



HSIGAN: A Conditional Hyperspectral Image Synthesis Method With Auxiliary Classifier

Wei Liu , Jie You, and Joonwhoan Lee 

Abstract—In this article, we explore a conditional hyperspectral image (HSI) synthesis method with generative adversarial networks (GAN). A new multistage and multipole generative adversarial network, which is suitable for conditional HSI generation and classification (HSIGAN), is proposed. For HSIs synthesis, it is crucial to learn a great deal of spatial–spectral distribution features from source data. The multistage progressive training makes the generator effectively imitate the real data by fully exploiting the high-dimension learning capability of GAN models. The coarse-to-fine information extraction method helps the discriminator to understand the semantic feature better while the multiscale classification prediction presents a positive impact on results. A spectral classifier joins the adversarial network, which offers a helping hand to stabilize and optimize the model. Moreover, we apply the 3-D DropBlock layer in the generator to remove semantic information in a contiguous spatial–spectral region and avoid model collapse. Experimental results of the quantitative and qualitative evaluation show that HSIGAN could generate high-fidelity, diverse hyperspectral cubes while achieving top-ranking accuracy for supervised classification. This result is encouraging for using GANs as a data augmentation strategy in the HSI vision task.

Index Terms—Classification, generative adversarial network (GAN), hyperspectral image (HSI), synthesis.

I. INTRODUCTION

THE hyperspectral image (HSI) is a 3-D data cube, where each pixel of the image contains tens to hundreds of narrow-band spectral information and produces a complete and continuous spectral curve. Different substances show different radiation intensity in HSI and present different spectral response curves. The spectral information of each pixel is added to the 2-D spatial image to generate a 3-D data cube. With the improvement of the spectral resolution of HSIs, the ability of detecting attribute information of ground objects is enhanced compared with panchromatic and multispectral imaging.

In recent years, HSIs have been rapidly developed and widely used in various fields. HSI technology not only plays an increasingly important role in the remote sensing field, but also

causes great interest in other areas, such as agriculture, medical diagnosis, atmosphere and environment, automated detection, and so on.

In the past 30 years, hyperspectral remote sensing technology has developed rapidly, and new breakthroughs have been made in HSI processing and information extraction technology. The research of HSI mainly includes denoising and recovery, dimensionality reduction, resolution enhancement, spectral decomposition, classification, change detection, fast calculation, and so on [1].

The development of high-performance computing technology has significantly improved the efficiency of data processing and analysis, and has been widely applied in the HSI information extraction. However, the rich spectral band, high dimension, and less supervised information not only bring difficulties to data storage and transmission, but also bring challenges to HSI processing and analysis.

The advantages introduced with deep learning solutions lie in the automatic and hierarchical learning process from data itself (or spatial–spectral portions of it) which enables to build a model with increasingly higher semantic layers until a representation suitable to the task at hand (e.g., classification, regression, segmentation, detection, etc.) is reached [2].

Deep learning has been very successful for HSI analysis tasks while being robust against the noise and uncertainties in spectral and ground truth measurements. Accurate classification of each pixel from HSI is a crucial step for land-cover evaluation and segmentation [3]. Nowadays, the automatic classification methods which achieve state-of-the-art results are often those using the deep learning approach [4]. However, accurately delineating the pixelwise land-cover region of HSIs is time consuming, expensive, and complicated. This problem leads to a lack of annotated training datasets [5], which often hinders research in the HSI domain.

Although deep discriminative models have provided significant impetus for the progress of deep learning, over the past few years, many of the notable advances in the field of image processing have come from new applications of deep generative modeling. The small sample size problem of HSIs can be alleviated by utilizing the synthesis sample of generative models.

Recently, researchers have made impressive progress on synthesis over natural images. There has been an increasing interest in synthesizing 3-D shapes with deep generative models, such as voxels [6], point clouds [7], [8], and octave trees [9]. Due to the high spectral resolution, the computational complexity of HSI is much higher than the 3-D voxelized data that are sparse and binary (e.g., 3-D point cloud). It is a challenging problem to learn the complex spatial–spectral distribution and generate realistic synthesis of hyperspectral samples. Therefore, we are committed to researching HSI generative models.

Manuscript received December 3, 2020; revised January 18, 2021; accepted February 17, 2021. Date of publication March 4, 2021; date of current version March 31, 2021. This work was supported in part by the Natural Science Foundation of Hebei Province under Grant F2020403030. (Corresponding author: Joonwhoan Lee.)

Wei Liu is with the Intelligent Sensor Network Engineering Research Center of Hebei Province, Hebei GEO University, Shijiazhuang 050031, China, and also with the Artificial Intelligence Laboratory, Jeonbuk National University, Jeonju 54896, South Korea (e-mail: marinlwhb@gmail.com).

Jie You and Joonwhoan Lee are with the Artificial Intelligence Laboratory, Jeonbuk National University, Jeonju 54896, South Korea (e-mail: youjie80@gmail.com; chlee@chonbuk.ac.kr).

Digital Object Identifier 10.1109/JSTARS.2021.3063911

In many deep learning generation models, the generative adversarial networks (GAN) are an outstanding representative. The GAN framework was first introduced by Goodfellow *et al.* [10]. Recently in the HSI research area, some GAN-based models have been proposed. In [11] and [12], the semisupervised 1-D learning methods were introduced, in which the unlabeled samples are used to train the network. These two models only focus on the spectral features and do not make full use of spatial information. Moreover, a 3-D GAN-CRF framework [13] was proposed by integrating a semisupervised deep learning and a probabilistic graphical model. However, as presented in [14], an excellent semisupervised GAN requires a bad generator because this generator produces data outside the real data distribution. Therefore, training a semisupervised GAN harms its image generation capability.

In addition, there are other GAN-based HSI classification models focusing on spatial–spectral features. Zhu *et al.* [15] proposed a 3-D-GAN method to stabilize the GAN training procedure by retaining only three principal components in HSIs. MSGAN [16] used a 1-D and 2-D convolutional structure to extract spatial and spectral features. Then, these extracted features were concatenated at the last fully connected layer of the discriminator to realize the spatial–spectral classification of HSIs. Later, a symmetric convolutional GAN based on collaborative learning and attention mechanism (CA-GAN) was proposed for HSI classification [17]. The aforementioned three methods use PCA to reduce the dimensionality and retain a few principal components of HSIs. They ignore the inherent dimension reduction capability of deep learning models, and the substantial correlation information between adjacent bands would be lost. The generated samples of these GAN-based HSI classification models are just with a few bands, so the generators cannot be directly used to generate the high fidelity 3-D HSI cubes. Moreover, the purpose of these models is to improve the classification accuracy rather than to solve the problem of HSI synthesis, so they did not evaluate the fidelity and diversity of the generated samples. For all we know, there has been very limited published research in trying to generate high-fidelity and diverse HSIs with high spectral resolution. There is not any existing effective generation model for HSI, by far.

It is difficult to learn the spatial–spectral features of HSIs due to the small number of labeled samples and the high feature dimensions, which will make synthesis models face an enormous challenge. In addition, compared with the natural images, the ground objects in the hyperspectral remote sensing image have no obvious and specific features. Although the combination of spatial and spectral features is helpful for the improvement of classification performance [18], [19], the high-resolution spectral information still plays a leading role. So, ensuring the high fidelity of synthetic spectral is an issue of crucial importance while learning real spatial distribution. In this article, we put forth a novel automatic data augmentation framework and the generative method that produces 3-D synthetic samples.

In more detail, the main contributions of this article are summarized as follows.

- 1) Our HSI-GAN model could generate high-fidelity and diverse HSI 3-D cube with all-band spatial–spectral information instead of 2-D patch with a few bands.
- 2) The proposed model copes with generation and classification tasks jointly and explores the mutual influence between them to achieve mutual benefits. The generator (G) can be treated as an automatic data augmentation

network to improve the insufficient labeled samples of the classifier (C). With the guidance of C, G produces more visually appealing results.

- 3) We apply the 3-D DropBlock (DB) layer in the generator to increase the diversity and evaluate the validity of the DB layer.
- 4) The coarse-to-fine and multiscale training strategy proposed not only opens the door to the spatial–spectral feature synthesis of HSI, but also could be used to generate other complicated high-dimensional images.

The remaining article is organized as follows. The related work is introduced in Section II, the details of our proposed method are given in Section III, and the performed experiments and resulting discussions are presented in Section IV. Finally, several concluding remarks are given in Section V.

II. RELATED WORK

A. Class Conditional GAN

In theory, the GAN allows the random distribution to approximate the real data completely. It has been the most prominent contemporary approach of image synthesis because the GAN model sidesteps the difficulty of many intractable probabilistic approximating computations. However, the downside of the original GAN is too much freedom and it is not controllable. The natural idea is to put some constraints on GAN, and then, there is the conditional GAN. Conditional generative models enjoy remarkable progress over the past few years. CGAN [20] directs the data generation process by conditioning the model on additional information, such as simply feeding class labels into both the generator and discriminator. In the AC-GAN [21], an auxiliary classifier GAN, every generated sample has a corresponding class label, and the discriminator gives both a probability distribution over sources and a probability distribution over the class labels. The unsupervised InfoGAN [22] decomposes an input vector into a standard noise latent vector and another conditional latent variable to capture salient semantic features of real samples. The automatically inferred additional information in InfoGAN has much more freedom to obtain certain features of real data than using the class label in CGAN which is restricted to known information [23]. In semisupervised Triple-GAN [24], the generator and the classifier characterize the conditional distributions between images and labels, and the discriminator solely focuses on identifying fake image–label pairs.

In our task, the aim of the generator is to generate HSI samples according to a specified class label, while the discriminator needs to not only determine whether each sample is true or false but also perform the classification task.

