IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 16 December 2022, accepted 21 December 2022, date of publication 26 December 2022,
date of current version 25 January 2023.

Digital Object Identifier 10.1109/ACCESS.2022.3232152

RESEARCH ARTICLE

Two-Branch Generative Adversarial Network With Multiscale Connections for Hyperspectral Image Classification

DONGMEI SONG, YUNHE TANG, BIN WANG, JIE ZHANG, AND CHANGLONG YANG
College of Oceanography and Space Informatics, China University of Petroleum (East China), Qingdao 266580, China

Corresponding author: Dongmei Song (songdongmei@upc.edu.cn)

This work was supported by the Key Program of Joint Fund of the National Natural Science Foundation of China and Shandong Province under Grant U1906217 and U22A20586, the Natural Science Foundation of Shandong Province under Grant ZR2022MD015, the National Natural Science Foundation of China under Grant 41701513, 41772350 and 61371189, and the Key Research and Development Program of Shandong Province under Grant 2019GGX101033.

ABSTRACT Hyperspectral image (HSI) classification has always drawn great attention in the field of remote sensing. Various deep learning models are in the ascendant and gradually applied to HSI classification. Nevertheless, limited-labeled and class-imbalanced datasets largely make the classifier prone to overfitting. To address the above problem, this article proposes a two-branch generative adversarial network with multiscale connections (TBGAN), which includes two generators to produce the spectral and spatial samples, respectively. Thereinto, the spectral generator is imbued with the self-attention mechanism to maximumly capture the long-term dependencies across the spectral bands. And meanwhile, an elaborated discriminator with two branches is devised in TBGAN for extracting the joint spectral-spatial features. Besides, the multiscale connections are placed between the discriminator and two generators to alleviate the instability problems caused by the inherently backward propagation of gradients in GAN. Furthermore, a feature-matching term is added to the loss function to prevent the generators from overtraining upon the current discriminator, thereby further improving the stability of the network. Experiments upon three benchmark datasets demonstrate that TBGAN achieves an extremely competitive classification accuracy and exerts lower sensitivity to the training sample size compared with several state-of-the-art methods.

INDEX TERMS Hyperspectral image classification, generative adversarial network, multiscale connections, joint spectral-spatial features.

I. INTRODUCTION

As an essential observation technology, hyperspectral remote sensing can simultaneously capture the spectral and spatial features of ground objects in a scene. By far, hyperspectral images (HSI) are widely used in urban planning [1], agricultural monitoring [2], mineral exploration [3], and military reconnaissance [4]. HSI classification is a prerequisite for many applications of HSI, nevertheless.

Various supervised methods have been applied to the HSI classification field in the machine learning community. Generally, these algorithms divide the spectral space with the obtained decision plane as the boundary, such as K-nearest neighbor (KNN) [5], maximum likelihood [6], neural network [7], logistic regression [8], random forest (RF) [9], and support vector machine (SVM) [10], [12]. However, the feature extraction and classifier design in these methods are conducted independently, which makes the extracted features may not be optimal for the classifier, thus reducing the classification accuracy. Additionally, due to the high dimensionality of HSI, most of these algorithms dramatically suffer from the so-called curse of dimensionality.

Compared with the abovementioned shallow models, deep learning-based models can train the classifier in a data-driven

The associate editor coordinating the review of this manuscript and approving it for publication was Yi Zhang.

manner and extract the hierarchical features simultaneously, forming a unified end-to-end framework. Consequently, deep learning has gradually become a powerful tool for HSI classification in recent years [13]. Typical deep learning models include convolutional neural network (CNN) [14], [24], stacked autoencoder (SAE) [25], recurrent neural network (RNN) [26], and deep belief network (DBN) [27]. Among the abovementioned deep learning-based methods, the inputs of RNN, SAE, and DBN are composed of spectral vectors, without containing the spatial features, thereby largely giving rise to unsatisfactory classifications. Nevertheless, CNN can simultaneously extract the spectral and spatial features of HSI and adopt the strategies of local connections and weight sharing to reduce the number of parameters, drawing great attention in the field of HSI classification. Hu et al. [14] first designed a five-layer CNN with the spectrum of each pixel as input and extracted the spectral features to perform the classification. Besides, a pixel-pair voting strategy enabled a one-dimensional convolutional neural network (1D-CNN) to achieve a promising classification result in the case of limited training samples [15]. However, due to the lack of texture and context information of the samples, 1D-CNN is prone to suffer from misclassification. Therefore, some scholars [16], [21] have introduced spatial features into the network to construct joint spectral-spatial frameworks, which can be roughly divided into two categories. One paradigm is to construct a two-branch structure, in which each branch extracts the spectral or spatial features respectively, and then concatenates these features for the classification [16], [18]. For example, Xu et al. [16] developed a spectral-spatial unified network (SSUN), employing a long short-term memory model (LSTM) and multiscale convolutional neural network to respectively extract the spectral and spatial features. The other paradigm receives the 3D cubes containing the spectral and spatial information and extracts the joint features by one or more convolutional operators [19], [21]. For instance, the multiscale 3D deep convolutional neural network (M3D-DCNN) [20] utilized 3D convolutional operators to extract the multiscale spatial and spectral features, announcing impressive results. In addition, there are also some studies [22], [24] that combine CNN with self-supervised learning, using a large number of unlabeled data and achieving promising classification results.

Although CNN-based methods have achieved excellent classification results, they are prone to overfitting when tuning the substantial learnable parameters with limited training data [28], [29]. However, gathering data is expensive and time-consuming in the field of remote sensing, and the obtained data generally take on long-tail distribution, which hinders the application of CNN.

Generative adversarial network (GAN) [30] was put forward to generate high-quality images through its unique adversarial training process between the generator and discriminator. With the advancement of GAN, hundreds of its variations have been derived. Among them, the relatively popular models are conditional generative adversarial network (CGAN) [31], deep convolutional generative adversarial network (DCGAN) [32], and Wasserstein GAN [33]. To alleviate the above overfitting problem of CNN-based methods, some scholars [34], [44] introduced GAN into HSI classification, which yields encouraging classification results under the circumstance of small-size samples. Zhan et al. [34] proposed a semi-supervised classification method based on 1D-GAN, which is the first application of GAN for HSI classification. A DCGAN-based method was proposed, in which the discriminator leveraged the first three principal components after the operation of principal components analysis (PCA) upon the original image as the inputs, with commendable classification results obtained [35]. Zhan et al. [36] further classified the samples via the voting mechanism of the dynamic neighborhood after the first classification using the spectral feature only. A novel multiclass spatial-spectral GAN (MSGAN) method [37] was developed with two generators to produce the fake spectral and spatial samples, respectively, and defined the novel adversarial objectives for multiclass, which achieves astounding results. For the sake of excavating the rich information from unlabeled samples, the generator network in multitask GAN (MTGAN) [38] was designed to simultaneously undertake the reconstruction and the classification tasks. To improve the generalization performance, a self-attention-based GAN [39] was combined with the variational auto-encoder (VAE) [45], in which the generator received the encoder-generated and random latent vectors to produce more enhanced virtual samples.

Even if the above GAN-based models have gained satisfying HSI classification performance, the training quality of models hinges on the gradients transmitted from the discriminator to the generator. Hence, the gradients may disappear due to accumulation when the layer of GAN is too deep. Furthermore, Arjovsky and Bottou [46] put forward the point that when there is an insubstantial overlap between the distribution of the real and the generated data, the discriminator will pass uninformative gradients to the generator. The above problems are the major contributors to the training instability of GAN, which hinders its classification accuracy. To improve the training stability, a multiscale gradients GAN (MSG-GAN) [47] was developed for synthesizing the high-resolution faces, which connected the intermediate layers of the generator with that of the discriminator, making the multiscale gradients can be directly passed from the discriminator to the generator. To solve the training instability problem of GAN for the task of HSI classification, this article establishes the multiscale connections between the discriminator and generators inspired by MSG-GAN. The main contributions of this article are summarized as follows.

