

Adaptive DropBlock-Enhanced Generative Adversarial Networks for Hyperspectral Image Classification

Junjie Wang, Feng Gao^{ID}, *Member, IEEE*, Junyu Dong^{ID}, *Member, IEEE*, and Qian Du^{ID}, *Fellow, IEEE*

Abstract—In recent years, the hyperspectral image (HSI) classification based on generative adversarial networks (GANs) has achieved great progress. GAN-based classification methods can mitigate the limited training sample dilemma to some extent. However, several studies have pointed out that existing GAN-based HSI classification methods are heavily affected by the imbalanced training data problem. The discriminator in GAN always contradicts itself and tries to associate fake labels to the minority-class samples and, thus, impair the classification performance. Another critical issue is the mode collapse in GAN-based methods. The generator is only capable of producing samples within a narrow scope of the data space, which severely hinders the advancement of GAN-based HSI classification methods. In this article, we proposed an Adaptive DropBlock-enhanced Generative Adversarial Networks (ADGANs) for HSI classification. First, to solve the imbalanced training data problem, we adjust the discriminator to be a single classifier, and it will not contradict itself. Second, an adaptive DropBlock (AdapDrop) is proposed as a regularization method employed in the generator and discriminator to alleviate the mode collapse issue. The AdapDrop generated drop masks with adaptive shapes instead of a fixed size region, and it alleviates the limitations of DropBlock in dealing with ground objects with various shapes. Experimental results on three HSI data sets demonstrated that the proposed ADGAN achieved superior performance over state-of-the-art GAN-based methods. Our codes are available at https://github.com/summitgao/HC_ADGAN.

Index Terms—Adaptive DropBlock (AdapDrop), deep learning, generative adversarial network (GAN), hyperspectral image (HSI) classification.

I. INTRODUCTION

BENEFITING from the advancement of Earth observation programs, hyperspectral sensors have received great

attention over the past few years. A great number of hyperspectral images (HSIs) captured by spaceborne or airborne sensors are available [1]. These images have high spectral resolutions and abundant spatial information, which brings opportunities to a wide variety of applications, such as urban development [2], land cover change monitoring [3], environmental pollution monitoring [4], and resource management [5]. Among these applications, classification has become one of the most critical topics in the hyperspectral application community.

HSIs classification aims to assign a distinct label to each pixel vector so that it is well defined by a given class. Plenty of methods have been proposed to solve the problem. In the early days of HSI classification, researchers mainly focused on spectral information [6]–[8]. However, the same object in different locations may exhibit different spectral features, while different objects may emerge with similar spectral features [9]. It is commonly difficult to classify such objects by using spectral features alone. To solve the problem, many studies combined spectral features with spatial features to establish spectral–spatial models for HSI classification. Benediktsson *et al.* [2] proposed a classification method based on mathematical morphology profiles, which uses both the spatial and spectral features for classification. Fauvel *et al.* [10] established a framework that fused the morphological information and the original HSI. Li *et al.* [11] presented a classification framework that integrates the local binary patterns (LBPs), global Gabor features, and spectral features. In [12], spatial–spectral information was transposed into a sparse model for classification. Pan *et al.* [13] developed the hierarchical guidance filtering to obtain a set of spectral–spatial features from different scales, and then, an ensemble model is established to utilize these features simultaneously. Besides these techniques, morphological kernel [14], [15], edge-preserving filter [16], extinction profile [17], and superpixel segmentation [18] are also employed to explore spectral information and spatial information for HSI classification. The combination of spectral information and spatial information improves the classification performance [19]–[22]. Although these techniques have achieved excellent performance, they are mainly based on handcrafted descriptors. However, most handcrafted descriptors heavily depend on prior knowledge to obtain optimal parameters, which limits the applicability of these methods in various

Manuscript received December 4, 2019; revised April 3, 2020 and June 27, 2020; accepted July 19, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0100602, in part by the National Natural Science Foundation of China under Grant U1706218, and in part by the Key Research and Development Program of Shandong Province under Grant 2019GHY112048. (*Corresponding author: Feng Gao.*)

Junjie Wang, Feng Gao, and Junyu Dong are with the Qingdao Key Laboratory of Mixed Reality and Virtual Ocean, School of Information Science and Engineering, Ocean University of China, Qingdao 266100, China (e-mail: gaofeng@ouc.edu.cn).

Qian Du is with the Department of Electrical and Computer Engineering, Mississippi State University, Starkville, MS 39762 USA.

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TGRS.2020.3015843

scenarios. Robust feature extraction is widely acknowledged as a critical step in HSI classification.

Deep learning has become the most impactful developments in artificial intelligence and big data analysis over the past few years. It has been demonstrated that deep models are capable of extracting the invariant and discriminant features efficiently in computer vision and natural language processing tasks [23]–[26]. Inspired by these flourishing techniques, deep models have been designed to classify HSIs. Chen *et al.* [27] first presented a deep learning-based HSI classification method and employed a stacked autoencoder (SAE) as a classifier. In [28], the deep belief network (DBN) is introduced for spectral–spatial information exploration. Pan *et al.* [29] proposed a vertex component analysis network (VCANet), which takes the physical characteristics of HSIs into account. VCANet is capable to exploit discriminative features when training samples are limited.

Recently, convolutional neural networks (CNNs) have been widely used in HSI classification. CNNs make use of regional connections to extract contextual features and have shown outstanding classification performance. In [30], spectral features are extracted via 1-D CNN, and spatial features are exploited via 3-D-CNN. Then, spectral and spatial features are combined for classification. Chen *et al.* [31] developed a 3-D-CNN HSI classification model, in which L_2 regularization is used in the training procedure to mitigate the overfitting problem. Zhong *et al.* [32] presented an end-to-end CNNs that take the 3-D cube as input data. Residual learning is introduced to solve the exploding gradient problem. In [33], Gabor filters are combined with convolutional filters to alleviate the overfitting problem in CNN training. Inspired by the inception module [34], Gong *et al.* [35] proposed a CNNs with multiscale convolution. The multiscale filter banks enrich the representation power of the deep model. Ma *et al.* [36] designed an end-to-end deconvolution network with skip architecture for spatial and spectral features' extraction. The network is capable to recover the lost information in the pooling operation of CNN via unpooling and deconvolution layers.

CNN-based methods have achieved tremendous progress in HSI classification. However, the performance of these techniques heavily depends on the number of training samples. Commonly, it is a challenging task to collect lots of training samples from HSIs. This problem can be alleviated by data augmentation. Cropping, horizontal flipping, and generative model are typical data augmentation techniques. Recently, the generative model has drawn a lot of attention since it is able to generate high-quality samples to alleviate the overfitting problem. Goodfellow *et al.* [37] designed the generative adversarial network (GAN), which is comprised of a generator G and a discriminator D . The generator G captures the data distribution, while D judges that whether a sample comes from G or from the training data. The generator G can be considered as a regularization method that can effectively alleviate the overfitting problem to a great extent.

Researchers have made efforts to design GAN-based models to alleviate the limited high-quality sample problem. Zhan *et al.* [38] proposed a semisupervised framework based on 1-D-GAN. After that, Zhu *et al.* [39] proposed a 3-D-GAN

for HSI classification. The spatial information is taken into consideration, and a softmax classifier is employed in the discriminator D to auxiliary classification. Feng *et al.* [40] proposed a multiclass GAN for HSI classification. Two generators are designed in multiclass GAN to generate HSI patches, and a discriminator is devised to output multiclass probabilities. Zhong *et al.* [41] integrated GAN and conditional random field (CRF) together, where dense CRFs impose graph constraints on the discriminators of GAN to refine the classification results.

Although these GAN-based models have achieved satisfying performance over their contemporary baselines, there still exist two drawbacks over HSI classification, which are urgently needed to be solved.

The first challenge is the imbalanced training data. The accuracy of classification is likely to deteriorate when available training samples are not uniformly distributed among different classes. However, the imbalanced training data problem is fundamental in HSIs since objects with different sizes present in a typical scene [42]. In Zhu's work [39], auxiliary classifier GAN (ACGAN) [43] is employed for HSI classification. In ACGAN, the discriminator has two outputs: one to discriminate real and fake samples, and the other to classify samples. It seems that ACGAN is capable of producing samples of a specific class. In practice, it is observed that two loss functions of the discriminator turnout to be flawed when generating the minority-class samples. The reason for this phenomenon is that when minority-class samples are passed to the discriminator, they are likely to be assigned the fake label. Therefore, the discriminator intends to associate a fake label to the minority-class samples. At this point, the generators produce samples that look real but not represent the minority class. The quality of generated samples is deteriorated, and hence, the classification performance is impaired.

Another critical issue is mode collapse. The generator fools the discriminator by only producing data from the same data mode [44]. It leads to a weak generator that can generate samples within a narrow scope of the data space. Therefore, the generated samples are too similar for the model to learn the true data distribution, and the model can hardly learn the full data distribution. The model collapse can be considered as a consequence of overfitting to the feedback of the discriminator. In a disparate line of work, DropBlock [45] was designed in CNNs to alleviate overfitting. In DropBlock, features in a square mask from one feature map are dropped together during training. It is demonstrated that DropBlock can learn more spatially distributed representations. However, when dealing with objects with various shapes, the fixed square masks are inflexible. We argue that if irregularly shaped masks are taken into account, the mode collapse problem can be alleviated.

