Generative Adversarial Networks for Hyperspectral Image Classification

Article *in* IEEE Transactions on Geoscience and Remote Sensing · February 2018
DOI: 10.1109/TGRS.2018.2805286

CITATIONS
2

READS
1,117

4 authors:

Zhu Lin
Harbin Institute of Technology
9 PUBLICATIONS **5** CITATIONS
SEE PROFILE

Yushi Chen
Harbin Institute of Technology
50 PUBLICATIONS **889** CITATIONS
SEE PROFILE

Pedram Ghamisi
Helmholtz-Zentrum Dresden-Rossendorf
92 PUBLICATIONS **1,316** CITATIONS
SEE PROFILE

Jon Atli Benediktsson
University of Iceland
472 PUBLICATIONS **17,394** CITATIONS
SEE PROFILE

Generative Adversarial Networks for Hyperspectral Image Classification

Lin Zhu, Yushi Chen, *Member, IEEE*, Pedram Ghamisi, *Member, IEEE*, and Jón Atli Benediktsson, *Fellow, IEEE*

***Abstract*— A generative adversarial network (GAN) usually contains a generative network and a discriminative network in competition with each other. The GAN has shown its capability in a variety of applications. In this paper, the usefulness and effectiveness of GAN for classification of hyperspectral images (HSIs) are explored for the first time. In the proposed GAN, a convolutional neural network (CNN) is designed to discriminate the inputs and another CNN is used to generate so-called fake inputs. The aforementioned CNNs are trained together: the generative CNN tries to generate fake inputs that are as real as possible, and the discriminative CNN tries to classify the real and fake inputs. This kind of adversarial training improves the generalization capability of the discriminative CNN, which is really important when the training samples are limited. Specifically, we propose two schemes: 1) a well-designed 1D-GAN as a spectral classifier and 2) a robust 3D-GAN as a spectral–spatial classifier. Furthermore, the generated adversarial samples are used with real training samples to fine-tune the discriminative CNN, which improves the final classification performance. The proposed classifiers are carried out on three widely used hyperspectral data sets: Salinas, Indiana Pines, and Kennedy Space Center. The obtained results reveal that the proposed models provide competitive results compared to the state-of-the-art methods. In addition, the proposed GANs open new opportunities in the remote sensing community for the challenging task of HSI classification and also reveal the huge potential of GAN-based methods for the analysis of such complex and inherently nonlinear data.**

***Index Terms*— Convolutional neural network (CNN), deep learning, generative adversarial network (GAN), hyperspectral image (HSI) classification.**

I. Introduction

HYPERSPECTRAL sensors simultaneously capture the spatial and spectral information of the observing target. Such data are characterized as a valuable source of information for earth observation, which brings both challenges and opportunities to develop new data processing techniques [1]. The classification of hyperspectral images (HSIs), which is a crucial step for the plethora of applications, tries to assign a specific class to each pixel in the scene. HSI classification is widely used, e.g., in urban development, land change monitoring, scene interpretation, and resource management. Among hyperspectral data processing techniques, classification is one of the most vibrant topics in the remote sensing community [2].

Most of the supervised methods that have been applied in the machine learning community have been explored for HSI classification in the last few decades. In general, there are two kinds of data classification methods: spectral classifiers and spectral–spatial classifiers.

Hyperspectral data usually contain hundreds of spectral channels of the same scene, which provide abundant spectral information. Traditional spectral classification algorithms typically include K nearest neighbors (KNN), maximum likelihood, neural network, and logistic regression [3]. Most of those algorithms dramatically suffer from the so-called curse of dimensionality (Hughes phenomenon) [4]. Support vector machines (SVMs) have shown their low sensitivity to high dimensionality with respect to the number of training samples and are unlikely to suffer from the Hughes phenomenon. Furthermore, SVM-based classifiers usually obtain good classification performance in terms of when a limited number of training samples is available compared with other widely used supervised techniques [5], [6]. During the first decade of this century, SVM-based spectral classifiers were considered as the state-of-the-art methods in the hyperspectral community.

With the advancement of the sensor and imaging systems, the spatial resolution of hyperspectral data is becoming finer and finer. With the help of spatial information, which can be extracted by diverse methodologies such as filtering or segmentation approaches, the classification performance can be significantly improved. Among those approaches, morphological profiles have been widely used to extract the spatial features of HSIs (usually followed by SVMs or random forest to obtain the resulting classification map) [7]. Many extensions have been developed for hyperspectral data feature extraction based on morphological operators, which demonstrate that the morphological profile is still under further development [8]. On the other hand, multiple kernel learning has been applied

Manuscript received November 22, 2017; revised January 13, 2018; accepted February 6, 2018. This work was supported in part by the Natural Science Foundation of China under Grant 61771171 and in part by the National Natural Science Foundation of Key International Cooperation under Grant 61720106002. *(Corresponding author: Yushi Chen.)*

L. Zhu and Y. Chen are with the School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China (e-mail: 17s105139@stu.hit.edu.cn; chenyushi@hit.edu.cn).

P. Ghamisi is with the German Aerospace Center (DLR), Remote Sensing Technology Institute (IMF), 82234 Wessling, Germany, and also with the Signal Processing in Earth Observation, Technical University of Munich, 80333 Munich, Germany (e-mail: p.ghamisi@gmail.com).

J. A. Benediktsson is with the Faculty of Electrical and Computer Engineering, University of Iceland, IS-107 Reykjavik, Iceland (e-mail: benedikt@hi.is).

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

Digital Object Identifier 10.1109/TGRS.2018.2805286

to hyperspectral data classification because of its powerful capability to handle heterogeneous spectral and spatial features in an efficient manner [9]. Spectral–spatial classification approaches are regarded as the mainstream of HSI classification. Many methods have been proposed in recent years on the spectral–spatial classification of hyperspectral data. As an example, in [10], a method, which combines sparse representation and Markov random fields, was proposed. The method utilizes Markov random fields to explore spatial correlation, which significantly improves the classification performance. Furthermore, a new spectral–spatial classifier based on discriminant analysis has been proposed for HSI classification in [11]. The proposed method learned a representative subspace from the spectral and spatial domains and produced a good classification performance.

Most of the traditional spectral and spectral–spatial classifiers do not classify the hyperspectral data in a "deep" manner. In [12], it is demonstrated that the classifiers like linear SVM and logistic regression can be attributed to single-layer classifiers, while decision tree or SVM with kernels is believed to have two layers. Deep models, which contain two or more hidden layers, tend to extract the invariant and discriminant features of the input data. Deep learning methods have been actively studied in a wide variety of research areas such as image classification, language processing, and speech recognition in recent years [13], [14]. Recently, deep learning has also been utilized for remote sensing data processing including HSI classification. A survey of deep learning-based approaches for remote sensing data can be found in [15].

In 2014, a deep learning-based method, stacked autoencoders, has been proposed for hyperspectral feature extraction and classification [16]. Deep belief network, another kind of deep models, was introduced and modified for HSI classification as well [17], [18].

In recent years, lots of deep models, especially deep convolutional neural networks (CNNs), have been proposed in the remote sensing community. In [19], a deep CNN with five layers was employed to extract the spectral features of HSIs leading to a promising classification performance. After that, in [20], a novel pixel-pair method is proposed as a deep spectral classifier, and the model achieved good performance when the number of training samples was not sufficient.

The aforementioned methods in [19] and [20] are spectral classifiers. In order to utilize the spatial and spectral information of HSIs simultaneously, some spectral–spatial classifiers based on deep CNNs have been proposed [21], [22]. For example, in [22], a framework based on principal component analysis (PCA), deep CNN, and logistic regression was used for HSI classification. HSIs are inherently 3-D data, so it is reasonable to design 3-D CNNs to effectively extract the spectral–spatial features of HSIs [23], [24]. On the other hand, CNNs can be combined with other techniques such as sparse representation approach and Gabor filtering. In [25], CNN is followed by sparse representation to refine the learned features. Recently, a method based on the combination of Gabor filtering and CNN was introduced to extract the features of HSIs, which leads to a performance improvement [26].

Furthermore, there are new types of deep models which have also been investigated for HSI classification. Very recently, a new deep dictionary learning has been proposed for HSI classification, which has shown its capability under the condition of limited training samples [27].

Although great progress has been achieved in HSI classification using deep learning-based methods, deep models usually face a serious problem known as overfitting. The reason behind this problem is that deep learning-based methods need to train a large number of learnable parameters, which requires a lot of training samples. The problem becomes serious when the number of training samples is limited. Unfortunately, the limited availability of training samples is a common situation in the remote sensing community since the collection of such data is either time demanding or expensive. Deep models are often over-trained if there are not sufficient training samples available, which means that the performance is good in the training stage but relatively poor in the test stage. Therefore, new and effective training strategies for deep models are needed to address the issue of overfitting. [The generative adversarial network (GAN), which this paper investigates, is one of those strategies.] Actually, the generator network of GAN can be regarded as a regularization method, which can mitigate the overfitting phenomena to a great extent.

GAN is a new kind of model, which usually contains a generative model G and a discriminative model D [28]. The models G and D are trained in an adversarial manner, in which G tries to generate the fake inputs as real as possible, and D tries to classify the real and fake inputs. In this adversarial game, both participants wish to get optimized results (i.e., D can achieve the best classification results, and G can generate the fake data which possess the most similar distribution with real data). Through the adversarial manner and competition of two networks, the training process of the discriminator will proceed both continuously and effectively instead of getting trapped into overfitting immediately when we use a limited number of training samples.

Furthermore, GAN generates samples which can be used as virtual samples. The proper usage of the virtual samples improves the classification performance. In this paper, the generated samples are used to boost the classification accuracy, and the experimental results prove the effectiveness of the usage of such samples.

Due to the advantages of CNN, deep convolutional GAN is well suited for image data processing [29]. GAN can be used in a variety of applications such as data synthesis, style transfer, and image super-resolution and classification [30].

In this paper, the use of the GAN for hyperspectral data classification is explored for the first time. With the help of GANs, the deep CNN achieves better performance in terms of classification accuracy and the overfitting problem raised by CNNs can be considerably mitigated. Two frameworks, 1D-GAN and 3D-GAN, are proposed for HSI classification, and the classification results obtained by these two frameworks showed that our GANs are superior to traditional CNNs even under the condition of limited training samples. In more detail, the main contributions of this paper are summarized as follows.

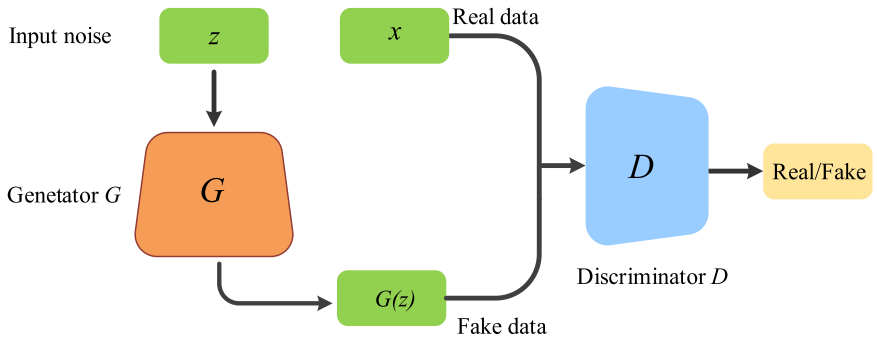


Fig. 1. Architecture of GAN.