1) We propose a two-branch generative adversarial network with multiscale connections (TBGAN) for HSI classification. Generators in TBGAN will produce the virtual spectrums and spatial patches to alle-viate the small-size sample problem.
2) To improve the training stability, the multiscale connections are established between the discriminator and two

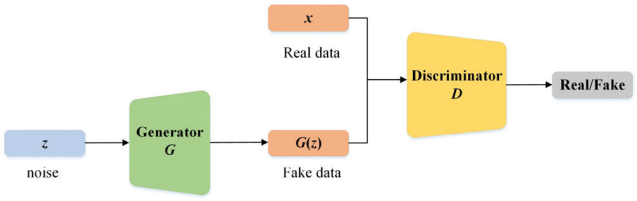


FIGURE 1. Framework of GAN.

generators. Moreover, a feature-matching term is added to the loss function to further increase the stability.

3) The discriminator with two branches is designed in TBGAN to extract the joint spectral-spatial features. The trained discriminator can be employed as a classifier.

II. METHODOLOGY

A. BASIC FRAMEWORK OF GAN

Before formally introducing the TBGAN method, we first review the basics of GAN. Motivated by the two-person zero-sum game theory, the GAN model [30] is proposed by taking the adversarial training process to optimize deep learning models as a new framework, which consists of a generator G and a discriminator D, as exhibited in Fig. 1. G tries to capture the potential distribution of real data and output the fake data, while D undertakes a binary classification task that can judge whether the input sample is real or not. Specifically, G takes a random noise z as input and attempts to generate the fake data $G(z)$. D uses the real data x or the fake data $G(z)$ as input and outputs the probability of the input attributable to the real data. The objective function of GAN is defined as follows:

$$\min_G \max_D V(D, G) = E_{x\sim p_{data}(x)}\left[\log D(x)\right] + E_{z\sim p_z(z)}\left[\log\left(1 - D(G(z))\right)\right] \quad (1)$$

where E denotes the expectation operator, $p_{\text{data}}(x)$ and $p_z(z)$ indicate the distributions of the real data and the noise, respectively. In the optimized procedure, D desires to distinguish as precisely as possible, that is, maximizing $V(D, G)$. While G has the opposite objective, which attempts to fool D by generating real-like data to minimize $V(D, G)$. The parameter updating of G relies on the backward propagation of D. As one module is updated, the other is fixed, and they evolve alternately until their capabilities reach an equilibrium.

B. PROPOSED METHOD

Inspired by the adversarial training mechanism of GAN, this article proposes a TBGAN framework for the classification of ground objects by extracting the joint spectral-spatial features. Similar to the traditional GAN, TBGAN also consists of the generator and the discriminator. As can be seen from Fig. 2, there are two branches devised in TBGAN, which is specifically composed of three modules: the spectral generator G_{spec}, the spatial generator G_{spat}, and the discriminator D. To generate the corresponding virtual samples, two

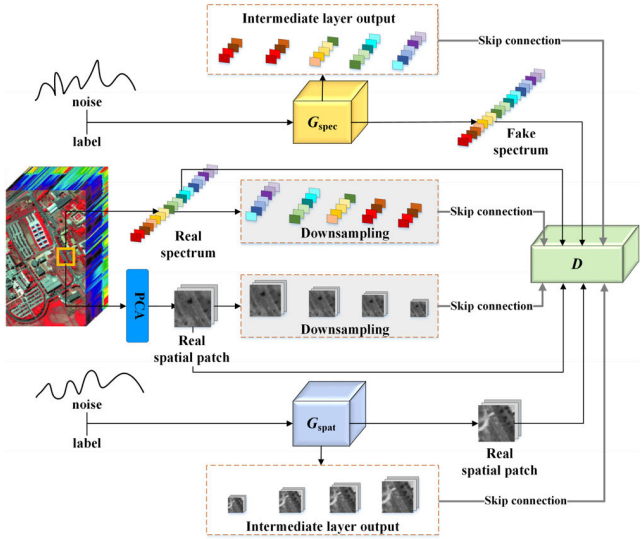


FIGURE 2. The framework of the proposed TBGAN.

generators receive both noises and labels as input and learn spectral and spatial data distribution of real images respectively. D employs both real and virtual samples as the inputs, which aims to extract the joint spectral-spatial features and eventually achieve the classification task. Here the real spatial samples are the cubes cropped around each pixel upon the first three principal components through PCA transformation. By picking out only the first few dominant components, the spectral dimension can thus be reduced.

It is worth noting that the intermediate layers of D are connected with their counterparts in two generators and the multiscale spectral/spatial features of real samples after downsampling. This kind of skip connection allows D to consider the multiscale features of both the real and the fake samples, thus enhancing its discriminative ability. Besides, such multiscale connections make the gradients be passed directly from D to the intermediate layers of two generators, which effectively avoids the circumstance of training instability caused by gradient accumulation in the previous GAN models.

1) SPECTRAL AND SPATIAL GENERATORS OF TBGAN

The generators G_{spec} and G_{spat} are employed to generate virtual samples containing spectral and spatial information, respectively. As shown in Fig. 3, the inputs of two generators are (z_{spec}, y) and (z_{spat}, y), where z_{spec} and z_{spat} represent the noise vectors and y denotes the one-hot coded labels. By concatenating labels and noises as input, the generators can learn the class-specific features during training, thus reducing the possibility of model collapse [31]. The virtual spectrums generated by G_{spec} are depicted by $G_{\text{spec}}(z_{\text{spec}}, y)$, while the virtual spatial patches generated by G_{spat} are depicted as $G_{\text{spat}}(z_{\text{spat}}, y)$. In addition, each virtual sample is assigned into class $n+1$ (n is the number of dataset classes) and endowed with an artificial label $y_{fake} = \frac{1}{n}(1, 1, \ldots, 1)_n^{\mathrm{T}}$.

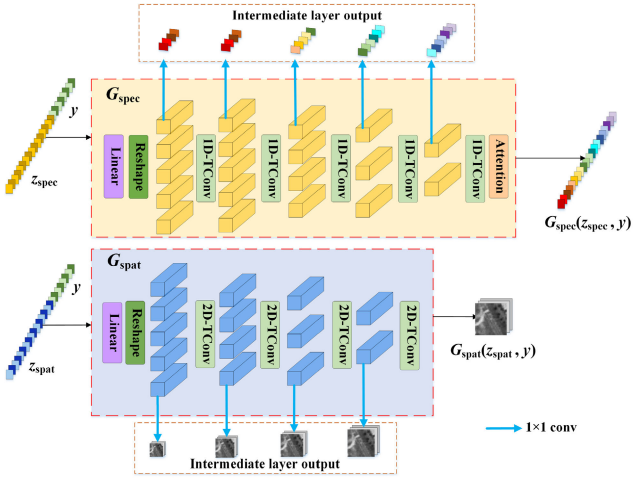


FIGURE 3. The architectures of the two generators in TBGAN.

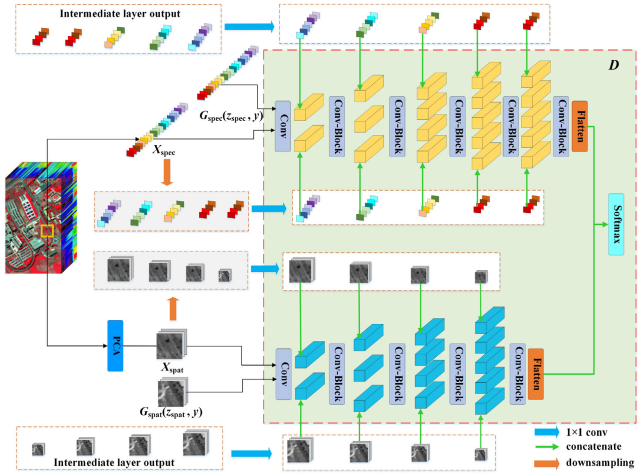


FIGURE 4. The architecture of the discriminator in TBGAN.

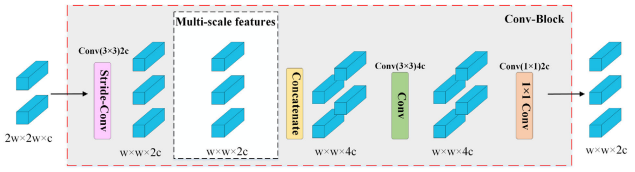


FIGURE 5. The construction of Conv-Block in discriminator.