To tackle the aforementioned limitations of GAN-based classification methods, we established an Adaptive Dropblock-enhanced GANs (ADGANs) for HSI classification. On the one hand, considering the contradiction between the loss functions of the discriminator in ACGAN, the discriminator in ADGAN is adjusted to be one single output that returns either the specific class label or the fake label. The generator is trained to avoid the

fake label and match the desired class labels. Since the discriminator is now defined as one single objective, it will not contradict itself. On the other hand, we propose adaptive DropBlock (AdapDrop) as a regularization method used in the generator and the discriminator. Instead of dropping a fixed size region, the AdapDrop generated drop masks with adaptive shapes, relaxing the limitations of DropBlock in dealing with objects with various shapes.

To validate the effectiveness of the proposed method, extensive experiments are conducted on three data sets. Experimental results demonstrate that the proposed ADGAN yields better performance than state-of-the-art GAN-based methods. In summary, our contributions are threefold.

- 1) We develop a novel GAN-based HSI classification model that contains a single output in discriminator. The contradictions in ACGAN when dealing with minority-class samples are mitigated.
- 2) For the purpose of alleviating the mode collapse problem, we propose the AdapDrop for regularization. The AdapDrop generates masks with adaptive shapes, which can boost the classification performance.
- 3) We conducted extensive experiments on three well-known hyperspectral data sets under the condition of limited training samples to validate the effectiveness of the proposed method. The experimental results achieve competitive results compared with other state-of-the-art classification methods.

The rest of this article is organized as follows. In Section II, the basic concepts of ACGAN and DropBlock are briefly reviewed. The scheme of the proposed method and its components are introduced in Section III. Experimental results and analysis are presented in Section IV. Finally, conclusions are drawn in Section V.

II. BACKGROUND

A. Generative Adversarial Networks

In recent years, GANs provide a solution to estimate input data distribution and correspondingly generate synthetic samples [37]. GAN is comprised of two parts: the generator G and the discriminator D . The generator G attempts to learn the distribution of real data and generate data that subject to this distribution. The discriminator D judges whether the input is real or fake. The generator G takes a random noise vector z as input and outputs an image $X_{\text{fake}} = G(z)$. The discriminator D receives a real image or a synthesized image from G as inputs and outputs a probability distribution $P(S|X) = D(X)$. The discriminator D is trained to maximize the log-likelihood it assigns to the correct source as follows:

$$L_D = E[\log P(S = \text{real}|X_{\text{real}})] + E[\log P(S = \text{fake}|X_{\text{fake}})]. \quad (1)$$

The generator G is trained to minimize the following likelihood:

$$L_G = E[\log P(S = \text{fake}|X_{\text{fake}})]. \quad (2)$$

When training GAN, an alternating optimization technique is employed. Especially, D is optimized by maximizing L_D

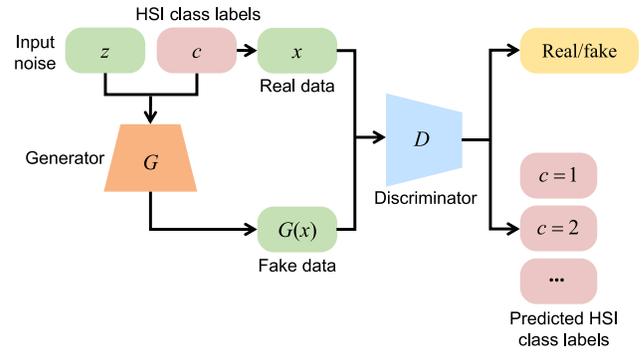


Fig. 1. Architecture of ACGAN employed in [39] for HSI classification.

with G fixed in one iteration. After that, G is optimized with minimizing L_G with D fixed. In such adversarial training, the discriminator D and the generator G promote each other. After many iterations, G captures the distribution of real data. At the same time, the capability of D to distinguish real data and fake data is enhanced.

B. Auxiliary Classifier GANs

In naive GAN, the discriminator only judges whether the input samples are real or fake. Therefore, they are not suitable for multiclass image classification. To tackle the limitations of naive GAN, Odena *et al.* [43] proposed ACGAN. In ACGAN, the discriminator D is a softmax classifier that can output multiclass label probabilities.

The architecture of ACGAN employed in [39] is illustrated in Fig. 1. The generator G accepts the class label c as input. The real data with the corresponding label and the data generated by G are regarded as the input of D . The discriminator D has two outputs: one to discriminate the real and fake data and the other to classify input in terms of its class c . The loss function of ACGAN is comprised of two parts: the log-likelihood of the right source of input L_S and the log-likelihood of the right class labels L_C . The L_S and L_C are computed as

$$L_S = E[\log P(S = \text{real}|X_{\text{real}})] + E[\log P(S = \text{fake}|X_{\text{fake}})] \quad (3)$$

$$L_C = E[\log P(C = c|X_{\text{real}})] + E[\log P(C = c|X_{\text{fake}})]. \quad (4)$$

During training, the generator G is optimized to maximize $L_C - L_S$, and the discriminator D is optimized to maximize $L_S + L_C$. Therefore, the generator G can be conditioned to draw a sample of the desired class.

C. DropBlock

Most current deep models are inclined to suffer from overparameterization and, therefore, give rise to the overfitting problem. In this regard, regularization methods are harnessed to mitigate this issue. To date, dropout [46] is a widely used regularization method and has been proven to be rather effective for fully connected layers. However, features in

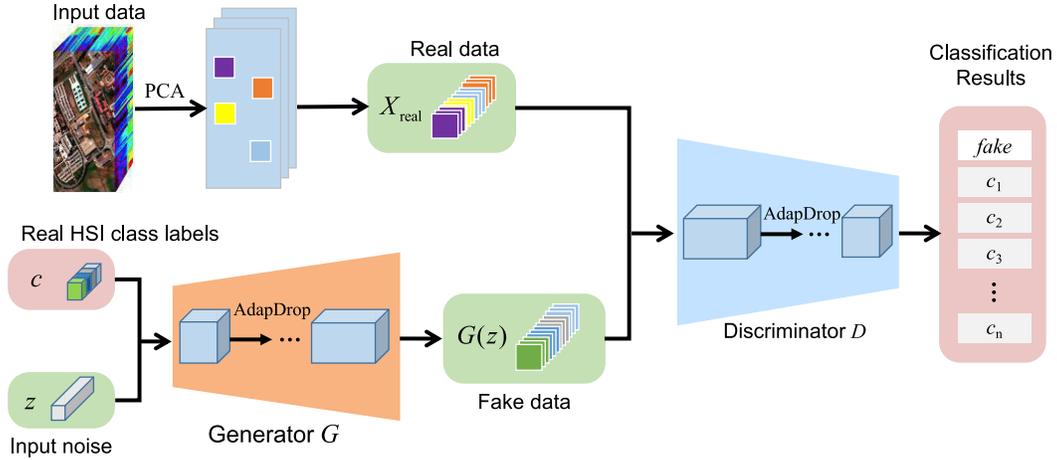


Fig. 2. Framework of ADGAN for HSI classification.

convolutional layers are highly spatially correlated. Dropout becomes less effective since it does not take image spatial information into account.

Recently, Ghisasi *et al.* [45] proposed DropBlock that is particularly effective to regularize the convolutional layers. Rather than dropping out random units, DropBlock drops contiguous regions from a feature map in the convolutional layer. It can be considered as a form of structured dropout. It is demonstrated that employing DropBlock in convolutional layers and skip connections effectively improves the classification performance.

III. METHODOLOGY

A. Framework of the Proposed ADGAN

The framework of the proposed ADGAN is illustrated in Fig. 2. The input HSI contains hundreds of bands, and there is a lot of redundancy among these spectral bands. It is rather difficult to obtain a robust generator G since the generator can hardly imitate the real data when high redundancy exists. Therefore, the number of spectral bands of the input HSI is reduced to three components by PCA [47]. The spectral information can be condensed to a suitable scale by PCA. This operation is a nontrivial step since PCA can not only dramatically reduce the computational complexity but also contribute to training a robust generator G .

From Fig. 2, it can be observed that the input of the generator G includes both noise z and class labels c . The discriminator D receives the image patches X_{real} with labels c and some fake patches $X_{\text{fake}} = G(z, c)$. It should be noted that in the proposed ADGAN, the discriminator D has only one single output that returns either a specific class c or the fake label. Then, the generator G is trained to generate image patches that match the desired class label. To this end, the discriminator D is trained to maximize the log-likelihood as follows:

$$L_D = E[\log P(C = c|X_{\text{real}})] + E[\log P(C = \text{fake}|X_{\text{fake}})]. \quad (5)$$

The generator G is trained to maximize the log-likelihood as follows:

$$L_G = E[\log P(C = c|X_{\text{fake}})]. \quad (6)$$

The first term of L_D encourages the discriminator D to assign a true label for real samples, and the second term expects to assign a fake label to the generated samples. On the contrary, the generator G expects to draw a sample of the desired class. By adversarial learning, the generator G captures the real data distribution of the desired class.