1) The use of GAN for hyperspectral data processing is explored for the first time in this paper. The adversarial samples including 1-D spectra and 3-D spectral–spatial patches are generated by a well-designed GAN.
2) Two GAN frameworks including spectral classifiers and spectral–spatial classifiers are proposed for HSI classification. The frameworks use well-designed CNNs to classify HSIs, and the adversarial training is adopted by a regularization technique.
3) Adversarial samples are used for HSI classification for the first time, which is investigated to fine-tune the aforementioned GANs for an improved classification performance.
4) The proposed methods are tested on three well-known hyperspectral data sets under the condition of having limited training samples available.

The rest of this paper is organized as follows. Section II presents the background of GAN. Section III presents the details of the proposed GAN frameworks including spectral and spectral–spatial architectures for HSI classification, and the introduction of adversarial samples for classification. The details of experimental results are reported in Section IV. In Section V, conclusions and discussions are presented.

II. Background

Supervised learning methods in machine learning can in general be divided into two parts: generative approaches and discriminate approaches. Generative approaches involve the learning of distribution parameters from data samples, and they can generate new samples according to the learned models. Typically, generative methods calculate the distribution assumption of the explicit or implicit variables of the real data. Then, they can generate new data through the learned assumptions, which have a similar distribution as the real data.

GAN which was proposed by Goodfellow *et al.* [28] is a novel way to train a generative model and a promising method to train classifiers. Commonly, GAN is composed of two parts: the generative network G and the discriminative model D. Generator G can capture the potential distribution of real data and output the new data while discriminator D is a binary classifier which can judge whether the input samples are real or not. The information flow in GAN is a feed forward pass from one model generator G, which generates the fake data, to the second model discriminator D, which evaluates the output of the first model. The architecture of GAN is shown in Fig. 1.

In order to learn the generator's distribution p_g over data x, we supposed that the true samples are equipped with data distribution $p(x)$ and the input noise variable has a prior $p(z)$. The generator accepts a random noise z as input and produces a mapping to data space $G(z)$. The $D(x)$ estimates the probability that x is the true sample from training data [31]. In the optimized procedure, discriminator D is trained to maximize $\log(D(x))$, which is the probability of assigning the correct labels to the correct sources while generator G is trained to minimize $\log(1 - D(G(z)))$. Therefore, the ultimate aim of the optimization is to solve the minimax problem

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

where E is the expectation operator. Through the calculation and evaluation, one found that when discriminator D has a high probability distribution of real samples, the gradients in D may disappear and cause the training process to stop. To make sure that the generator has proper gradients when the classification accuracy for the discriminator is high, the generator's loss function is usually formulated to maximize the probability of classifying a sample as true instead of minimizing the probability of it to be classified as false [30]. Therefore, the modified loss function can be written in the following optimization form:

$$\begin{aligned} F(D,G) &= \min_D V(D,G) \\ &= -E_{x\sim p(x)}[\log(D(x))] \\ &\quad - E_{z\sim p(z)}[\log(1 - D(G(z)))] \quad (2) \\ f(D,G) &= \min_G V(D,G) = -E_{z\sim p(z)}[\log(D(G(z)))]. \quad (3) \end{aligned}$$

The parameters' update of G is based on the back-propagation of D instead of directly using the real data samples. In addition, there is no requirement that the latent code has any specific dimensionality nor on the generator net being invertible. GANs can learn models that generate points only on a thin manifold that goes near the data, so the GAN framework can train any kind of generator net.

On one hand, the adversarial model is most straightforward when the G and D are both multiply perceptrons [28]. On the

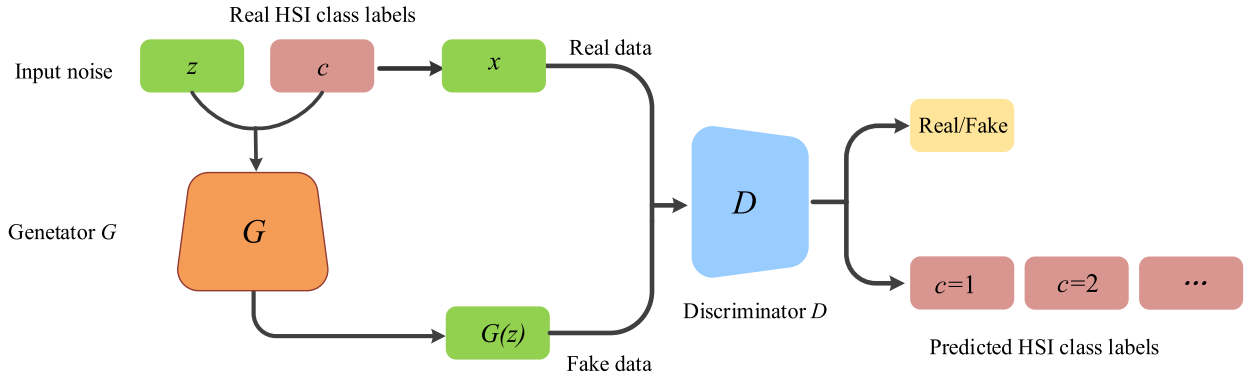


Fig. 2. General architecture of the proposed method.

other hand, the shallow multiply perceptrons usually have limited capabilities in coping with complex data. Recently, deep learning methods have achieved a substantial success and they can build a network with several layers instead of having shallow net layer connections [33]. Therefore, they can leverage the nonlinear mapping abilities and capture more crucial features of input data (i.e., especially in coping with the complicated and inherently nonlinear data) [34], [35]. As a special type of deep learning-based method, the convolution network (i.e., CNN) [36], which is inspired by neuroscience [37], utilizes two important strategies which make it different from other deep learning methods. Those two strategies are local connections and shared weights. In more detail, in the CNN, the local correlation information is explored by considering the local connectivity between nearby neurons, instead of the full connection of all neural units as in other deep methods. In this context, CNNs can reduce the computational complexity and redundancy in exploiting the data features. Especially, CNNs can alleviate the overfitting phenomenon in the deep learning process. Recently, the deep methods and CNNs have shown their excellent performance in HSI classification and feature extraction [23].

Although GANs have shown their promising usages in a variety of aspects, they also suffer from stability issues due to its adversarial idea [28]. To solve this issue, some techniques for stabilizing the training process of the GAN have been developed. Among those approaches, the convolution network can address the instability issue to some extent. Recently, the deep convolutional generative adversarial network (DCGAN) architecture, which uses deep convolution networks in G and D, was proposed in [29]. Generator G, which takes a uniform noise distribution z and class labels c as input and output, respectively, can be reshaped into a 3-D tensor. Discriminator D has a similar architecture to common CNNs except that the pooling layers are replaced with stride convolutions and all the activation functions are the leaky rectified linear unit [38]. Although GANs have shown promising performance in image synthesis [39] and many other aspects [40], as unsupervised generative models, the D networks of the original GANs only estimate whether the input samples are either true or fake. Therefore, they are not suitable for the purpose of multiclass image classification. Recently, the concept of GAN has been extended to be a conditional model with semisupervised methods. Odena *et al.* [41] proposed semi-GAN whose labels of true training data were imported to discriminator D.

Furthermore, the conditional GAN requires some additional information for both G and D where the extra information can be class labels or other data modalities [42]. Therefore, the conditional GAN can leverage the generative adversarial nets to generate the specific labels. In addition, Odena *et al.* [43] presented auxiliary classifier GAN (AC-GAN), which can be used in image classifications, whose discriminator D is modified to be a softmax classifier that can output multiclass label probabilities.

III. Proposed Method

A. Framework of the Proposed Method

Our networks are shaped with respect to the theory of AC-GAN while the objective function is modified to take into account the likelihood of the correct source and likelihood of the correct HSI classes. The parameters are optimized depending on the multiclassification loss, so they can optimize the loss function more appropriately than can traditional GANs. The labeled information is regarded as the input of both generator and discriminator, and discriminator D contains an auxiliary decoder network which can output the respective training data labels. Definitely, the extra-label information can leverage both the classification abilities of discriminator D and generative abilities of generator G. The architecture of our method is shown in Fig. 2.

From the original GAN, we can see that in training process, discriminator D receives both real training data and fake data generated by generator G and outputs the probability distribution $P(S|X) = D(X)$. Therefore, the aim of the D network is to try to maximize the log-likelihood of the right source

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

Similarly, the aim of the G network is to try to minimize the same quantity.

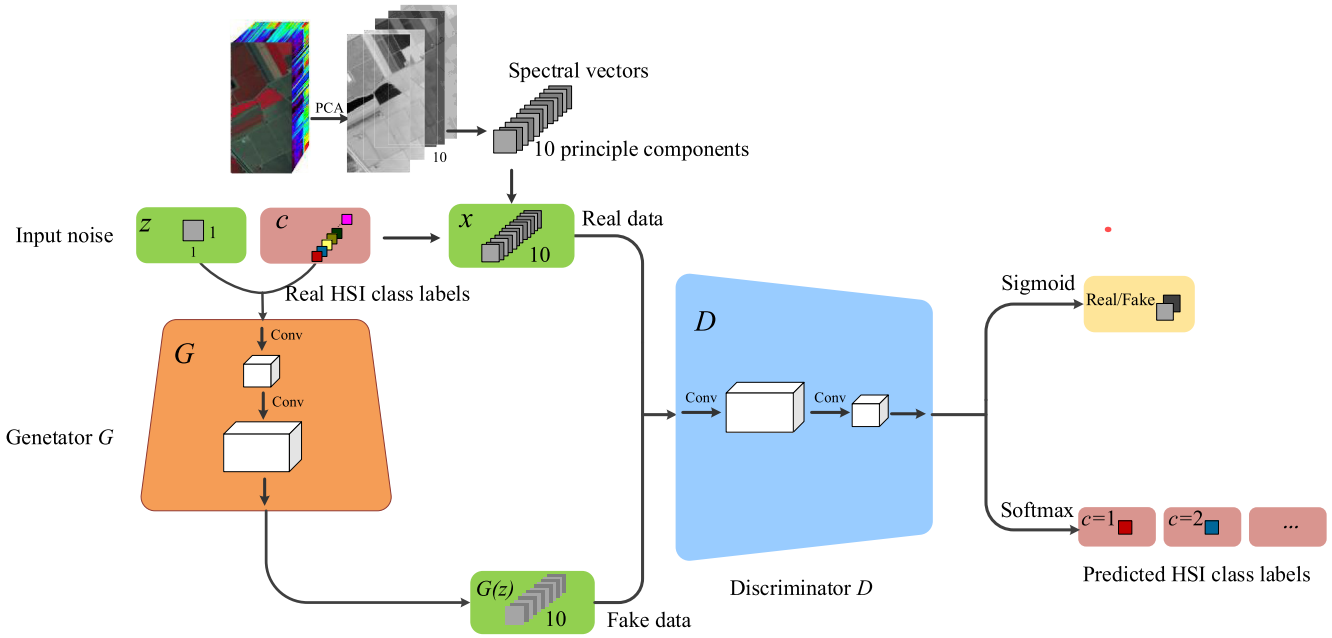


Fig. 3. Framework of 1D-GAN for HSI classification.