G_{spec} contains five 1-D transposed convolutional layers (1D-TConv), whose kernel size is 5. G_{spat} is stacked by four 2D-TConv, and the kernel size of each layer is 3×3. Except for the last TConv layer that takes tanh as the activation function, each TConv layer in both generators utilizes the rectified linear units (ReLUs) as the nonlinear activation function and adopts batch normalization strategy.

However, existing models still find obstacles when capturing long-term dependencies across the spectral bands due to extensive bands in HSI [48]. Recently, the self-attention mechanism [49] has become a breakthrough with high hope to effectively address the above issue by obtaining global information of the feature maps through simple query and assignment operations [50]. Therefore, self-attention is drawn into G_{spec} to calculate the response of all bands in the spectral sequence to a certain band. The self-attention layer is placed at the end of G_{spec}, because the feature maps achieve the largest after five 1D-TConv operations, thus making the self-attention mechanism perform well.

2) DISCRIMINATOR OF TBGAN

In this article, a two-branch discriminator D is designed to fulfill the task of ground object classification by exploiting the joint spectral-spatial features. The architecture of discriminator D is depicted in Fig. 4. There are two sources of the input samples for D: one is the spectrums and spatial patches of the real images, denoted by (X_{spec}, X_{spat}), and the other is the virtual samples generated by two generators, represented by $(G_{spec}(z_{spec}, y), G_{spat}(z_{spat}, y))$. In particular, each branch of the discriminator D consists of several Conv-Blocks, which can excavate the spectral or spatial features of the input samples. Besides, the pooling operation is replaced by the strided convolution in all Conv-Blocks so as to achieve the adaptive learning of downsampling.

Fig. 5 exhibits the structure of Conv-Block in the spatial branch, which is nearly consistent with that in the spectral branch. For the input feature maps, its height and width are labeled as 2w, and c is the number of channels. To obtain the spatial features, the Conv-Block first performs strided convolution to halve the size of feature maps and double the number of channels and then concatenates the handled feature maps with the multiscale features. These multiscale features consist of the intermediate layer outputs of the generator and the downsampled versions of the real data. After that, the concatenated feature maps are delivered into a convolution layer, whose kernel size is 3×3 (5×1 in the spectral branch). During this convolution, the size and quantity of feature maps remain unchanged. Finally, the number of channels is halved by further executing 1×1 convolution. After the implementation of four successive Conv-Blocks, the output spatial features are flattened as a one-dimensional vector. Similarly, the spectral features can also be flattened into a vector after adopting five successive Conv-Blocks in the spectral branch. By further concatenating these two vectors, the joint spectral-spatial features are extracted. Then, these joint features are delivered into a softmax layer to achieve the classification of ground objects.

To avoid the network training instability caused by gradient accumulation, the intermediate layers of D are connected with their counterparts in two generators in a manner of combining 1×1 convolution with concatenation. For matching the size of the feature maps from the intermediate layers of generators, the real spectrums, and spatial patches are downsampled in an interlaced fashion.

Meanwhile, the quantity of the intermediate layer outputs in G_{spec} and G_{spat} are reduced by 1×1 convolution corresponding to different downsampled versions of the real

samples. Subsequently, the acquired feature maps and the down-sampled real data are extended by 1×1 convolution respectively to produce the multiscale features, which have the same channels as the intermediate layer outputs in D. Finally, the multiscale features are concatenated with the counterparts from the intermediate layers in D and then delivered into the corresponding Conv-Blocks.

Here $D(X_{\text{spec}}, X_{\text{spat}})$ and $D(G_{\text{spec}}(z_{\text{spec}}, y), G_{\text{spat}}(z_{\text{spat}}, y))$ denote the discriminant results of D for the real and virtual samples, respectively. Note that each convolutional layer in D employs the leaky-rectified linear units (Leaky-ReLUs) as the nonlinear activation function. Meanwhile, each layer applies the batch normalization strategy except for the input and the output layers.

3) LOSS FUNCTION OF TBGAN

The discriminator in the classical GAN utilizes the sigmoid classifier to distinguish whether an input is true or false, which pertains to binary classification. For the circumstance of multi-classification, the discriminator in ACGAN [51] method is imbued with a softmax classifier to undertake the multi-classification task. In recent years, to improve the adversarial training effects upon multi-classification, a multi-class adversarial strategy [37] is devised, which enables the softmax layer simultaneously complete the discrimination of input sources and the classification task. For this reason, this multi-class adversarial strategy is also introduced into TBGAN. Meanwhile, a feature-matching term is also added to the loss function, thus facilitating the generated samples preferably subject to the distribution of the real data. Consequently, the loss function of TBGAN can be defined as follows:

$$\begin{cases} L_G = L_c + \lambda L_s \\ L_D = L_{real} + L_{fake} \end{cases} \tag{2}$$

where L_G and L_D represent the loss functions corresponding to the generators and discriminator, respectively. L_c denotes the categorical loss of the virtual samples corresponding to the true labels y, and L_s is the summation of the matching losses of the spectral and spatial features. The hyperparameter λ is utilized to trade off L_c and L_s. L_{real} depicts the categorical loss of the real samples, and L_{fake} represents the categorical loss of the virtual samples with the artificial label $y_{fake} = \frac{1}{n}(1, 1, \ldots, 1)_n^{\mathrm{T}}$. Specifically, these losses can be calculated as follows:

$$\begin{cases} L_c = CE\left(D(G_{\text{spec}}(z_{\text{spec}}, y), G_{\text{spat}}(z_{\text{spat}}, y)), y\right) \\ L_s = \left\| f_1(X_{\text{spec}}) - f_1(G_{\text{spec}}(z_{\text{spec}}, y)) \right\|_2^2 \\ \qquad + \left\| f_2(X_{\text{spat}}) - f_2(G_{\text{spat}}(z_{\text{spat}}, y)) \right\|_2^2 \\ L_{real} = CE\left(D(X_{\text{spec}}, X_{\text{spat}}), y\right) \\ L_{fake} = CE\left(D(G_{\text{spec}}(z_{\text{spec}}, y), G_{\text{spat}}(z_{\text{spat}}, y)), y_{fake}\right) \end{cases} \tag{3}$$

where $\mathrm{CE}(\cdot)$ denotes the cross entropy, $f_1(\mathrm{x})$ and $f_2(\mathrm{x})$ depict the output of the flatten layers in the spectral and spatial branches of D, respectively. The objective of the generators is to make the discriminator distinguish the virtual samples as a certain class in the dataset and match the expected value of the features from flatten layers. Whereas D aims to furthest improve the multi-classification accuracy of the real samples and classify the virtual samples as the class of $n + 1$.

Besides, to alleviate the overconfidence of the discriminator, the labels in (3) can be smoothed complying with the strategy adopted in [52]. Concretely, by introducing a hyperparameter of ε, the elements of 0 and 1 in vector y are substituted with ε and 1-ε, respectively.

4) PROCEDURE OF TBGAN

As shown in Table 1, the specific procedure of the TBGAN method consists of the virtual sample generation, extracting the joint spectral-spatial features, and the ground object classification.

TABLE 1. Procedure of the proposed TBGAN method.

Input: the training sample $x_{train}=(x_{\text{spec}}, x_{\text{spat}})_{i=1}^{N}$ and test samples $x_{test}=(x_{\text{spec}}, x_{\text{spat}})_{i=1}^{M}$ from n classes, the label-smoothed one-hot labels $(y_i)_{i=1}^{N}$ of training samples, batch size b, the number of training epochs e, noise dimensions a_1 and a_2, updating times k of the discriminator
Initialize all the weight matrices and biases
for every epoch
  for every batch
    sample a_1 dimensional noises $z_{\text{spec}} = (z_{1_i})_{i=1}^{b}$
    sample a_2 dimensional noises $z_{\text{spat}} = (z_{2_i})_{i=1}^{b}$
    concatenate z_{spec} and z_{spat} with the smoothed class labels $(y_i)_{i=1}^{b}$
    generate b samples $\{G_{\text{spec}}(z_{\text{spec}}, y_i), G_{\text{spat}}(z_{\text{spat}}, y_i)\}_{i=1}^{b}$
    for k steps do
      update parameters of discriminator by minimizing L_D
    end
    update parameters of G_{spec} and G_{spat} by minimizing L_G
  end
end
Output: the labels of test samples x_{test} predicted by the trained discriminator in TBGAN

III. EXPERIMENTS

To demonstrate the classification performance of the proposed TBGAN, the experiments are conducted upon the Pavia University, the Salinas, and the Indian Pines dataset. In the experiments, 10% of the labeled samples are randomly selected for training, and the remainder is used for testing. Besides, class accuracy, average accuracy (AA), overall accuracy (OA), and Kappa coefficient are employed as indicators for measuring the classification results.