B. Supervised and Semisupervised GAN

Recently, the various supervised GAN models are demonstrated to be able to generate visually realistic images. The semisupervised GAN methods proposed in [25] and [26] make full use of both labeled and unlabeled data to obtain reliable empirical results. The collection of HSI ground-truth labels is expensive and difficult. In order to make full use of a lot of unlabeled samples in HSI data, some researchers try to use the semisupervised GAN to create HSI classifier. However, the following questions remain open. First, the land cover of HSIs is

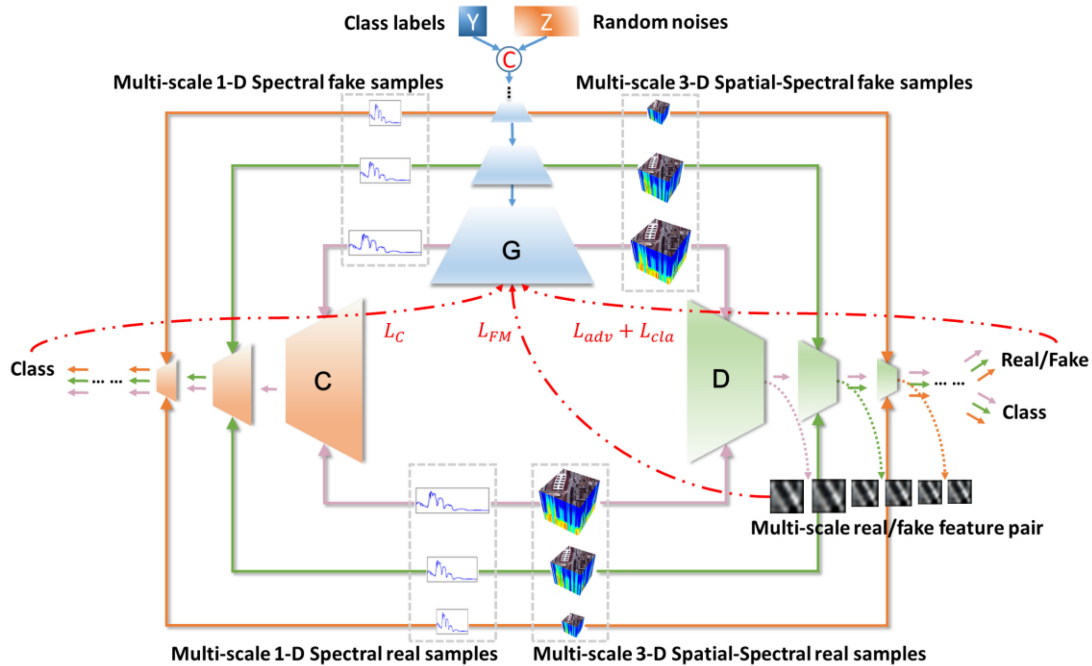


Fig. 1. Overall schematic of the HSI-GAN model. The solid lines denote the forward pass and each color line represents a different scale. The red dashed lines denote the flow of gradients. The losses include the spectral classification loss L_C , the feature matching loss L_{FM} , the adversarial loss L_{adv} , and the multiple classification loss L_{cla} .

complex and diverse. The unlabeled samples may not be consistent with labeled data in the types of land cover. This category inconsistency will lead to different data distributions between unlabeled and labeled samples, and could affect the generation of synthetic samples. Second, it seems that good semisupervised learning and a sound generator cannot be obtained at the same time. A good semisupervised GAN requires a bad generator because the generator produces data outside the real-data distribution, as presented in [14]. The same phenomenon was also observed in [27], where the model generated better images but failed to improve the performance of semisupervised learning. For these reasons, we introduce a supervised GAN model with an eye on the generation of high fidelity 3-D HSIs. The experimental results demonstrated that training a small number of labeled samples could achieve the top-ranking quality of reproduction.

C. Three-Dimensional Generative Adversarial Network

Recently, with the advances in deep representation learning, there is more and more relevant literature on studying 3-D object generation. 3-D image understanding and generation in the graphics and vision community has been paid more and more attention [28]. In [6], the authors propose a novel framework which generates 3-D objects from a probabilistic space by leveraging recent advances in volumetric convolutional networks and generative adversarial nets.

Although the HSI is the usage of 3-D field representation, it is different from 3-D shape data that are sparse and binary [29] because the HSIs include tens or hundreds of consecutive spectral bands, and the computational complexity is much higher than the voxelized shapes.

So, it is a challenging problem that the HSI cube is synthesized based on learned feature representations. Compared with RGB images and 3-D voxelized objects [30], it is more difficult to

model the abundant spatial feature and high-resolution spectral information of HSI cube. The research on depth feature representation of HSI generation has just started, and there are still many problems to be explored.

III. PROPOSED METHOD

A. Framework of the Proposed Method

Our goal is to explore a learning data augmentation method via adversarial networks. The proposed GAN model enables the machine learning-based approach to increase the available annotated samples more efficiently, and the generated generic augmented data could be utilized to improve the HSI classification performance. To address both problems, we present a novel multistage and multipole supervised conditional HSI GAN.

Compared with the traditional bipolar adversarial network, we add a spectral classifier to the opposing camps as the third pole, which offers a helping hand to stabilize and optimize the model. The flowchart of the HSI-GAN model is shown in Fig. 1. HSI-GAN consists of three parts—a multistage 3-D spatial-spectral feature generator G , a multistage spatial-spectral feature discriminator-classifier D , and a multistage 1-D spectral feature classifier C .

The HSI undergoes the detailed division on the spectral dimension, and each pixel has a continuous spectrum of B bands. Let a set of pixels with the high-resolution spectral information be denoted by $X^{\text{spe}} = \{x_i^{\text{spe}}\}_{i=1}^N$, where $x_i^{\text{spe}} \in \mathbb{R}^B$ is a 1-D spectral vector and N is the number of pixels. The corresponding spatial neighborhood pixels that are centered at x_i^{spe} are sampled from HSI, and these pixels compose a 3-D cube which is denoted by $X = \{x_i\}_{i=1}^N$, where $x_i \in \mathbb{R}^{H \times W \times B}$. H and W is the spatial size of the 3-D cube. $Y = \{y_i\}_{i=1}^N$ represents the

corresponding one-hot coded labels of the center pixel of each cube, where $y_i \in \mathbb{R}^C$ and C is the number of land cover classes.

As common conditional generative models, G uses both random noises $z \in Z$ and class labels $y \in Y$ to generate images with stabilized training, and then every generated sample has its corresponding class label. After the training, the spatial-spectral 3-D cube with label $X_{\text{fake}} = \{G(z_i, y_i)\}_{i=1}^N$ is generated. Then, the synthesized samples are fed to the other two poles D and C . The model training enters the feature extraction and classification phase. One pole is the discriminator D that provides a probability distribution over sources and the class labels. Another pole is the spectral classifier C that outputs the class label of the 1-D spectral samples.

The proposed model performs 3-D convolution to extract features along both spatial and spectral dimensions. The 3-D convolution is enforced by convolving with 3-D kernels and then outputs a volume formed by stacking multiple contiguous spectral bands together. 3-D convolution is well-suited for HSI and can preserve the spectral information as much as possible.

B. Multistage 3-D Spatial-Spectral Feature Generator

Due to hundreds of spectral bands in HSIs, in the training phase for synthesis HSIs, each occupied pixel in the HSI volume fetches a great deal of spatial-spectral distribution feature from real source data. In other words, the generator needs to effectively imitate the real data by fully exploiting the high-dimensional learning capability of GAN models.

Our goal is to find a solution that breaks through the limitation of high spectral dimensions and synthesizes spatial-spectral features with high fidelity. Inspired by [31] and [32], we proposed a multistage generator to generate high spectral resolution HSI cube. For soothing the training difficulty of high-dimensional data, our generator is divided into several coarse-to-fine stages, and the generator outputs different scale cubes at each stage, which is denoted by $\{x_{\text{fake}_i}\}_{i=1}^S$ and S is the number of stages. The coarse output of the primary stage tries to sketch the primitive feature at a low spectral resolution, and the fine output of the superior stage aims to improve the spectral details of the coarse cube from the previous stages. The objective of multistage generators is to learn the distribution of real samples progressively. In [31], the network adds new layers that model increasingly fine details as training progresses, starting from a low resolution. In [32], the model includes two-scale subgenerators and trains them in two steps. The main difference from the above models is that we use an integrated model. The samples of multiscales are trained synchronously.

In Fig. 1, G receives both noise z and label y as input, and the various scale fake samples x_{fake} are generated at various stages of G , $x_{\text{fake}} = G(z, y)$. Then, x_{fake} is fed into the corresponding stage of D and C . The generator is constructed by a series of convolutional-transpose layers, batch normalization layers, and ReLU activations [33]. The input is a class label y and a latent vector z that is drawn from a standard normal distribution. The output is a 3-D cube with the spatial-spectral feature. The strided transposed convolution layers allow the latent vector to be transformed into a volume close to the distribution of training samples by using 3-D filters. This generator could aggregate multistage information for the HSIs synthesis task. For RGB images, such a multiresolution pipeline is a well-established practice, such as recent unconditional GANs [34], [35] and conditional image generation [36], [37]. Now, we bring similar ideas from the high-resolution spatial domain to the high-resolution spectral

domain, and use the multiresolution pipeline to generate the 3-D HSIs.

C. Multistage 3-D Spatial-Spectral Feature Discriminator and Classifier

The synthesis of high-dimensional and high-spectral resolution 3-D cube poses a significant challenge to the GAN discriminator design. Recent work has shown that GAN can produce convincing image samples with high spatial resolution [36], [38]. As improving of the spatial resolution, differentiating real and synthesized images is getting harder and harder for the discriminator. To enhance discriminator ability, [39] introduced multiple discriminators at the same image scale. In [40], the global and local context discriminators are trained to distinguish real images from completed ones. And then, [32] extend the design to multiple discriminators with identical architecture at different image scales. In view of multiple discriminators having the same structure in [32], we naturally employ an integrated discriminator. This discriminator could take concerted action to the multistage structure of the generator. The generated and real samples, $\{x_{\text{fake}_i}\}_{i=1}^S$ and $\{x_{\text{real}_i}\}_{i=1}^S$, flow into the corresponding stages of the discriminator. As shown in Fig. 1, G and D achieve docking at various stages. Although there is only one discriminator, D can extract multiscale features from coarse to fine scale and guide the generator to produce finer details.

If the generator with a class label creates unreasonable wrong images, the classifier will not be able to tell which class the generator was attempting to sample from. This strategy encourages the generator to stay away from awful, hard-to-classify images and attracts it toward easy-to-classify images [41]. The auxiliary classifier stabilizes training and makes G produce excellent results. In our model, D could perform double duty as the discriminator and classifier.

The discriminator is made up of strided convolution layers with 3-D kernels, batch normalization layers, and LeakyReLU activations. The inputs are multiscale 3-D cubes and contain real and fake training samples. There are two outputs of D —one is a scalar probability that indicates the real or fake data distribution; another is the multiclass probability.

D. Multistage 1-D Spectral Feature Classifier

The learning ability of G depends entirely on discriminators. Inspired by [42] and [43], further, an auxiliary spectral feature classifier C joins the opposing camps of G and D . This spectral classifier could offer a helping hand to improve the performance of the generator. On the other hand, adding extra loss could stabilize training because adversarial training is known to be unstable and sensitive to hyperparameters [44]. The primary task of C is to adapt the spectral feature from the synthesis domain to make them appear as if they are sampled from the real domain.