As mentioned earlier, the discriminator in Zhu's work [39] has two outputs: one to discriminate the real and fake data, and the other to classify the input in term of its class c . In the training phase, the generator aims to draw images belonging to class c . Therefore, the parameters of the generator are optimized to maximize the superposition of two components. The first is the log-likelihood of generating an image that the discriminator considers real. The second component is the log-likelihood of generating an image that the discriminator considers it to be class c . However, there exists a contradiction between two components when dealing with the minority class. Especially, when a generated minority-class image is fed into the discriminator, it is likely to be judged as a fake image since the minority-class images are scarce in the training set. To optimize its loss function, the discriminator prefers to associate a fake label to the minority-class images. Then, the two components of the generator will contradict each other, and two components can hardly be optimized at once. This phenomenon deteriorates the quality of generated images, which severely limits the performance of GAN-based approaches for HSI classification. To solve this problem, we proposed the ADGAN that alleviates the contradiction in ACGAN and achieves the balance in training samples to some extent.

In the proposed ADGAN, the discriminator D has one output that returns either a specific class c or the fake label, as shown in Fig. 2. The discriminator D is trained to associate the real samples with their class label c . Meanwhile, D also tries to associate the samples generated by G with the fake

label. On the contrary, the generator G is trained to avoid the fake label and match the generated samples to the desired class. By doing so, the balance of training samples can be balanced to some extent. Besides, since the discriminator in ADGAN is now defined as one single objective rather than a combination of two objectives, it will not contradict itself.

In ADGAN, the network extracts the spectral and spatial features simultaneously. The input hyperspectral data are condensed by PCA, and only three components are reserved. Through PCA, an optimal representation of the input HSI is achieved, and the computational burden is also dramatically reduced. From Fig. 2, we can observe that the generator G accepts random Gaussian noise z as input. The noise z is transformed to the same size as the real input data with three bands in the spectral domain. After that, the discriminator D accepts the generated fake samples together with the real samples as input. The output of D indicates the probability that the input sample belongs to class c or fake. After many iterations, both G and D achieve optimized results. Especially, G can generate fake data that subject to the distribution of real data, while D can hardly discriminate it. The competition between both networks can promote HSI classification performance. The key idea of ADGAN lies in restoring the balance of data set, and through the design of novel adversarial objective functions, the contradiction in ACGAN can be alleviated.

B. Structured Dropout With Attention Mechanism: Adaptive DropBlock

In the proposed ADGAN, G and D are both in the form of convolutional networks. Pooling layers are replaced with strided convolutions. In addition, batch normalization is employed in both the G and D . To further improve the classification performance, we proposed AdapDrop as the regularization method to enhance the spatially correlated feature representations.

Deep neural networks generally suffer from overparameterization and, thus, give rise to the overfitting problem. Regularization methods, such as batch normalization and dropout, are harnessed to mitigate the problem. In this article, the proposed ADGAN is comprised of two convolutional networks, which makes it more complex to regularize.

Many regularization methods have been proposed to alleviate the overfitting problem, such as DropBlock [45], DropPath [48], and DropConnect [49]. DropBlock is one of the most commonly used regularization methods in CNNs. It operates on the feature maps, and then, the units in a random contiguous region of feature maps are dropped together. As DropBlock discards features in a correlated area, the remaining semantic information will be used to fit the training data. However, DropBlock backfires when it drops overmuch information in the feature map since it drops the whole blocks with a fixed shape, which may contain essential features for training, as shown in Fig. 3.

To mitigate the drawback of DropBlock, we propose AdapDrop that is a structured regularization method with an attention mechanism. AdapDrop first randomly selects some blocks in the feature map. Then, it produces an adaptive mask with

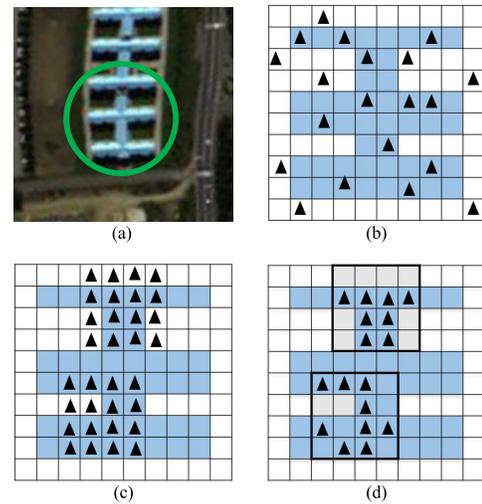


Fig. 3. Illustration of dropout, DropBlock, and AdapDrop. (a) Input image. (b) Dropout. (c) DropBlock. (d) Proposed AdapDrop. The regions with meaningful information are marked by blue squares, and dropped operations are marked by black triangles. Dropout can hardly utilize spatial information. DropBlock takes advantage of the spatial information, but it breaks the meaningful semantic information for learning. The proposed AdapDrop drops pixels based on the activations of the input and improves the model's capability of focusing on nontrivial spatial information.

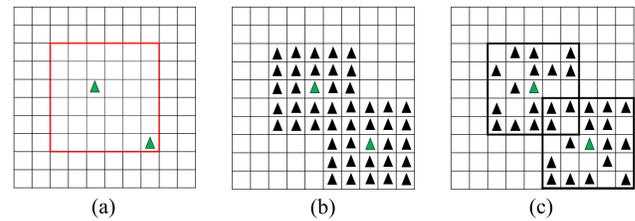


Fig. 4. Illustration of the proposed AdapDrop. (a) Similar to DropBlock, two elements are sampled from each feature map. (b) Every element is expanded to a $b_size \times b_size$ block. (c) In each block, the largest k th percentile elements are dropped.

irregular shapes in the selected blocks by dropping the top- k th percentile elements. The top- k th percentile elements are chosen according to the values in the feature map. Due to the continuity of image pixel values, the neighboring pixels in the feature map have similar values. Hence, when we drop the top- k th percentile elements, it is usually expressed as an irregular shape according to the spatial characteristics of the target object, such as the roof of a building and lawn garden. Therefore, the AdapDrop effectively removes maximally activated regions and encourages the network to consider less prominent features.

The three phases of AdapDrop are shown in Fig. 4. An adaptive mask is applied to the feature map and scale the output. The AdapDrop algorithm is shown in Algorithm 1. It has three parameters: b_size , k , and γ . b_size is the size of the mask block, and k denotes that the top- k th percentile elements in the mask block will be dropped. γ controls the number of features to drop, and the computation of γ can be found in [45]. The current feature map $A^{(n)}$ is first normalized, and a new feature map $A'^{(n)}$ is generated. A set of pixels are sampled with the Bernoulli distribution. For each position $M_{i,j}$, create

Algorithm 1 Workflow of AdapDrop**Input:** Feature of the current layer $A^{(n)}$, b_size , γ , k **Output:** Feature of the next layer $A^{(n+1)}$

1: Normalize the feature:

$$A'^{(n)} = \frac{A^{(n)} - A_{\min}^{(n)}}{A_{\max}^{(n)} - A_{\min}^{(n)}}$$

2: Randomly sample some pixels: $M_{i,j} \sim \text{Bernoulli}(\gamma)$ 3: For each $M_{i,j}$, create a block centered at $M_{i,j}$. The size of the block is $b_size \times b_size$. In each block, set the top- k th percentile elements to be zero, and set the rest elements to be one.4: Apply the mask: $A'^{(n)} = A'^{(n)} \times M$

5: Scale the output feature:

$$A^{(n+1)} = A'^{(n)} \times \text{count}(M) / \text{count_ones}(M)$$

TABLE I
IMPLEMENTATION DETAILS OF THE PROPOSED ADGAN

Nets	No.	Type	Kernel size	AdapDrop	Stride	Padding	Activation function
G	1	Conv	4×4×512	no	1	no	ReLU
	2	Conv	4×4×256	yes	2	yes	ReLU
	3	Conv	4×4×128	no	2	yes	ReLU
	4	Conv	4×4×64	no	2	yes	ReLU
	5	Conv	4×4×3	no	2	yes	Tanh
D	1	TConv	4×4×64	no	2	yes	LeakyReLU
	2	TConv	4×4×128	no	2	yes	LeakyReLU
	3	TConv	4×4×256	no	2	yes	LeakyReLU
	4	TConv	4×4×512	yes	2	yes	LeakyReLU
	5	TConv	4×4×128	no	1	no	—
	6	FC	—	no	—	—	Softmax

a spatial square block centered at $M_{i,j}$. The size of the block is $b_size \times b_size$. In each block, the top- k th percentile elements are set to be zero, and the rest elements are set to be one. Next, apply the mask, and scale the output as

$$A^{(n+1)} = A'^{(n)} \times \text{count}(M) / \text{count_ones}(M). \quad (7)$$

$\text{count}(M)$ denotes the number of elements in the masks, and $\text{count_ones}(M)$ denotes the number of one in the masks.