A. DATA DESCRIPTION

1) PAVIA UNIVERSITY DATASET

The Pavia University dataset actually captured pictures of Pavia, an Italian city, by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor during a flight campaign in 2003. The imaging wavelength of ROSIS ranges from 430 to 860 nm, in which 103 spectral bands are retained after removing 12 bands significantly affected by noise. This

dataset contains 610 × 340 pixels, where 9 classes of ground objects are labeled, including Trees, Asphalt, Brick, etc. Table 2 shows the specific sample distribution on the Pavia University dataset used for training and testing.

TABLE 2. Sample distribution on the Pavia University Dataset.

NO.	Class	Training	Testing
1	Asphalt	663	5968
2	Meadows	1865	16784
3	Gravel	210	1889
4	Trees	306	2758
5	Metal Sheets	134	1211
6	Bare Soil	503	4526
7	Bitumen	133	1197
8	Bricks	368	3314
9	Shadows	95	852
Total		4277	38499

2) SALINAS DATASET

The Salinas dataset was gathered by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor over the Salinas Valley. The imaging wavelength range of AVIRIS is from 400 to 2500 nm, in which 204 bands are available after eliminating the bands absorbed by water. This dataset is in size of 512 × 217 pixels with a resolution of 3.7m. Among them, the labeled pixels are divided into 16 categories, including Fallow, Celery, Stubble, etc. Table 3 reports the number of samples used for training and testing.

TABLE 3. Sample distribution on the Salinas Dataset.

NO.	Class	Training	Testing
1	Brocoli_green_weeds_1	201	1808
2	Brocoli_green_weeds_2	372	3354
3	Fallow	197	1779
4	Fallow_rough_plow	139	1255
5	Fallow_smooth	268	2410
6	Stubble	396	3563
7	Celery	358	3221
8	Grapes_untrained	1127	10144
9	Soil_vinyard_develop	620	5583
10	Corn_senesced_green	328	2950
11	Lettuce_romaine_4wk	107	961
12	Lettuce_romaine_5wk	193	1734
13	Lettuce_romaine_6wk	91	825
14	Lettuce_romaine_7wk	107	963
15	Vinyard_untrained	727	6541
16	Vinyard_vertical	181	1626
Total		5412	48717

3) INDIAN PINES DATASET

The Indian Pines dataset was also collected by AVIRIS sensors, with a size of 145 × 145 pixels. After eliminating 24 bands absorbed by water, 200 spectral bands are reserved. This dataset contains 10,249 labeled pixels, which are divided into 16 categories, including Alfalfa, Corn-notill, Corn-mintill, etc. Table 4 shows the number of samples used for training and testing.

B. EXPERIMENTAL SETTING

To evaluate the performance of the proposed TBGAN model, the experiments are designed comparative to six representative HSI classification methods, including RBF-SVM [12], RF [9], LSTM [14], SSUN [14], M3D-DCNN [20], and DCGAN [35]. Meanwhile, the exploratory experiments are additionally conducted for the models of TBGAN containing the spectral branch or spatial branch only, which are named TB-SPE and TB-SPA, respectively. In the comparison models, the hyper-parameters, such as gamma and C in RBF-SVM and the number of decision trees in RF, are optimally sought out by grid-search, while the parameter configurations of other deep learning models comply with their sources. The detailed configurations of TBGAN are illustrated in Table 5 and Table 6, where Conv represents the convolutional layer, Tconv expresses the transposed convolutional layer, Atten represents the self-attention layer, and BN indicates the batch normalization. The learning rate of both the generators and the discriminator is set at 0.0002, the epoch is set to 500, and the batch size is 64. TBGAN adopts the Adam method [53] as the optimizer to adaptively adjust the learning rate. The dimensions of both z_{spec} and z_{spat} are set as 100, and the smoothing parameter ε is empirically set to 0.1. The parameter k is set to 3, indicating that, the discriminator will be updated three times when the generators are updated once. All hyper-parameters of TB-SPE and TB-SPA are configured the same as TBGAN.

Besides, the experiments are proceeded in the Pytorch backend with NVIDIA 1080Ti (number of cores: 1, RAM:11GB, Cuda version: 11.0). Since the models may be

TABLE 4. Sample distribution on the Indian Pines Dataset.

NO.	Class	Training	Testing
1	Alfalfa	5	41
2	Corn-notill	143	1285
3	Corn-mintill	83	747
4	Corn	24	213
5	Grass-pasture	48	435
6	Grass-trees	73	657
7	Grass-pasture-mowed	3	25
8	Hay-windrowed	48	430
9	Oats	2	18
10	Soybean-notill	97	875
11	Soybean-mintill	245	2210
12	Soybean-clean	59	534
13	Wheat	20	185
14	Woods	126	1139
15	Buildings-Grass-Trees-Drives	39	347
16	Stone-Steel-Towers	9	84
Total		1024	9225

TABLE 5. Detailed configurations of G_{spec} and G_{spat}.

Name	Layer	Kernel	Stride	Padding	BN	Activation
G_{spec}	Tconv1	5×16	2	2	Yes	ReLu
	Tconv2	5×8	2	2	Yes	ReLu
	Tconv3	5×4	2	2	Yes	ReLu
	Tconv4	5×2	2	2	Yes	ReLu
	Tconv5	5×1	2	2	No	Tanh
	Atten	1×1	1	-	-	-
G_{spat}	Tconv1	3×3×64	2	1	Yes	ReLu
	Tconv2	3×3×32	2	1	Yes	ReLu
	Tconv3	3×3×16	2	1	Yes	ReLu
	Tconv4	3×3×3	2	1	No	Tanh

TABLE 6. Detailed configurations of *D*.

Name	Layer		Kernel	Stride	Padding	BN	Activation
D	Conv-block1	Conv1-1	5×2/3×3×16	2/2	2/1	No	
		Conv1-2	3×4/3×3×32	1/1	1/1	Yes	LeakyRelu
		Conv1-3	1×2/1×1×16	1/1	0/0	Yes	
	Conv-block2	Conv2-1	5×4/3×3×32	2/2	2/1	Yes	
		Conv2-2	3×8/3×3×64	1/1	1/1	Yes	LeakyRelu
		Conv2-3	1×4/3×3×32	1/1	0/0	Yes	
	Conv-block3	Conv3-1	5×8/3×3×64	2/2	2/1	Yes	
		Conv3-2	3×16/3×3×128	1/1	1/1	Yes	LeakyRelu
		Conv3-3	1×8/1×1×64	1/1	0/0	Yes	
	Conv-block4	Conv4-1	5×16/3×3×128	2/2	2/1	Yes	
		Conv4-2	3×32/3×3×256	1/1	1/1	Yes	LeakyRelu
		Conv4-3	1×16/1×1×128	1/1	0/0	Yes	
	Conv-block5	Conv5-1	5×16	2	2	Yes	LeakyRelu
		Conv5-2	3×32	1	1	Yes	

spectral branch / spatial branch

influenced by random initializations, the mean and standard deviation of classification results after ten runs are taken as the final experimental basis.

