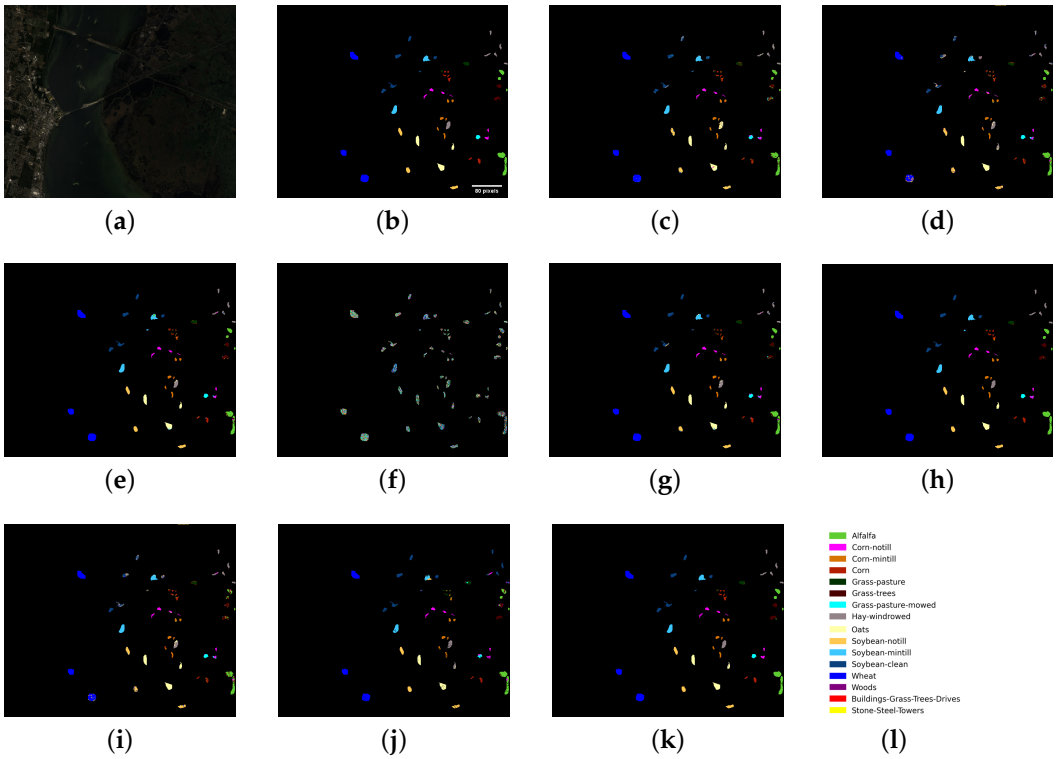

Figure 9. The classification maps for the KSC with compared methods. (**a**) Source Image. (**b**) Ground Truth. (**c**) SVM. (**d**) 2D-CNN. (**e**) 3D-CNN. (**f**) HSI-BERT. (**g**) CA-GAN. (**h**) DCFSL. (**i**) VSCNN. (**j**) S-DMM. (**k**) Our proposed method. (**l**) Legend.

Table 14. The ablation experiments on Indian Pines (OA) (%).

Number	Without Generator	With Generator
5	74.24 ± 2.15	76.25 ± 0.92
10	84.52 ± 1.43	86.28 ± 0.77
15	85.76 ± 1.87	87.47 ± 1.45
20	87.15 ± 1.51	89.26 ± 0.53
25	90.43 ± 1.65	93.01 ± 1.14

Table 15. The ablation experiments on the Pavia University dataset (OA) (%).

Number	Without Generator	With Generator
5	83.34 ± 2.12	85.95 ± 0.58
10	88.04 ± 2.21	91.40 ± 1.39
15	91.56 ± 2.38	93.20 ± 0.59
20	92.14 ± 2.98	94.69 ± 0.76
25	94.89 ± 2.12	96.38 ± 0.42

Table 16. The ablation experiments on the Kennedy Space Center (OA) (%).

Number	Without Generator	With Generator
15	96.82 ± 1.32	98.39 ± 0.63
20	97.54 ± 1.63	99.54 ± 0.40
25	98.12 ± 1.32	99.84 ± 0.15

5. Conclusions

In this paper, we propose a new method that combines a Generative Adversarial Network, convolution block and Transformer Encoder in a unified framework. The proposed method has both a global receptive field provided by the Transformer Encoder and a local receptive field provided by the convolution block. In order to perform better in the few-shot learning problem, the Generative Adversarial Network is used to provide more training data. Experiments conducted on the Indian Pines, PaviaU and KSC datasets demonstrate that our method exceeds the results of existing deep learning methods for hyperspectral image classification in the few-shot learning problem.