The classifier structure consists of several strided convolution layers with 1-D kernels, batch normalization layers, LeakyReLU activations layers, and the fully connected layer. The inputs of C are 1-D spectral vectors that cover all bands. The vectors are extracted from the spatial center of the 3-D HSI samples, which is denoted by $\{x_{\text{real}_i}^{\text{spe}}\}_{i=1}^S$ and $\{x_{\text{fake}_i}^{\text{spe}}\}_{i=1}^S$ and S is the number of stages. The output of C is the multiclass probability of spectral information. As shown in Fig. 1, the multistage classifier C acts in cooperation with the multistage structure of the generator. By applying auxiliary C , we reinforce the learning effort of G on the spectral features.

E. Losses for Optimization

Initially, we tend to understand the GAN model from a game theory perspective. However, WGAN [45] guides us forward to reflect the GAN problem from the view of data distribution. The goal of the generator is to eventually output different data samples that conform to the real data distribution. The comparisons based on explicit expressions are not possible due to the resulting distribution being too complex. So, the task of the discriminator is to compare two probability distributions based on samples. That is, the discriminator could be tread as a learnable loss function of the generator, which can effectively close the distance of generated distribution and the real distribution.

In this article, the objective function of D has two parts—the adversarial loss L_{adv} and the multiple classification loss L_{cla} . As shown in (1), L_{adv} uses the adversarial objective function proposed by WGAN-GP [46], which introduces a gradient penalty norm to the previous WGAN

$$L_{\text{adv}} = \mathbb{E}_{\hat{x} \sim \mathbb{P}_g} [D(\hat{x})] - \mathbb{E}_{x \sim \mathbb{P}_r} [D(x)] + \lambda \mathbb{E}_{\hat{x} \sim \mathbb{P}_{\hat{x}}} \left[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2 \right] \quad (1)$$

where $\hat{x} = G(z, y)$, \mathbb{P}_g and \mathbb{P}_r are the generated and real distribution, respectively. The third term is a penalty on the gradient norm for random samples $\hat{x} \sim \mathbb{P}_{\hat{x}}$. In the case of classification, we use a typical softmax cross-entropy loss

$$L_{\text{cla}} = E[P(\text{class} = c|x)] + E[P(\text{class} = c\hat{x})]. \quad (2)$$

The loss of multistage discriminator D is represented as L_D

$$L_D = \sum_{i=1}^S (L_{\text{adv}} + L_{\text{cla}}). \quad (3)$$

L_C is the spectral classification loss, and we also use a softmax cross-entropy loss

$$L_C = L_C^{\text{real}} + L_C^{\text{fake}} \quad (4)$$

where we introduce

$$L_C^{\text{real}} = \sum_{i=1}^S E[P(\text{class} = c x^{\text{spe}})] \quad (5)$$

$$L_C^{\text{fake}} = \sum_{i=1}^S E[P(\text{class} = c \hat{x}^{\text{spe}})]. \quad (6)$$

The goal of G is to optimize the following objective:

$$L_G = L_G^{\text{adv}} + L_G^{\text{cla}} + L_G^C. \quad (7)$$

Here, we define

$$L_G^{\text{adv}} = \sum_{i=1}^S -E[D(\hat{x})] \quad (8)$$

$$L_G^{\text{cla}} = \sum_{i=1}^S E[P(\text{class} = c\hat{x})] \quad (9)$$

$$L_G^C = \sum_{i=1}^S E[P(\text{class} = c|\hat{x}^{\text{spe}})]. \quad (10)$$

In addition, a feature matching loss [25] is proposed to address the instability of GANs and prevent the generator from overtraining. This new objective requires the generator to produce data that matches the statistics of the real data. The feature value on an intermediate layer of the discriminator is extracted to optimize G . There is a similar concept in [47] that calculates the matching loss using the feature value from pretrained VGG net, which is helpful for improving performance.

We use the discriminator to specify the statistics that we think are worth matching. Here, the feature map is obtained by the last convolution (after activation) before the fully connected layer within each stage of D . For ease of presentation, we define L_{FM} as the total feature matching loss. Then, we get the feature matching loss of each stage by using the Huber loss function with a parameter $\delta = 1$, and then denote the i th stage Huber loss as $\text{HL}^{(i)}$

$$L_{\text{FM}} = \sum_{j=1}^S \sum_{i=1}^j \text{HL}^{(i)} \quad (11)$$

where $\sum_{i=1}^j \text{HL}^{(i)}$ is the feature matching loss of the j th stage and S is the total number of stages of D . Note that L_{FM} is employed to optimize only the generator rather than discriminator.

The full objective of G combines adversarial loss, classification loss, and feature matching loss as follows:

$$L_G = \lambda_1 L_G^{\text{adv}} + \lambda_2 L_G^{\text{cla}} + \lambda_3 L_G^C + \lambda_4 L_{\text{FM}} \quad (12)$$

where λ_i are weights that control the interaction of all losses. The losses mentioned above could anchor the generated image to the original distribution in some meaningful way. However, to tradeoff the training between discriminator and the generator and achieve better convergence, the hyperparameter tuning needs to take time and a lot of patience, particularly when many loss functions are introduced in GAN.

F. Diversity of Augmented Data

There are prodigious differences between HSIs and RGB images. That is, the intraclass spatial–spectral information is very similar, and the variations in the interclass feature are not significant. This characteristic dramatically increases the risk of model collapse. Thus, a very challenging task for the HSI GAN model is to avoid model collapse and increase the diversity of synthetic samples in the process of mapping a low-dimensional manifold to a high-dimensional space. In the research of GAN, model collapse is a widely concerned issue. There are many related papers to study and solve this problem from different perspectives [25], [48]–[51]. Here, we employ an efficient and low-cost method in which the DB [52] layer is introduced into the GAN model. DB is a form of structured dropout, where units in a contiguous region of a feature map are dropped together. Its effectiveness has been verified in RGB image classification and detection task. So, we have reason to believe that a superior D can be obtained by appending DB. Then, the well-performing D is the best motivator for G learning. In previous research, most of the optimization strategies are only for D . Here, we apply DB not only in the discriminator but also in the generator. 3-D DB is effective in removing semantic information in a contiguous spatial–spectral region. Randomly dropping continuous regions enforces the remaining units to synthesize more diverse features, which can avoid the model collapses. The effect of using DB in

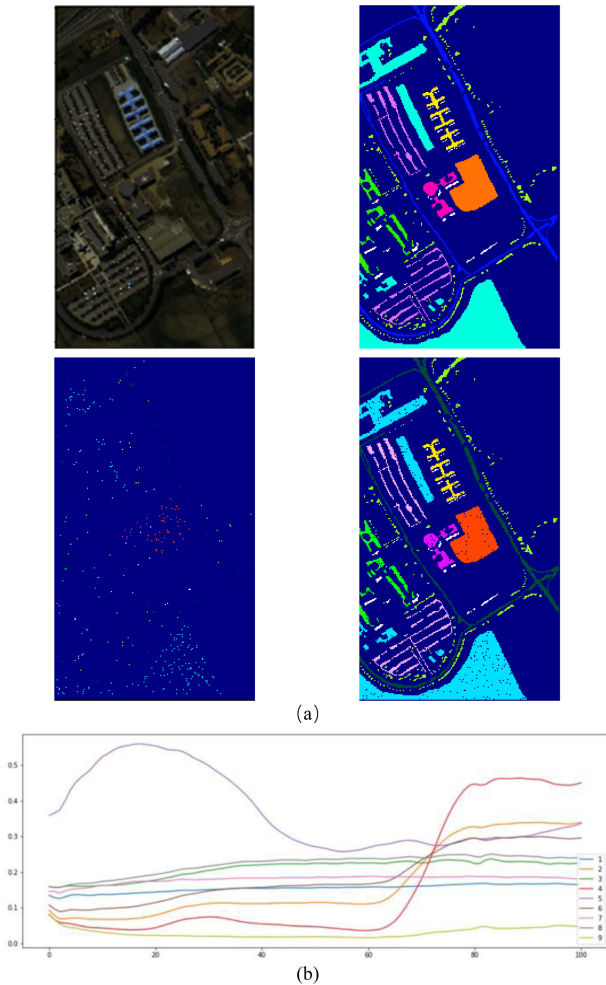


Fig. 2. Pavia University dataset. (a) From left to right: False-color composite, ground truth (gt), training_gt, and test_gt. (b) Mean spectrum curve of per class.

the generator to improve performance is notable, especially if the samples are insufficient and have similar characteristics.

IV. EXPERIMENTS

A. Dataset Description

In the experiment, we evaluate the performance of the proposed technique on the real HSIs dataset. These datasets are well-known standard remote sensing HSIs datasets with various image properties and size, which include the Pavia University (PU), and Indian Pines (IP). The PU scenes (see Fig. 2) acquired by the ROSIS sensor during a flight campaign over Pavia, Northern Italy, is a 610×340 pixel image and the number of spectral bands is 103. The image ground truths differentiate nine land cover types. IP scene (see Fig. 3) was gathered by AVIRIS sensor over the Indian Pines test site in North-Western Indiana and consists of 145×145 pixels and 200 spectral reflectance bands. The ground truth available is designated into 16 classes and is not all mutually exclusive. The number of training and test samples for each class is listed in Tables I and II. In consideration of the computation complexity, down-sampling is done in the spectral domain, and the image is cropped into small cubes. The size of each 3-D cube is $101 \times 17 \times 17$ (band \times height \times width).