C. CLASSIFICATION RESULTS

Tables 7 - 9 present the classification results of nine methods upon the Pavia University, the Salinas, and the Indian Pines dataset, respectively. Each table records, from top to bottom, the means of class accuracy, AA, OA, and Kappa coefficients after ten runs, as well as the standard deviations of the latter three evaluation metrics. As can be seen from Tables 7 - 9, the deep learning methods generally behave better in classification performance than traditional machine learning methods by the exploitation of hierarchical features. Furthermore, the GAN-based methods can generate more training samples, which is very helpful for network training and makes them achieve higher accuracy compared with other deep learning methods. Among these GAN-based methods, TBGAN exceeds TB-SPE and TB-SPA, which strongly demonstrates the superiority of utilizing the joint spectral-spatial features. DCGAN and TB-SPA achieve encouraging results, which can be attributed to the utilization of PCA transformation, partly introducing spectral information. By virtue of the multiscale connections and the two-branch structure, TBGAN obtains the best classification results among these nine methods. For the Pavia University dataset, TBGAN increases by 5.67%, 1.78%, and 0.13% respectively in terms of the OA index compared with LSTM, M3D-DCNN, and SSUN. For the Salinas dataset, TBGAN attains the optimal class accuracy for 12 classes, 7 of which reach 100% and the 10 times average of OA reaches 99.98%. In addition, TBGAN also achieves surpassing performance on the imbalanced Indian Pines dataset. For example, the prediction accuracy of 96.80% is given for the Grass-pasture-mowed class in the scenario of only 3 training samples.

In addition to the quantitative comparisons of the classification results in Tables 7 - 9, the qualitative visualization is also provided by creating the classification maps for each method on three HSI datasets. As exhibited in Fig. 6-8,

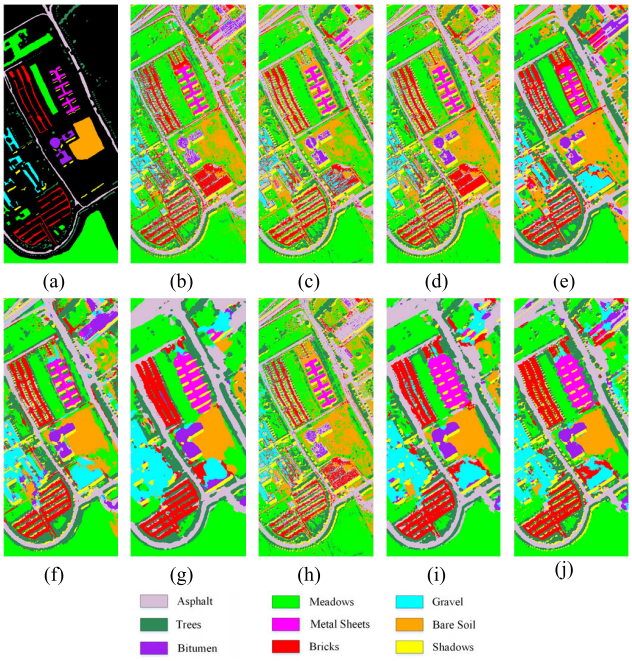


FIGURE 6. Classification maps of different models on the Pavia University dataset. (a) Ground truth, (b) RBF-SVM, (c) RF, (d) LSTM, (e) M3D-DCNN, (f) SSUN, (g) TB-SPE, (h) TB-SPA, and (i) TBGAN.

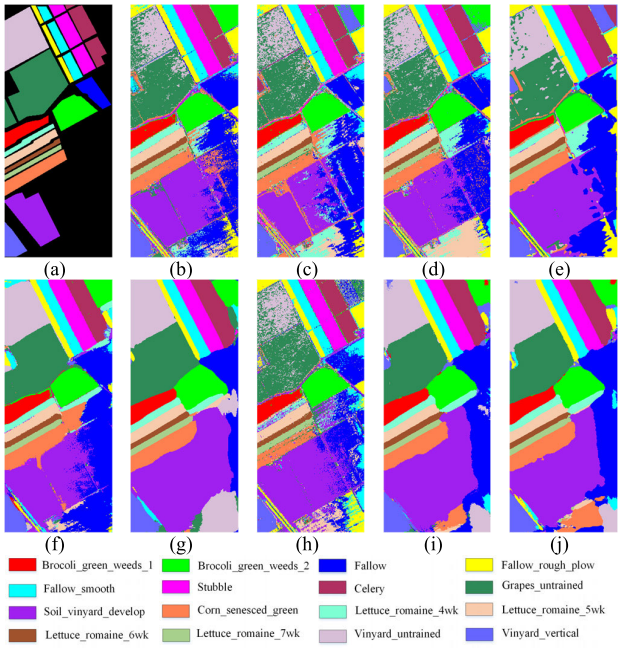


FIGURE 7. Classification maps of different models on the Salinas dataset. (a) Ground truth, (b) RBF-SVM, (c) RF, (d) LSTM, (e) M3D-DCNN, (f) SSUN, (g) TB-SPE, (h) TB-SPA, and (i) TBGAN.

it can be observed that the classification maps obtained by the TBGAN are closer to the ground truths and have fewer outliers compared with other methods, which further confirms the effectiveness of the proposed method. Moreover, because the input of TB-SPA is a 3D cube in size of 47 × 47×3, its prediction may be interfered by the substantial spatial homogeneity, thus making the classification results

TABLE 7. Comparative classification results of the models upon the pavia university dataset.

No.	RBF-SVM	RF	LSTM	M3D-DCNN	SSUN	DCGAN	TB-SPE	TB-SPA	TBGAN
1	90.70	91.26	93.80	98.02	99.65	99.95	90.76	99.95	**99.97**
2	93.77	97.54	97.96	99.01	99.95	**100**	93.12	99.98	99.99
3	83.89	66.28	80.38	97.78	98.62	**99.64**	63.06	99.31	99.41
4	96.22	89.15	93.03	99.02	99.61	98.50	90.53	99.09	**99.41**
5	99.25	98.90	99.29	99.90	99.94	99.96	99.39	**100**	**100**
6	91.21	64.54	90.60	97.51	99.97	**100**	85.21	99.99	99.99
7	83.24	76.75	85.82	99.17	98.97	99.95	72.87	**99.96**	**99.96**
8	84.61	89.48	89.40	96.77	99.35	99.21	85.15	**99.48**	99.37
9	**99.97**	99.37	99.54	99.13	99.73	98.14	99.87	99.51	99.87
OA（%）	91.95±1.34	89.29±0.25	94.19±0.22	98.46±0.47	99.73±0.05	99.75±0.03	89. 2±1.44	99.82±0.05	**99.86±0.06**
AA（%）	91.43±0.02	85.92±0.05	92.20±1.13	97.92±0.45	99.53±0.27	99.48±0.08	86.66±2.35	99.69±0.09	**99.77±0.11**
Kappa（%）	89.24±0.02	85.57±0.32	92.28±0.30	97.96±0.62	99.64±0.07	99.67±0.03	85.72±1.89	99.77±0.06	**99.82±0.08**

TABLE 8. Comparative classification results of the models upon the Salinas dataset.

No.	RBF-SVM	RF	LSTM	M3D-DCNN	SSUN	DCGAN	TB-SPE	TB-SPA	TBGAN
1	99.96	99.42	99.62	**100**	99.97	**100**	98.58	**100**	**100**
2	99.74	99.92	99.89	99.78	**100**	**100**	90.02	**100**	**100**
3	98.64	98.66	99.05	99.59	99.90	**100**	84.02	**100**	**100**
4	99.14	99.21	99.31	99.66	99.54	99.76	96.76	99.84	**99.90**
5	99	98.24	98.85	99.74	99.61	99.81	76.02	99.79	**99.84**
6	99.87	99.69	99.84	99.97	**100**	**100**	99.11	99.99	**100**
7	99.94	99.40	99.63	**100**	99.91	**100**	99.11	99.98	**100**
8	80.76	86.84	86.31	95.09	99.79	**100**	68.41	99.97	99.98
9	99.49	99.30	99.86	99.75	99.99	99.99	98.68	**100**	99.99
10	98.51	92.53	95.96	98.50	99.75	99.98	86.34	**99.99**	99.91
11	99.48	95.10	98.05	**100**	99.72	**100**	75.83	99.95	**100**
12	99.42	99.22	99.61	99.64	99.92	**100**	85.66	**100**	**100**
13	98.67	97.67	98.88	98.47	99.66	**99.99**	74.01	99.96	**99.99**
14	97.92	93.47	97.11	99.55	99.85	99.97	86.98	99.97	**99.98**
15	83.59	65.90	77.94	92.24	98.69	99.98	64.84	**99.99**	**99.99**
16	99.77	97.78	98.81	**100**	99.77	**100**	94.02	**100**	99.95
OA（%）	93.31±0.07	91.54±0.15	93.59±0.36	97.71±0.63	99.70±0.41	99.97±0.02	83.12±5.64	99.97±0.02	**99.98±0.01**
AA（%）	97.12±0.08	95.15±0.15	96.40±1.07	98.76±0.34	99.75±0.22	**99.97±0.02**	86.15±5.90	99.96±0.02	**99.97±0.02**
Kappa（%）	92.53±0.08	90.56±0.17	92.86±0.44	97.45±0.70	99.66±0.45	99.97±0.02	81.25±6.19	99.97±0.02	**99.98±0.01**

TABLE 9. Comparative classification results of the models upon the indian pines dataset.