In the network, we can see that generator G also accepts HSI class labels $c \sim p_c$; in addition to the noise z, the output of G can be defined by $X_{\text{fake}} = G(z)$. The real training data with corresponding class labels and the fake data generated by G are regarded as the input of discriminator D. The probability distribution over sources $P(S|X)$ along with the probability distribution over class labels $P(C|X)$ is fed into the network D. The objective function contains two parts: the log-likelihood of the right source of input data L_s and the log-likelihood of the right class labels L_c

$$L_s = E[\log P(S = \text{real}|X_{\text{real}})] + E[\log P(S = \text{fake}|X_{\text{fake}})] \quad (5)$$

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

So D is optimized to maximize the $L_S + L_C$, while G is optimized to maximize $L_C - L_S$.

In this experiment, the two frameworks (i.e., the combination of the spectral vector and the spectral–spatial information) are designed, while the former is called 1D-GAN and the latter is called 3D-GAN. The G and D in the aforementioned two frameworks are both in the form of convolutional networks. The proposed GANs are not only used to classify the real and fake inputs but also to predict the corresponding labels of the input data. G has the form of fractionally strided CNN and D adopts CNNs. Finally, discriminator D is followed by sigmoid classifier and softmax classifiers applied in parallel, which are used to classify the real/fake samples and the HSIs, respectively.

With the continuation of the training process, both G and D will theoretically achieve the most optimized results when G can generate fake data that are most similar to real data and D cannot distinguish between fake data and real data. In this manner, we can prove that the whole network achieves the Nash equilibrium condition, and the adversarial behavior and competition between two networks can promote the classification performance. Therefore, the crucial idea of GAN lies in the adversarial training and through the continuous competition, we can obtain superior classification results in comparison to traditional CNN methods.

B. 1D-GAN

The main framework of 1D-GAN is shown in Fig. 3, which is built based on HSI spectral features only where all the input noise and training data are spectral vectors.

Due to the availability of hundreds of bands in original HSI, which means that the input has high dimensions, it is difficult to train generator G (i.e., the training stage is not stable). In the other words, the generator cannot effectively imitate the real data because of high redundancy. Thus, we extract the spectral features of HSIs based on PCA [44]. PCA can condense the whole image by reducing the spectral dimensions to a suitable scale, and it is a crucial step to formulate a robust GAN.

Pertinently, in order to promote the generator to exert superior abilities on capturing crucial features and generating realistic-looking samples as the best possible and preserve most important spectral information of original HSI bands, we preserve ten principal components, which contain the 99.99% energy of the inputs.

In Fig. 3, we show the framework of 1D-GAN on the Salinas data set as input. From Fig. 3, one can see that the G receives both noise z and labels c as input, while D receives as input the real spectral vectors x with labels c and some batches of fake spectral vectors $G(z)$. Then, through the D network, one can obtain the final results including both the class labels and the source labels. In our networks, any pooling layer is replaced

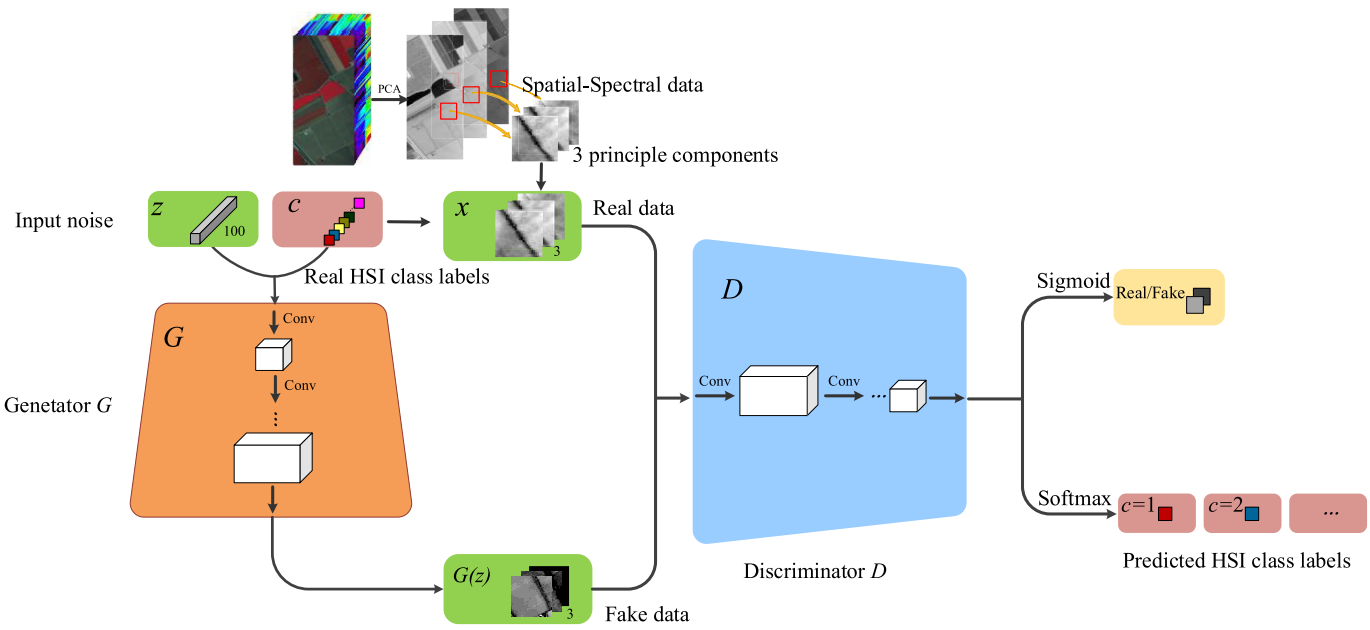


Fig. 4. Framework of 3D-GAN for HSI classification.

with stride convolutions (discriminator) and fractional-strided convolutions (generator).

C. 3D-GAN

The main framework of 3D-GAN is shown in Fig. 4. In 3D-GAN, the network extracts spectral–spatial features effectively, which achieves better classification performance. Here, we only describe the special parts of the proposed 3D-GAN. Different from 1D-GAN which only utilizes the spectral information of HSIs, the spatial features are also taken into consideration in addition to the spectral features in this section.

The number of spectral bands is reduced to three components by PCA, which condense the whole image by reducing the data dimensions to a suitable scale, and in the meantime, it reserves the spatial information. Due to PCA, the computational complexity is dramatically reduced, and it is really important to stabilize the GAN training procedure. In 3D-GAN, the generator accepts noise as input and transforms its shape to the same size as real data with three principal components in the spectral domain. Then, the discriminator accepts the real data or the generated fake samples as input data, and it uses the sigmoid classifier to give the real and fake classification results and softmax classifier to give the classification map. The first layer of the GAN, which takes a uniform noise distribution z as input, is just a matrix multiplication. The result is then reshaped into a tensor and used as the start of the convolution stack. In the proposed frameworks, we indicate the hyperspectral ground labels in multicolor squares but the source labels in gray squares.

In 3D-GAN, like the basic allocations in 1D-GAN, pooling layers are replaced with strided convolutions (in the discriminator) and fractional-strided convolutions (in the generator). In addition, batch normalization is used in both the generator and the discriminator and the fully connected hidden layers are removed.

D. Generative Adversarial Samples for Classification

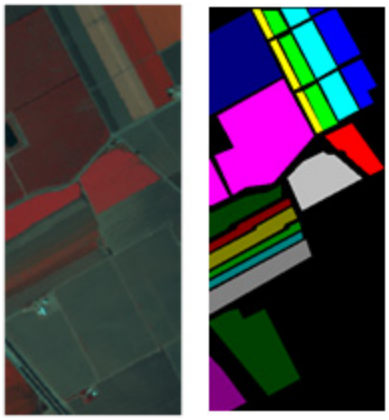
Now let us consider that the GAN consists of both a generator and a discriminator. The classification abilities of the discriminator have been discussed above. On the other hand, using the fake samples from the generator also has a potential in improving the final classification results in terms of accuracies. In fact, when the discriminator cannot distinguish the real data from the synthetic fake data, we can also conclude that the generated abilities of G are superior to that of D. Consequently, the whole adversarial network achieves the global optimality in theory [28].

In this paper, the generated fake samples can be regarded as augmentation data [45] that increase the number of training samples. Then, both the fake samples and the true samples are fed into the networks in order to optimize the nets. Let us suppose that the original data set has N classes, during network training. To begin with, each generated sample is passed forward through the network and assigned a label by taking the maximum value of the probability prediction vector. These fake samples can consequently be used for training in the network with these labels. In addition, the generated samples do not belong to any class of real samples. Due to the difference of real samples and generated samples, a new class label is then created (i.e., $N+1$) and every fake sample is endowed with this new label. In this paper, we adopted this method to assign $N+1$ to the labels of fake samples. In the experimental part of this paper, we qualitatively visualize the adversarial samples through the false-color map.

IV. Experimental Results

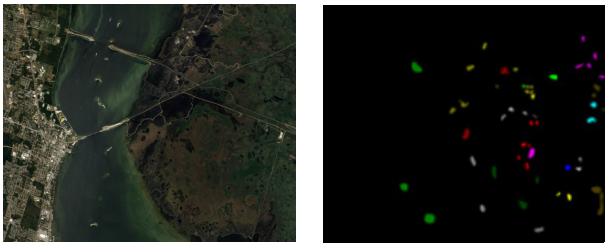
A. Data Description

In our study, three widely used hyperspectral data sets with different environmental settings were adopted to validate the proposed methods. They are captured over Salinas Valley in California (Salinas), Kennedy Space Center (KSC) in Florida,



(a) (b)

Fig. 5. Salinas data set. (a) False-color composite (bands 50, 28, and 15). (b) Ground reference map.



(a) (b)

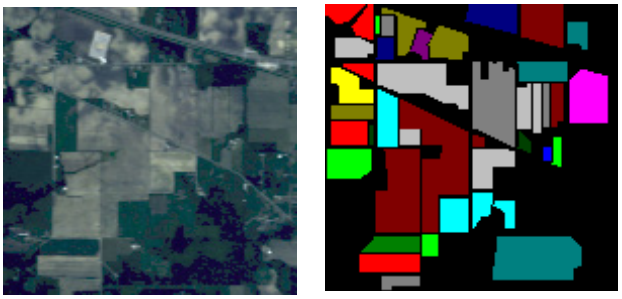
Fig. 6. KSC data set. (a) False-color composite (bands 28, 19, and 10). (b) Ground reference map.

and a mixed vegetation site over the Indian Pines test area in Northwestern Indiana (Indian Pines).

The first data set, Salinas, was collected by the 224-band Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor over Salinas Valley. The available data set is composed of 512 × 217 pixels with 204 bands [after the removal of low signal-to-noise ratio (SNR) bands], and the available ground reference map covers 16 classes of interest. The HSI is characterized by having a high spatial resolution (3.7-m pixels). The false-color composite image and the corresponding ground reference map are demonstrated in Fig. 5.

The second data set was captured by the NASA AVIRIS instrument over the KSC. It is called here the KSC data set. The KSC data set has an altitude of approximately 20 km, with a spatial resolution of 18 m. This data set is composed of 512 × 614 pixels. After removing water absorption and low SNR bands, 176 bands were used for the corresponding false-color composite map. In the classification, 13 classes were defined for the site. Fig. 6 demonstrates the classes of the KSC data set and the corresponding false-color composite map.