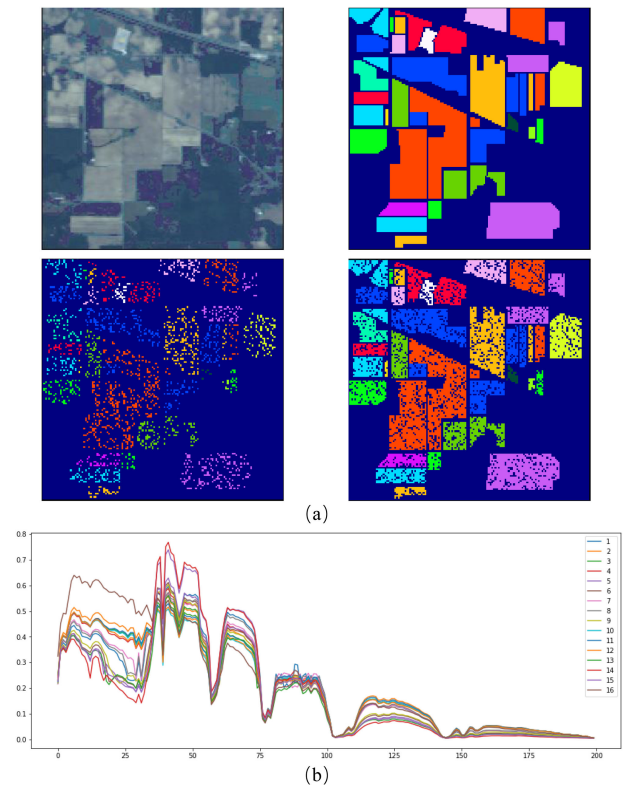


Fig. 3. Indian Pines dataset. (a) From left to right: False-color composite, ground truth (gt), training_gt, and test_gt. (b) Mean spectrum curve of per class.

TABLE I
PU DATASET: THE LAND COVER CLASSES AND THE NUMBERS OF SAMPLES

No.	Name	Training (3%)	Test	Total
1	Asphalt	199	6432	6631
2	Meadows	599	18050	18649
3	Gravel	59	2040	2099
4	Trees	97	2967	3064
5	Metal sheets	40	1305	1345
6	Bare soil	154	4875	5029
7	Bitumen	45	1285	1330
8	Bricks	104	3578	3682
9	Shadow	48	899	947
		1345	41431	42776

TABLE II
IP DATASET: THE LAND COVER CLASSES AND THE NUMBERS OF SAMPLES

No.	Name	Training (20%)	Test	Total
1	Alfalfa	9	37	46
2	Corn-notill	279	1149	1428
3	Corn-mintill	176	654	830
4	Corn	41	196	237
5	Grass-pasture	118	365	483
6	Grass-trees	161	569	730
7	Grass-pasture-mowed	8	20	28
8	Hay-windrowed	89	389	478
9	Oats	5	15	20
10	Soybean-notill	193	779	972
11	Soybean-mintill	493	1962	2455
12	Soybean-clean	114	479	593
13	Wheat	43	162	205
14	Woods	256	1009	1265
15	Build-grass-trees	85	301	386
16	Stone-steel-towers	19	74	93
		2089	8160	10249

TABLE III
ARCHITECTURES OF THE GENERATOR

Layer	Kernel size (3D)	$C_{in} \times C_{out}$	BN	Activation	Stride	Padding	DropBlock	Output size
FC.1		$200 + n_{class} \times 1024$	N	LeakyReLU				
Deconv.2	3×3×3	1024×512	Y	ReLU	2,2,2	0,0,0		
Deconv.3	3×3×3	512×512	Y	ReLU	2,2,2	1,1,1		
Deconv.4	3×3×3	512×512	Y	ReLU	2,2,2	1,1,1		
Deconv.5	3×3×3	512×256	Y	ReLU	2,1,1	1,1,1		
Deconv.6	3×3×3	256×256	Y	ReLU	1,1,1	1,1,1	5×5×5, 0.2	
Deconv.7	3×3×3	256×128	Y	ReLU	2,1,1	1,1,1		
Deconv.8	3×3×3	128×128	Y	ReLU	1,1,1	1,1,1		
Output	1×1×1	128×1	Y	No	1,1,1	0,0,0		33×9×9
Deconv.9	3×3×3	128×128	Y	ReLU	2,2,2	1,3,3		
Deconv.10	3×3×3	128×128	Y	ReLU	1,1,1	1,1,1		
Output	1×1×1	128×1	Y	No	1,1,1	0,0,0		65×13×13
Deconv.11	3×3×3	128×64	Y	ReLU	2,2,2	15,5,5		
Deconv.12	3×3×3	64×64	Y	ReLU	1,1,1	1,1,1		
Output	1×1×1	64×1	Y	No	1,1,1	0,0,0		101×17×17

TABLE IV
ARCHITECTURES OF THE DISCRIMINATOR

Layer	Kernel size (3D)	$C_{in} \times C_{out}$	BN	Activation	Stride	Padding	DropBlock	Input size
Input	1×1×1	1×64	N	LeakyReLU	1,1,1	0,0,0		101×17×17
Conv. 1	3×3×3	64×64	Y	LeakyReLU	2,2,2	15,5,5		
Conv. 2	3×3×3	64×128	Y	LeakyReLU	1,1,1	1,1,1		
Input	1×1×1	1×128	N	LeakyReLU	1,1,1	0,0,0		65×13×13
Conv. 3	3×3×3	128×128	Y	LeakyReLU	2,2,2	1,3,3		
Conv. 4	3×3×3	128×128	Y	LeakyReLU	1,1,1	1,1,1		
Input	1×1×1	1×128	N	LeakyReLU	1,1,1	0,0,0		33×9×9
Conv. 5	3×3×3	128×128	Y	LeakyReLU	2,1,1	1,1,1		
Conv. 6	3×3×3	128×256	Y	LeakyReLU	1,1,1	1,1,1	5×5×5, 0.2	
Conv. 7	3×3×3	256×256	Y	LeakyReLU	2,1,1	1,1,1		
Conv. 8	3×3×3	256×512	Y	LeakyReLU	2,2,2	1,1,1		
Conv. 9	3×3×3	512×512	Y	LeakyReLU	2,2,2	1,1,1		
AvgPool.								3×3×3
FC.10		512×1						1×1×1
FC.10		512× n_{class}						1×1×1

TABLE V
ARCHITECTURES OF THE SPECTRAL CLASSIFIER

Layer	Kernel size (1D)	$C_{in} \times C_{out}$	Stride	BN	Activation	Padding	Input size
Input	1	1×64	1	N	LeakyReLU	0	101
Conv. 1	5	64×64	2	Y	LeakyReLU	16	
Input	1	1×64	1	N	LeakyReLU	0	65
Conv. 2	5	64×64	2	Y	LeakyReLU	2	
Input	1	1×64	1	N	LeakyReLU	0	33
Conv. 3	5	64×128	2	Y	LeakyReLU	2	
Conv. 4	5	128×128	2	Y	LeakyReLU	2	
Conv. 5	5	128×128	2	Y	LeakyReLU	2	
AvgPool.							5
FC.6		128× n_{class}					1

B. Experimental Setting

For the generator, the dimensions of input noise vector z are 200 and the dimensions of label vector y depend on the number of land cover classes. The generator G outputs three-scale HSI cubes. They are from coarse to fine $33 \times 9 \times 9$, $65 \times 13 \times 13$, and $101 \times 17 \times 17$, respectively. The DB, which is inserted between deconvolution and BN layers, has two parameters corresponding to block size and drop probability. The main network structure and parameters are shown in Tables III–V.

Through observing the training process, we find that the learning rate has a great influence on the training stability of the adversarial network. D and C are trained with a learning rate two times greater than G . They are $50e-4$ and $25e-4$, respectively. Because the different learning rates are used, D and C do not need more training per iteration before optimizing G 's weights.

In other words, D , C , and G just need to be trained once in each iteration as long as the appropriate learning rate is set. We fully exploited the different learning rates and the gradient penalty [46] to keep training stable.

The batch size is set as 20 with shuffle enabled. We pass the loss to RMSProp for parameter optimization. For the coefficients of generator loss λ_1 , λ_2 , λ_3 , and λ_4 are set as 1, 0.2, 0.2, and 10, respectively. To avoid classification overfitting of real training samples, we feed a mixup of two real samples into D and C

$$x_{\text{mix}} = \alpha \cdot x_i + (1 - \alpha) \cdot x_j \quad (13)$$

where α is a random value that obeys beta distribution with a parameter 0.1. Corresponding classification loss of x_{mix} is

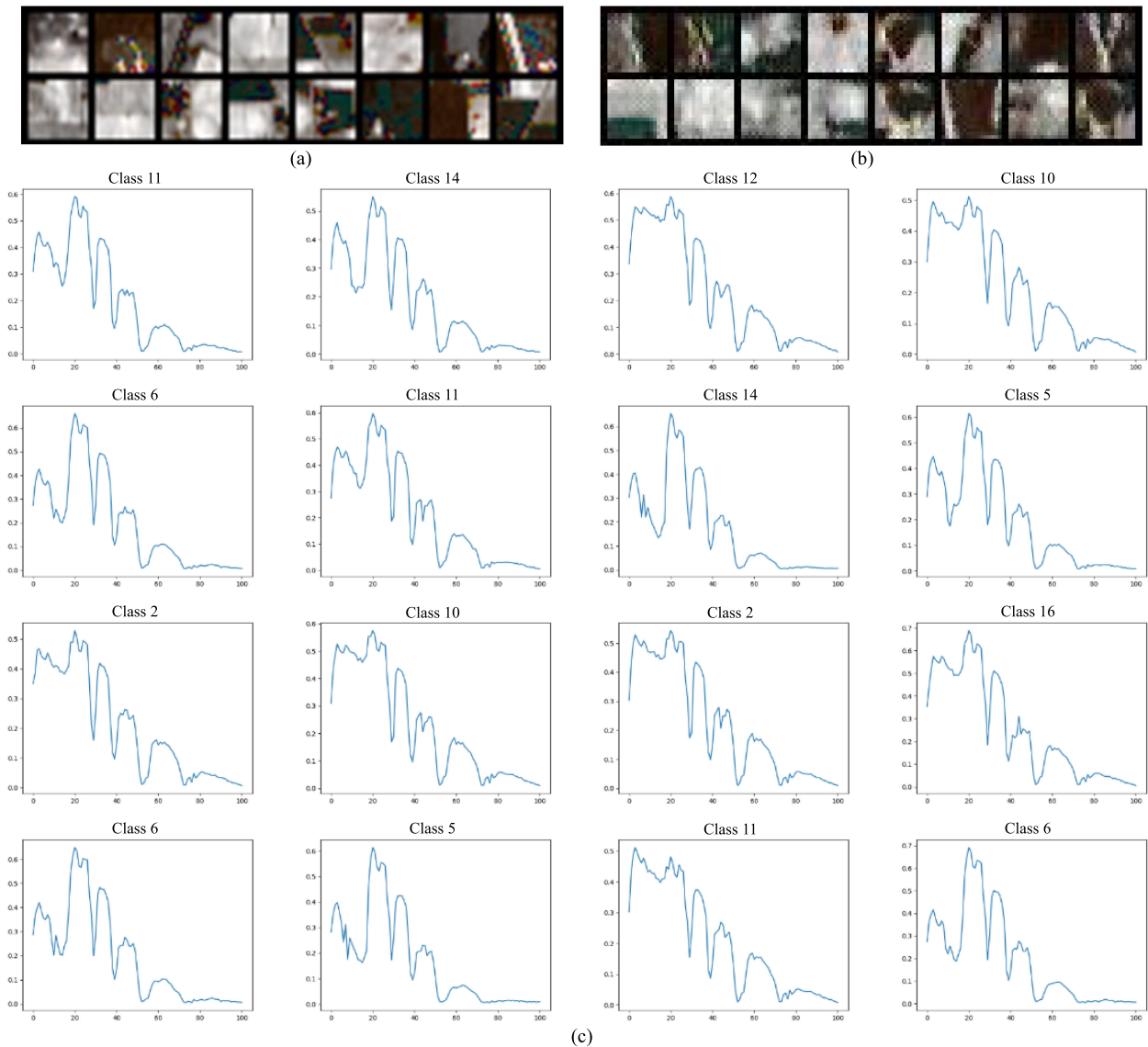


Fig. 4. Samples of IP dataset. (a) Real patch. (b) Fake patch. (c) Spectral curve of generated patches. The patches of the corresponding position in (a) and (b) have the same class label.