C. Implementation Details of the Proposed ADGAN

Table I shows the implementation details of the proposed ADGAN. The generator G and discriminator D are CNNs with five convolutional layers. The size of the input noise is $100 \times 1 \times 1$. The generator G converts the inputs to fake samples with a size of $64 \times 64 \times 3$. In the generator G , the AdapDrop is employed in the second convolutional layer, while the AdapDrop is employed in the fourth transposed convolutional layer in the discriminator D .

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we first describe the data sets used in our experiments. Then, an exhaustive investigation of several important parameters of the AdapDrop is presented. Besides,

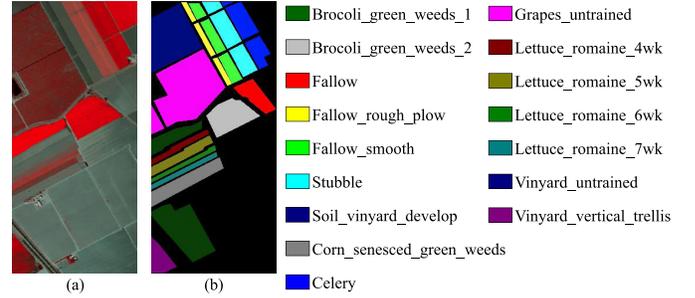


Fig. 5. Salinas data set. (a) False-color composite. (b) Ground reference map.

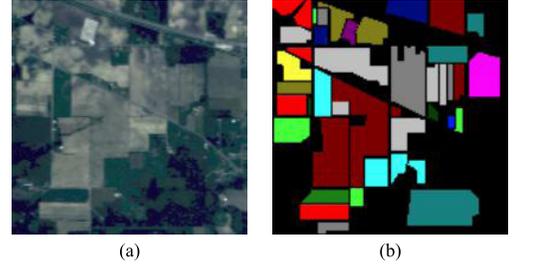


Fig. 6. Indian Pines data set. (a) False-color composite. (b) Ground reference map.

we tested the impact of different regularization methods on classification results in the network. Next, the comparisons with five closely related HSI classifiers are provided. After that, we compared the running time of different classification methods. Finally, we visualized the generated samples to show the advantage of ADGAN.

A. Data Sets' Description

To evaluate the performance of the ADGAN on HSI classification, three representative HSI data sets are used, including the Salinas, Indian Pines, and Pavia University data sets.

- 1) The Salinas data set was captured by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor. The size of the image is 512×217 pixels. The data set is comprised of 204 spectral bands. Some low signal-to-noise ratio (SNR) bands are removed. The data set has a high spatial resolution of 3.7 m per pixel. The false-color composite image (bands 50, 170, and 190) and the corresponding ground reference map are illustrated in Fig. 5.
- 2) The Indian Pines data set is a mixed vegetation site in Northwestern Indiana, and it was collected by the AVIRIS sensor. The size of the data set is 145×145 pixels. It is comprised of 220 spectral bands in the wavelength range of 0.4–2.5 μm . The false-color composite image and the ground reference map are shown in Fig. 6. It should be noted that the water absorption bands are removed in our implementations.



Fig. 7. Pavia University data set. (a) False-color composite. (b) Ground reference map.

TABLE II
SAMPLES' DISTRIBUTION FOR THE SALINAS DATA SET

No.	Class	Training	Test
1	Brocoli_green_weeds_1	11	1998
2	Brocoli_green_weeds_2	20	3706
3	Fallow	11	1965
4	Fallow_rough_plow	8	1386
5	Fallow_smooth	14	2664
6	Stubble	22	3937
7	Celery	19	3560
8	Grapes_untrained	62	11209
9	Soil_vinyard_develop	34	6169
10	Corn_senesced_green_weeds	18	3260
11	Lettuce_romaine_4wk	9	1059
12	Lettuce_romaine_5wk	11	1916
13	Lettuce_romaine_6wk	5	911
14	Lettuce_romaine_7wk	6	1064
15	Vinyard_untrained	40	7228
16	Vinyard_vertical_trellis	10	1797
Total		300	53829

3) The Pavia University data set was acquired by the Reflective Optics System Imaging Spectrometer (ROSIS) in Northern Italy in 2001. The data set converts nine urban land-cover types. The size of the data set is 610×340 pixels, and the resolution of the image is 1.3 m per pixel. The data set is comprised of 103 spectral bands in the wavelength range from 430 to 860 nm. Fig. 7 illustrates the data set and the corresponding ground reference map.

For all three data sets, the labeled samples were split into two parts: the training set and the test set. Because of the relatively higher computational complexity of the GANs, we try to control the number of training samples to ensure stable experimental results. After numerous experiments, we found that randomly selecting 300 training samples on the Salinas data set, 1000 training samples on the Indian Pines data set, and 1000 training samples on the Pavia University data set can ensure stable results. For the Salinas data set, the number of training and test samples for each class are listed in Table II. For the Indian Pines and Pavia University data sets, sample

TABLE III
SAMPLES' DISTRIBUTION FOR THE INDIAN PINES DATA SET

No.	Class	Training	Test
1	Alfalfa	5	41
2	Corn-notill	139	1289
3	Corn-min	81	749
4	Corn	23	214
5	Grass-pasture	47	436
6	Grass-trees	71	659
7	Grass-pasture-mowed	3	25
8	Hay-windrowed	46	432
9	Oats	2	18
10	Soybean-notill	95	877
11	Soybean-mintill	240	2215
12	Soybean-clean	58	535
13	Wheat	20	185
14	Woods	123	1142
15	Buildings-Grass-Trees	38	348
16	Stone-Steel-Towers	9	84
Total		1000	9249

TABLE IV
SAMPLES' DISTRIBUTION FOR THE PAVIA UNIVERSITY DATA SET

No.	Class	Training	Test
1	Asphalt	155	6476
2	Meadows	436	18213
3	Gravel	49	2050
4	Trees	72	2992
5	Painted metal sheets	31	1314
6	Bare soil	118	4911
7	Bitumen	31	1299
8	Self-Blocking Bricks	86	3596
9	Shadows	22	925
Total		1000	41776

distribution is listed in Tables III and IV, respectively. The training set adjusts the parameters during the training process by testing the classification accuracies and the losses of the temporary model generated during training. The network with the lowest loss is selected for testing. In the test process, all the test samples in the data set are used to estimate the capability of the trained network. Three evaluation criteria, including overall accuracy (OA), average accuracy (AA), and Kappa coefficient (κ) are presented for all test samples.

B. Parameter Analysis

1) *Analysis of b_size* : The size of block b_size in AdapDrop is an important parameter that affects the classification accuracy. The contextual information in classification is sensitive to neighborhood noise. Fig. 8 shows the classification performance on three data sets under a different b_size . In our implementations, b_size varies from 3 to 11. We can observe that when b_size increases from 3 to 7, the classification accuracy improves since more contextual information is taken

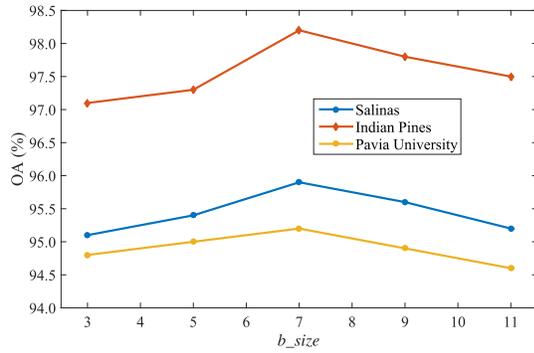


Fig. 8. Relationship between the classification accuracy and b_size on three data sets.

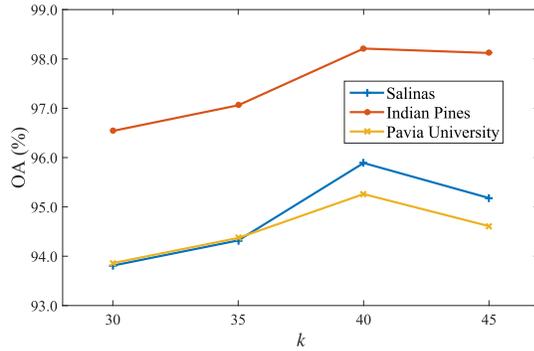


Fig. 9. Relationship between the classification accuracy and k on three data sets.

into account. However, when the larger block size is selected, continuous blank areas affect the robust training of the network. When $b_size = 7$, the best accuracy is achieved. Therefore, b_size is set to 7 in the following experiments.

2) *Analysis of k* : The parameter k denotes that the top- k th percentile elements in the mask block will be dropped, and it is a critical parameter in AdapDrop. We evaluate the classification performance by take $k = 30, 35, 40$, and 45 , respectively. Fig. 9 illustrates the influence of k on classification accuracy on three data sets. It can be seen that when $k = 40$, the OA achieves the best performance. When smaller k is selected, the dropped features can hardly achieve the goal of mitigating overfitting. When it gets bigger, the network dropped too many features, and the network inclines to learn incorrect representations provided by the irrelevant background. Therefore, k is chosen as 40 in our following experiments.