The third data set is a mixed vegetation site over the Indian Pines test area in Northwestern Indiana (Indian Pines) which was collected by the AVIRIS. The data set was obtained by an aircraft flown, with a size of 145 pixels × 145 pixels and 220 spectral bands in the wavelength range of 0.4–2.5 μm. The false-color image is shown in Fig. 7(a). The number of



(a) (b)

Fig. 7. Indian Pines data set. (a) False-color composite image (bands 28, 19, and 10). (b) Ground reference map.

TABLE I

LAND-COVER CLASSES AND NUMBERS OF PIXELS ON SALINAS DATA SET

Class			Samples
No.	Color	Name	Numbers
1		Brocoli_green_weeds_1	1977
2		Brocoli_green_weeds_2	3726
3		Fallow	1976
4		Fallow_rough_plow	1394
5		Fallow_smooth	2678
6		Stubble	3959
7		Celery	3579
8		Grapes_untrained	11213
9		Soil_vinyard_develop	6197
10		Corn_senesced_green_weeds	3249
11		Lettuce_romaine_4wk	1058
12		Lettuce_romaine_5wk	1908
13		Lettuce_romaine_6wk	909
14		Lettuce_romaine_7wk	1061
15		Vinyard_untrained	7164
16		Vinyard_vertical_trellis	1737
Total			53785

bands is reduced to 200 by removing water absorption bands. Sixteen different land-cover classes are provided in the ground reference map, as shown in Fig. 7.

For all three data sets, the labeled samples were split into two subsets which contain training and test samples, and the details are listed in Tables I–III. During the training procedure of GAN, we use 200 training samples, which are truly limited, to learn weights and biases of each neuron, and 100 training samples are used to guide the design of proper architectures, which can identify whether the network is overfitted or not. In the test process, all the samples in the data set are used to estimate the capability of the trained network. This is important for designing the proper convolution network and we can use the test samples to assess the final classification performance.

B. Classification Results for 1D-GAN

In this experiment, our networks are also compared with other former methods used in HSI feature extraction and classifications such as the original CNN to validate the quality of

TABLE II
LAND-COVER CLASSES AND NUMBERS OF PIXELS ON KSC DATA SET

Class			Samples
No.	Color	Name	Numbers
1		Scrub	761
2		Willow swamp	243
3		CP hammock	256
4		Slash pine	252
5		Oak/Broadleaf	161
6		Hardwood	229
7		Swamp	105
8		Graminoid marsh	431
9		Spartina marsh	520
10		Cattail marsh	404
11		Salt marsh	419
12		Mud flats	503
13		Water	927
Total			5211

TABLE III
LAND-COVER CLASSES AND NUMBERS OF PIXELS ON INDIAN PINES DATA SET

Class			Samples
No.	Color	Name	Numbers
1		Alfalfa	46
2		Corn-notill	1428
3		Corn-min	830
4		Corn	237
5		Grass-pasture	483
6		Grass-trees	730
7		Grass-pasture-mowed	28
8		Hay-windrowed	478
9		Oats	20
10		Soybean-notill	972
11		Soybean-mintill	2455
12		Soybean-clean	593
13		Wheat	205
14		Woods	1265
15		Buildings-Grass-Trees	386
16		Stone-Steel-Towers	93
Total			10249

classification results. The training and test data are randomly chosen among the whole data set, and we only use 200 training samples for each data sets and the epoch is set to 500 for our methods. The GANs have a relatively higher computational complexity compared with that of other networks considering the fact that both G and D need to be trained in each epoch. The classification results are given in the form of mean $\pm$ standard deviation. The 1D-GAN framework is built only based on HSI spectral feature extraction. Therefore, all the input noise and training data are spectral vectors. First, the number of training data spectral bands in each data set is reduced to 10 by PCA. The size of noise z sent to the generator networks is $1 \times 1 \times 1$, and it is then converted to $10 \times 1 \times 1$ through the fractionally strided convolution of generator G.Then, the generated fake samples and true training samples are imported to discriminator D, which can give the corresponding image labels c and source labels s. The main architectures of the 1D-GANs and PCA-CNN for each data set are shown in Table IV, and the n_class shown in Table IV represents the number of classes in each data set

In order to have a fair comparison, the architecture of PCA-CNN is designed in such a way to be the same as discriminator D. The stride is 1 and the padding is set to 0 in this experiment. Similarly, we apply the same activation function as used in DCGAN. The batch normalization is also introduced to the networks for the specific layers. The batch normalization can stabilize training by normalizing the input for each unit to have zero mean and unit variance.

At the end of the common CNN, the fully connected neural nodes are activated by a linear function. Then, the corresponding labels are given through a softmax classifier. After the full connection and activation, the last layer neural nodes of discriminator D are connected with both a softmax classifier and a sigmoid classifier. From the softmax classifier, one can obtain the predicted label c. From the sigmoid classifier, one can obtain the labels s (fake or real) of the GAN.

In this experiment, our methods are compared with other feature extraction methods such as the PCA, factor analysis (FA) [46], locally linear embedding (LLE) [47], linear discriminant analysis (LDA) [48], independent component analysis (ICA) [49], PCA-CNN, and recurrent neural network (RNN) [50]. FA is a linear statistical method designed for potential factors from observed variables to replace original data. LLE seeks a lower dimensional projection of the data which preserves distances within local neighborhoods. LDA can be used to perform supervised dimensionality reduction, by projecting the input data to a linear subspace consisting of the directions which maximize the separation between classes. ICA separates a multivariate signal into additive subcomponents that are maximally independent. An RNN is a network which uses hidden layers or memory cells to learn features, and it adopts recurrent connections between neural activations at consecutive time steps. The effectiveness of different feature extraction methods is evaluated mainly through classification results. Additional classifiers such as the KNN classifier and SVM with a kernel like the radial basis function kernel with SVM (RBF-SVM) for the evaluation.

In order to have a fair comparison, we use the grid-search method to find the best parameters for these feature extraction methods. The results listed in Tables V–VII are the best results obtained in the searching process. For the methods with the RBF-SVM classifiers, we use a "grid-search" [51] method to define the most appropriate C and γ [52]. In this manner, pairs of (C, γ) are tried and the one with the best classification accuracy on the validation samples is picked. This method is convenient and straightforward. Moreover, the computational time to find the best parameters by grid search is not much since there are only two parameters that need to be estimated. In SVM-based experiments, we do the search by exponentially growing sequences of C and γ, as done in [23].

For PCA, FA, LDA, and ICA, we select the number of features in the range of 10–N (i.e., the number of hyperspectral bands). The number of the neighbors in LLE is chosen from 1 to 15, and for KNN, the range of the nearest

TABLE IV
ARCHITECTURES OF THE 1D-GAN

Nets	NO.	Convolution	BN	Stride	Padding	Activation function
G	1	4×1×512	YES	1	0	ReLu
	2	4×1×128	YES	1	0	ReLu
	3	4×1×1	NO	1	0	Tanh
D	1	4×1×256	NO	1	0	LeakyReLu
	2	4×1×512	YES	1	0	LeakyReLu
	3	4×1×128	NO	1	0	NO
	4	128×n_class	NO	-	-	Softmax
		128×2	NO	-	-	Sigmoid

TABLE V
CLASSIFICATION RESULTS OBTAINED BY DIFFERENT APPROACHES ON SALINAS DATA SET

Dataset	Classifier	KNN								
	FE methods	PCA	FA	LLE	LDA	ICA	CNN	PCA-CNN	RNN	1D-GAN
Salinas	OA(%)	75.20 ±1.42	71.27 ±2.17	74.50 ±1.58	76.53 ±2.17	80.12 ±1.90	83.32 ±1.90	84.14 ±2.17	77.46 ±2.27	85.64 ±2.96
	AA(%)	70.29 ±0.98	65.10 ±2.52	68.63 ±1.36	78.28 ±2.52	79.93 ±3.20	76.10 ±2.03	77.13 ±2.52	77.05 ±3.80	79.86 ±3.19
	K×100	72.14 ±1.38	68.20 ±2.48	70.13 ±1.72	72.82 ±1.38	77.68 ±2.13	81.34 ±1.42	83.67 ±1.38	74.81 ±2.51	83.67 ±2.03
	Classifier	Logistic Regression								
	OA(%)	79.12 ±0.98	76.57 ±2.52	76.70 ±1.76	79.22 ±2.67	80.35 ±2. 23	88.29 ±1.73	88.94 ±2.17	78.46 ±2.15	89.13 ±2.19
	AA(%)	78.29 ±1.48	76.10 ±2.52	75.63 ±1.36	68.94 ±3.47	75.14 ±3.45	84.10 ±2.03	87.13 ±2.52	80.92 ±2.61	88.13 ±3.50
	K×100	76.14 ±1.72	74.20 ±2.48	74.13 ±1.72	72.57 ±3.07	78.04 ±2.09	87.34 ±1.42	88.45 ±1.38	76.01 ±2.41	89.03 ±2.62
	Classifier	RBF-SVM								
	OA(%)	78.20 ±2.42	75.27 ±3.17	76.50 ±2.58	78.53 ±2.17	83.04 ±2.17	87.32 ±1.90	87.98 ±2.17	77.23 ±2.23	88.45 ±2.80
	AA(%)	82.29 ±0.98	81.10 ±3.52	79.63 ±2.36	75.51 ±2.52	80.48 ±2.17	84.10 ±2.03	85.13 ±2.52	78.02 ±3.82	84.63 ±3.64
	K×100	75.14 ±1.38	73.20 ±3.48	72.89 ±2.72	77.05 ±1.38	81.98 ±2.17	85.34 ±1.72	85.45 ±1.38	74.56 ±2.41	88.45 ±2.62

neighbors is varied between 1 and 10. Tables V–VII show the results obtained from the condition when the GAN models are trained using the ten principal components of the original HSIs. Compared with the PCA, FA, LLE, LDA, ICA, and RNN, the CNN-based methods show better performances in terms of accuracies, especially the 1D-GAN methods which all achieve the best accuracy on three different classifiers. Furthermore, Tables V–VII illustrate that the classification results usually have a superior performance when feature extraction methods are followed by either *LR* or RBF-SVM classifiers. For the Salinas data set, 1D-GAN-*LR* exhibits the best overall accuracy (OA), average accuracy (AA), and kappa (K), with improvements of 0.84%, 4.03%, and 0.0169 over CNN-*LR*, respectively. Our approach outperforms 1D-GAN-KNN by 3.49%, 8.27%, and 0.0536 in terms of OA, AA, and K, respectively. For the KSC data set, as can be seen, the OA of 1D-GAN-*LR* is 89.64%, which is increased by 3.50% and 0.48% compared with that of KNN and RBF-SVM, respectively; it also increases 1.97%, 3.29%, and 0.0308 in OA, AA, and K compared with that of 1D-PCA-CNN. For the Indian Pines data set, the best performance is achieved by 1D-GAN followed by RBF-SVM. In this context, 1D-GAN followed by RBF-SVM improves the OA, AA, and K of CNN-RBF-SVM by 3.32%, 4.35%, and 0.0391, respectively. The results show that the 1D-GAN method gave the best performance in terms of OA, AA, and K for all three data sets.