calculated by

$$L_{\text{cla}} = \alpha \cdot E [P(\text{class} = y_i | x_i)] + (1 - \alpha) \cdot E [P(\text{class} = y_j | x_j)]. \quad (14)$$

C. Experimental Results of Synthesis

The conditional GAN could learn complex and high-dimensional distributions of HSIs. Here, we verify the architecture plausibility for the synthetic 3-D cube. So far, there are many available GAN model evaluation indexes [53]. However, these indexes are designed for RGB images and unsuitable for HSIs. Evaluating the generated fake data from HSI-GAN is inherently challenging due to the nonsignificant spatial feature of the ground-object and the high spectral dimension. Here, we try to validate and compare the similarity of real and synthetic distributions under several criteria.

1) *Fidelity of Spatial-Spectral Feature*: First, we illustrate the visual results of real and generated samples. In general, the quality of the generated images could be validated by visual inspection. To an observer, a well-trained generator produces synthetic image patches which are visually similar to real ones, as shown in Figs. 4 and 5. The patches of the corresponding position in (a) and (b) have the same class label. The corresponding spectral curves at the center of each fake patch are presented in Figs. 4 and 5(c). The visual result and spectral curve demonstrate that the fake samples are comparable to the real ones for each class. In HSIs, each kind of ground-object is made up of irregular geometry and has little distinct spatial characteristics. Because the spatial visual discriminability of HSIs is very weak, we argue here that it is insufficient to evaluate the authenticity of the generated image according to only the visual result of spatial information.

Next, the distribution of the synthetic samples in spatial-spectral space should be followed. We apply principal

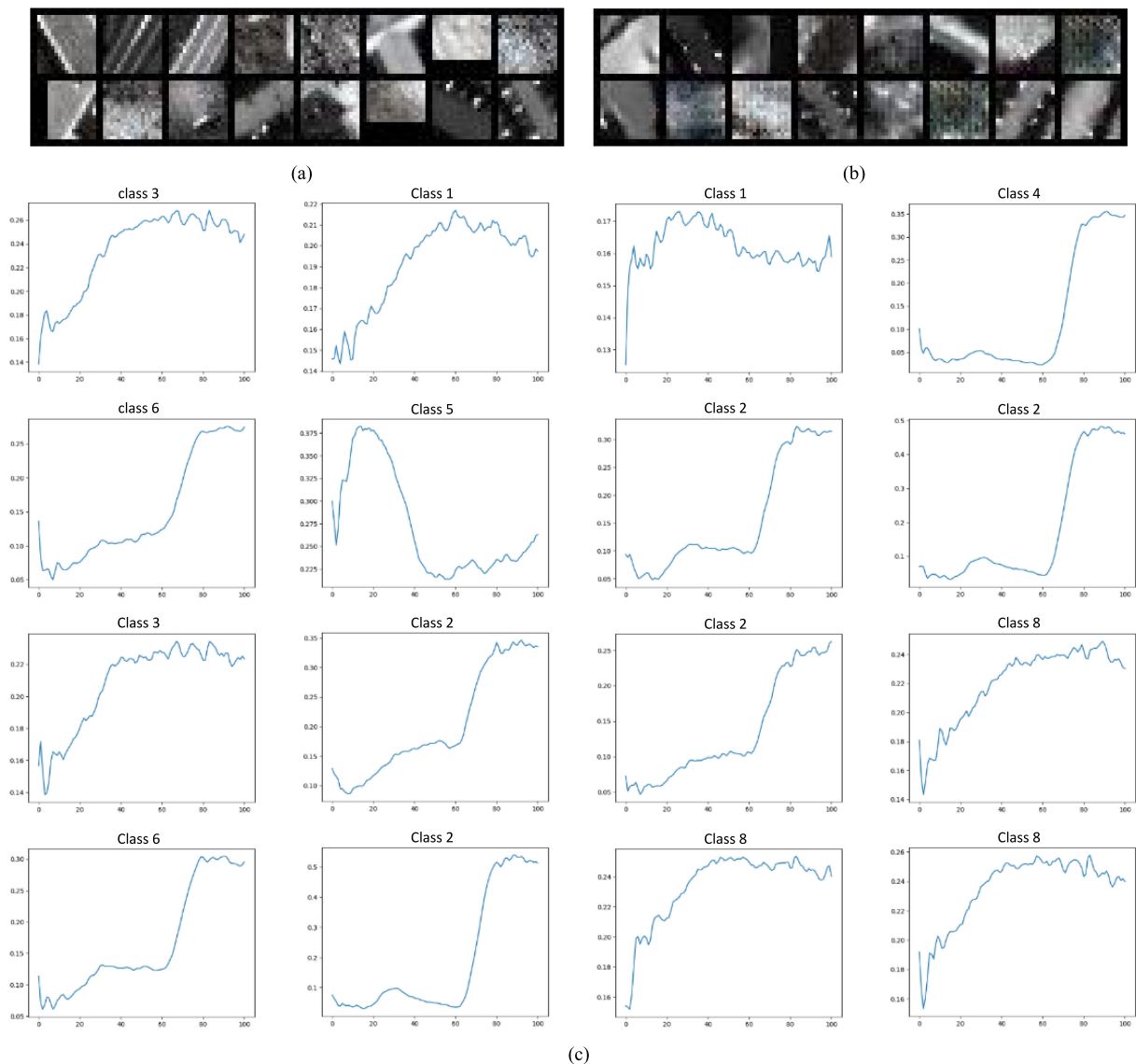


Fig. 5. Samples of PU dataset. (a) Real patch. (b) Fake patch. (c) Spectral curve of generated patches. The patches of the corresponding position in (a) and (b) have the same class label.

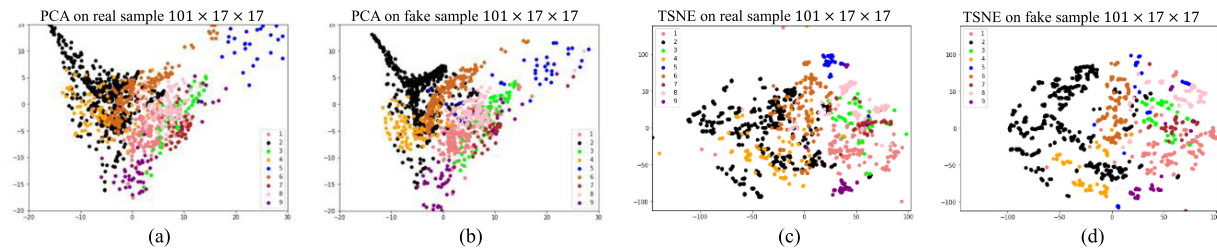


Fig. 6. Sample distribution of PU dataset. (a) and (b) PCA applied to the real and fake spatial-spectral patch. (c) and (d) TSNE applied to real and fake spatial-spectral patch. The real and fake sets have the same number of samples.

component analysis (PCA) and T-stochastic neighbor embedding (T-SNE) [54] to map the spatial-spectral information into a 2-D space (see Fig. 6). The cluster of the different classes is reproduced truthfully by synthetic samples. It is clear from this figure that although there are slight deformations between the real and synthetic data, the generator can well-capture the

classwise specificities. Through the observation of Fig 6, the class boundaries of fake distribution are clearer than the real ones, which implies that the fake data are easier to separate.

2) *Fidelity of Spectral Feature*: Due to the weak spatial feature of the ground-object in HSIs, spectral information is more critical for semantic analysis. The quality of the synthetic

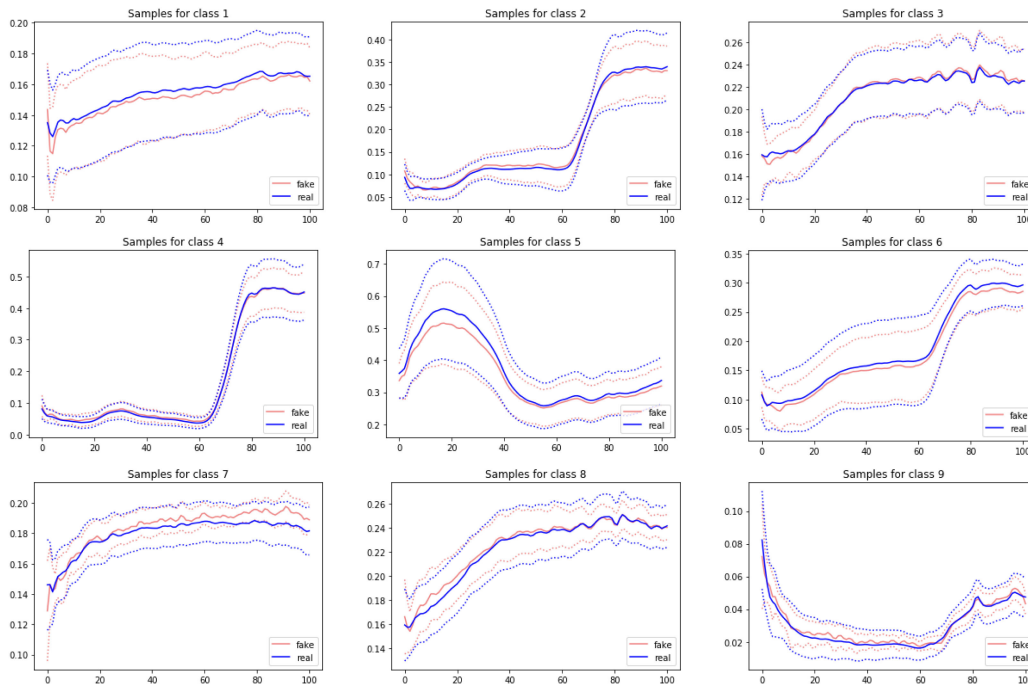


Fig. 7. Comparison of the mean spectrum and standard deviation per class on the PU dataset. The mean spectrum and standard deviation are indicated by the solid lines and dotted lines, respectively.

spectral curve is assessed visually by comparing the spectral characteristics between real and fake samples. The spectral curve produced by the generator is very similar to the real data curve, as shown in Fig. 7. The proposed GAN model could capture the significant spectral characteristics and the spectral shapes have been learned effectively by the generator. Fig. 8 represents the real and fake spectral feature in each category in more detail. Each dotted line in olive and pink is for the spectral curve of each real and generated samples, in which the number of real and fake samples is the same. We can see that most of the fake samples are within the standard deviation range of the real data. This shows that the generator learns well the distribution of training data, and then, the synthetic samples from a well-trained generator simulate the real data well. However, according to the observation of the spectral curve, we find out that the fake standard deviation of certain classes is slightly lower than the real one, which means that the synthetic spectrum is a little less diverse than the real one. Note that the x -axis indicates the number of spectral bands and the y -axis shows the normalized reflectance of spectral bands in Figs. 7 and 8.