3) *Analysis of the Size of the Patch*: The size of the input image patch is an important parameter. As illustrated in Fig. 10, the input image patches are set to 11, 15, 19, 23, 27, and 31, respectively. It can be observed that the classification accuracy sharply increases when the patch size ranges from 11 to 27 on three data sets. When patch size grows larger than 27, the classification accuracy tends to decrease. It is due to the reason that a larger image patch takes pixels of different classes into account, and hence, some negative effects are incurred. In the meanwhile, valuable spatial information is not exploited effectively when the patch size is rather small. Therefore, the extracted features are not representative of the

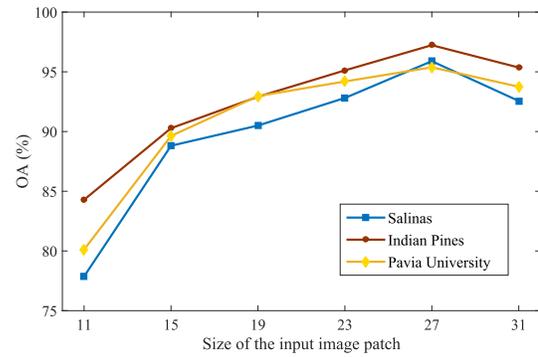


Fig. 10. Classification performance versus different input image patch sizes on three data sets.

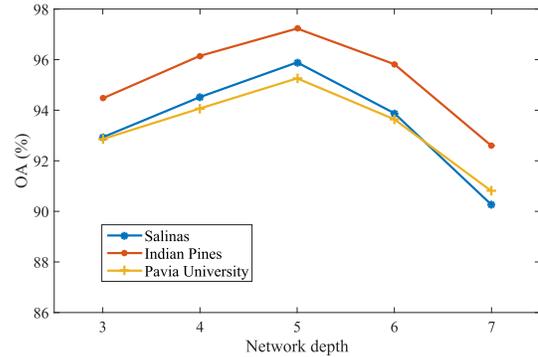


Fig. 11. Classification performance versus network depths on three data sets.

central pixel. Therefore, in our implementations, the input image patch size is set to 27×27 pixels on three data sets.

4) *Analysis of Network Depth*: It is widely acknowledged that the network depth of current deep learning-based methods is getting deeper and deeper. However, when training samples are relatively limited, the parameters in deeper models can hardly be optimized, and the model is unable to work well. As illustrated in Fig. 11, when the network depth is set to 5, the best performance is obtained on three data sets. It is reasonable that deeper architectures may suffer from the exploding gradients problem. Especially, error gradients accumulate quickly and, thus, result in an unstable network. Hence, the network depth is set to be 5.

C. Comparison With Different Regularization Methods

In this section, we empirically investigate the effectiveness of the proposed AdapDrop for HSI classification. Dropout and DropBlock are employed for comparison in extensive experiments. As shown in Fig. 12, the proposed AdapDrop has a superior performance compared with dropout and DropBlock. Among these methods, dropout is less effective since it randomly drops separate pixels in the feature map, and the dropped information can be easily retrieved through neighborhood pixels. DropBlock removes the entire blocks, and the network's learning capabilities may be affected. The proposed AdapDrop removes highly informative regions in the feature map, and the network can effectively learn robust features of the ground objects in HSI classification.

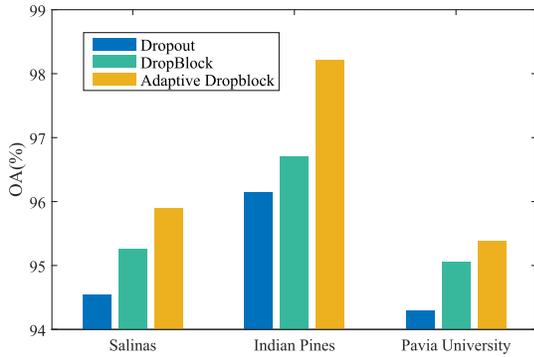


Fig. 12. Classification accuracy by employing different regularization methods on three data sets.

D. Classification Results on Three Data Sets

In order to verify the effectiveness of the proposed ADGAN, we compare it with five closely related methods, such as the random forest (RF) [6], contextual SVM (CSVM) [50], CNN with extinction profiles (EP-CNN) [51], spectral-spatial ResNet (SS-ResNet) [52], and 3-D-ACGAN [39]. In order to ensure a fair comparison, all the methods use default parameters and the same proportion of training sets. All the experimental results are obtained by running ten times independently with a random division for training and test sets.

RF investigates an RF of binary classifiers as a means of increasing the diversity of hierarchical classifiers. N_f is set to be 20, and 100 trees are grown for each experiment. For CSVM, both local spectral information and spatial information in a reproducing kernel Hilbert space are jointly exploited. A neighborhood of 9×9 pixels is employed, and default parameters of SVM are used as mentioned in [50]. EP-CNN fuses the hyperspectral and light detection and ranging-derived data using extinction profiles and deep learning. A neighborhood with a size of 27×27 pixels is considered. $\alpha = 3$ and $s = 7$ are employed as described in [51]. The SS-ResNet combines spatial information and spectral information, and it takes advantage of residual learning. A neighborhood with a size of 11×11 is employed. In addition, 300 epochs and the Adam optimizer are used. For 3-D-ACGAN, the source code provided by Prof. Chen is used, and default parameters are chosen as mentioned in [39]. Especially, 64×64 neighborhood of each pixel is used, and the input images are normalized into the range $[-0.5, 0.5]$. The size of the minibatch is 100, and Adam optimizer is employed. For data preprocessing, three components are utilized as inputs. The generator G and discriminator D are designed with five convolutional layers. The size of the input noise is $100 \times 1 \times 1$, and the generator converts the inputs to fake samples with a size of $64 \times 64 \times 3$. In order to fairly compare the proposed ADGAN with 3-D-ACGAN, both methods have similar architectures except for the output of the discriminator.

Both visual and quantitative analyses are provided in our experiments. For visual analysis, the classification maps generated by different methods are illustrated in figure form. For

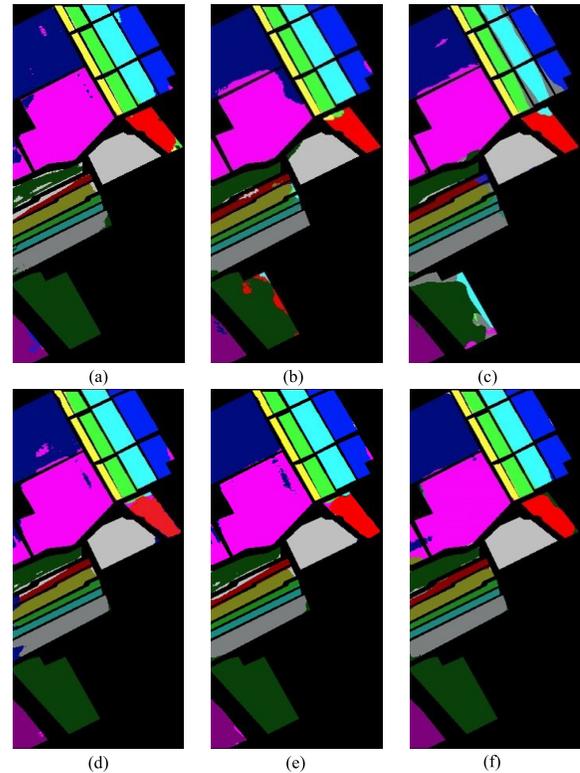


Fig. 13. Visualized results of different classification methods on the Salinas data set. (a) Result by RF [6]. (b) Result by CSVM [50]. (c) Result by EP-CNN [51]. (d) Result by SS-ResNet [52]. (e) Result by 3-D-ACGAN [39]. (f) Result by the proposed ADGAN.

quantitative analysis, the classification maps are illustrated in tabular form.