TABLE VI
CLASSIFICATION RESULTS OBTAINED BY DIFFERENT APPROACHES ON KSC DATA SET

Dataset	Classifier	KNN								
	FE methods	PCA	FA	LLE	LDA	ICA	CNN	PCA-CNN	RNN	1D-GAN
KSC	OA(%)	68.20 ±3.42	67.27 ±2.27	69.50 ±1.87	76.61 ±3.42	76.70 ±2.72	83.32 ±2.30	84.14 ±3.42	78.26 ±2.23	86.14 ±3.80
	AA(%)	58.29 ±4.28	54.10 ±2.72	59.63 ±1.71	64.02 ±4.28	66.23 ±3.42	78.10 ±2.43	79.13 ±4.28	68.10 ±3.75	79.58 ±3.21
	K×100	65.14 ±4.38	64.20 ±2.07	67.13 ±2.56	74.32 ±2.63	73.95 ±3.04	81.34 ±2.63	85.45 ±3.52	75.73 ±2.67	83.45 ±3.62
	Classifier	Logistic Regression								
	OA(%)	71.20 ±0.98	70.27 ±2.52	72.50 ±1.36	75.42 ±2.35	76.90 ±2.86	85.32 ±2.03	87.67 ±2.42	80.82 ±1.99	89.64 ±2.47
	AA(%)	74.29 ±2.16	67.10 ±2.52	70.63 ±1.36	70.34 ±3.71	72.89 ±3.15	80.10 ±2.03	81.13 ±1.28	71.79 ±2.87	84.42 ±3.19
	K×100	70.14 ±2.38	68.20 ±2.48	71.13 ±1.72	71.96 ±2.71	74.00 ±2.05	83.34 ±1.42	85.15 ±1.52	78.63 ±2.21	88.23 ±2.62
	Classifier	RBF-SVM								
	OA(%)	70.20 ±2.10	69.27 ±2.17	69.50 ±1.58	71.61 ±3.42	78.21 ±2.97	86.32 ±1.90	86.14 ±1.42	80.30 ±3.07	88.16 ±2.80
	AA(%)	60.29 ±2.98	59.10 ±3.52	60.63 ±1.36	66.13 ±4.28	74.65 ±2.45	81.80 ±2.03	81.02 ±1.28	70.66 ±5.35	83.13 ±3.54
	K×100	68.14 ±2.46	62.20 ±2.15	64.13 ±1.72	70.17 ±2.63	75.58 ±2.63	84.89 ±1.52	86.23 ±1.51	78.04 ±3.53	85.10 ±2.03

TABLE VII
CLASSIFICATION RESULTS OBTAINED BY DIFFERENT APPROACHES ON INDIAN PINES DATA SET

Dataset	Classifier	KNN								
	FE methods	PCA	FA	LLE	LDA	ICA	CNN	PCA-CNN	RNN	1D-GAN
Indian Pines	OA(%)	48.20 ±5.42	47.27 ±4.17	47.50 ±3.58	55.52 ±2.36	57.60 ±1.21	61.32 ±1.90	63.14 ±1.52	53.61 ±4.84	64.39 ±2.46
	AA(%)	40.29 ±5.98	40.10 ±4.92	39.63 ±3.36	40.34 ±4.43	41.80 ±2.82	55.10 ±1.03	58.89 ±1.42	37.57 ±5.13	56.13 ±3.19
	K×100	44.14 ±5.38	43.20 ±4.48	42.13 ±3.72	48.26 ±2.61	50.63 ±1.52	58.34 ±1.42	60.45 ±1.04	45.67 ±5.52	60.45 ±3.14
	Classifier	Logistic Regression								
	OA(%)	51.20 ±1.59	47.27 ±2.23	46.50 ±1.97	54.67 ±1.98	56.57 ±2.78	64.32 ±1.47	66.14 ±1.46	57.51 ±2.53	67.24 ±2.80
	AA(%)	43.29 ±2.98	45.10 ±3.52	41.63 ±2.36	39.37 ±2.76	42.69 ±3.14	57.10 ±3.03	59.13 ±2.09	44.52 ±4.03	59.27 ±3.19
	K×100	49.14 ±1.68	46.20 ±2.48	43.13 ±1.28	49.05 ±3.05	51.39 ±3.01	61.16 ±1.98	64.10 ±1.52	51.24 ±3.12	63.45 ±2.62
	Classifier	RBF-SVM								
	OA(%)	50.20 ±1.42	43.27 ±2.17	44.50 ±1.58	53.65 ±2.48	58.47 ±2.14	65.32 ±1.90	67.32 ±2.09	53.66 ±5.18	68.64 ±2.14
	AA(%)	47.29 ±0.98	38.10 ±2.52	40.63 ±1.36	47.34 ±1.42	52.48 ±2.98	60.10 ±2.03	62.13 ±1.90	39.17 ±6.88	64.45 ±3.25
	K×100	49.14 ±1.38	40.20 ±2.48	42.13 ±1.72	50.89 ±2.13	57.47 ±2.47	62.45 ±1.42	65.36 ±1.35	46.08 ±7.02	66.36 ±3.08

Furthermore, detailed experiments with different principal components of 1D-GAN have been investigated to give a comprehensive comparison. Table VIII shows the classification results of 1D-GAN with different principal components; the experiments utilized the whitening PCA, which is a modified PCA with identity covariance

TABLE VIII
CLASSIFICATION RESULTS OF 1D-GAN WITH DIFFERENT PRINCIPAL COMPONENTS ON THREE DATA SETS

Dataset	Method	PCA-3	PCA-10	PCA-50	PCA-100	PCA-All
Salinas	OA(%)	85.32±1.54	89.13±2.19	83.46±1.95	83.32±1.32	81.89±2.24
	AA(%)	89.15±0.39	88.13±3.50	74.39±1.47	74.09±0.97	73.14±3.10
	K×100	82.07±1.22	90.45±2.62	75.20±1.42	76.60±1.03	78.52±2.26
	Time(s)	29.6	46.1	66.8	86.6	106.2
KSC	OA(%)	76.89±2.24	89.64±2.47	82.74±1.89	82.45±1.45	82.33±1.34
	AA(%)	70.14±1.34	84.13±3.19	78.28±2.04	77.28±2.04	76.15±1.45
	K×100	74.52±1.26	88.15±2.62	83.41±1.23	81.24±0.89	81.48±1.26
	Time(s)	19.5	32.7	68.2	142.4	178.5
Indian Pines	OA(%)	64.09±3.27	67.24±2.80	64.35±2.09	62.30±1.85	59.58±2.56
	AA(%)	58.14±2.56	59.27±3.19	56.75±2.56	53.75±1.56	48.78±3.09
	K×100	61.32±2.21	63.45±2.62	61.53±1.03	60.07±1.03	57.00±2.13
	Time(s)	23.2	37.2	85.4	115.6	153.4

TABLE IX
ARCHITECTURES OF THE 3D-GAN

Nets	No.	Convolution	BN	Stride	Padding	Activation function
G	1	4×4×512	YES	1	0	ReLU
	2	4×4×256	YES	2	1	ReLU
	3	4×4×128	YES	2	1	ReLU
	4	4×4×64	YES	2	1	ReLU
	5	4×4×3	NO	2	1	Tanh
D	1	4×4×64	NO	2	1	LeakyReLU
	2	4×4×128	YES	2	1	LeakyReLU
	3	4×4×256	YES	2	1	LeakyReLU
	4	4×4×512	YES	2	1	LeakyReLU
	5	4×4×64	NO	1	0	NO
	6	64×n_class	NO	-	-	Softmax
		64×2	NO	-	-	Sigmoid

matrix. In Table VIII, the PCA-3, PCA-10, PCA-50, PCA-100, and PCA-All represent the situations where we preserve 3, 10, 50, 100, and all principal components, respectively.

In 1D-GAN, ten principal components are used to condense the spectral information. Due to the fact that we only use spectral information in 1D-GAN, we try to preserve sufficient components (i.e., 10). Furthermore, compared with 3D-GAN, the computational complexity of 1D-GAN is relatively low. We use relatively more components without high computational complexity. From Table VIII, one can see that the ten principal components achieved the best classification accuracy on three data sets. If the number of the selected principal components is not sufficient, the classification results tend to be bad (e.g., when three are selected). Because of the dimensionality of the input is low (e.g., three PCs), one cannot formulate a deep network to capture the discriminant and nonlinear features efficiently. If the number of the selected principal components is too many (e.g., 100), the classification results tend to be bad too. In order to obtain a good classification performance, ten principal components are selected in 1D-GAN. Because of the dimensionality of the input is too high (e.g., 100), it may cause the serious overfitting problem under the condition of limited training samples (i.e., 200).

C. Classification Results for 3D-GAN

In 3D-GAN, the network considers both spectral and spatial features effectively, which can lead to a better performance in terms of classification accuracies than the ones obtained by 1D-CNN. As mentioned before, we preserve three principal components as the inputs of 3D-GAN. The architecture of the 3D-GAN is shown in Table IX. The number of classes for each data set is denoted by n_class. The networks G and D of 3D-GAN are deep CNNs with five convolution layers and proper activation functions. Discriminator D is followed by a sigmoid classifier and a softmax classifier, which are used to classify the real/fake samples and the HSIs, respectively. 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 $64 \times 64 \times 3$. Then, the generated samples are sent to discriminator D. Bach normalization is used in some specific layers to boost the classification accuracy. In our practice, the model leads to instability if batch normalization is used in all layers.

TABLE X
CLASSIFICATION WITH SPECTRAL–SPATIAL FEATURES ON THE SALINAS DATA SET

Method	3D-RBF-SVM	EMP-SVM	EMP-CNN	3D-CNN	PCA-CNN	3D-GAN
OA(%)	83.09±1.08	85.90±1.26	87.04±0.16	88.15±0.24	91.26±0.48	93.02±1.54
AA(%)	85.46±2.06	82.53±1.38	74.02±0.36	77.76±0.82	84.47±0.43	89.15±0.39
K×100	81.07±1.19	84.02±1.46	85.43±0.19	86.05±0.13	90.88±0.52	92.07±1.22
Brocoli_green_weeds_1	94.15±0.50	77.97±0.69	96.33±1.20	60.17±0.36	84.32±1.13	98.12±1.02
Brocoli_green_weeds_2	98.57±0.89	99.75±0.37	93.86±0.23	95.04±0.17	98.81±0.08	94.11±0.12
Fallow	90.56±0.50	50.40±0.51	65.54±1.29	84.69±0.32	75.92±0.15	76.46±0.28
Fallow_rough_plow	98.93±0.40	98.72±0.82	94.33±0.47	97.63±0.15	89.84±0.09	100.00±0.47
Fallow_smooth	95.23±0.63	97.44±0.36	77.01±1.01	99.95±0.08	80.00±0.00	88.25±1.89
Stubble	99.25±0.91	99.94±0.36	94.02±0.01	99.96±0.10	96.88±0.07	99.34±0.36
Celery	98.82±0.33	99.88±0.02	90.00±0.50	87.50±0.00	97.96±0.19	99.90±0.67
Grapes_untrained	78.50±0.57	98.50±0.22	75.47±0.69	89.65±0.22	83.64±0.13	89.44±1.13
Soil_vinyard_develop	94.11±0.50	99.33±0.37	95.19±0.01	99.38±0.61	100.00±0.00	100.00±0.00
Corn_senesced_green_weeds	85.56±0.36	93.99±0.56	85.41±0.04	98.20±0.95	100.00±0.00	98.13±1.00
Lettuce_romaine_4wk	90.63±0.78	82.30±1.23	92.03±0.16	92.20±0.36	98.87±0.12	96.69±2.12
Lettuce_romaine_5wk	99.48±0.03	100.00±0.00	89.45±1.23	34.54±4.36	99.35±0.04	99.06±1.04
Lettuce_romaine_6wk	20.08±2.47	99.12±0.16	40.80±1.89	19.20±5.36	74.65±0.68	77.92±1.68
Lettuce_romaine_7wk	66.29±1.67	97.64±0.45	20.00±2.48	90.38±1.36	70.73±0.05	78.21±0.67
Vinyard_untrained	59.14±1.06	0.0138±2.56	52.64±2.96	91.59±1.36	65.13±0.20	70.88±0.45
Vinyard_vertical_trellis	66.96±0.78	83.79±0.48	55.97±1.87	18.49±6.36	83.42±0.94	90.0±0.12
Runtime (s.)	49.86	287.54	28.14	96.15	23.30	55.10