No.	RBF-SVM	RF	LSTM	M3D-DCNN	SSUN	DCGAN	TB-SPE	TB-SPA	TBGAN
1	59.62	19.02	35.95	91.20	**97.62**	95.61	24.88	90.73	93.66
2	76.19	67.25	76.51	87.21	97.38	**98.57**	60.02	98.19	98.35
3	75.79	54.24	61.78	86.57	97.79	98.39	36.75	97.51	**98.51**
4	69.87	37.51	51.17	86.47	**96.73**	94.27	28.69	94.60	95.31
5	87.51	81.40	86.28	96.34	96.41	96.14	69.26	**96.57**	96.44
6	89.06	94.28	94.58	96.39	**98.74**	96.16	86.91	97.58	98.51
7	78.32	10.8	47.31	83.64	95.00	96	30.8	96.4	**96.80**
8	93.72	98.02	97.33	93.61	**99.86**	99.12	93.81	99.35	99.63
9	63.24	6.67	21.11	97.50	**95.55**	72.22	13.33	81.67	82.78
10	74.61	67.26	77.17	84.95	95.42	96.36	43.53	97.55	**97.62**
11	76.57	86.02	82.02	84.80	98.87	99.25	79.33	99.12	**99.28**
12	80.55	46.18	63.30	83.76	96.83	98.01	31.76	97.62	**97.85**
13	91.81	92.16	98.59	99.33	99.35	97.40	81.19	99.46	**99.57**
14	91.98	95.32	94.40	96.30	99.02	99.72	95.04	99.97	**99.99**
15	74.13	39.02	57.70	86.11	98.79	99.02	40.03	98.36	**99.08**
16	97.56	75.60	87.98	**97.50**	95.47	91.19	83.57	91.31	91.90
OA（%）	80.87±1.14	75.31±0.76	79.74±0.75	88.77±2.71	97.98±0.42	98.11±0.26	66.65±6.48	98.26±0.27	**98.54±0.31**
AA（%）	75.17±1.53	60.67±1.67	70.77±1.61	82.28±3.30	**97.42±0.67**	95.46±1.80	56.18±7.31	96.02±1.12	96.58±1.66
Kappa（%）	78.09±1.34	71.51±0.88	76.82±0.87	87.15±3.12	97.69±0.48	97.85±0.30	61.21±7.83	98.02±0.31	**98.33±0.35**

tend to be over-smoothed. Owing to the designed two-branch structure, TBGAN can extract the spectral information more thoroughly, which makes the classification results more refined compared with TB-SPA.

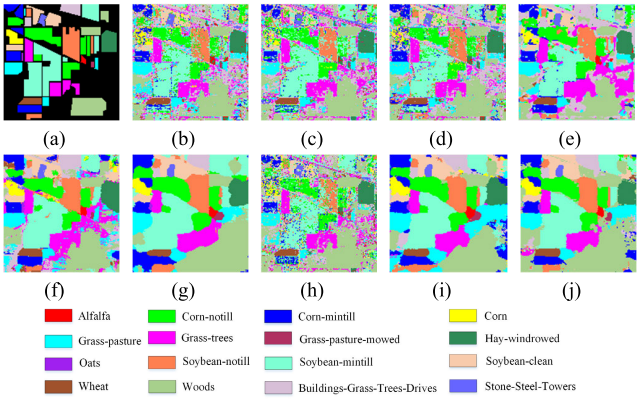


FIGURE 8. Classification maps of different models on the Indian Pines dataset. (a) Ground truth, (b) RBF-SVM, (c) RF, (d) LSTM, (e) M3D-DCNN, (f) SSUN, (g) TB-SPE, (h) TB-SPA, and (i) TBGAN.

TABLE 10. Model complexity of models (Taking the Salinas dataset as example).

Methods	Params (M)	FLOPs (M)
LSTM	0.19	0.93
M3D-DCNN	0.27	45.64
SSUN	1.22	2.7
DCGAN	10.5	20.92
TB-SPE	0.03	0.2
TB-SPA	1.18	67.88
TBGAN	1.21	68.28

D. MODEL COMPLEXITY

To assess the complexity of the proposed TBGAN, Table 10 presents the number of parameters (Params) and floating-point operations (FLOPs) of seven deep learning methods. The results suggest that TBGAN has fewer parameters than advanced SSUN and DCGAN, but the actual computation is slower than that of other models due to the two-branch structure.

For a more comprehensive evaluation, the running time of nine methods upon each dataset is provided in Table 11. Generally speaking, shallow models in the machine learning community are more efficient than deep learning algorithms. More significantly, the four GAN-based models take more time during the training stage than other deep learning models because both the generator and the discriminator need to be trained simultaneously. In particular, the proposed TBGAN and its sub-models TB-SPE and TB-SPA all adopt such a training strategy of updating the discriminator three times while updating the generator once. TBGAN requires longer training time than TB-SPE and TB-SPA, this is probably because TBGAN needs to extract the joint spectral-spatial features and update the two generators in each training iteration.

IV. DISCUSSION

The relevant experiments are carried out for exploring the impacts of some significant influencing factors such as patch size, hyper-parameter λ, self-attention mechanism, and the number of training samples upon the model of TBGAN.

TABLE 11. Training and testing time of models upon three datasets.

Algorithms	Time(s)	Pavia University	Salinas	Indian Pines
RBF-SVM	Training	17.43	35.21	2.49
	Testing	10.04	23.90	2.14
RF	Training	51.20	55.74	15.24
	Testing	3.81	0.72	0.32
LSTM	Training	88.98	110.44	23.45
	Testing	7.97	5.55	0.80
M3D-DCNN	Training	1942.03	2970.75	206.56
	Testing	10.96	9.22	1.99
SSUN	Training	211.38	266.04	54.52
	Testing	20.59	12.15	2.17
DCGAN	Training	2828.46	3322.45	764.82
	Testing	13.69	15.88	2.78
TB-SPE	Training	2859.80	3964.1	683.82
	Testing	18.71	13.27	1.36
TB-SPA	Training	2564.67	3293.59	673.22
	Testing	28.13	15.83	2.37
TBGAN	Training	5310.14	7865.55	1262.02
	Testing	13.26	16.59	2.81

A. IMPACTS OF THE PATCH SIZES

Obviously, the performance of TBGAN is susceptible to the patch size. The larger patches may contain redundant information resulting in lower classification accuracy and heavier computation. In contrast, the smaller patches may provide insufficient spatial features for training the model, leading to false discriminants. In the experiments, four spatial neighborhoods of 31 × 31, 39 × 39, 47 × 47 and 55 × 55 are adopted, and the classification results on three datasets are presented in Table 12. For the Salinas and Indian Pines datasets, TBGAN achieves the best classification results both in the patch size of 47 × 47, while for the Pavia University dataset, the best OA of TBGAN corresponds to the patch size of 55 × 55. Thus, the patch size in the formal experiment is set to 47 × 47 by a majority of all datasets.

TABLE 12. Overall accuracy upon three datasets with different patch size.