1) *Results on the Salinas Data Set:* Table V lists the corresponding evaluation criteria of six algorithms. The first 16 rows illustrate the results of each class, and the last three rows show the OA, AA, and Kappa coefficients. The best classification results are emphasized by bolding. As shown in Table V, deep learning-based methods, EP-CNN, SS-ResNet, 3-D-ACGAN, and ADGAN, are superior to RF and CSVM because of the hierarchical nonlinear feature extraction. Compared with EP-CNN and SS-ResNet, 3-D-ACGAN improves the classification performance with the assistance of generated samples. Among the six methods, ADGAN achieves the best classification results in most cases since it not only generated high-quality samples but also alleviated the drawback of ACGAN as mentioned earlier. In addition, compared with other methods, ADGAN improves at least 1.11% in OA, 0.79% in AA, and 0.41% in Kappa. Fig. 13 shows the classification maps of different methods on the Salinas data set. As illustrated in Fig. 13(a)–(d), RF, CSVM, EP-CNN, and SS-ResNet misclassify many samples at the boundary of different classes. Compared with these methods, 3-D-ACGAN achieves better classification results on majority classes because of data augmentation. Compared with 3-D-ACGAN, the proposed ADGAN performs better in minority classes, for example, *Lettuce_roumaine_4wk*, *Lettuce_roumaine_6wk*, and *Lettuce_roumaine_7wk*. It is

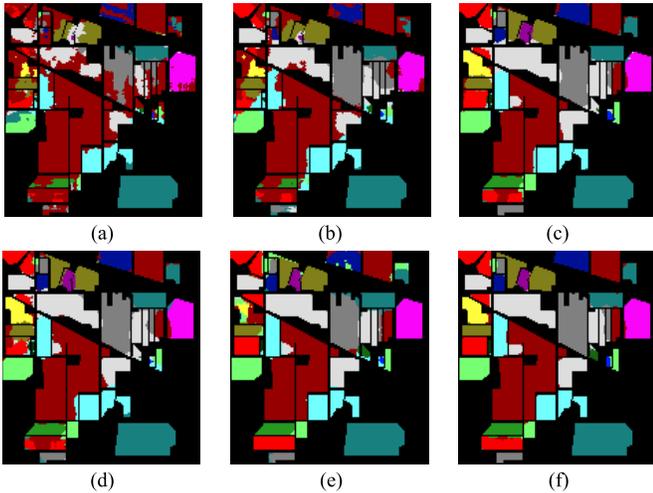


Fig. 14. Visualized results of different classification methods on the Indian Pines data set. (a) Result by RF [6]. (b) Result by CSVM [50]. (c) Result by EP-CNN [51]. (d) Result by SS-ResNet [52]. (e) Result by 3-D-ACGAN [39]. (f) Result by the proposed ADGAN.

demonstrated that the proposed ADGAN achieves the best performance on the Salinas data set.

2) *Results on the Indian Pines Data Set:* The statistical classification results on the Indian Pines data set are summarized in Table VI, and Fig. 14 illustrates the classification results of different methods. As can be observed from Table VI, SS-ResNet, 3-D-ACGAN, and ADGAN are superior to RF, CSVM, and EP-CNN by introducing attention mechanism or extra generated training samples. For minority classes, such as Alfalfa, Grass-pasture-mowed, Oats, and Stone-Steel-Towers, the classification performance of the proposed ADGAN is better than 3-D-ACGAN. It is demonstrated that ADGAN has better classification performance when handling minority class samples on this data set. Among all these methods, ADGAN obtains the best statistical results in terms of the OA, AA, and Kappa. As shown in Fig. 14(a)–(d), many samples belonging to the Soybean-clean and Building-Grass-Trees are falsely assigned the neighboring labels by RF, CSVM, EP-CNN, and SS-ResNet. Compared with them, ADGAN achieves better region uniformity in the Soybean-notill class. Moreover, ADGAN obtains better performance in the boundary pixel classification of the Stone-Steel-Towers, which is limited in the training set. It is evident that the proposed ADGAN obtains the best performance on the Indian Pines data set.

3) *Results on the Pavia University Data Set:* The quantitative criteria of different methods on the Pavia University data set are shown in Table VII. The corresponding classification maps on the data set are illustrated in Fig. 15. As can be observed in Table VII, the Gravel and Self-Blocking Bricks classes are misclassified by RF, CSVM, EP-CNN, and SS-ResNet. Compared with these methods, 3-D-ACGAN and ADGAN obviously improve the classification performance by generating high-quality training samples. When handling the minority classes, such as Painted metal sheets, Bitumen, and Shadows, the proposed ADGAN performs better than 3-D-ACGAN. From visual comparisons, the proposed

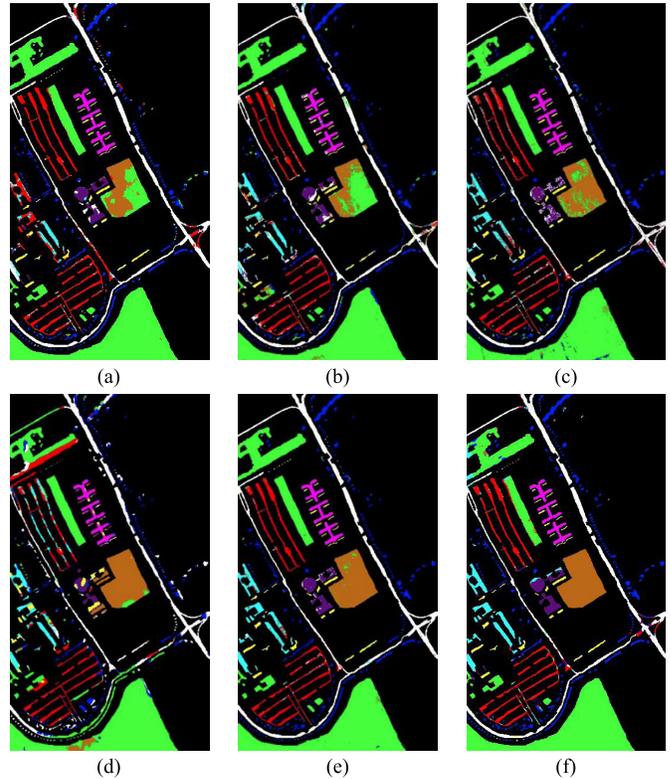


Fig. 15. Visualized results of different classification methods on the Pavia University data set. (a) Result by RF [6]. (b) Result by CSVM [50]. (c) Result by EP-CNN [51]. (d) Result by SS-ResNet [52]. (e) Result by 3-D-ACGAN [39]. (f) Result by the proposed ADGAN.

ADGAN obtains the best classification results. The proposed ADGAN surpasses 3-D-ACGAN by 1.13%, 1.20%, and 1.25% in terms of OA, AA, and Kappa. As shown in Fig. 15(a)–(d), there are many noisy scattered points in the Bare soil and Gravel in the classification results by RF, CSVM, EP-CNN, and SS-ResNet. Compared with them, 3-D-ACGAN and ADGAN provide better results with little noise. It should be noted that, when handling the minority classes, such as Bitumen and Shadows, ADGAN performs better and is the closest to the ground truth map. The experimental results on this data set demonstrate that the ADGAN exhibits good classification performance by capturing the intrinsic interclass discriminative features.

From visual comparisons, the classification results by the proposed ADGAN are less noisy than the other methods. The quantitative criteria in Tables V–VII are consistent with the visual comparisons. It should be noted that deep learning-based methods generally perform better than shallow architectures. Especially, the GAN-based methods indeed obtain better classification results when the training samples are limited. The proposed ADGAN is capable of achieving better classification accuracy than 3-D-ACGAN in minority class classification due to the newly designed discriminator and AdapDrop.

E. Investigation on Running Time

Table VIII lists the running time of different classification methods on three data sets. Compared with RF and CSVM,

TABLE V
CLASSIFICATION RESULTS OBTAINED BY DIFFERENT CLASSIFICATION METHODS ON THE SALINAS DATA SET

Class	RF	CSVM	EP-CNN	SS-ResNet	3D-ACGAN	ADGAN
Brocoli_green_weeds_1	89.64±0.41	91.46±0.40	92.96±0.30	94.13±0.47	94.82±0.42	95.95±0.44
Brocoli_green_weeds_2	89.70±0.35	91.73±0.78	91.57±0.32	93.46±0.51	94.19±0.54	95.86±0.51
Fallow	91.82±0.12	93.70±0.19	93.20±0.16	95.32±0.48	96.04±0.41	95.01±0.35
Fallow_rough_plow	85.46±0.52	87.84±0.33	89.23±0.25	91.89±0.20	92.61±0.36	93.32±0.27
Fallow_smooth	88.90±0.38	90.52±0.27	93.61±0.47	95.16±0.61	95.87±0.58	96.92±0.49
Stubble	87.17±0.60	89.05±0.56	89.64±0.47	91.28±0.72	92.92±0.75	93.86±0.20
Celery	90.24±0.29	92.18±0.17	93.52±0.16	94.66±0.29	95.36±0.35	96.21±0.13
Grapes_untrained	87.63±0.93	89.67±0.40	94.07±0.33	95.12±0.51	96.02±0.68	96.37±0.49
Soil_vinyard_develop	90.72±0.37	92.82±0.53	93.20±0.18	94.16±0.12	95.33±0.39	96.50±0.21
Corn_senesced_green_weeds	85.86±1.09	88.08±0.84	88.13±0.94	89.31±0.93	92.05±0.94	92.25±0.87
Lettuce_roumaine_4wk	86.94±0.89	89.25±0.39	90.08±0.13	91.80±0.43	92.56±0.46	93.68±0.36
Lettuce_roumaine_5wk	89.31±0.83	91.37±0.72	92.59±0.58	93.22±0.22	93.91±0.44	95.07±0.28
Lettuce_roumaine_6wk	88.50±0.92	90.40±0.98	91.52±0.80	92.74±0.39	93.62±0.29	94.23±0.35
Lettuce_roumaine_7wk	89.59±1.77	91.68±1.34	92.89±0.82	94.65±0.52	91.13±0.81	93.02±0.49
Vinyard_untrained	87.41±2.51	89.24±1.26	90.08±0.57	91.23±0.56	95.84±0.73	95.68±0.53
Vinyard_vertical_trellis	90.53±0.84	90.89±0.58	93.57±0.26	94.02±0.14	94.30±0.58	95.26±0.41
OA(%)	89.12±0.43	91.64±0.35	92.86±0.22	94.06±0.87	94.78±1.24	95.89±0.54
AA(%)	88.71±0.51	90.62±0.40	91.86±0.17	93.26±0.41	94.16±0.49	94.95±0.35
$\kappa \times 100$	88.84±0.58	90.82±0.47	92.25±0.24	93.38±0.37	94.50±0.62	94.91±0.56