TABLE XI
CLASSIFICATION WITH SPECTRAL–SPATIAL FEATURES ON THE KSC DATA SET

Method	3D-RBF-SVM	EMP- SVM	EMP-CNN	3D-CNN	PCA-CNN	3D-GAN
OA(%)	76.13±0.24	90.59±0.25	96.94±0.13	95.63±0.26	96.02±0.42	96.89±1.24
AA(%)	64.60±0.34	85.83±0.28	91.39±0.02	89.65±0.21	93.17±0.14	94.14±0.40
K×100	73.52±1.30	89.41±1.39	96.60±0.14	94.95±1.13	95.27±0.49	96.52±0.26
Scrub	88.68±0.69	87.82±4.24	90.14±0.89	91.71±3.71	96.189±0.56	98.292±0.42
Willow swamp	69.77±2.67	80.14±7.06	57.28±2.04	89.73±10.1	71.605±1.96	79.835±1.45
CP hammock	70.73±1.45	87.64±6.19	97.24±0.23	92.16±5.33	76.828±1.24	98.438±0.14
Slash pine	57.32±0.63	86.64±4.79	95.60±0.19	86.94±6.27	100.00±0.00	86.508±1.12
Oak/Broadleaf	42.85±2.56	77.02±8.87	63.75±1.56	94.79±6.89	100.00±0.00	98.758±0.14
Hardwood	32.24±0.62	89.66±5.82	92.07±1.03	90.92±7.90	96.943±0.78	100.00±0.00
Swamp	10.00±3.77	83.37±17.8	100.00±0.00	91.57±6.14	77.143±1.47	97.143±1.06
Graminoid marsh	42.23±3.64	91.78±2.82	100.00±0.00	96.22±3.13	79.814±1.89	72.949±2.10
Spartina marsh	84.00±0.25	97.12±1.57	91.37±0.56	99.53±0.99	100.00±0.00	99.231±0.09
Cattail marsh	81.88±1.35	97.06±1.85	99.56±0.01	99.81±0.37	99.752±0.04	100.00±0.00
Salt marsh	96.75±0.45	99.64±1.12	100.00±0.00	99.79±0.30	100.00±0.00	100.00±0.00
Mud flats	86.32±1.26	99.24±2.11	98.21±0.17	97.69±2.31	99.205±0.03	96.481±1.23
Water	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.000±0.00	100.00±0.00
Runtime (s.)	47.94	292.78	27.68	57.53	24.56	48.57

The size of the mini batch was 100 and the learning rate was 0.0002. In this set of experiments, the number of training epochs for the CNNs and GANs is 600.

The SVM-based and CNN-based five methods are included to give a comprehensive comparison. The classification results are shown in Tables X–XII. For the three data sets, we use $64 \times 64 \times n$_band, where the n_band represents the number of spectral bands and $64 \times 64 \times 3$ neighbors of each pixel as the input 3-D images in these methods without PCA and using PCA, respectively. The input images are normalized into the range $[-0.5, 0.5]$.

Due to the advantages of SVM, some SVM-based HSI classifiers are included for comparison. The 3D-RBF-SVM receives the 3-D images as inputs. The extended morphological

TABLE XII
CLASSIFICATION WITH SPECTRAL–SPATIAL FEATURES ON THE INDIAN PINES DATA SET

Method	3D-RBF-SVM	EMP-SVM	EMP-CNN	3D-CNN	PCA-CNN	3D-GAN
OA(%)	58.01±1.08	69.34±1.06	86.48±0.13	86.47±0.26	87.27±1.01	89.09±1.97
AA(%)	50.56±2.06	52.63±1.28	68.19±0.26	70.41±0.42	80.17±0.48	83.14±1.58
$K\times 100$	52.07±1.19	64.51±0.56	84.23±0.16	84.12±0.19	85.24±1.29	87.32±1.21
Alfalfa	77.35±1.50	15.94±0.39	10.43±1.78	14.70±1.40	15.00±1.34	30.21±1.03
Corn-notill	18.52±1.06	39.16±1.28	86.71±1.25	86.34±1.11	86.98±0.96	81.79±0.26
Corn-mintill	54.66±0.57	70.75±0.59	84.25±1.47	89.49±0.70	85.27±1.34	75.93±1.26
Corn	32.30±0.39	54.74±0.00	76.92±2.69	42.00±1.43	94.09±1.63	90.08±1.23
Grass-pasture	9.73±0.54	68.37±0.47	84.96±1.59	85.91±0.23	92.75±1.87	86.39±2.12
Grass-trees	85.53±2.96	96.21±1.16	88.33±0.23	92.75±0.30	79.04±1.03	93.28±0.23
Grass-pasture-mowed	10.27±1.36	100.00±0.00	10.67±2.35	12.00±0.50	53.57±1.65	40.71±1.05
Hay-windrowed	78.50±0.23	85.00±0.56	100.00±0.00	100.00±0.00	91.59±0.48	98.11±0.21
Oats	12.32±0.51	15.78±1.19	12.46±2.56	10.00±1.20	10.20±2.05	20.00±1.96
Soybean-notill	62.28±3.36	75.92±1.39	89.85±0.45	78.72±0.95	73.25±1.14	74.28±0.89
Soybean-mintill	66.86±1.61	81.23±0.96	91.55±1.04	95.52±1.23	90.30±0.78	91.12±0.25
Soybean-clean	28.25±0.42	31.85±0.51	86.76±0.69	89.47±0.39	80.43±0.96	84.99±1.46
Wheat	99.18±1.47	98.23±0.28	87.50±1.48	80.00±0.00	56.34±2.48	49.75±2.45
Woods	85.98±1.28	90.85±1.19	88.37±1.67	84.55±0.58	91.62±0.69	94.38±0.26
Buildings-Grass-Trees	13.82±0.35	94.28±1.01	53.53±2.48	69.54±1.31	74.58±0.78	94.47±0.79
Stone-Steel-Towers	87.78±0.32	95.24±0.67	95.65±0.57	89.34±1.08	88.17±1.02	88.22±1.16
Runtime (s.)	68.80	315.12	42.18	93.51	37.95	72.46

profile with SVM (EMP-SVM) is a widely used spectral–spatial classifier for [5]. In the EMP-SVM method, three principal components from HSIs are computed and then the opening and closing operations are used to extract spatial information on the first three components. In the experiments, the shape structuring element is set as disk with an increasing size from 1 to 4. Therefore, 24 spatial features are generated. The learned features are fed to an RBF-SVM to obtain the final classification results.

Furthermore, 3D-CNN [21], which has the same architecture as discriminator D in 3D-GAN, PCA-CNN, and EMP-CNN are also used for comparison. For the PCA-CNN, the CNN is conducted on the three principal components, which is useful when the training samples are limited [20]. From Tables X–XII, one can see that for Salinas data set, the 3D-GAN exhibits the highest OA, AA, and K, with an improvement of 1.76%, 3.68%, and 0.0119 over PCA-CNN, respectively. On the other hand, our 3D-GAN approach outperforms 3D-RBF-SVM by 9.93%, 2.69%, and 0.09 in terms of OA, AA, and K, respectively. Furthermore, the proposed 3D-GAN obtains a better classification performance on Salinas data set compared with those of the 3D-CNN and EMP-CNN. For the KSC and Indian Pines data sets, we can obtain the similar results. Compared with these state-of-the-art methods, the 3D-GAN demonstrates the best performance.

In terms of the running time of different networks, the 3D-GAN needs a longer time to optimize a new network, which is caused by the update on both the generator network and discriminator network in one training epoch. Therefore, the computational complexity of GAN is approximately twice as much as the CNN-based methods, but our method possesses a superior ability in terms of classification accuracy.

Like 1D-GAN, detailed experiments with different principal components of 3D-GAN have been investigated to give a comprehensive comparison. Table XIII shows the classification results of 3D-GAN. In the experiment reported in Table XIII, three principal components are investigated to obtain the main spectral information and their spatial neighborhood pixels are used to obtain the spatial information. Due to the fact that the spatial information is also included, we use relatively fewer components in the spectral domain compared with those in 1D-GAN.

From Table XIII, one can observe that the PCA operation can improve the accuracy to some degree. In the experiment of 3D-GAN, we chose only three principal components because the classification results are relatively good compared with those of other traditional methods, although the classification performance may be better if more components are selected. One should note that more principal components lead to a higher computational complexity and a longer training time. Furthermore, the classification accuracies descended when too many components (e.g., 100 components) were selected.

Taking the aforementioned factors into consideration, after performing the whiten PCA, we preserved three components in 3D-GAN.

To explore the effect of the adversarial component of GANs in classification, we also conduct experiments on all three data sets to see whether the classifier component of the 3D-GAN would perform better than that of an isolated GAN

TABLE XIII
CLASSIFICATION RESULTS OF 3D-GAN WITH DIFFERENT PRINCIPAL COMPONENTS ON THREE DATA SETS

Dataset	Method	PCA-3	PCA-10	PCA-50	PCA-100	PCA-All
Salinas	OA(%)	93.32±1.54	95.12±0.75	95.03±0.95	94.53±1.02	94.89±1.24
	AA(%)	89.15±0.39	94.83±0.28	94.39±0.47	93.09±0.97	93.14±0.40
	K×100	92.07±1.22	94.41±1.39	94.20±0.42	93.60±0.45	94.52±0.26
	Time(s)	35.6	66.7	102.9	185.6	366.2
KSC	OA(%)	96.89±1.24	97.56±1.04	97.14±0.89	93.45±1.21	93.33±1.34
	AA(%)	94.14±0.40	92.14±1.06	94.28±2.04	89.28±2.04	88.15±1.45
	K×100	96.52±0.26	96.45±0.54	95.41±0.23	92.24±0.59	91.48±1.26
	Time(s)	25.05	49.5	108.1	200.4	330.8
Indian Pines	OA(%)	89.09±1.97	90.04±1.29	88.97±1.87	85.30±1.45	83.58±1.12
	AA(%)	83.14±1.58	85.02±1.54	82.75±1.56	81.75±1.56	77.78±2.14
	K×100	87.32±1.21	89.16±0.89	86.53±1.03	83.07±1.03	79.00±1.63
	Time(s)	40.2	60.7	167.9	348.6	532.7

TABLE XIV
CLASSIFICATION RESULTS BETWEEN NORMAL AND ISOLATED GAN NETWORKS ON THE THREE DATA SETS

Dataset		3D-GAN	3D-Isolated-GAN
Salinas	OA(%)	93.02±1.54	90.09±0.56
	AA(%)	88.15±0.39	85.76±1.31
	K×100	92.07±1.22	88.05±0.96
KSC	OA(%)	97.69 ±0.24	94.20±0.49
	AA(%)	94.14±0.40	84.34±1.37
	K×100	96.52±0.26	89.25±0.98
Indian Pines	OA(%)	90.69 ±0.86	87.23±1.75
	AA(%)	83.14±1.82	80.14±2.04
	K×100	89.62±0.26	84.32±0.47

network. The isolated GAN network means that discriminator D updates its parameters in the training process while generator G does not update its parameters. In other words, the gradient flow from discriminator D does not broadcast to generator G, and actually the two networks are not really in an adversarial manner. We call this training process 3D-Isolated-GAN, and the results are listed in Table XIV.