Author Contributions: Conceptualization, Z.X.; Data curation, Z.X. and Z.C.; Formal analysis, J.L.; Funding acquisition, J.B. and L.J.; Methodology, J.B.; Resources, Z.C.; Supervision, L.J.; Visualization, Z.X.; Writing—original draft, J.L.; Writing—review & editing, J.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 61772401, in part by the Key Research and Development Program of Shaanxi under Grant 2022GY-062 and Grant 2020GXLH-Y-023, in part by the Science and Technology Project of Hunan Provincial Water Resources Department under Grant XSKJ2021000-39, in part by the Scientific Research Project of Department of the Natural Resources of Hunan Province under Grant 202211, in part by the Fund of National Key Laboratory of Science and Technology on Remote Sensing Information and imagery Analysis, Beijing Research Institute of Uranium Geology under Grant 6142A010409. This paper is also supported by the Science and Technology on Communication Information Security Control Laboratory.

Data Availability Statement: Not Applicable.

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

References

1. Chang, C.I. *Hyperspectral Data Exploitation: Theory and Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2007.
2. Bai, J.; Ding, B.; Xiao, Z.; Jiao, L.; Chen, H.; Regan, A.C. Hyperspectral image classification based on deep attention graph convolutional network. *IEEE Trans. Geosci. Remote. Sens.* **2021**, *60*, 1–16. [CrossRef]
3. Bai, J.; Yuan, A.; Xiao, Z.; Zhou, H.; Wang, D.; Jiang, H.; Jiao, L. Class incremental learning with few-shots based on linear programming for hyperspectral image classification. *IEEE Trans. Cybern.* **2020**, *52*, 5474–5485. [CrossRef] [PubMed]
4. Makki, I.; Younes, R.; Francis, C.; Bianchi, T.; Zucchetti, M. A survey of landmine detection using hyperspectral imaging. *ISPRS J. Photogramm. Remote Sens.* **2017**, *124*, 40–53. [CrossRef]
5. Gevaert, C.M.; Suomalainen, J.; Tang, J.; Kooistra, L. Generation of spectral–temporal response surfaces by combining multispectral satellite and hyperspectral UAV imagery for precision agriculture applications. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 3140–3146. [CrossRef]
6. Brown, A.J.; Walter, M.R.; Cudahy, T. Hyperspectral imaging spectroscopy of a Mars analogue environment at the North Pole Dome, Pilbara Craton, Western Australia. *Aust. J. Earth Sci.* **2005**, *52*, 353–364. [CrossRef]
7. Kuflik, P.; Rotman, S.R. Band selection for gas detection in hyperspectral images. In Proceedings of the 2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel, Eilat, Israel, 14–17 November 2012; pp. 1–4.
8. Salem, F.; Kafatos, M.; El-Ghazawi, T.; Gomez, R.; Yang, R. Hyperspectral image analysis for oil spill detection. In Proceedings of the Summaries of NASA/JPL Airborne Earth Science Workshop, Pasadena, CA, USA, 27 February–2 March 2001; pp. 5–9.
9. Awad, M. Sea water chlorophyll-a estimation using hyperspectral images and supervised artificial neural network. *Ecol. Inform.* **2014**, *24*, 60–68. [CrossRef]
10. Jay, S.; Guillaume, M. A novel maximum likelihood based method for mapping depth and water quality from hyperspectral remote-sensing data. *Remote Sens. Environ.* **2014**, *147*, 121–132. [CrossRef]
11. Jänicke, C.; Okujeni, A.; Cooper, S.; Clark, M.; Hostert, P.; van der Linden, S. Brightness gradient-corrected hyperspectral image mosaics for fractional vegetation cover mapping in northern california. *Remote Sens. Lett.* **2020**, *11*, 1–10. [CrossRef]
12. Li, J.; Pang, Y.; Li, Z.; Jia, W. Tree species classification of airborne hyperspectral image in cloud shadow area. In *International Symposium of Space Optical Instrument and Application*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 389–398.
13. Li, J.; Bioucas-Dias, J.M.; Plaza, A. Semisupervised hyperspectral image classification using soft sparse multinomial logistic regression. *IEEE Geosci. Remote Sens. Lett.* **2012**, *10*, 318–322.
14. Zhong, Y.; Zhang, L. An adaptive artificial immune network for supervised classification of multi-/hyperspectral remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2011**, *50*, 894–909. [CrossRef]