3) *Tradeoff Between Diversity and Realism*: We apply the DB layer to increase the samples diversity. Through comparing results between adding/removing the DB layer, this approach seems to be beneficial to improving model performance.

First, we observe the effect of the DB layer for the generator from the spectral perspective. We select a certain class and generate 32 samples by using two trained generators with and without the DB layer, respectively. The comparison results of spectral curves at center pixel are shown in Fig. 9. We find out that the standard deviation of spectral curves from the right column (without DB layer) is lower than the one from the left column (with DB layer), which means that employing the DB layer in the generator could contribute to generating diverse samples and avoid model collapse.

Then, we observe the distribution of the synthetic samples. In Figs. 10 and 11, we can see that the synthetic samples of each class tend to cluster close together when using the generator without the DB layer. That is, the ability to capture inner class diverse characteristics is weak in Figs. 10(c) and 11(c).

Next, we try to improve the realism of the generated samples by increasing the weight of classification loss. However, in the training process, we find that the model intentionally avoids generating samples near the class decision boundary in order to pursue higher classification accuracy. That is why fake samples usually have a clearer boundary than the real one. We could do a tradeoff between the diversity and realism of generated samples by adjusting the classification coefficient λ_2 and λ_3 . The greater the coefficient, the more we expect the model to reduce the samples near the auxiliary classifier decision boundary. As shown in Figs. 10 and 11, (d) is more concentrated than (b).

D. Experimental Results of Classification

To verify that the synthesized hyperspectral samples are realistic and diverse, we use the fake data to augment existing hyperspectral datasets. In this part of the experiments, the advantages of synthetic samples for feature extraction and classification are investigated. First, we fix the trained G and only employ the classification loss L_{cla} to optimize D without considering the adversarial loss L_{adv} . For comparison, the discriminator is trained on the augmented dataset (AD), the real dataset (RD), and the fake dataset (FD), respectively. Finally, we use the same number of test samples from RD and FD to evaluate the classification performance. Overall accuracy (OA%), average accuracy (AA%), and kappa statistic ($K \times 100$) are adopted to assess the classification performance.

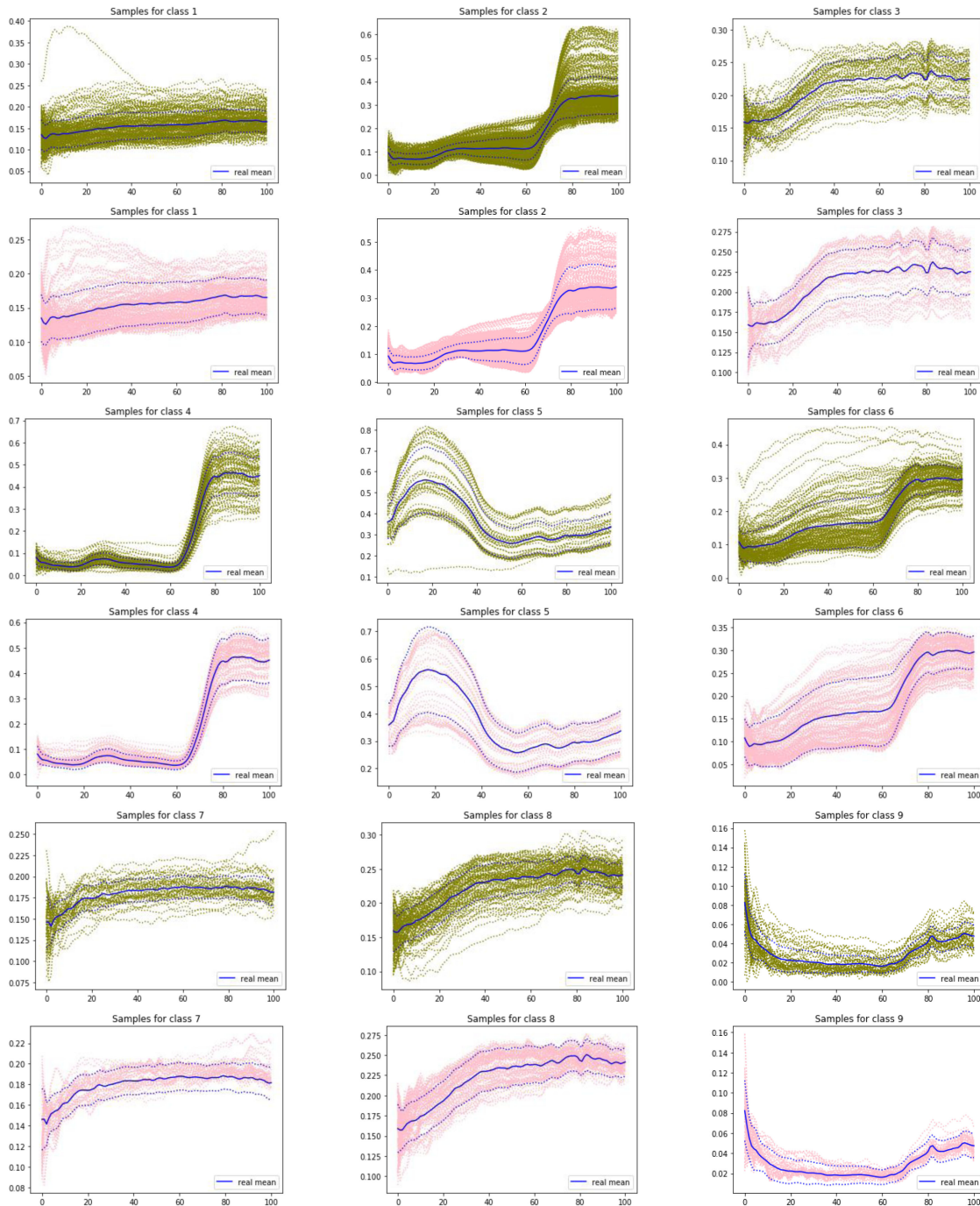


Fig. 8. Comparison of the spectral curve on the PU dataset. The solid and dotted lines in blue are used for the mean spectrum and standard deviation of real samples, respectively. The real and fake samples in each category are presented in olive and pink, respectively.

Each training minibatch is made up of real–fake mixed samples with a 50–50 ratio when using the augmented dataset. Note that, when the training samples were input in the discriminator, each fake sample is generated dynamically by trained G according to the label of the real training sample. That is, the fake samples are different in every training minibatch instead of using pregenerated fixed fake samples set. This strategy dramatically enriches the quantity of trainable samples.

As shown in Tables VI and VII, we train D by using the RD and AD, and then test the classification performance on AD and

FD. The D network trained by real–fake mixed dataset shows better performance on both test sets. It can also be found from the classification results that the fake data have more clear class boundaries than real samples and would be easier to distinguish.

Next, we try to train the D network only using FD. According to the test result provided by Table VII, the performance of fake samples is not bad in both the training and testing phases. The augmented HSI samples have a similar distribution with real data and could help the classifier to learn more robust and discriminative features. The superior classification performance

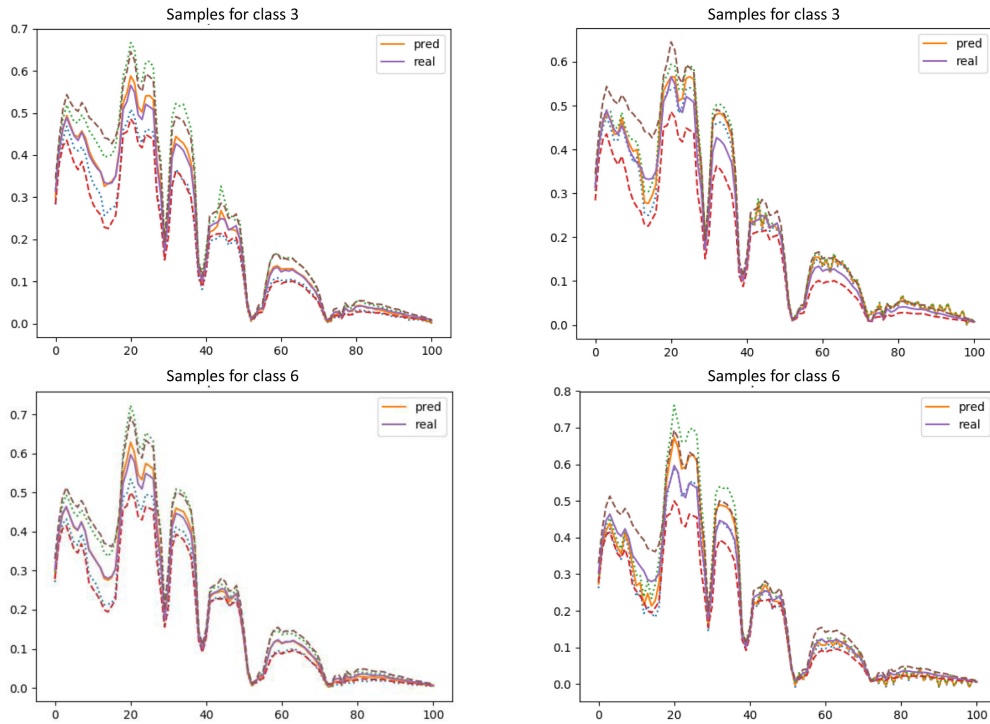


Fig. 9. Comparison of the spectral curve on the IP dataset. The left column is with DB layer, and the right column is without DB layer. The dashed lines and dotted lines are used for the standard deviation of real and fake data, respectively.