In this experiment, 200 training samples are randomly chosen for each data set, and from Table XIV, we can see that the normal 3D-GAN outperforms the 3D-Isolated-GAN about 2.93%, 3.49%, and 2.46% in terms of OA for the three data sets. Therefore, we can make the conclusion that the adversarial action between two networks is highly important and influential in the training process.

D. Visualization of Adversarial Samples and 3D-GAN Augmentation

The GAN consists of a generator and a discriminator, which is a two-player minimax game between G and D. If the discriminator cannot distinguish the real data from the synthetic fake data, we can conclude that the generated ability of G has superior performance, and the whole adversarial network achieves the global optimality in theory [28].

The discriminator can be regarded as a classifier to get the classification results. On the other hand, the synthetic fake samples from the generator can be used to increase the number of training samples. The discriminator model is the one which directly accesses information in the data set (e.g., the real samples from the distribution and the generator model learn based on error signals from the discriminator).

In the visualization experiment, the fake samples are generated by the generator (i.e., G) of 3D-GAN. Some selected fake and real samples on three hyperspectral data sets are shown in Fig. 8. In fact, it is difficult to distinguish the fake samples from the real samples. From Fig. 8, one can see that at the end of the adversarial training, the generated samples get more and more details from the real data.

To verify the superior performance of the 3D-GAN augmentation methods, we feed the 100 fake samples, which are generated by 3D-GAN, into the training data set. We suppose here that the original data set has N classes and that the fake samples can be endowed with the label $N+1$. Then, the real and fake samples are classified by the PCA-CNN and the 3D-GAN, which are called the Aug-PCA-CNN and the 3D-Aug-GAN, respectively. The classification results from these data augmentation methods and the original methods

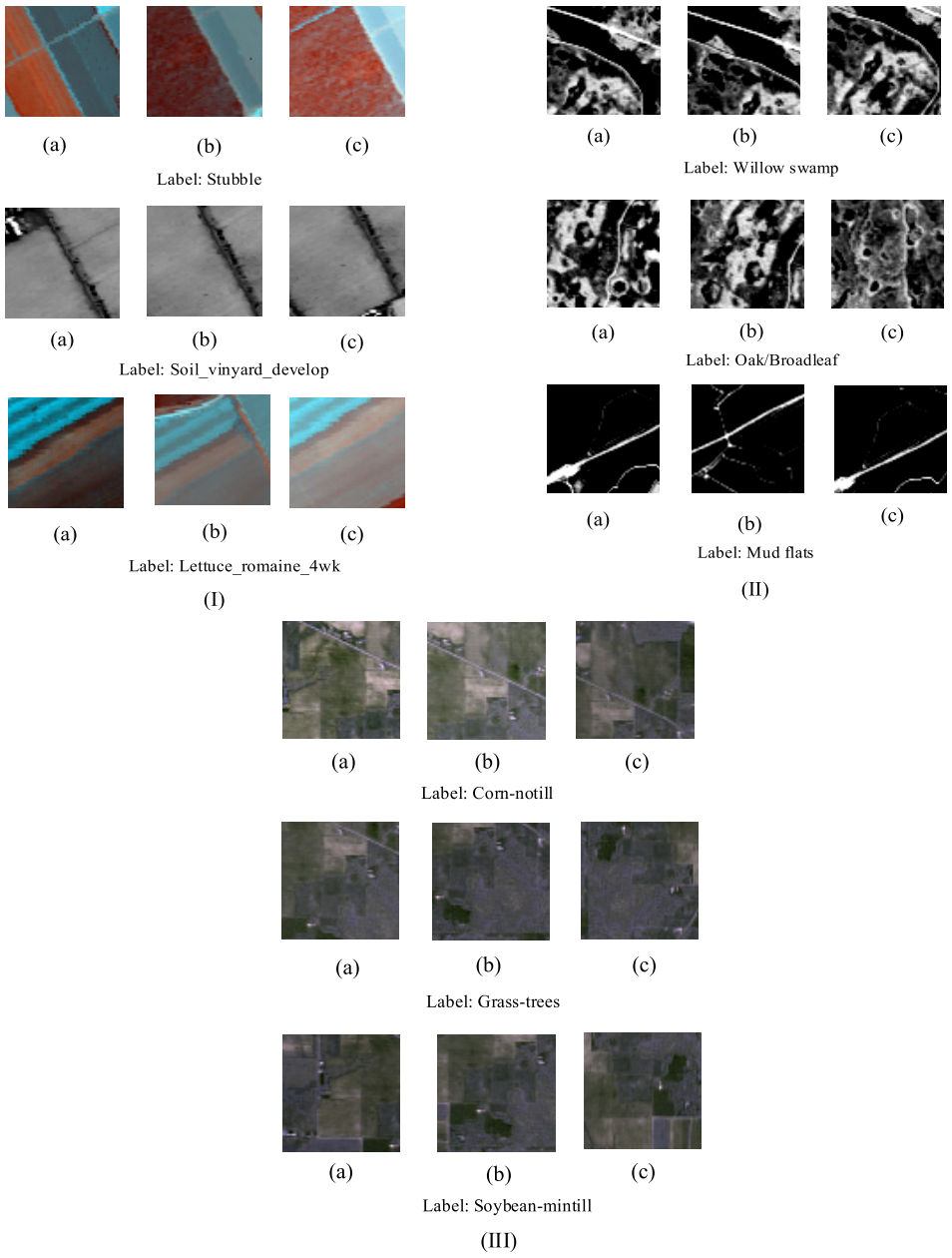


Fig. 8. Real data and generated fake data with same labels in different classes on three data sets: (I) Salinas data set, (II) KSC data set, and (III) Indian Pines data set. (a) Real training data. (b) First corresponding fake data. (c) Second corresponding fake data.

are shown in Tables XV–XVII. From Tables XV–XVII, one can see that the OA of the methods after data augmentation on three data sets outperforms the methods without data augmentation. It is obvious that the additional fake samples can improve the classification performance compared with original methods.

E. Classification Maps

At last, in this section, the classification accuracies are evaluated from a visual perspective. The methods including EMP-SVM, 3D-CNN, 3D-GAN, and 3D-Aug-GAN with virtual samples are selected to classify the whole images. Figs. 9–11 show the classification maps for different methods on the three data sets. All parameters in these models are optimized. From Figs. 9–11, one can figure out how the different methods affect the classification results. The SVM-based methods always have the most errors for the three data sets [see Figs. 9(b), 10(b), and 11(b)]. Especially for the Salinas data set [Fig. 9(b)], many pixels are misclassified at the top of the image with the SVM-based method. By making

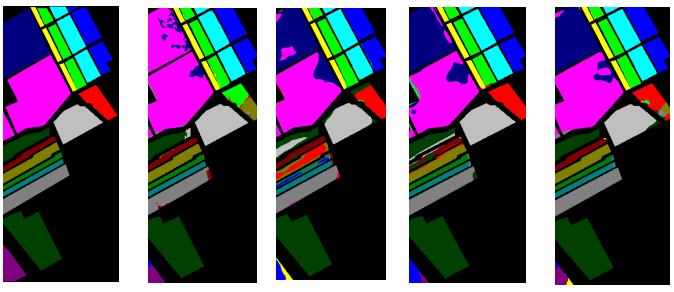


Fig. 9. Salinas: (a) Ground reference map. Classification maps for different classifiers: (b) EMP-SVM, (c) 3D-CNN, (d) 3D-GAN, and (e) 3D-Aug-GAN.

TABLE XV
CLASSIFICATION WITH DATA AUGMENTATION ON THE SALINAS DATA SET

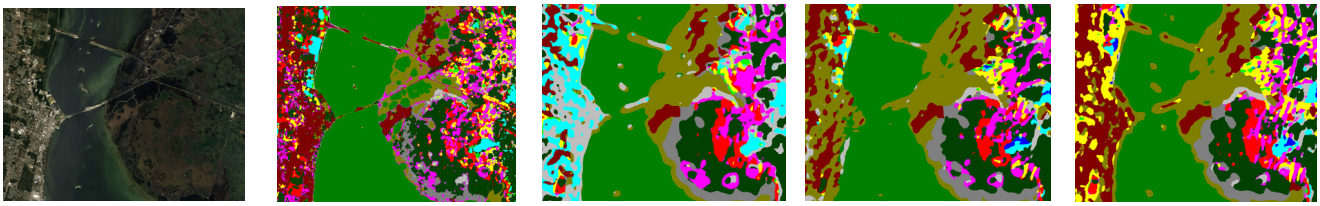
Method	PCA-CNN	Aug-PCA-CNN	3D-GAN	3D-Aug-GAN
OA(%)	91.26±0.48	92.04±0.16	93.02±1.54	93.67±0.56
AA(%)	84.47±0.43	85.02±0.36	88.15±0.39	90.89±1.31
K×100	90.88±0.52	91.13±0.19	92.07±1.22	92.55±0.96
Train time(s.)	23.30	20.84	55.10	57.15
Test time(s.)	26.45	26.63	26.56	26.12

TABLE XVI
CLASSIFICATION WITH DATA AUGMENTATION ON THE KSC DATA SET

Method	PCA-CNN	Aug-PCA-CNN	3D-GAN	3D-Aug-GAN
OA(%)	96.02±0.42	96.85±0.32	97.69 ±0.24	98.12±0.63
AA(%)	93.17±0.14	93.29±0.48	94.14±0.40	94.76±0.12
K×100	95.27±0.49	95.71±0.56	96.52±0.26	98.05±0.20
Train time(s.)	24.56	19.96	48.57	57.15
Test time(s.)	2.32	2.19	2.75	2.01

TABLE XVII
CLASSIFICATION WITH DATA AUGMENTATION ON THE INDIAN PINES DATA SET

Method	PCA-CNN	Aug-PCA-CNN	3D-GAN	3D-Aug-GAN
OA(%)	88.72±0.94	89.34±1.45	90.69 ±0.86	91.10±0.56
AA(%)	80.17±0.48	82.62±2.19	83.14±1.82	83.76±1.52
K×100	86.64±0.49	87.43±1.98	89.62±0.26	89.95±0.13
Train time(s.)	37.95	41.04	72.46	57.15
Test time(s.)	4.67	4.18	5.46	5.09



(a) (b) (c) (d) (e)

Fig. 10. KSC: (a) False-color image. Classification maps for different classifiers: (b) EMP-SVM, (c) 3D-CNN, (d) 3D-GAN, and (e) 3D-Aug-GAN.

(a) (b) (c) (d) (e)

Fig. 11. Indian Pines: (a) Ground reference map. Classification maps for different classifiers: (b) EMP-SVM, (c) 3D-CNN, (d) 3D-GAN, and (e) 3D-Aug-GAN.

the comparison between the true ground reference and the classification maps, one can see that the obtained classification results from CNNs and GANs are more precise, which show that the CNNs and GANs are promising methods for HSI classification. Furthermore, the proposed GAN with data augmentation (3D-Aug-GAN) demonstrates an excellent visual classification performance, and it works pretty well under the condition of limited raining samples.

V. CONCLUSION

In this paper, the usefulness and effectiveness of GAN for HSI classification are explored for the first time. With the help

of GANs, the deep CNN achieves better performance in terms of classification accuracy compared with that of the traditional CNN. Furthermore, the overfitting problem raised by CNNs is mitigated. In more detail, two frameworks are designed: 1) the first one, called the 1D-GAN, is based on spectral vectors and 2) the second one, called the 3D-GAN, combines the spectral and spatial features. These two architectures demonstrated excellent abilities in feature extraction and image classification compared with other state-of-the-art methods. In the proposed GANs, PCA is used to reduce the high dimensionality of inputs, which is really important to stabilize the training procedure. Due to the large number of learnable parameters (e.g., weights) in deep models, deep CNNs suffer a lot from the problem of overfitting, while GAN can be regarded as a regularization technique that can mitigate the overfitting phenomenon in the training process. Furthermore, the HSI samples that are generated by GAN are illustrated for the first time here, possibly opening a new window for generation of hyperspectral data generation. More importantly, generated adversarial samples are for the first time used as training samples for HSI classification in this paper. These samples significantly improve the classification performance. The aforementioned techniques show the huge potential of GANs for HSI classification.

REFERENCES

[1] C. Chang, *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*. Dordrecht, The Netherlands: Kluwer, 2003.

[2] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proc. IEEE*, vol. 101, no. 3, pp. 652–675, Mar. 2013.

[3] P. Ghamisi, J. Plaza, Y. Chen, J. Li, and A. J. Plaza, "Advanced spectral classifiers for hyperspectral images: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 1, pp. 8–32, Mar. 2017.

[4] G. F. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Inf. Theory*, vol. IT-14, no. 1, pp. 55–63, Jan. 1968.

[5] J. A. Gualtieri and R. F. Cromp, "Support vector machines for hyperspectral remote sensing classification," *Proc. SPIE*, vol. 3584, pp. 221–232, Jan. 1999.

[6] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[7] 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.

[8] P. Ghamisi, M. D. Mura, and J. A. Benediktsson, "A survey on spectral–spatial classification techniques based on attribute profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2335–2353, May 2015.

[9] Y. Gu, J. Chanussot, X. Jia, and J. A. Benediktsson, "Multiple Kernel learning for hyperspectral image classification: A review," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 11, pp. 6547–6565, Nov. 2017.

[10] Y. Yuan, J. Lin, and Q. Wang, "Hyperspectral image classification via multitask joint sparse representation and stepwise MRF optimization," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2966–2977, Dec. 2016.

[11] Q. Wang, Z. Meng, and X. Li, "Locality adaptive discriminant analysis for spectral–spatial classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 11, pp. 2077–2081, Nov. 2017.

[12] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[13] 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.

[14] 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.

[15] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the Art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.

[16] 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.

[17] 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. 1–12, Jan. 2015.

[18] P. Zhong, Z. Gong, S. Li, and C.-B. Schönlieb, "Learning to diversify deep belief networks for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 55, no. 6, pp. 3516–3530, Jun. 2017.

[19] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors.*, vol. 2015, Jan. 2015, Art. no. 258619.

[20] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 844–853, Feb. 2017.

[21] W. Shao and S. Du, "Spectral-spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Oct. 2016.

[22] J. Yue, W. Zhao, S. Mao, and H. Liu, "Spectral–spatial classification of hyperspectral images using deep convolutional neural networks," *Remote Sens. Lett.*, vol. 6, no. 6, pp. 468–477, 2015.

[23] 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.

[24] Y. Li, H. Zhang, and Q. Shen, "Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sens.*, vol. 9, no. 1, p. 67, 2017.

[25] H. Liang and Q. Li, "Hyperspectral imagery classification using sparse representations of convolutional neural network features," *Remote Sens.*, vol. 8, no. 2, p. 99, 2016.

[26] 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.

[27] V. Singhal, H. K. Aggarwal, S. Tariyal, and A. Majumdar, "Discriminative robust deep dictionary learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5274–5283, Sep. 2017.

[28] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. NIPS*, Montreal, QC, Canada, 2014, pp. 2672–2680.

[29] A. Radford, L. Metz, and S. Chintala. (2016). "Unsupervised representation learning with deep convolutional generative adversarial networks." [Online]. Available: https://arxiv.org/abs/1511.06434

[30] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process.*, vol. 35, no. 1, pp. 53–65, Jan. 2017.

[31] M. Mirza and S. Osindero. (2014). "Conditional generative adversarial nets." [Online]. Available: https://arxiv.org/abs/1411.1784

[32] D. Pfau and O. Vinyals. (2017). "Connecting generative adversarial networks and actor-critic methods." [Online]. Available: https://arxiv.org/abs/1610.01945

[33] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[34] I. Sutskever and G. E. Hinton, "Deep, narrow sigmoid belief networks are universal approximators," *Neural Comput.*, vol. 20, no. 11, pp. 2629–2636, Nov. 2008.

[35] N. Le Roux and Y. Bengio, "Deep belief networks are compact universal approximators," *Neural Comput.*, vol. 22, no. 8, pp. 2192–2207, Aug. 2010.

[36] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[37] N. Kruger *et al.*, "Deep hierarchies in the primate visual cortex: What can we learn for computer vision?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1847–1871, Aug. 2013.

[38] A. L. Maas, A. Y. Hannu, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Atlanta, GA, USA, 2013, pp. 1–6.

[39] M. Arjovsky, S. Chintala, and L. Bottou. (2017). "Wasserstein GAN." [Online]. Available: https://arxiv.org/abs/1701.07875
[40] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. (2017). "Unpaired image-to-image translation using cycle-consistent adversarial networks." [Online]. Available: https://arxiv.org/abs/1703.10593
[41] A. Odena. (2016). "Semi-supervised learning with generative adversarial networks." [Online]. Available: https://arxiv.org/abs/1606.01583
[42] M. Mirza and S. Osindero. (2014). "Conditional generative adversarial nets." [Online]. Available: https://arxiv.org/abs/1411.1784
[43] A. Odena, C. Olah, and J. Shlens. (2016). "Conditional image synthesis with auxiliary classifier GANs." [Online]. Available: https://arxiv.org/abs/1610.09585
[44] 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 2011.
[45] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, Lake Tahoe, CA, USA, 2012, pp. 1097–1105.
[46] D. J. Bartholomew, F. Steele, J. Galbraith, and I. Moustaki, "Analysis of multivariate social science data," *Struct. Equation Model. Multidisciplinary J.*, vol. 18, no. 4, pp. 686–693, Apr. 2011.
[47] H. Yang *et al.*, "LLE-PLS nonlinear modeling method for near infrared spectroscopy and its application," *Spectrosc. Spectral Anal.*, vol. 27, no. 10, pp. 1955–1958, Oct. 2007.
[48] W. Li, S. Prasad, J. E. Fowler, and L. M. Bruce, "Locality-preserving dimensionality reduction and classification for hyperspectral image analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 4, pp. 1185–1198, Apr. 2012.
[49] T. M. Tu, "Unsupervised signature extraction and separation in hyperspectral images: A noise-adjusted fast independent component analysis approach," *Opt. Eng.*, vol. 39, no. 4, pp. 897–906, Apr. 2000.
[50] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.
[51] J. Weston and C. Watkins, "Multiclass support vector machines," Dept. Comput. Sci., Univ. London, London, U.K., Tech. Rep. CSD-TR-98-04, 1998.
[52] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.

Lin Zhu is currently pursuing the master's degree with the Department of Information Engineering, School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin, China.

Her research interests include hyperspectral image classification, machine learning, and deep learning.

Yushi Chen (M'11) received the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 2008.

He is currently an Associate Professor with the School of Electronics and Information Engineering, Harbin Institute of Technology. His research interests include remote sensing data processing and machine learning.

Pedram Ghamisi (S'12–M'15) is currently a Research Scientist with the German Aerospace Center, Remote Sensing Technology Institute, Wessling, Germany. His research interests include remote sensing and image analysis, with a special focus on the spectral and spatial techniques for hyperspectral image classification, multisensor data fusion, machine learning, and deep learning.

Jón Atli Benediktsson (S'84–M'90–SM'99–F'04) received the Cand.Sci. degree in electrical engineering from the University of Iceland, Reykjavik, Iceland, in 1984, and the M.S.E.E. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1987 and 1990, respectively.

From 2009 to 2015, he was the Pro Rector of science and academic affairs and a Professor of electrical and computer engineering at the University of Iceland, where he became the President and the Rector in July 2015. He is a Co-Founder of the biomedical startup company Oxymap, Reykjavik. His research interests include remote sensing, biomedical analysis of signals, pattern recognition, image processing, and signal processing. He has many publications extensively in those fields.

Dr. Benediktsson is a fellow of SPIE. He was a member of the 2014 IEEE Fellow Committee. He is a member of the Association of Chartered Engineers in Iceland (VFI), Societas Scinetiarum Islandica, and Tau Beta Pi. He was a recipient of the Stevan J. Kristof Award from Purdue University in 1991 for being the Outstanding Graduate Student in remote sensing, the Icelandic Research Council's Outstanding Young Researcher Award in 1997, the IEEE Third Millennium Medal in 2000, the Yearly Research Award from the Engineering Research Institute, University of Iceland, in 2006, the Outstanding Service Award from the IEEE Geoscience and Remote Sensing Society (GRSS) in 2007, the IEEE/VFI Electrical Engineer of the Year Award in 2013, and the OECE Award from the School of Electrical and Computer Engineering, Purdue University, in 2016. He was a co-recipient of the University of Iceland's Technology Innovation Award in 2004, the 2012 IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (TGRS) Paper Award, the IEEE GRSS Highest Impact Paper Award in 2013, and the International Journal of Image and Data Fusion Best Paper Award in 2014. He has been on the GRSS AdCom since 2000. He was the 2011–2012 President of the IEEE GRSS. He was the Editor-in-Chief of the IEEE TGRS from 2003 to 2008. He has been an Associate Editor of the IEEE TGRS since 1999, the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS since 2003, and the IEEE ACCESS since 2013. He was the Chairman of the Steering Committee of IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING from 2007 to 2010. He is on the Editorial Board of the PROCEEDINGS OF THE IEEE, the International Editorial Board of the *International Journal of Image and Data Fusion*, and the Editorial Board of *Remote Sensing*.