15. 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]
16. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [CrossRef]
17. Zhang, H.; Li, Y.; Zhang, Y.; Shen, Q. Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network. *Remote Sens. Lett.* **2017**, *8*, 438–447. [CrossRef]
18. Li, Y.; Zhang, H.; Shen, Q. Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sens.* **2017**, *9*, 67. [CrossRef]
19. 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]
20. Xu, Y.; Li, Z.; Li, W.; Du, Q.; Liu, C.; Fang, Z.; Zhai, L. Dual-Channel Residual Network for Hyperspectral Image Classification With Noisy Labels. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5502511. [CrossRef]
21. Bai, J.; Huang, S.; Xiao, Z.; Li, X.; Zhu, Y.; Regan, A.C.; Jiao, L. Few-shot hyperspectral image classification based on adaptive subspaces and feature transformation. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–17. [CrossRef]
22. Hu, L.; Luo, X.; Wei, Y. Hyperspectral Image Classification of Convolutional Neural Network Combined with Valuable Samples. *J. Phys. Conf. Ser.* **2020**, *1549*, 052011. [CrossRef]
23. Li, Z.; Liu, M.; Chen, Y.; Xu, Y.; Li, W.; Du, Q. Deep Cross-Domain Few-Shot Learning for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5501618. [CrossRef]
24. Liu, S.; Shi, Q.; Zhang, L. Few-shot hyperspectral image classification with unknown classes using multitask deep learning. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 5085–5102. [CrossRef]
25. Miao, J.; Wang, B.; Wu, X.; Zhang, L.; Hu, B.; Zhang, J.Q. Deep Feature Extraction Based on Siamese Network and Auto-Encoder for 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. 397–400.
26. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
27. Wang, Q.; Liu, S.; Chanussot, J.; Li, X. Scene classification with recurrent attention of VHR remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 1155–1167. [CrossRef]
28. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
29. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
30. Zhang, H.; Wu, C.; Zhang, Z.; Zhu, Y.; Lin, H.; Zhang, Z.; Sun, Y.; He, T.; Mueller, J.; Manmatha, R.; et al. Resnest: Split-attention networks. *arXiv* **2020**, arXiv:2004.08955.
31. Xu, Z.; Zhang, W.; Zhang, T.; Yang, Z.; Li, J. Efficient transformer for remote sensing image segmentation. *Remote Sens.* **2021**, *13*, 3585. [CrossRef]
32. Zhang, J.; Zhao, H.; Li, J. TRS: Transformers for Remote Sensing Scene Classification. *Remote Sens.* **2021**, *13*, 4143. [CrossRef]
33. Qing, Y.; Liu, W.; Feng, L.; Gao, W. Improved Transformer Net for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 2216. [CrossRef]
34. He, J.; Zhao, L.; Yang, H.; Zhang, M.; Li, W. HSI-BERT: Hyperspectral image classification using the bidirectional encoder representation from transformers. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 165–178. [CrossRef]
35. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
36. Wu, H.; Xiao, B.; Codella, N.; Liu, M.; Dai, X.; Yuan, L.; Zhang, L. Cvt: Introducing convolutions to vision transformers. *arXiv* **2021**, arXiv:2103.15808.
37. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*.
38. Feng, J.; Feng, X.; Chen, J.; Cao, X.; Zhang, X.; Jiao, L.; Yu, T. Generative adversarial networks based on collaborative learning and attention mechanism for hyperspectral image classification. *Remote Sens.* **2020**, *12*, 1149. [CrossRef]
39. Zhao, W.; Chen, X.; Chen, J.; Qu, Y. Sample generation with self-attention generative adversarial Adaptation Network (SaGAAN) for Hyperspectral Image Classification. *Remote Sens.* **2020**, *12*, 843. [CrossRef]
40. Archibald, R.; Fann, G. Feature selection and classification of hyperspectral images with support vector machines. *IEEE Geosci. Remote Sens. Lett.* **2007**, *4*, 674–677. [CrossRef]
41. Makantasis, K.; Karantzalos, 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; pp. 4959–4962.