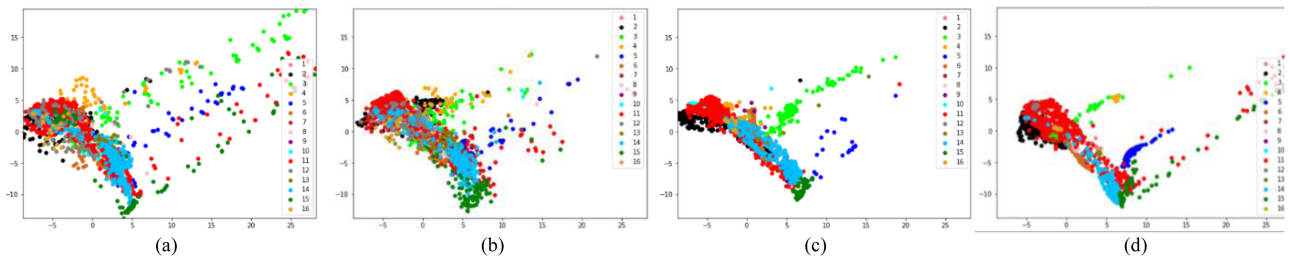


Fig. 10. Comparison of PCA applied to the spatial-spectral patch for IP dataset. (a) Real sample. (b) With DB layer, $\lambda_2 = \lambda_3 = 0.2$. (c) Without DB layer, $\lambda_2 = \lambda_3 = 0.2$. (d) With DB layer, $\lambda_2 = \lambda_3 = 1$.

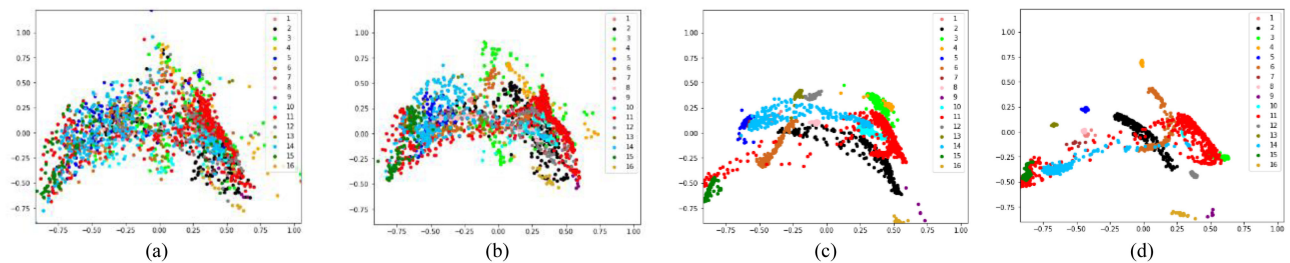


Fig. 11. Comparison of PCA applied to the spectral feature of IP data set. (a) Real sample. (b) With DB layer, $\lambda_2 = \lambda_3 = 0.2$. (c) Without DB layer, $\lambda_2 = \lambda_3 = 0.2$. (d) With DB layer, $\lambda_2 = \lambda_3 = 1$.

further verifies that the proposed GAN model could generate high fidelity synthetic 3-D HSI cubes. Under the condition of limited HSI training samples, the AD is a powerful and useful tool to improve classification accuracy. This is encouraging for using GANs as a data augmentation strategy in the HSI vision task.

Now, we compare our proposed classifier with other convolutional networks. Several supervised deep feature classification architectures based on CNN were conducted to extract the spectral-spectral feature for comparison. A baseline neural network (NN) is built using four fully connected layers with Dropout. In Lee *et al.* [55], a model is built through fully

TABLE VI
COMPARISON OF CLASSIFICATION RESULT ON THE IP DATASET

	Test on RD		Test on FD		Support
	Train on		Train on		
	RD	AD	RD	AD	
OA	94.63	96.04	88.15	99.99	8160
AA	94.59	95.93	87.90	99.99	8160
K	93.87	95.48	86.49	99.99	8160
1	95.77	80.64	100.00	100.00	37
2	91.07	94.40	90.86	100.00	1149
3	96.65	98.16	844.47	100.00	654
4	96.79	98.74	95.15	100.00	196
5	98.48	96.33	76.78	100.00	365
6	92.55	98.06	65.65	100.00	569
7	95.24	100.00	95.00	100.00	20
8	93.86	80.67	92.03	100.00	389
9	100.00	100.00	90.91	100.00	15
10	92.85	96.60	94.08	100.00	779
11	94.98	98.33	89.18	100.00	1962
12	91.30	94.99	91.62	100.00	479
13	94.81	96.66	89.49	100.00	162
14	98.15	96.11	90.96	99.95	1009
15	97.09	95.88	84.29	99.83	301
16	93.51	91.72	95.17	100.00	74

TABLE VII
COMPARISON OF CLASSIFICATION RESULT ON THE PU DATASET

	Test on RD			Test on FD			Support
	Train on			Train on			
	RD	FD	AD	RD	FD	AD	
OA	98.82	91.55	99.24	97.37	99.96	99.98	41431
AA	98.81	92.02	99.24	97.40	99.96	99.98	41431
K	98.44	88.87	98.99	96.50	99.95	99.97	41431
1	99.17	82.27	99.14	98.03	99.89	99.99	6432
2	99.35	99.36	99.64	99.38	99.97	99.96	18050
3	96.33	79.12	97.24	96.77	99.93	100.00	2040
4	97.65	97.01	99.41	99.44	100.00	100.00	2967
5	99.05	97.04	99.96	71.14	99.96	99.96	1305
6	98.47	98.76	99.09	98.82	100.00	99.93	4875
7	99.11	57.37	97.98	93.01	99.96	100.00	1285
8	98.23	75.38	98.60	93.59	99.99	100.00	3578
9	98.47	98.85	99.94	99.61	100.00	100.00	899

convolutional layers that eventually predict the corresponding label of each pixel vector. Also, three other networks use 3-D convolution and fully connected layers, Hamida *et al.* [56], Li *et al.* [57], and He *et al.* [58]. We perform comparison experiments using the default parameters and only adjust the training the number of iterations to optimize the result.

In Tables VIII and IX, we can see that the classification accuracy of our proposed model is higher than those of the other methods. The results show that the designed classifier can achieve the desired performance even without data augmentation. We think this mainly benefits from the multistage network structure. In the training phase, the classification losses from three stages are aggregated to optimize the parameters. The learning of coarse-to-fine helps the classifier to understand the semantic feature better. In the test phase, we have also integrated the results of various stages that come from three input patches instead of only the fine-grained prediction. As shown in Table X, in comparison with the single fine-grained prediction, the classifier with integrated prediction gives us a performance boost. The advantages of this strategy are even more apparent for the data with small interclass differences, such as the IP dataset. The

TABLE VIII
COMPARISON OF CLASSIFICATION RESULT OBTAINED BY DIFFERENT APPROACHES ON THE PU DATASET (3% TRAINING SAMPLE)

	NN	Hamida	Li	Lee	He	Train on RD	Train on AD
OA	92.84	93.45	93.92	92.25	93.51	98.82	99.24
AA	93.16	94.81	95.28	92.50	95.40	98.81	99.24
K	90.50	91.43	92.03	89.72	91.54	98.44	98.99
1	93.17	94.60	95.07	92.42	95.69	99.17	99.14
2	96.06	95.76	95.87	95.10	94.90	99.35	99.64
3	81.22	89.62	91.44	84.22	92.18	96.33	97.24
4	94.49	94.72	95.18	92.45	97.06	97.65	99.41
5	99.35	99.47	98.49	99.89	99.92	99.05	99.96
6	90.41	95.87	96.51	87.55	96.09	98.47	99.09
7	85.77	89.17	89.33	83.61	92.15	99.11	97.98
8	86.59	91.44	93.18	89.68	95.53	98.23	98.60
9	99.84	97.89	99.51	99.73	98.76	98.47	99.94

TABLE IX
COMPARISON OF CLASSIFICATION RESULT OBTAINED BY DIFFERENT APPROACHES ON THE IP DATASET (20% TRAINING SAMPLE)

	NN	Hamida	Li	Lee	He	Train on RD	Train on AD
OA	87.23	88.83	86.29	87.37	90.54	94.63	96.04
AA	87.08	89.54	87.01	87.21	91.59	94.59	95.93
K	85.42	87.27	84.42	85.52	89.22	93.87	95.48
1	77.42	83.78	94.29	84.85	60.71	95.77	80.64
2	83.14	84.84	80.07	81.26	90.57	91.07	94.40
3	82.89	82.45	79.65	82.72	86.79	96.65	98.16
4	76.47	86.89	82.26	75.76	94.15	96.79	98.74
5	93.06	92.98	92.89	91.02	93.99	98.48	96.33
6	94.92	98.64	98.21	96.22	98.63	92.55	98.06
7	93.33	95.65	92.68	80.85	97.67	95.24	100.00
8	97.31	98.69	99.48	98.32	97.19	93.86	80.67
9	77.78	84.21	78.57	78.57	100.00	100.00	100.00
10	83.48	84.79	78.70	82.91	87.80	92.85	96.60
11	85.99	88.64	84.76	88.69	91.71	94.98	98.33
12	84.28	83.02	85.43	76.25	86.15	91.30	94.99
13	98.77	99.69	98.16	98.48	97.91	94.81	96.66
14	95.06	98.19	97.80	95.01	97.90	98.15	96.11
15	69.24	83.19	80.48	76.50	78.65	97.09	95.88
16	96.55	98.65	97.37	94.44	84.37	93.51	91.72

TABLE X
CLASSIFICATION ACCURACY VALUES ON THE IP AND PU DATASETS: TRAINING AND TEST ON REAL DATA

	IP data set			PU data set		
	OA	AA	K	OA	AA	K
Fine-grained prediction	81.16	81.92	78.81	98.72	98.72	98.31
Integrated prediction	94.63	94.59	93.87	98.82	98.81	98.44

integrated prediction is a powerful tool, which means that the multistage method has a positive impact on HSI classification.