TABLE VI
CLASSIFICATION RESULTS OBTAINED BY DIFFERENT CLASSIFICATION METHODS ON THE INDIAN PINES DATA SET

Class	RF	CSVM	EP-CNN	SS-ResNet	3D-ACGAN	ADGAN
Alfalfa	31.20±8.09	68.05±9.06	85.38±3.60	84.65±3.62	92.68±0.28	93.46±1.29
Corn-notill	76.99±1.69	85.45±1.15	91.45±0.76	94.37±1.48	96.74±0.19	97.38±0.83
Corn-min	68.35±2.16	79.56±2.40	88.69±3.59	95.24±3.16	97.81±0.36	99.82±0.06
Corn	54.63±1.95	83.93±4.93	89.56±2.02	84.51±5.83	94.69±1.79	95.65±1.62
Grass-pasture	86.98±4.29	93.17±2.26	88.05±2.40	96.72±0.55	97.26±0.88	98.37±0.75
Grass-trees	90.94±0.93	97.26±0.21	96.13±1.92	97.98±0.97	94.95±2.42	95.63±2.38
Grass-pasture-mowed	37.69±9.74	84.48±3.48	92.61±2.57	86.31±5.22	94.72±0.99	98.06±0.60
Hay-windrowed	89.92±2.98	98.32±0.66	97.80±0.40	98.69±0.11	98.43±0.79	99.54±0.12
Oats	58.64±8.35	65.68±6.86	70.88±7.18	72.24±6.98	93.97±2.89	95.80±2.98
Soybean-notill	66.57±2.63	79.87±3.28	89.85±2.91	94.65±1.65	96.96±1.91	97.83±0.51
Soybean-mintill	79.59±3.07	87.72±1.74	95.28±1.45	97.57±0.77	96.81±1.45	98.71±0.36
Soybean-clean	68.33±4.62	82.99±0.99	88.69±2.04	90.38±3.77	95.73±1.01	96.24±1.05
Wheat	93.59±2.68	98.39±1.23	97.86±1.42	99.81±0.02	96.40±0.97	97.08±0.51
Woods	91.76±0.59	95.54±0.54	98.40±0.57	99.14±0.31	95.80±1.86	97.96±0.82
Buildings-Grass-Trees	69.30±5.01	79.28±3.45	89.91±6.02	93.54±3.53	97.99±1.02	98.79±0.79
Stone-Steel-Towers	85.68±1.63	87.60±5.74	88.53±6.27	89.96±6.46	95.12±2.53	96.53±1.68
OA(%)	77.76±0.48	86.48±0.23	92.65±0.53	95.84±1.32	96.31±0.51	97.23±0.41
AA(%)	71.89±0.98	85.46±1.44	90.57±2.01	92.23±2.27	96.00±0.36	97.30±1.08
$\kappa \times 100$	74.65±0.54	85.62±0.28	91.75±0.60	95.21±1.51	96.24±0.60	96.57±0.60

deep learning-based methods cost more training time because of the construction of a deep network. 3-D-ACGAN and ADGAN are time-consuming on the training time because adversarial learning needs more time to converge. For the

test time, the proposed ADGAN has an obvious advantage than EP-CNN and SS-ResNet because of the simpler network structure of the discriminator. Furthermore, we can observe that ADGAN has a competitive performance compared with

TABLE VII
CLASSIFICATION RESULTS OBTAINED BY DIFFERENT CLASSIFICATION METHODS ON THE PAVIA UNIVERSITY DATA SET

Class	RF	CSVM	EP-CNN	SS-ResNet	3D-ACGAN	ADGAN
Asphalt	87.28±0.58	90.75±0.47	94.27±0.52	94.85±0.27	93.73±0.62	92.36±1.33
Meadows	93.16±0.36	93.88±0.12	96.61±0.41	97.26±0.13	98.31±0.41	96.18±0.45
Gravel	63.02±2.73	76.24±1.30	79.92±1.98	80.42±1.02	88.33±1.87	91.49±1.45
Trees	85.65±0.25	90.89±0.54	92.49±0.53	93.20±0.40	96.94±0.71	97.29±0.25
Painted metal sheets	94.60±0.39	98.85±0.31	98.54±0.34	98.35±0.13	92.22±0.36	98.54±0.16
Bare soil	61.38±1.37	85.59±0.71	90.65±0.48	90.76±0.52	96.24±1.12	96.37±0.27
Bitumen	72.69±1.29	83.32±0.12	86.52±1.66	86.69±0.83	85.58±2.10	88.40±1.54
Self-Blocking Bricks	84.51±0.77	84.87±0.48	84.55±0.89	84.82±1.06	97.65±1.14	95.75±1.25
Shadows	95.74±0.54	95.83±0.29	98.06±0.32	98.71±0.26	92.56±0.32	96.02±0.26
OA(%)	85.37±0.51	90.32±0.30	93.07±0.21	93.69±0.36	94.25±1.24	95.38±0.54
AA(%)	82.00±1.29	88.91±0.26	91.29±0.30	91.67±0.22	93.51±1.40	94.71±0.35
$\kappa \times 100$	83.43±0.62	88.93±0.14	92.73±0.28	92.61±0.29	93.96±1.26	95.21±0.41

TABLE VIII
RUNNING TIME OF DIFFERENT CLASSIFICATION METHODS ON THREE DATA SETS

Training time (s)			
Methods	Salinas	Indian Pines	Pavia University
RF	3.34	2.13	3.15
CSVM	13.52	10.92	12.92
EP-CNN	271.64	243.29	251.68
SS-ResNet	277.73	229.04	254.72
3D-ACGAN	395.31	358.34	379.16
ADGAN	408.19	367.84	383.13
Test time (s)			
Methods	Salinas	Indian Pines	Pavia University
RF	1.53	1.12	1.62
CSVM	3.66	3.45	3.96
EP-CNN	7.14	5.53	7.43
SS-ResNet	13.82	9.65	11.10
3D-ACGAN	7.39	6.11	7.12
ADGAN	6.91	6.23	6.95

3-D-ACGAN in test time. It means that the proposed ADGAN is capable of real-time applications.

F. Visualization of Adversarial Samples

In the visualization experiment, some representative fake samples generated by the 3-D-ACGAN and the proposed ADGAN on the Salinas data set are illustrated in Fig. 16. As can be observed, the proposed ADGAN can generate high-quality samples that have similar structures compared with the real image samples. On the contrary, 3-D-ACGAN sometimes fails to generate good samples for the minority class and collapse toward learning the basic structures of the real samples. As mentioned earlier, because of the self-contradiction in 3-D-ACGAN's discriminator, the generator G prefers to generate samples belonging to the majority

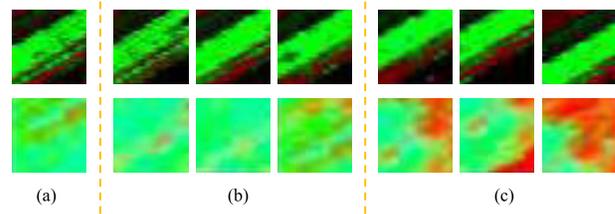


Fig. 16. Representative samples generated for the minority classes on the Salinas data set. (a) Real samples. (b) ADGAN. (c) 3-D-ACGAN.

class. Therefore, the performance of 3-D-ACGAN in generating minority class samples is affected. The proposed ADGAN models the classification task and the discrimination task into one single objective. Hence, the mode collapse issue can be alleviated to some extent. The proposed ADGAN is superior in adversarial learning when aiming at the generation of minority class samples and can be employed to improve the HSI classification accuracy.

V. CONCLUSION

In this article, an AdapDrop-enhanced framework for HSI classification is proposed. The proposed ADGAN can effectively alleviate the following two problems: 1) the imbalanced training data in HSI and 2) the mode collapse problem in GAN-based classification methods. First, the discriminator is adjusted to be a single output that returns either the fake label or the specific class label. The discriminator will not contradict itself when training samples are imbalanced. Second, AdapDrop is proposed as a regularization method to mitigate the mode collapse problem. Instead of dropping a fixed size region, the proposed AdapDrop generates drop masks with adaptive shapes, which can better deal with ground objects with various shapes. To evaluate the proposed framework, extensive experiments are performed on three hyperspectral data sets. The results show that the proposed ADGAN can achieve better performance compared with the state-of-the-art baselines.