Patch size	31×31	39×39	47×47	55×55
Pavia University	99.50	99.84	99.88	**99.90**
Salinas	99.90	99.97	**99.99**	99.98
Indian Pines	97.83	98.36	**98.84**	98.75

B. OPTIMAL CHOICE OF HYPER-PARAMETER λ IN L_G

The hyper-parameter of λ in (2) is a weight factor to trade off L_c and L_s. To explore the influence of λ upon the classification results, the value of λ is selected from the range of [0, 0.5] at 0.1 intervals. As shown in Table 13, for the first two datasets, the overall accuracies of TBGAN performance are less affected by the parameter λ. However, for the Indian Pines dataset, the performance of TBGAN varies significantly with different values of λ and the best overall accuracy is acquired when λ equals 0.3. In view of this, the

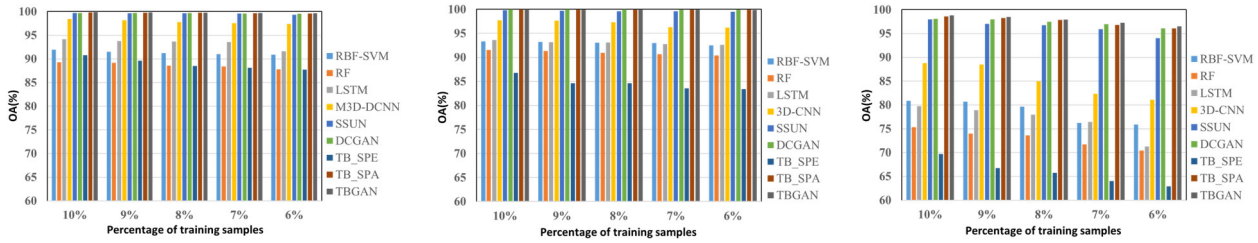


FIGURE 9. OA of RBF-SVM, RF, LSTM, M3DCNN, SSUN, TB-SPE, TB-SPA, and TBGAN with different ratios of training samples on (a) Pavia University dataset, (b) Salinas dataset, and (c) Indian Pines dataset.

TABLE 13. Overall accuracy upon three datasets with different values of λ.

λ	0	0.1	0.2	0.3	0.4	0.5
Pavia University	99.88	**99.89**	99.86	99.88	99.87	99.85
Salinas	99.97	99.97	99.98	**99.99**	99.98	99.98
Indian Pines	98.01	98.41	98.56	**98.84**	98.62	98.77

TABLE 14. Overall accuracy upon three datasets with/without self-attention mechanism.

Self-attention	with	without
Pavia University	**99.88**	99.87
Salinas	**99.99**	99.95
Indian Pines	**98.84**	98.74

value of hyper-parameter λ is set to 0.3 accordingly in the experiments.

C. ADVANTAGES OF SELF-ATTENTION MECHANISM

To capture the long-term dependencies in the spectral sequences, G_{spec} is integrated with the self-attention mechanism, whose effectiveness is validated by training the TBGAN model with or without the self-attention, respectively. As depicted in Table 14, the classification performances of TBGAN are significantly improved upon the three datasets with the addition of the self-attention mechanism.

D. SENSITIVITY TO THE NUMBER OF TRAINING SAMPLES

To investigate the sensitivity of different classification methods to the number of training samples, 10%, 9%, 8%, 7%, and 6% of the labeled samples are successively picked out from the three datasets in the experiments. As shown in Fig. 9, with the reduction of training samples, the classification accuracy of all nine methods declines to varying degrees. As well known to all, deep learning methods require extensive training samples to optimize the parameters, and insufficient samples tend to result in overfitting of the model, thus reducing the classification accuracy. Whereas, the four GAN-based models, by virtue of generating real-like samples, can alleviate the overfitting problem caused by the reduction of training samples. Specifically, as the ratio of training samples from three datasets decreases from 10% to 6%, the OA of TBGAN declines by 0.2%, 0.09%, and 2.3%, respectively, which are significantly slower than the other methods.

V. CONCLUSION

This article proposes a novel TBGAN model for HSI classification. Specifically, there are two generators devised in TBGAN to produce the spectral and spatial real-like data, respectively, which alleviates the small sample size problems. Furthermore, the spectral generator is integrated with the self-attention mechanism, ameliorating the manipulation ability of the long-term dependency relationship. For the multi-classification task, an elaborate discriminator with two branches is designed in TBGAN to extract the spectral and spatial features more thoroughly. It is particularly worth mentioning that the multiscale connections are placed between the discriminator and two generators in TBGAN to improve the network stability and the classification capability. Meanwhile, a feature-matching term is added to the loss function to make the training process more stable. The experimental results demonstrate that TBGAN behaves the superior classification performance and shows lower sensitivity to the number of training samples, which exerts great potential for classification under the circumstance of small size samples. In future research, more innovative strategies are highly expected to be developed in GAN-based supervised frameworks for further improving the performance of HSI classification.

ACKNOWLEDGMENT

The authors gratefully appreciate the editor and anonymous reviewers for their efforts and constructive comments, which have greatly improved the technical quality and presentation of this study.

REFERENCES

[1] L. Liu, N. C. Coops, N. W. Aven, and Y. Pang, "Mapping urban tree species using integrated airborne hyperspectral and LiDAR remote sensing data," *Remote Sens. Environ.*, vol. 200, pp. 170–182, Oct. 2017.

[2] C. M. Gevaert, J. Suomalainen, J. Tang, and L. Kooistra, "Generation of spectral–temporal response surfaces by combining multispectral satellite and hyperspectral UAV imagery for precision agriculture applications," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 3140–3146, Jun. 2015.

[3] C. Tao, Y. J. Wang, B. Zou, Y. L. Tu, and X. L. Jiang, "Assessment and analysis of migrations of heavy metal lead and zinc in soil with hyperspectral inversion model," *Spectrosc. Spectr. Anal.*, vol. 38, no. 6, pp. 1850–1855, Jun. 2018.

[4] I. Makki, R. Younes, C. Francis, T. Bianchi, and M. Zucchetti, "A survey of landmine detection using hyperspectral imaging," *ISPRS J. Photogramm. Remote Sens.*, vol. 124, pp. 40–53, Feb. 2017.

[5] Y. Guo, H. Cao, S. Han, Y. Sun, and Y. Bai, "Spectral–spatial hyperspectralimage classification with K-nearest neighbor and guided filter," *IEEE Access*, vol. 6, pp. 18582–18591, 2018.
[6] X. Jia, "Block-based maximum likelihood classification for hyperspectral remote sensing data," in *Proc. Int. Geosci. Remote Sens. Symp.*, vol. 2, Aug. 1997, pp. 778–780.
[7] F. Ratle, G. Camps-Valls, and J. Weston, "Semisupervised neural networks for efficient hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 5, pp. 2271–2282, May 2010.
[8] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral–spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 809–823, Mar. 2012.
[9] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 492–501, Mar. 2005.
[10] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.
[11] C. G. Dai, X. B. Huang, and G. J. Dong, "Support vector machine for classification of hyperspectral remote sensing imagery," in *Proc. Int. Conf. Fuzzy Syst. Knowl. Discov. (FSKD)*, Aug. 2007, pp. 77–80.
[12] S. Zhong, C.-I. Chang, and Y. Zhang, "Iterative support vector machine for hyperspectral image classification," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 3309–3312.
[13] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019.
[14] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, pp. 1–12, Jan. 2015.
[15] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 844–853, Feb. 2017.
[16] Y. Xu, L. Zhang, B. Du, and F. Zhang, "Spectral–spatial unified networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 5893–5909, Oct. 2018.
[17] J. Feng, J. Chen, L. Liu, X. Cao, X. Zhang, L. Jiao, and T. Yu, "CNN-based multilayer spatial–spectral feature fusion and sample augmentation with local and nonlocal constraints for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 4, pp. 1299–1313, Apr. 2019.
[18] B. Wang, Q. Shao, D. Song, Z. Li, Y. Tang, C. Yang, and M. Wang, "A spectral–spatial features integrated network for hyperspectral detection of marine oil spill," *Remote Sens.*, vol. 13, no. 8, p. 1568, Apr. 2021.
[19] M. Seydgar, A. A. Naeini, M. Zhang, W. Li, and M. Satari, "3-D convolution-recurrent networks for spectral–spatial classification of hyperspectral images," *Remote Sens.*, vol. 11, no. 7, p. 883, Apr. 2019.
[20] M. He, B. Li, and H. Chen, "Multi-scale 3D deep convolutional neural network for hyperspectral image classification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3904–3908.
[21] M. Rao, P. Tang, and Z. Zhang, "A developed Siamese CNN with 3D adaptive spatial–spectral pyramid pooling for hyperspectral image classification," *Remote Sens.*, vol. 12, no. 12, pp. 1–28, Jun. 2020.
[22] S. Hou, H. Shi, X. Cao, X. Zhang, and L. Jiao, "Hyperspectral imagery classification based on contrastive learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–13, 2022.
[23] B. Liu, A. Yu, X. Yu, R. Wang, K. Gao, and W. Guo, "Deep multiview learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7758–7772, Sep. 2021.
[24] M. Zhu, J. Fan, Q. Yang, and T. Chen, "SC-EADNet: A self-supervised contrastive efficient asymmetric dilated network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–17, 2022.
[25] C. Kwan, B. Ayhan, B. Budavari, Y. Lu, D. Perez, J. Li, S. Bernabe, and A. Plaza, "Deep learning for land cover classification using only a few bands," *Remote Sens.*, vol. 12, no. 12, p. 2000, Jun. 2020.
[26] F. Zhou, R. Hang, Q. Liu, and X. Yuan, "Hyperspectral image classification using spectral–spatial LSTMs," *Neurocomputing*, vol. 328, no. 7, pp. 39–47, 2019.
[27] Y. Chen, X. Zhao, and X. Jia, "Spectral–spatial classification of hyperspectral data based on deep belief network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2381–2392, Jun. 2015.
[28] W. Lv and X. Wang, "Overview of hyperspectral image classification," *J. Sensors*, vol. 2020, pp. 1–13, Jul. 2020.
[29] X. Li, Z. Li, H. Qiu, G. Hou, and P. Fan, "An overview of hyperspectral image feature extraction, classification methods and the methods based on small samples," *Appl. Spectrosc. Rev.*, 2021, doi: 10.1080/05704928.2021.1999252.
[30] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 2672–2680.
[31] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.
[32] J. Li, J. Jia, and D. Xu, "Unsupervised representation learning of image-based plant disease with deep convolutional generative adversarial networks," in *Proc. 37th Chin. Control Conf. (CCC)*, Jul. 2018, pp. 1–16.
[33] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*.
[34] Y. Zhan, D. Hu, Y. Wang, and X. Yu, "Semisupervised hyperspectral image classification based on generative adversarial networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 212–216, Feb. 2018.
[35] L. Zhu, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Generative adversarial networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5046–5063, Sep. 2018.
[36] Y. Zhan, Y. Medjadba, G. Wang, X. Yu, J. Qin, T. Huang, K. Wu, D. Hu, Z. Zhao, Y. Wang, Y. Cao, and R. Jiao, "Hyperspectral image classification based on generative adversarial networks with feature fusing and dynamic neighborhood voting mechanism," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2019, pp. 811–814.
[37] J. Feng, H. Yu, L. Wang, X. Cao, X. Zhang, and L. Jiao, "Classification of hyperspectral images based on multiclass spatial–spectral generative adversarial networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5329–5343, Mar. 2019.
[38] R. Hang, F. Zhou, Q. Liu, and P. Ghamisi, "Classification of hyperspectral images via multitask generative adversarial networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 2, pp. 1424–1436, Feb. 2021.
[39] C. Tao, H. Wang, J. Qi, and H. Li, "Semisupervised variational generative adversarial networks for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 914–927, 2020.
[40] L. Zhang, Q. Nie, H. Ji, Y. Wang, Y. Wei, and D. An, "Hyperspectral imaging combined with generative adversarial network (GAN)-based data augmentation to identify haploid maize kernels," *J. Food Composition Anal.*, vol. 106, Mar. 2022, Art. no. 104346.
[41] T. Dam, S. G. Anavatti, and H. A. Abbass, "Mixture of spectral generative adversarial networks for imbalanced hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022, Art. no. 5502005, doi: 10.1109/LGRS.2020.3041864.
[42] J. Wang, S. Guo, R. Huang, L. Li, X. Zhang, and L. Jiao, "Dual-channel capsule generation adversarial network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2022.
[43] J. Wang, F. Gao, J. Dong, and Q. Du, "Adaptive DropBlock-enhanced generative adversarial networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 5040–5053, Jun. 2021.
[44] Z. Zhong, J. Li, D. A. Clausi, and A. Wong, "Generative adversarial networks and conditional random fields for hyperspectral image classification," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3318–3329, Jul. 2020.
[45] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, vol. 2014, pp. 1–14.
[46] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–17.
[47] A. Karnewar and O. Wang, "MSG-GAN: Multi-scale gradients for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7796–7805.
[48] W. Zhao, X. Chen, J. Chen, and Y. Qu, "Sample generation with self-attention generative adversarial adaptation network (SaGAAN) for hyperspectral image classification," *Remote Sens.*, vol. 12, no. 5, pp. 1–13, Mar. 2020.
[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5999–6009.
[50] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 12744–12753.

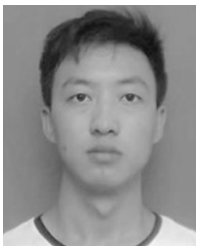
[51] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 4043–4055.
[52] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 2234–2242.
[53] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Feb. 2011.

DONGMEI SONG received the B.S. and M.Sc. degrees in ecology from Shenyang Agricultural University, Shenyang, China, in 1997 and 2000, respectively, and the Ph.D. degree in landscape ecology from the Institute of Applied Ecology, Chinese Academy of Sciences, Shenyang, in 2003. She then studied as a Postdoctoral Researcher at the Institute of Geographic Sciences and Natural Resources Research, CAS, China, in 2005. She is currently a Professor with the College of Oceanography and Space Informatics, China University of Petroleum (East China), Qingdao, China. Her major research interest includes remote sensing image processing.

YUNHE TANG received the B.S. degree from the China University of Petroleum (East China), Qingdao, China, in 2020, where she is currently pursuing the master's degree with the College of Oceanography and Space Informatics. Her research interests include hyperspectral image classification and oil spill detection.

BIN WANG received the B.S. degree in information and computational science from the China University of Mining and Technology, Xuzhou, China, in 2008, the M.Sc. degree in computational mathematics from the Dalian University of Technology, Dalian, China, in 2011, and the Ph.D. degree in geographic information science from The Hong Kong Polytechnic University, Hong Kong. He is currently with the College of Oceanography and Space Informatics, China University of Petroleum (East China), Qingdao, China. His major research interests include spatial analyses and mathematical modeling, and intelligence algorithms for remote sensing image processing.

JIE ZHANG received the B.S. and M.S. degrees in mathematics from Inner Mongolia University, Urumqi, China, in 1984 and 1987, respectively, and the Ph.D. degree in applied mathematics from Tsinghua University, Beijing, China, in 1993. He is the Director of the Laboratory of Marine Physics and Remote Sensing, Ministry of Natural Resources, First Institute of Oceanography, Qingdao, China. He has a broad interest in marine physics and remote sensing applications. His research mainly focuses on the following: the SAR retrieval of ocean dynamics processes and the SAR detection of marine targets, ocean hyperspectral remote sensing, high-frequency surface-wave radar ocean detection techniques, and the integration of marine remote sensing application systems. He has served as a member of multiple domestic/international committees and a principal investigator/coinvestigator of many projects from the National Science Foundation of China, the State High-Tech Development Plan (863), and other funding agencies. He has been the supervisor of nearly 40 Ph.D. degree students and has published more than 200 articles.



CHANGLONG YANG received the B.S. degree from the China University of Petroleum (East China), Qingdao, China, in 2020, where he is currently pursuing the master's degree with the College of Oceanography and Space Informatics. His research interest includes hyperspectral images classification with machine learning, especially for deep learning methods.