V. CONCLUSION

In this article, we present a novel multistage and multipole supervised conditional HSI-GAN, which demonstrates excellent abilities in feature learning and can be used to generate realistic synthesis samples.

Our model can output 3-D spatial-spectral HSI cube with all bands by taking full advantage of the capability of deep learning models. The multistage generator and discriminator are essential to stabilize the progressive training procedure, especially in high spectral resolutions. Our network structure employs an auxiliary spectral classifier which is mainly employed to strengthen the learning performance in the spectral domain.

In hyperspectral remote sensing cases, limited training samples and extremely similar interclass features could easily lead to model collapse. To solve this problem, we insert the DB layer into the generator. The experiment shows that the DB layer shows powerful capabilities and helps produce diverse synthesis samples. In addition, adjusting the classification loss weight is an efficient path for the tradeoff between the diversity and realism of generated samples. By using the proper architecture and training strategy, HSI-GAN could learn complex spatial-spectral features and generate high-fidelity 3-D HSIs.

We use the discriminator network as a classifier to evaluate the fidelity of fake data, while investigating the advantages of synthetic samples for feature extraction and classification. Extensive experimental results have demonstrated that synthetic samples can serve as a data augmentation strategy for hyperspectral datasets. Moreover, the discriminator integrates the multistage predicted result, and achieves positive performance improvements in terms of classification accuracy.

Our proposed model opens the door to new possibilities in spatial-spectral feature synthesis of HIS, and this method also could be extended to other complicated high-dimensional images.

REFERENCES

- [1] A. Signoroni, M. Savardi, A. Baronio, and S. Benini, "Deep learning meets hyperspectral image analysis: A multidisciplinary review," *J. Imag.*, vol. 5, no. 5, 2019, Art. no. 52.
- [2] P. Ghamisi, N. Yokoya, J. Li, W. Liao, and A. Plaza, "Advances in hyperspectral image and signal processing: A comprehensive overview of the state-of-the-art," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 37–78, Dec. 2018.
- [3] K. Tan, J. Hu, J. Li, and P. Du, "A novel semi-supervised hyperspectral image classification approach based on spatial neighborhood information and classifier combination," *ISPRS J. Photogramm. Remote Sens.*, vol. 105, pp. 19–29, 2015.
- [4] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 145PA, pp. 120–147, 2017.
- [5] B. Pan, Z. Shi, and X. Xu, "MugNet: Deep learning for hyperspectral image classification using limited samples," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 108–119, 2017.
- [6] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," *Adv. Neural Inf. Process. Syst.*, vol. 29, pp. 82–90, 2016.
- [7] M. Gadelha, S. Maji, and R. Wang, "Shape generation using spatially partitioned point clouds," 2017, *arXiv:1707.06267*.
- [8] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 40–49.
- [9] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2088–2096.
- [10] I. Goodfellow *et al.*, "Generative adversarial nets," *Adv. Neural Inf. Process. Syst.*, vol. 27, pp. 2672–2680, 2014.
- [11] 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. 1–5, Feb. 2018.
- [12] N. Audebert, B. L. Saux, and S. Lefèvre, "Generative adversarial networks for realistic synthesis of hyperspectral samples," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2018, pp. 4359–4362.
- [13] 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.
- [14] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. R. Salakhutdinov, "Good semi-supervised learning that requires a bad GAN," in *Proc. Neural Inf. Process. Syst. Conf.*, 2017, pp. 6510–6520.
- [15] Z. Lin, 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.
- [16] 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, Aug. 2019.
- [17] J. Feng, X. Feng, J. Chen, X. Cao, and T. Yu, "Generative adversarial networks based on collaborative learning and attention mechanism for hyperspectral image classification," *Remote Sens.*, vol. 12, no. 7, 2020, Art. no. 1149.
- [18] F. Luo, B. Du, L. Zhang, L. Zhang, and D. Tao, "Feature learning using spatial-spectral hypergraph discriminant analysis for hyperspectral image," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2406–2419, Jul. 2019.
- [19] L. Zhang, Q. Zhang, B. Du, X. Huang, Y. Y. Tang, and D. Tao, "Simultaneous spectral-spatial feature selection and extraction for hyperspectral images," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 16–28, Jan. 2018.
- [20] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.
- [21] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2642–2651.
- [22] X. Chen, Y. Duan, R. Houthoofd, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," *Adv. Neural Inf. Process. Syst.*, vol. 29, pp. 2172–2180, 2016.
- [23] Y. Hong, U. Hwang, J. Yoo, and S. Yoon, "How generative adversarial nets and its variants work: An overview of GAN," *ACM Comput. Surv.*, vol. 52, no. 1, 2017, Art. no. 10.
- [24] L. Chongxuan, T. Xu, J. Zhu, and B. Zhang, "Triple generative adversarial nets," in *Proc. Neural Inf. Process. Syst. Conf.*, 2017, pp. 4088–4098.
- [25] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," 2016, *arXiv:1606.03498*.
- [26] A. Odena, "Semi-supervised learning with generative adversarial networks," 2016, *arXiv:1606.01583*.
- [27] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "It takes (only) two: Adversarial generator-encoder networks," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1250–1257.
- [28] X. Xu, D. Corrigan, A. Dehghani, S. Caulfield, and D. Moloney, "3D object recognition based on volumetric representation using convolutional neural networks," in *Proc. Int. Conf. Articulated Motion Deformable Objects*, 2016, pp. 147–156.
- [29] Y. Li, S. Pirk, H. Su, C. R. Qi, and L. J. Guibas, "FPNN: Field probing neural networks for 3D data," *Adv. Neural Inf. Process. Syst.*, vol. 29, pp. 307–315, 2016.
- [30] J.-Y. Zhu *et al.*, "Visual object networks: Image generation with disentangled 3D representations," *Adv. Neural Inf. Process. Syst.*, vol. 31, pp. 118–129, 2018.
- [31] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," 2017, *arXiv:1710.10196*.
- [32] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8798–8807.
- [33] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.
- [34] E. L. Denton, S. Chintala, and R. Fergus, "Deep generative image models using a Laplacian pyramid of adversarial networks," *Adv. Neural Inf. Process. Syst.*, vol. 28, pp. 1486–1494, 2015.
- [35] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie, "Stacked generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5077–5086.
- [36] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1511–1520.
- [37] H. Zhang *et al.*, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5907–5915.
- [38] H. Zhang *et al.*, "StackGAN++: Realistic image synthesis with stacked generative adversarial networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1947–1962, Aug. 2019.
- [39] I. Durugkar, I. Gemp, and S. Mahadevan, "Generative multi-adversarial networks," 2016, *arXiv:1611.01673*.
- [40] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–14, 2017.
- [41] R. Shu, H. Bui, and S. Ermon, "AC-GAN learns a biased distribution," in *Proc. NIPS Workshop Bayesian Deep Learn.*, 2017.

- [42] Y. Taigman, A. Polyak, and L. Wolf, "Unsupervised cross-domain image generation," 2016, *arXiv:1611.02200*.
- [43] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3722–3731.
- [44] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," *Adv. Neural Inf. Process. Syst.*, vol. 29, pp. 658–666, 2016.
- [45] S. M. Arjovsky and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 214–223.
- [46] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Neural Inf. Process. Syst. Conf.*, 2017, pp. 5767–5777.
- [47] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4681–4690.
- [48] A. Ghosh, V. Kulharia, V. P. Namboodiri, P. H. Torr, and P. K. Dokania, "Multi-agent diverse generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8513–8521.
- [49] I. O. Tolstikhin, S. Gelly, O. Bousquet, C.-J. Simon-Gabriel, and B. Schölkopf, "AdaGAN: Boosting generative models," in *Proc. Neural Inf. Process. Syst. Conf.*, 2017, pp. 5424–5433.
- [50] I. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," 2016, *arXiv:1701.00160*.
- [51] G.-J. Qi, L. Zhang, H. Hu, M. Edraki, J. Wang, and X.-S. Hua, "Global versus localized generative adversarial nets," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1517–1525.
- [52] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "Dropblock: A regularization method for convolutional networks," in *Proc. Neural Inf. Process. Syst. Conf.*, 2018, pp. 10727–10737.
- [53] Q. Xu *et al.*, "An empirical study on evaluation metrics of generative adversarial networks," 2018, *arXiv:1806.07755*.
- [54] V. D. M. Laurens and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 2605, pp. 2579–2605, 2008.
- [55] H. Lee and H. Kwon, "Going deeper with contextual CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4843–4855, 2017.
- [56] A. B. Hamida, A. Benoit, P. Lambert, and C. B. Amar, "3-D deep learning approach for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4420–4434, Aug. 2018.
- [57] Y. Li, H. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sens.*, vol. 9, no. 1, pp. 67, 2017.
- [58] 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.*, 2017, pp. 3904–3908.



Wei Liu received the B.S. degree in computer application and the M.S. degree in agricultural mechanization engineering from Hebei Agricultural University, Baoding, China, in 2002 and 2005, and the Ph.D. degree in electrical and computer engineering from Jeonbuk National University, Jeonju, South Korea, in 2020.

She is currently a Lecturer with the College of Information Engineering, Hebei GEO University, Shijiazhuang, China. From 2003 to 2005, she was with the Key Laboratory of Modern Precision Agriculture System Integration Research, China Agricultural University, Beijing, China. Her current research interests include remote sensing image processing, machine learning applications of hyperspectral images, and computer vision.



Jie You received the B.S. degree in computer science from the Jiangxi University of Science and Technology, Ganzhou, China, and the M.S. degree in computer science and engineering, in 2019 from Jeonbuk National University, Jeonju, South Korea, where he is currently working toward the Ph.D. degree with the Artificial Intelligence Laboratory.

His research interests include GIS image processing and analysis, object detection, and pattern recognition.



Joonwhoan Lee was born in Kyunggi-do, South Korea, in 1957. He received the B.S. degree in electronic engineering from Hanyang University, Seoul, South Korea, in 1980, the M.S. degree in electric and electronic engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 1982, and the Ph.D. degree in electrical and computer engineering from the University of Missouri, Columbia, MO, USA, in 1990, respectively.

In 1985, he joined the Department of Electronic Engineering, Jeonbuk National University, Jeonju, South Korea, where he has been a Professor with the Department of Computer Science and Engineering since 1998. He was a Visiting Scholar with the School of Computing Science, Simon Fraser University, Burnaby, BC, Canada, during his sabbatical leave from 2013 to 2014. He is currently the Chairman of the AI Research Institute, Jeonbuk National University. His research interests include machine learning applications of signal processing, and fuzzy and rough sets.