In the future, we plan to extend our work in two directions. First, several self-attention networks will be investigated to improve classification performance. In addition, more regularization techniques will be explored to alleviate the mode collapse problem and, therefore, further enhance the classification performance.

ACKNOWLEDGMENT

The authors would like to thank Prof. Y. Chen for sharing the source code of 3-D-ACGAN. They would also like to thank the Associate Editor and all the reviewers for their valuable comments and suggestions that have significantly improved the quality of this article.

REFERENCES

- [1] B. Tu, X. Zhang, X. Kang, J. Wang, and J. A. Benediktsson, "Spatial density peak clustering for hyperspectral image classification with noisy labels," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 5085–5097, Jul. 2019.
- [2] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 480–491, Mar. 2005.
- [3] Q. Wang, Z. Yuan, and X. Li, "GETNET: A general end-to-end two-dimensional CNN framework for hyperspectral image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 3–13, Jan. 2019.
- [4] P. Arellano, K. Tansey, H. Balzter, and D. S. Boyd, "Detecting the effects of hydrocarbon pollution in the amazon forest using hyperspectral satellite images," *Environ. Pollut.*, vol. 205, pp. 225–239, Oct. 2015.
- [5] Q. Wang, Y. Gu, and D. Tuija, "Discriminative multiple kernel learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 7, pp. 3912–3927, Jul. 2016.
- [6] 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.
- [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] M. Fauvel, J. Chanussot, and A. J. Benediktsson, "Evaluation of kernels for multiclass classification of hyperspectral remote sensing data," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, May 2006, p. II, doi: 10.1109/ICASSP.2006.1660467.
- [9] H. Zhang, Y. Li, Y. Jiang, P. Wang, Q. Shen, and C. Shen, "Hyperspectral classification based on lightweight 3-D-CNN with transfer learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5813–5828, Aug. 2019.
- [10] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 11, pp. 3804–3814, Nov. 2008.
- [11] W. Li, C. Chen, H. Su, and Q. Du, "Local binary patterns and extreme learning machine for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3681–3693, Jul. 2015.
- [12] J. Zou, W. Li, and Q. Du, "Sparse representation-based nearest neighbor classifiers for hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2418–2422, Dec. 2015.
- [13] B. Pan, Z. Shi, and X. Xu, "Hierarchical guidance filtering-based ensemble classification for hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 4177–4189, Jul. 2017.
- [14] L. Fang, S. Li, W. Duan, J. Ren, and J. A. Benediktsson, "Classification of hyperspectral images by exploiting spectral-spatial information of superpixel via multiple kernels," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 12, pp. 6663–6674, Dec. 2015.
- [15] M. Fauvel, J. Chanussot, and J. A. Benediktsson, "A spatial-spectral kernel-based approach for the classification of remote-sensing images," *Pattern Recognit.*, vol. 45, no. 1, pp. 381–392, 2012.
- [16] X. Kang, S. Li, and J. Atli Benediktsson, "Spectral-spatial hyperspectral image classification with edge-preserving filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2666–2677, May 2014.
- [17] L. Fang, N. He, S. Li, P. Ghamisi, and J. A. Benediktsson, "Extinction profiles fusion for hyperspectral images classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1803–1815, Mar. 2018.
- [18] S. Li, T. Lu, L. Fang, X. Jia, and J. A. Benediktsson, "Probabilistic fusion of pixel-level and superpixel-level hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7416–7430, Dec. 2016.
- [19] Y. Zhong, A. Ma, and L. Zhang, "An adaptive memetic fuzzy clustering algorithm with spatial information for remote sensing imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 4, pp. 1235–1248, Apr. 2014.
- [20] Y. Y. Tang, Y. Lu, and H. Yuan, "Hyperspectral image classification based on three-dimensional scattering wavelet transform," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2467–2480, May 2015.
- [21] W. Li and Q. Du, "Collaborative representation for hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1463–1474, Mar. 2015.
- [22] C. Chen, W. Li, E. W. Tramel, M. Cui, S. Prasad, and J. E. Fowler, "Spectral-spatial preprocessing using multihypothesis prediction for noise-robust hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 4, pp. 1047–1059, Apr. 2014.
- [23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [24] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
- [25] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.
- [26] K. Yao, G. Zweig, M.-Y. Hwang, Y. Shi, and D. Yu, "Recurrent neural networks for language understanding," in *Proc. Interspeech.*, 2013, pp. 2524–2528.
- [27] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.
- [28] 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.
- [29] B. Pan, Z. Shi, and X. Xu, "R-VCANet: A new deep-learning-based hyperspectral image classification method," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 5, pp. 1975–1986, May 2017.
- [30] H. Zhang, Y. Li, Y. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network," *Remote Sens. Lett.*, vol. 8, no. 5, pp. 438–447, May 2017.
- [31] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.
- [32] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral-spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2018.
- [33] Y. Chen, L. Zhu, P. Ghamisi, X. Jia, G. Li, and L. Tang, "Hyperspectral images classification with Gabor filtering and convolutional neural network," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2355–2359, Dec. 2017.
- [34] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
- [35] Z. Gong, P. Zhong, Y. Yu, W. Hu, and S. Li, "A CNN with multiscale convolution and diversified metric for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3599–3618, Jun. 2019.
- [36] X. Ma, A. Fu, J. Wang, H. Wang, and B. Yin, "Hyperspectral image classification based on deep deconvolution network with skip architecture," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4781–4791, Aug. 2018.
- [37] I. Goodfellow *et al.*, "Generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
- [38] 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.
- [39] 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.

- [40] 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.
- [41] 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, doi: [10.1109/TCYB.2019.2915094](https://doi.org/10.1109/TCYB.2019.2915094).
- [42] J. Li, Q. Du, Y. Li, and W. Li, "Hyperspectral image classification with imbalanced data based on orthogonal complement subspace projection," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 7, pp. 3838–3851, Jul. 2018.
- [43] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2642–2651.
- [44] T. Chavdarova and F. Fleuret, "SGAN: An alternative training of generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9407–9415.
- [45] G. Ghiasi, T. Lin, and Q. V. Le, "Dropblock: A regularization method for convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 10727–10737.
- [46] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [47] G. Licciardi, P. R. Marpu, J. Chanussot, and J. A. Benediktsson, "Linear versus nonlinear PCA for the classification of hyperspectral data based on the extended morphological profiles," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 3, pp. 447–451, May 2012.
- [48] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8697–8710.
- [49] L. Wan, M. D. Zeiler, S. Zhang, Y. LeCun, and R. Fergus, "Regularization of neural networks using dropconnect," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1058–1066.
- [50] P. Gurram and H. Kwon, "Contextual SVM using Hilbert space embedding for hyperspectral classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 5, pp. 1031–1035, Sep. 2013.
- [51] P. Ghamisi, B. Hofle, and X. X. Zhu, "Hyperspectral and LiDAR data fusion using extinction profiles and deep convolutional neural network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 6, pp. 3011–3024, Jun. 2017.
- [52] J. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza, and J. Li, "Visual attention-driven hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 8065–8080, Oct. 2019.



Junjie Wang received the B.Sc. degree in computer science from the Ocean University of China, Qingdao, China, in 2018, where he is pursuing the M.Sc. degree in computer science and applied remote sensing with the School of Information Science and Technology.

His research interests include computer vision and remote sensing image processing.



Feng Gao (Member, IEEE) received the B.Sc. degree in software engineering from Chongqing University, Chongqing, China, in 2008, and the Ph.D. degree in computer science and technology from Beihang University, Beijing, China, in 2015.

He is an Associate Professor with the School of Information Science and Engineering, Ocean University of China, Qingdao, China. His research interests include remote sensing image analysis, pattern recognition, and machine learning.



Junyu Dong (Member, IEEE) received the B.Sc. and M.Sc. degrees from the Department of Applied Mathematics, Ocean University of China, Qingdao, China, in 1993 and 1999, respectively, and the Ph.D. degree in image processing from the Department of Computer Science, Heriot-Watt University, Edinburgh, U.K., in 2003.

He is a Professor and the Vice-Dean of the School of Information Science and Engineering, Ocean University of China. His research interests include visual information analysis and understanding, machine learning, and underwater image processing.



Qian Du (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of Maryland at Baltimore, Baltimore, MD, USA, in 2000.

She is the Bobby Shackouls Professor with the Department of Electrical and Computer Engineering, Mississippi State University, Starkville, MS, USA. Her research interests include hyperspectral remote sensing image analysis and applications, pattern classification, data compression, and neural networks.

Dr. Du has served as the Co-Chair of the Data Fusion Technical Committee of the IEEE Geoscience and Remote Sensing Society (GRSS) from 2009 to 2013 and the Chair of the Remote Sensing and Mapping Technical Committee of the International Association for Pattern Recognition (IAPR) from 2010 to 2014. She also serves as the Chief Editor of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING.