HyperViTGAN: Semisupervised Generative Adversarial Network With Transformer for Hyperspectral Image Classification

Ziping He, Kewen Xia, Pedram Ghamisi, *Senior Member, IEEE*, Yuhen Hu, *Life Fellow, IEEE*, Shurui Fan, and Baokai Zu

***Abstract*—Generative adversarial networks (GANs) have achieved many excellent results in hyperspectral image (HSI) classification in recent years, as GANs can effectively solve the dilemma of limited training samples in HSI classification. However, due to the class imbalance problem of HSI data, GANs always associate minority-class samples with fake label. To address this issue, we first propose a semisupervised generative adversarial network incorporating a transformer, called HyperViTGAN. The proposed HyperViTGAN is designed with an external semisupervised classifier to avoid self-contradiction when the discriminator performs both classification and discrimination tasks. The generator and discriminator with skip connection are utilized to generate HSI patches by adversarial learning. The proposed HyperViTGAN captures semantic context and low-level textures to reduce the loss of critical information. In addition, the generalization ability of the HyperViTGAN is improved through the use of data augmentation. Experimental results on three well-known HSI datasets, Houston 2013, Indian Pines 2010, and Xuzhou, show that the proposed model achieves competitive HSI classification performance in comparison with the current state-of-the-art classification models.**

***Index Terms*—Generative adversarial network (GAN), hyperspectral image (HSI) classification, semisupervised learning, transformer.**

Manuscript received 30 April 2022; revised 18 June 2022 and 5 July 2022; accepted 15 July 2022. Date of publication 18 July 2022; date of current version 4 August 2022. This work was supported in part by the National Scholarship for Building High Level Universities, China Scholarship Council nder Grant 201906700002, in part by the National Natural Science Foundation of China under Grant U1813222 Grant 42075129, in part by Hebei Province Natural Science Foundation under Grant E2021202179, in part by Key Research and Development Project from Hebei Province under Grant 19210404D, Grant 20351802D, and Grant 21351803D. (*Corresponding author: Ziping He.)*

Ziping He is with the School of Electronics and Information Engineering, Hebei University of Technology, Tianjin 300401, China, and also with Helmholtz-Zentrum Dresden-Rossendorf, Helmholtz Institute Freiberg for Resource Technology, 09599 Freiberg, Germany (e-mail: hezipping1102@126.com).

Kewen Xia and Shurui Fan are with the School of Electronics and Information Engineering, Hebei University of Technology, Tianjin 300401, China (e-mail: kwxia@hebut.edu.cn; fansr@hebut.edu.cn).

Pedram Ghamisi is with the Helmholtz-Zentrum Dresden-Rossendorf, Helmholtz Institute Freiberg for Resource Technology, 09599 Freiberg, Germany, and also with the Institute of Advanced Research in Artificial Intelligence, 1030 Vienna, Austria (e-mail: p.ghamisi@gmail.com).

Yuhen Hu is with the Department of Electrical and Computer Engineering, University of Wisconsin-Madison, Madison WI 53706 USA (e-mail: hu@engr.wisc.edu).

Baokai Zu is with the Faculty of Information Technology, Beijing University of Technology, Beijing 100021, China (e-mail: zubaokai@163.com).

Digital Object Identifier 10.1109/JSTARS.2022.3192127

I. Introduction

A HYPERSPECTRAL sensor captures both spectral and spatial information of the observed target. Hyperspectral images (HSIs) are widely used in agriculture, mineralogy, urban development, scene interpretation, resource management, and other fields. HSI classification is a vibrant topic in the field of remote sensing [1], [2]: because HSIs have rich spectral bands and complex spatial structure with high-dimensional features and a limited number of labeled samples, HSI classification remains a challenging task [3].

HSI classification methods are broadly classified as unsupervised and supervised. Unsupervised methods group similar pixels together according to feature information that represents the characteristics of the pixels. Supervised methods use annotated hyperspectral pixels to learn the intrinsic association between pixel features and categories to classify unannotated pixels and determine pixel classes. Supervised hyperspectral classification methods are further grouped into methods based on spectral information and methods based on spectral-spatial information.

Over the past decades, numerous supervised machine learning algorithms have been used for HSI classification, such as support vector machines (SVMs) [4], [5], random forests [6], K nearest neighbors (KNN) [7], logistic regression [8], and neural network [9]. To date, many excellent traditional machine learning methods have been proposed from time to time in the field of hyperspectral image processing and have shown excellent classification performance [10]–[15].

Deep models have proven to be a powerful tool for data mining and analysis, especially in remote sensing, which is typically a data-intensive discipline; these models have become an indispensable tool for remote sensing data processing. For example, a novel recurrent network (RNN) that can efficiently analyze HSI pixels as sequence data was first proposed by Mou *et al.* [16] and has yielded competitive results. In [17], a semisupervised Siamese network integrating an autoencoder and a Siamese network was proposed. This network was trained simultaneously on massive unannotated samples and few annotated samples to obtain unsupervised feature with refinement representation and unsupervised features rectified by limited labeled samples, respectively. Tan *et al.* [18] proposed a deep HSI feature extraction method with multiple Gaussian–Bernoulli restricted Boltzmann machines (GBRBMs) with different hidden layers in parallel.

Shi *et al.* [19] proposed a dual-attention denoising network considering both the global dependence and correlation of spectral and spatial information for HSI denoising. Roy *et al.* [20] first introduced an attention-based adaptive spectral-spatial kernel component to obtain selective 3-D convolutional kernels, capturing differentiated spectral-spatial HSI features with end-to-end training resulted in better classification results compared to the existing methods they investigated. In [21], a cascaded RNN model consisting of two RNN layers for exploring redundant and complementary information of HSIs with gated recurrent units (GRUs) was proposed. Hong *et al.* [22] proposed a minibatch graph convolutional network (GCN) called miniGCN that can train large-scale GCNs in the form of minibatch and showed superior advantages of miniGCN over GCNs on HSI classification. A deeper and wider network called contextual deep CNN was proposed by Lee *et al.* [23] for HSI classification. Zhong *et al.* [24] proposed an end-to-end spectral-spatial residual network (SSRN) for HSI classification, which mitigates the declining-accuracy phenomenon of other deep learning models.

Despite the blossoming of machine learning and deep models for HSI classification, these algorithms are susceptible to the curse of dimensionality, also called the Hughes phenomenon [25]. Additionally, deep models often require plenty of training samples, and HSI annotation is time consuming and labor intensive, so HSI only has a limited number of annotated samples. As a result, deep models easily face the issue of overfitting. There are various solutions for overfitting and the curse of dimensionality, such as feature extraction [26], dropout [27], regularization, and data augmentation. The generator within GAN can be seen as a method of data augmentation. Therefore, GAN [28] is starting to emerge in HSI classification.

The generator and the discriminator are the two main components that constitute the GAN model, and they explore the real data distribution through a constant competitive game between the generator and the discriminator. The first approach to classify HSIs using GAN was proposed by Zhu *et al.* [29] in 2018, which proposed a total of two schemes, 1-D GAN and 3-D GAN, ultimately improving the classification performance and exploiting the potential of GAN for HSI classification. Wang *et al.* [30] proposed Caps-TripleGAN for HSI classification by exploring triple generative adversarial networks (TripleGAN) and combining capsule network (CapsNet). Another GAN, combining CapsNet and convolutional long short-term memory (ConvLSTM), was designed by Wang *et al.* [31] to tackle the insufficiency of annotated samples in HSI by generating artificial HSI samples for data augmentation. Wang *et al.* [32] proposed a dropblock-enhanced generative adversarial network (ADGAN) to address the model collapse problem of GANs and the persistent and challenging class imbalance of hyperspectral data, achieving remarkable performance compared to the state-of-the-art GAN-based models. Similarly, Roy *et al.* [33] proposed a novel 3D-HyperGAMO model, which employs generative adversarial minority oversampling to conquer the class imbalance issue in HSI. In the training process, the existing samples of the minority-class are used by 3D-HyperGAMO to automatically generate more minority-class samples, which tackles the class imbalance by leveraging the oversampling strategy.

Although the aforementioned GANs combined with convolutional neural networks (CNNs) and RNNs achieve competitive results in HSI classification, some limitations remain in their approaches to targeting of sequence data. Capturing the sequence attributes well in HSIs with their many categories and extremely similar spectral features is difficult for CNNs; furthermore, spatial information is given too much attention, distorting the sequence information in the learned features on the spectrum. And, even though RNNs was designed for sequential data, represented by long short-term memory (LSTM) [34] and GRUs [35]. RNNs are able to extract rich contextual semantics from sequential data like sequential networks. However, the effective spectral information in RNNs is stored in individual fragmented neurons, which cannot effectively preserve the ultralong data dependencies. Furthermore, the sequential network structure makes it difficult to efficiently scale and parallelize the computation of LSTM and GRUs. The emergence of the transformer [36] has successfully addressed the shortcomings of the CNN in capturing long-range information. Compared to RNNs, transformer allows parallel computation, which reduces training time and performance degradation due to long-term dependencies. The number of manipulations needed to calculate the correlation between two positions do not grow with distance, and its self-attention module captures long-range information more easily than the CNN, making the transformer one of the most cutting-edge models today. A vision transformer (ViT) [37] demonstrates that the transformer not only excels in natural language processing (NLP) but also achieves outstanding performance in image classification. The current cutting-edge transformer backbone network also shows excellent performance in the field of HSI classification. For example, Hong *et al.* first used the ViT for HSI classification by proposing a network called SpectralFormer [38], which solved the flaws of the ViT in the weakness of local detail spectral representation and jump connection design, and ultimately achieved a significant 10% improvement in overall classification accuracy over the classical ViT, which strongly demonstrated the potential of the transformer in HSI classification. Since the transformer has shown competitive performance in computer vision, the authors in [39] used the transformer for a more difficult vision task, GAN. The authors in [39] proposed the first pure transformer-based GAN with no convolution at all, named TransGAN, which achieved competitive results when compared with the current cutting-edge CNN-based GANs. Immediately afterward, the authors in[40] proposed the ViTGAN to solve the training stability problem that arises when combining GAN with the transformer. The ViTGAN was shown to yield comparable performance to the most advanced CNN-based StyleGAN2 [41].

In this article, we first propose a novel semisupervised GAN-based model, HyperViTGAN, for the HSI classification task, in combination with the cutting-edge and promising transformers currently used for HSI classification. Three well-designed hyperspectral ViT-based cascaded elements—a generator, discriminator, and external classifier—are designed to constitute HyperViTGAN. The design of a discriminator with a single discrimination output and an external classifier with a single classification output effectively eliminates self-contradiction when the

discriminator performs both classification and discrimination tasks. In addition, because the HyperViTGAN is specifically designed for HSI, it can better preserve spectral sequence information in order to avoid the loss of critical information. At the same time, HyperViTGAN yields better generalization ability via data augmentation. Three well-known HSI datasets, Indian Pines 2010, Houston 2013, and Xuzhou, are used to quantitatively and qualitatively validate the classification performance of HyperViTGAN, which outperforms the current state-of-the-art models for the HSI classification task. Our contribution to this work is as follows.

1) HyperViTGAN, a GAN-based entirely on transformers for HSI is first proposed. A discriminator and an external semisupervised classifier that do not share the architecture with each other are designed for a single discrimination task and a single classification task in HyperViTGAN, respectively. HyperViTGAN can generate hyperspectral HSI patches by adversarial learning and semisupervised learning while alleviating the challenge of class imbalance in HSI.
2) Cascaded architecture with skip connections are designed for a generator, discriminator, and classifier to deliver memory-like information, thus avoiding the loss of key components and boosting classification performance.

The rest of this article is organized as follows. Section II focuses on theoretical background related to GAN and transformer. Our proposed HyperViTGAN is introduced in detail in Section III. The specific experimental details and analysis are reported in Section IV. Finally, Section V concludes this article.

II. Related Works

A. Generative Adversarial Networks (GANs)

GANs [28] learn by pitting two neural networks against each other. Minimizing the divergence of distribution between the generated data p_z and the real data p_{data} is the target of GAN, e.g., via variety of f-divergences [42] or integral probability metrics (IPMs) [43]–[45].

The generator G and discriminator D are the two main elements that form the GAN model. The main purpose of G in GAN is to generate samples similar to real samples in order to fool D. The input of D consists of two parts, real samples and generated samples, and D aims to judge whether the input sample is real samples or generated samples generative by G. The GAN converges when D and G reach Nash equilibrium in the game theory, i.e., D cannot judge whether its input is real samples or generated samples generated by G. At this point, it can be assumed that G learned the distribution of real samples.

To enable the generator to learn on the data $\mathbf{X}_{\text{real}}$, random noise z is fed to the generator G and a mapping of the data space $\mathbf{X}_{\text{fake}} = G(z))$ is generated. Immediately afterward, D computes the probability that $\mathbf{X}_{\text{real}}$ is the real samples from training set and outputs a probability distribution $P(S|\mathbf{X}) = D(\mathbf{X})$ over the input samples. Therefore, maximizing the log-likelihood of its assignment to the right source is the ultimate aim of D

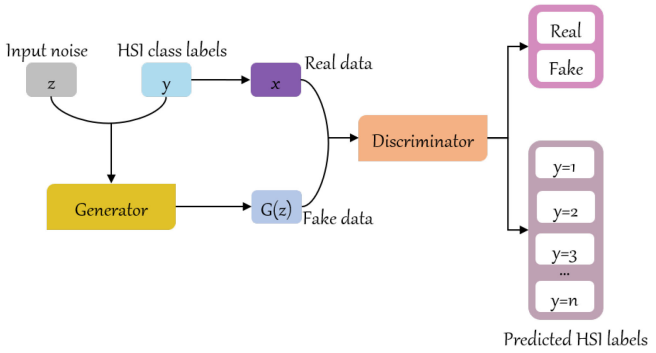

Fig. 1. Architecture of ACGAN employed in [29] for the HSI classification task. $y \in \{1, 2, 3, \ldots, n\}$ denote the HSI class labels. n represents the number of HSI classes.

$$L = E[\log P(S = \text{real}|\mathbf{X}_{\text{real}})] + E[\log P(S = \text{fake}|\mathbf{X}_{\text{fake}})]. \quad (1)$$

The use of side information applied to the GAN can effectively enhance existing generative models, thus forming two means of optimization. The first is to use auxiliary label information to augment the original GAN, and train both the generator and discriminator using labeled data, i.e., CGAN [46]. The CGAN was developed to better control the GAN using side information. It adds some prior conditions to the initial GAN model to make the GAN more controllable. Specifically, CGAN adds conditional constraint c to both G and D to guide the data generation process. The second way is to directly reconstruct the side information by modifying the discriminator to include an auxiliary decoder network, thereby improving the generation effect of the GAN, i.e., SGAN [47], [48]. Combining the advantages of the aforementioned two approaches, the ACGAN [49] shows that incorporating more architecture as well as specialized loss functions in the latent space of the GAN yields high-quality samples. The ACGAN is used in [29] for HSI classification; its framework is shown in Fig. 1. The generator of the ACGAN has two inputs, conditional constraint c and random noise z, and outputs the generated data $\mathbf{X}_{\text{fake}} = G(c, z)$. D performs two tasks simultaneously: to determine the probability distribution of whether the input data is true or not and the probability distribution of categories, $P(S|\mathbf{X}), P(O|\mathbf{X}) = D(\mathbf{X})$. The objective function of the ACGAN contains two parts, the first oriented to the loss function of whether the data are true or not, and the second part to the loss function of the data classification accuracy. The ultimate objective function of the ACGAN consists of two parts, L_S and L_O. Therefore, the objective function of D and G is to maximize $L_S + L_O$ and $L_O - L_S$, respectively .

$$L_S = E[\log P(S = \text{real}|\mathbf{X}_{\text{real}})] + E[\log P(S = \text{fake}|\mathbf{X}_{\text{fake}})] \quad (2)$$

$$L_O = E[\log P(O = \text{class}|\mathbf{X}_{\text{real}}] + E[\log P(O = \text{class}|\mathbf{X}_{\text{fake}})] \quad (3)$$

where L_S and L_O represent the log-likelihood of the exact origin and categories, respectively.

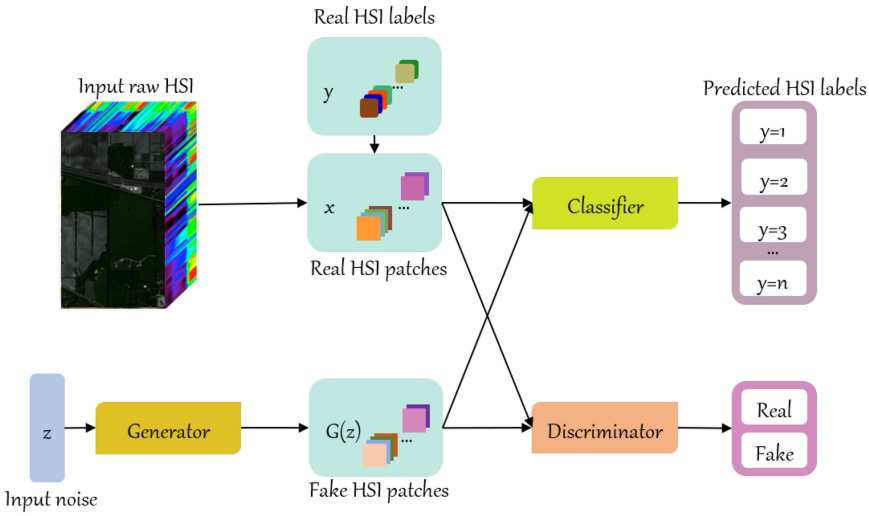

Fig. 2. Overview of the proposed HyperViTGAN for the HSI classification task.

B. Vision Transformer (ViT)

A ViT [37] is a convolution-free model proposed in 2020 to directly apply the transformer to image classification. The ViT divides images into many equally sized patches and obtains the patch embedding by a linear transformation. The specific realization of the ViT is described as follows.

The 2-D image $\mathbf{X} \in \mathbb{R}^{H\times W\times I}$ is split into a sequence of patches $\mathbf{X}_p \in \mathbb{R}^{N\times(P^2\cdot I)}$ in order to adapt the input of the transformer, where $N = \frac{H\times W}{P^2}$ denotes the resulting number of patches and $P^2 \cdot I$ is the dimension of each patch. H, W, I, and P represent height, width, number of channels, and the patch size of the image, respectively.

$\mathbf{E} \in \mathbb{R}^{P^2\cdot I\times d}$ and $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{(N+1)\times d}$ are a patch embedding and a position embedding, respectively. The number of units embedded in the spectrum is denoted by d. $\mathbf{X}_{\text{class}}$ denotes a learnable classification embedding. We prepend $\mathbf{X}_{\text{class}}$ to the sequence of embedded patches ($\mathbf{h}_0^0 = \mathbf{X}_{\text{class}}$), whose state at the output of the transformer encoder ($\mathbf{h}_{\mathcal{L}}^0$) serves as the patch representation $\mathbf{y}$ [see (7)]. $\mathbf{y}$ is the patch representation of $1 \times n$ dimensions, which is calculate by the multiple-layer perceptron (MLP) head (classification head) including LayerNorm (LN) and Linear layer. The ViT is dominated by MLP, multiheaded self-attention (MSA) and LN modules. The embedded patches $\mathbf{h}_0$ consists of a learnable classification embedding $\mathbf{X}_{\text{class}}$ and a 1-D positional embedding $\mathbf{E}_{\text{pos}}$ according to [50]. The architecture of the ViT is as follows:

$$\mathbf{h}_0 = [\mathbf{X}_{\text{class}}; \mathbf{X}_p^1\mathbf{E}; \mathbf{X}_p^2\mathbf{E}; \ldots; \mathbf{X}_p^N\mathbf{E}] + \mathbf{E}_{\text{pos}} \tag{4}$$

$$\mathbf{h}'_\ell = \text{MSA}(\text{LN}(\mathbf{h}_{\ell-1})) + \mathbf{h}_{\ell-1} \tag{5}$$

$$\mathbf{h}_\ell = \text{MLP}(\text{LN}(\mathbf{h}'_\ell)) + \mathbf{h}'_\ell \tag{6}$$

$$\mathbf{y} = \text{LN}(\mathbf{h}_{\mathcal{L}}^0,) \tag{7}$$

where $\mathbf{E} \in \mathbb{R}^{(P^2\cdot I\times d)}$, $\ell = 1, \ldots, \mathcal{L}$.

The query, key, and value representation are represented by the learnable matrices $\mathbf{W}_q \in \mathbb{R}^{d\times d_k}$, $\mathbf{W}_k \in \mathbb{R}^{d\times d_k}$, and $\mathbf{W}_v \in \mathbb{R}^{d_k\times d_v}$, respectively. Equation (8) shows the calculation of a single self-attention head (indexed by j).

$$\text{Attention}_j(\mathcal{X}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_j}}\right)\mathbf{V} \tag{8}$$

where $\mathbf{Q} = \mathcal{X}\mathbf{W}_q$, $\mathbf{K} = \mathcal{X}\mathbf{W}_k$, and $\mathbf{V} = \mathcal{X}\mathbf{W}_v$, $\mathcal{X} = \text{LN}(\mathbf{h}_\ell), \ell = 1, \ldots, \mathcal{L}$ represents the input of the transformer encoder.

In (9), MSA in (5) integrates information derived from J self-attention heads via concatenation and linear projection.

$$\text{MSA}(\mathcal{X}) = \text{concat}_{j=1}^{J}[\text{Attention}_j(\mathcal{X})]\mathbf{W} \tag{9}$$

where $\mathbf{W} \in \mathbb{R}^{d_v\times d}$ denotes the transformation matrix.

III. Methodology

In this section, the diagram of our HyperViTGAN is first illustrated in Fig. 2, and the design of the three cascaded ViT-based operation (i.e., discriminator, generator, and classifier) is then introduced. We introduce the following techniques to both the discriminator and generator to make them apply well to the highly accurate classification of HSIs:

1) design of hyperspectral discriminator and generator;
2) newly designed external classifier;
3) cross-layer adaptive fusion.

A. Discriminator Design

The discriminator of the ACGAN [29], [49] performs two different tasks at the same time, classification and discrimination, so there are some flaws in the design of the two losses. First, a discriminator of a single architecture cannot be optimal on two different tasks at the same time. Second, the loss function designed by the ACGAN for D is defective in generating minority-class samples. This design causes minority-class samples to be identified as fake samples by D. Therefore, the discriminator always treats minority-class samples as fake, damaging the

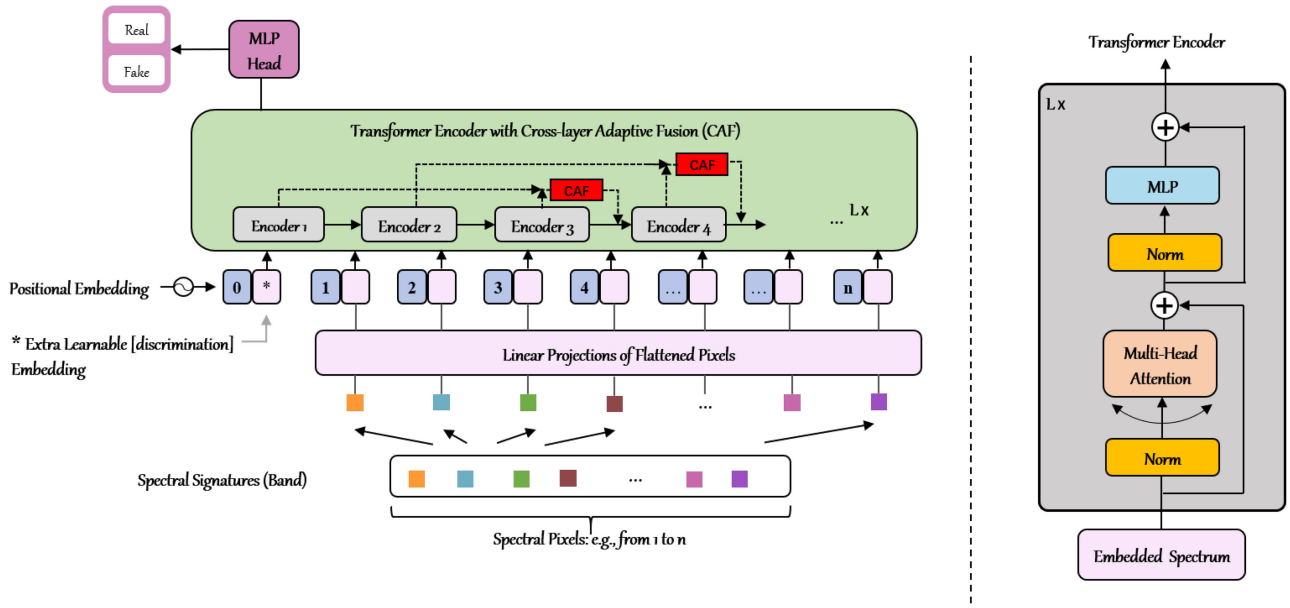

Fig. 3. Discriminator architecture of the proposed HyperViTGAN for HSI classification task. We take the HSI pixel points as input of D and perform a pixel linear embedding. The positional embedding is added and the resulting vector sequence is fed into a standard transformer encoder. The diagram on the left shows the model overview of D. On the right, we show the specific flow of the transformer block.

classification performance. Unlike the discriminator design in the ACGAN, we design the hyperspectral discriminator (HyperD) as a single output for the discrimination task only so that the discriminator does not contradict itself. Ultimately, the discriminator training is designed to maximize (10).

$$L_D = E[\log P(S = \text{real}|\mathbf{X}_{\text{real}})] + E[\log P(S = \text{fake}|\mathbf{X}_{\text{fake}})] \tag{10}$$

where $\mathbf{X}_{\text{real}}$ and $\mathbf{X}_{\text{fake}} = G(z)$ denote the real HSI patches and fake HSI patches generated by the generator, respectively. $P(S|\mathbf{X}) = D(\mathbf{X})$ represents the probability distribution of the discriminator for the real HSI patches $\mathbf{X}_{\text{real}}$ and fake HSI patches $\mathbf{X}_{\text{fake}}$.

The discriminator in our proposed HyperViTGAN is used only to judge the source of the input HSI patches. D is trained to maximize log-likelihood in (10) to its own ability to correctly assign sources of HSI.

The architecture of the HyperD in HyperViTGAN is designed based on [40]. HyperD is entirely a pure ViT-based network. Unlike the discriminator in [40], HyperD only discriminates the authenticity of the input and does not perform classification. The overview framework of the discriminator is shown in Fig. 3.

Next, two main modules used in HyperD, cross-layer adaptive fusion (CAF) and data augmentation, are introduced.

1) Cross-Layer Adaptive Fusion (CAF): Since the skip connection in the transformer is only used in a single block, this weakens the connections between different layers. Therefore, the CAF module is introduced to strengthen the association between different layers or blocks of the transformer [38]. A diagram of the CAF is shown in Fig. 4. The CAF is a module designed for learning cross-layer feature fusion, which uses a middle-range skip connection (SC) mechanism. Let $\mathbf{h}_{\ell-2} \in \mathbb{R}^{1\times d}$ and $\mathbf{h}_{\ell} \in \mathbb{R}^{1\times d}$ be the outputs (or representations) in the $(\ell-2)$th and (ℓ)th layers, respectively. d is set to 128. CAF can be then expressed by

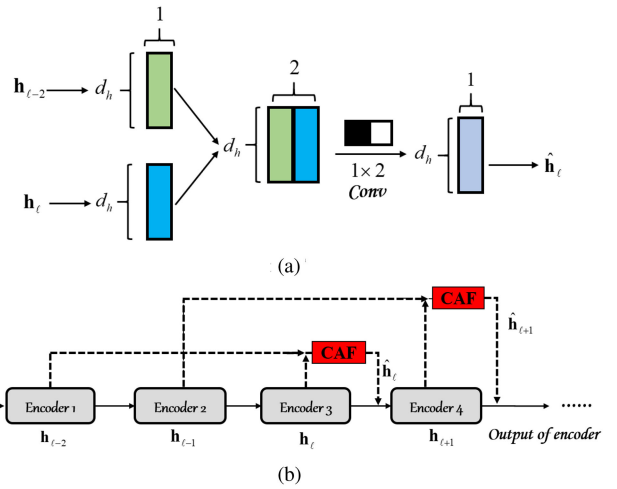

Fig. 4. Diagram of CAF module [38]. (a) Architecture of the CAF module. (b) CAF module in the HyperViTGAN.

$$\hat{\mathbf{h}}_{\ell} \leftarrow \ddot{\mathbf{w}} \begin{bmatrix} \mathbf{h}_{\ell} \\ \mathbf{h}_{\ell-2} \end{bmatrix} \tag{11}$$

where $\hat{\mathbf{h}}_{\ell}$ is the integrated representation in the (ℓ)th block with CAF, and $\ddot{\mathbf{w}} \in \mathbb{R}^{1\times 2}$ represents the parameter for adaptive fusion.

2) Data Augmentation: In order to reduce the GAN's memory and sensitivity to adversarial samples, we introduce mixup [51] for data augmentation of real samples. Specifically, the mixup mechanism trains the network model on convex

combinations of paired samples and their labels in order to regularize the network, which can increase the model's ability of generalize, and improve its robustness against adversarial attacks. We then design the GAN by linearly superimposing the real spectrometric characteristics by mixup during the training of the discriminator to get the augmented data, which effectively improves the generalization ability of the GAN on HSI. The labels for the augmented HSI data were obtained based on the calculation obtained in [51].

B. Generator Design

We followed the generator architecture in [40] to design a hyperspectral generator (HyperG) for the HSI generation task. Unlike in [40], we use HSI pixel embedding instead of patch embedding. This design enables the generator to better adapt to the hyperspectral data. The generator G generates samples that can fool the discriminator D. For this purpose, the HyperG G is trained to maximize as follows:

$$L_G = E[\log P(S = \text{real}|\mathbf{X}_{\text{fake}})]. \tag{12}$$

G strives to enable the generated $\mathbf{X}_{\text{fake}}$ to be assigned by D to a label belonging to class c, while the D aims to accurately identify the $\mathbf{X}_{\text{fake}}$ as the new class of fake. Through adversarial learning between G and D, G is able to generate fake data that D cannot discriminate as real or fake.

In this section, we design a generator specifically for HSI. A linear projection $\mathbf{E}_G \in \mathbb{R}^{d\times(B\cdot p^2)}$ is learned to generate HSI pixel vectors. Note that p denotes the size of the each pixel in the HSI cube, and p has a size of 1. $\mathbf{E}$ maps the d-dimensional output embedding to the HSI patch of $B \times p \times p$. The sequence of $N = \frac{W\times H}{p^2}$ HSI pixels $[\mathbf{X}_p^i]_{i=1}^N \in \mathbb{R}^{p^2\times B}$ is finally reshaped to form a whole HSI patch $\mathbf{X} \in \mathbb{R}^{W\times H\times B}$. The principles of the generator are shown as follows:

$$\mathbf{h}_0 = \mathbf{E}_{\text{pos}} \tag{13}$$

$$\mathbf{h}'_\ell = \text{MSA}(\text{SLN}(\mathbf{h}_{\ell-1}, \mathbf{w})) + \mathbf{h}_{\ell-1} \tag{14}$$

$$\mathbf{h}_\ell = \text{MLP}(\text{SLN}(\mathbf{h}'_\ell, \mathbf{w})) + \mathbf{h}'_\ell \tag{15}$$

$$\mathbf{y} = \text{SLN}(\mathbf{h}_{\mathcal{L}}, \mathbf{w}) = [\mathbf{y}^1, \ldots, \mathbf{y}^N] \tag{16}$$

$$\mathbf{X} = [\mathbf{X}_p^1, \ldots, \mathbf{X}_p^N] = [f_\theta(\mathbf{E}_{\text{fou}}, \mathbf{y}^1), \ldots, f_\theta(\mathbf{E}_{\text{fou}}, \mathbf{y}^N)] \tag{17}$$

where $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{N\times d}$ denotes the patch embedding and positional embedding unit in generator, $i = 1, \ldots, N$ represents the current sequence, and $\mathbf{w} \in \mathbb{R}^d$ is the intermediate latent embedding from random vector $\mathbf{z}$. $\mathbf{E}_{\text{fou}} \in \mathbb{R}^{p^2\cdot D}$ and $f_\theta(\cdot,\cdot)$ denote a Fourier encoding of $p \times p$ spatial locations and a two-layer MLP, respectively.

The SLN in (14) is calculated by

$$\begin{aligned}\text{SLN}(\mathbf{h}'_\ell, \mathbf{w}) &= \text{SLN}(\mathbf{h}_\ell, \text{MLP}(\mathbf{z})) \\ &= \gamma_\ell(\mathbf{w}) \odot \frac{\mathbf{h}_\ell - \boldsymbol{\mu}}{\boldsymbol{\sigma}} + \beta_\ell(\mathbf{w}).\end{aligned} \tag{18}$$

Among them, the symbol $\odot$ is the element-wise dot product. The mean and variance of the sum of inputs within the layer are tracked by $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, respectively. The adaptive normalization parameters governed by the noise $\mathbf{z}$ is calculated by γ_ℓ and β_ℓ.

Eventually, the HSI pixel values $\mathbf{X}_p^i \in \mathbb{R}^{p^2\times B}$ are mapped from an HSI patch embedding $y^i \in \mathbb{R}^d$ by an implicit neural representation unit, as shown in (17).

It is worth pointing out that the CAF module is also used in the design of the generator to strengthen the association between different blocks and reduce the extent to which key information is forgotten.

C. Classifier Design

While the design of a discriminator that performs both classification and discrimination tasks concurrently achieves an excellent classification performance on the HSI classification task [29], [32], such a design forces the discriminator to converge to separate data distributions for both classification and discrimination tasks, thereby corrupting the overall HSI classification performance of the GAN-based model. GANs that share a single architecture for classification and discrimination fail to solve the HSI class imbalance problem properly, and the discriminator is prone to associating fakes with minority-class(es), which results in weak abilities to classify minority-class(es) samples. Haque [52] used GAN and a semisupervised algorithm to supplement the supervised classifier with artificial data. This algorithm, named EC-GAN, was proved to be effective on small, realistic datasets. It consists of three parts: generator, discriminator, and classifier. The classifier does not share its architecture with the discriminator, which avoids self-contradiction when performing both discrimination and classification. Inspired by the aforementioned problems, we designed a semisupervised ViT-based cascaded external classifier C, called hyperspectral classifier (HyperC), specifically for HSI classification.

$$\begin{aligned}L_C = {} & E[\log P(O = \text{class}|\mathbf{X}_{\text{real}})] \\ & + \lambda E[\log P(O = \text{class}|\mathbf{X}_{\text{fake}})] \\ \text{argmax}(\mathbf{X}_{\text{fake}}) = {} & \text{argmax}(C(\mathbf{X}_{\text{fake}})) > t\end{aligned} \tag{19}$$

where $P(O|\mathbf{X}) = C(\mathbf{X})$ represents the probability distribution of the classifier C to assign sample $\mathbf{X}_{\text{real}}$ and $\mathbf{X}_{\text{fake}}$ to each specific HSI class, denoted $class$; λ denotes the unsupervised loss weight; and t is the confidence threshold of the pseudolabel of $\mathbf{X}_{\text{fake}}$.

In (19), the first term is supervised loss using real HSI samples and real HSI labels. The second term is the unsupervised loss in fake HSI samples and their corresponding pseudolabels. High-quality fake HSI samples are selectively selected to supplement supervised HSI classification. Due to the presence of the weight λ, the fake samples do not contribute much to the model update and classifier loss calculation. We set λ to 0.1 followed [52]. A smaller λ ensures that the model still learns mainly from real HSI samples, while the model is fine tuned using the high-quality fake HSI samples. The selection of high-quality fake samples is also critical to the training of the model. We followed the design of [52] and [53] and employed a confidence-based pseudolabeling scheme. The weak initial generation capability of the GAN leads to generation of low-quality samples by the generator. The pseudolabel confidence threshold t successfully prevents low-quality samples from joining the training process

TABLE I
LAND COVER OF THE HOUSTON 2013 DATASET, WITH THE STANDARD TRAINING AND TESTING SETS FOR EACH CLASS

Class No.	Class Name	Training	Testing
1	Healthy Grass	198	1053
2	Stressed Grass	190	1064
3	Synthetic Grass	192	505
4	Tree	188	1056
5	Soil	186	1056
6	Water	182	143
7	Residential	196	1072
8	Commercial	191	1053
9	Road	193	1059
10	Highway	191	1036
11	Railway	181	1054
12	Parking Lot1	192	1041
13	Parking Lot2	184	285
14	Tennis Court	181	247
15	Running Track	187	473
Total		2832	12197

TABLE II
LAND COVER OF THE INDIAN PINES 2010 DATASET, WITH THE STANDARD TRAINING AND TESTING SETS FOR EACH CLASS

Class No.	Class Name	Training	Testing
1	Corn: high	726	2661
2	Corn: mid	465	1275
3	Corn: low	66	290
4	Soybean-high	324	1041
5	Soybean-mid	2548	35317
6	Soybean-low	1428	27782
7	Residues	368	5427
8	Wheat	182	3205
9	Hay	1938	48107
10	Grass/Pasture	496	5048
11	Cover crop 1	400	2346
12	Cover crop 2	176	1988
13	Woodlands	1640	46919
14	Highway	105	4758
15	Local road	52	450
16	Buildings	40	506
Total		10954	187120

TABLE III
LAND COVER OF THE XUZHOU DATASET, WITH THE RANDOMLY SELECTED TRAINING AND TEST SETS FOR EACH CLASS

Class No.	Class Name	Training	Testing
1	Bareland1	10	26386
2	Lakes	10	4017
3	Coals	10	2773
4	Cement	10	5204
5	Crops-1	10	13174
6	Tress	10	2426
7	Bareland2	10	6980
8	Crops	10	4767
9	Red-title	10	3060
Total		90	68787

of the classifier model. As the generative power of the GAN gradually improves with training, more and more high-quality fake samples will be adopted into the GAN training process, bypassing the confidence threshold t. The unsupervised loss is obtained by calculating the fake samples and their corresponding pseudolabels. Here, t is set to 0.7, following [52].

IV. EXPERIMENTS

In this section, three typical HSI datasets are briefly introduced, followed immediately by the implementation details of the state-of-the-art models used for comparison. Finally, plenty of ablation and comparison experiments are shown to evaluate both quantitative and qualitative HSI classification performance of our HyperViTGAN.

A. Data Description

1) Houston 2013: This dataset was captured by ITRES CASI-1500 on June 23, 2012 over the University of Houston campus and adjacent urban areas in Texas, USA [54]. It comprises 349×1905 pixels and has 144 wavelength bands ranging from 364 to 1046 nm at 10-nm intervals. We reserved only ten principal components of the Houston 2013 dataset as spectral bands for our experiments using principal component analysis (PCA) [55]. The Houston 2013 dataset we adopted is a cloud-free version.[1] Table I lists 15 classes of interest and the division of training and test samples. Fig. 6 demonstrates the color composite representation and the training and test samples we used.

2) Indian Pines 2010: This dataset (see Fig. 7) was collected on May 24 and 25, 2010 by using the ProSpecTIR system in a region around Purdue University in West Lafayette, Indiana, USA. A subset of 445×750 pixels is used for our experiments, which has a spatial resolution of 2 m, a spectral width of 5 nm, and 360 spectral bands. Only ten principal components in the Indian Pines 2010 dataset were retained as spectral bands using PCA. There are 16 land cover classes in this studied scene (see Table II). Table II lists the class names and training and test samples used in the experiments.

3) Xuzhou: This dataset was captured in November 2014 by the airborne HYSPEX hyperspectral camera in the area around Xuzhou, China. The spatial resolution of Xuzhou is 0.73 m/pixel with 500×260 pixels 436 spectral bands. Nine classes are included in this dataset. It is worth mentioning that we use PCA to reduce the number of spectral bands to ten principal components for the Xuzhou dataset. Table III lists nine classes and the division between training and test samples used for the experiments. Ten samples are randomly selected from each class to form the training set, and the rest constitute the test set. Fig. 8 illustrates the color composite representation of Xuzhou dataset and its training and test samples.

B. Experimental Setup

1) Evaluation Metric: Three well-known evaluation metrics: *Overall accuracy (OA), average accuracy (AA), and Kappa coefficient (k)* are used to evaluate the HSI classification performance of our HyperViTGAN and comparison models.

[1]The data were provided by Prof. N. Yokoya from the University of Tokyo and RIKEN AIP.

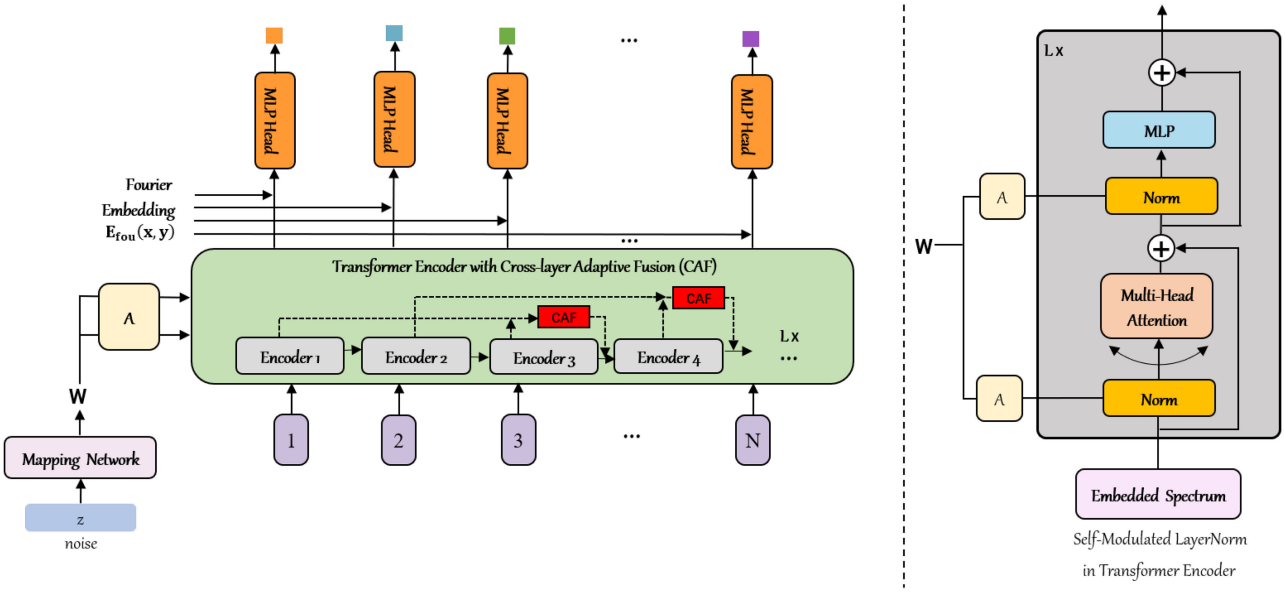

Fig. 5. Generator architecture of the proposed HyperViTGAN for the HSI classification task. We follow [40] to replace the normalization with the self-modulated layernorm (SLN) computed from the affine transform A learned from $\mathbf{w}$. The diagram on the left shows the overall architecture of the generator for generating HSI pixel points using random noise z. The right side of the diagram illustrates the details of the self-modulation operation applied in the transformer block.

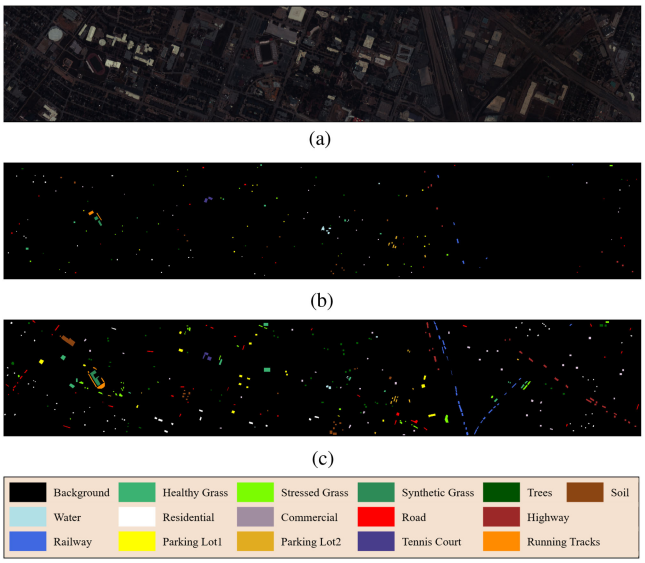

Fig. 6. Houston 2013 dataset. (a) RGB composition. (b) Training set. (c) Test set.

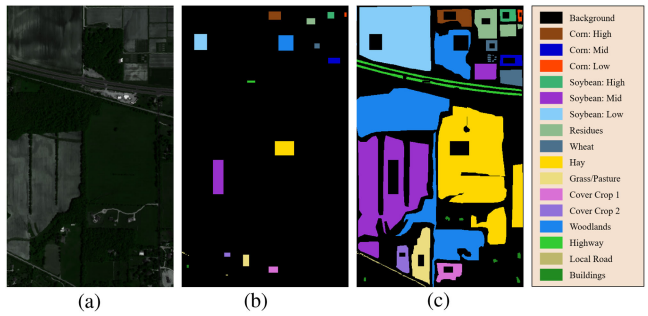

Fig. 7. Indian Pines 2010 dataset. (a) RGB composition. (b) Training set. (c) Test set.

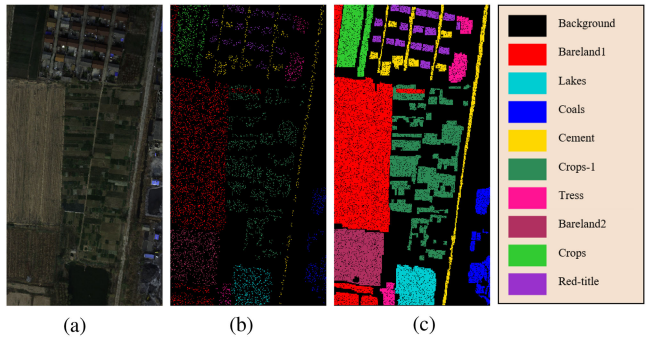

Fig. 8. Xuzhou dataset. (a) RGB composition. (b) Training set. (c) Test set.

2) Comparision With State-of-the-Art Backbone Networks: We have selected several well-known and representative algorithms and models for comparing our proposed HyperViTGAN. They are KNN, SAE [56], CNN [57], SSRN [24], RNN [21], ViT [36], SpectralFormer [38], EC-GAN [52], and ViTGAN [40]. The parameters of these comparison models are configured as listed in the following.

1) *KNN:* KNN with the nearest neighbor number of 10 is used for HSI classification.
2) *SAE:* Three hidden layers are used to constitute the SAE, with the number of neurons in layers one to three being 32, 64, and 128, respectively. The activation function used is rectified linear unit (ReLu).
3) *CNN:* Three 1×1 2-dimensional convolutional layers are used. The size of the neighbor region is set to 7×7. Based on [57], the dropout rate is set to 0.6.

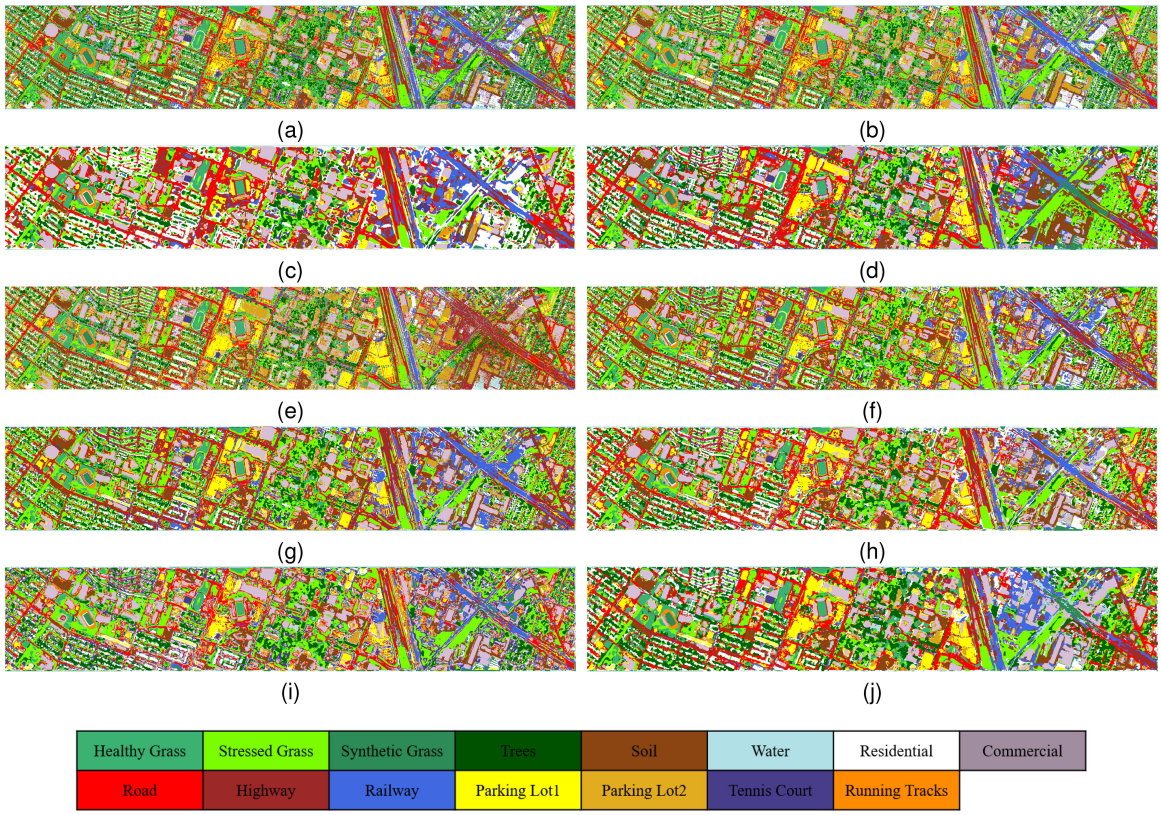

Fig. 9. Classification maps obtained on Houston 2013 dataset. (a) KNN. (b) SAE. (c) CNN. (d) SSRN. (e) RNN. (f) ViT. (g) SpectralFormer. (h) EC-GAN. (i) ViTGAN. (j) HyperViTGAN.

4) *SSRN:* The kernel numbers of convolutional filter banks are set to 28. The patch size is set to 7×7 for a fair comparison.
5) *RNN:* Two recursive layers with the GRU were used for the RNN. There are 128 neurons in each recursive layer.
6) *ViT:* A ViT network architecture containing only the transformer encoder is used, where the number of encoder blocks is 5.
7) *SpectralFormer:* Based on [38], five cascaded transformer encoder blocks with the embedded spectrum of 64 units are employed. The patch-wise SpectralFormer version with patch input is adopted for HSI classification.
8) *EC-GAN:* Based on [52], a generator, a discriminator, and an external classifier are employed to construct EC-GAN. The generator and discriminator use the architecture of the deep convolutional GAN (DCGAN) [58]. The patch size is set to 8×8.
9) *ViTGAN:* A discriminator and a generator are used to build ViTGAN, where both the discriminator and the generator are based on a five-block ViT architecture.
10) *The proposed HyperViTGAN:* For the proposed HyperViTGAN, a discriminator, a generator, and an external classifier are used to construct HyperViTGAN, where the discriminator, generator, and classifier are all designed based on the five-block ViT for a fair comparison. Each encoder consists of six attention heads, and the patch size is set to 7×7.

3) Implementation Details: All experiments were implemented on the PyTorch platform with a laptop configuration of Intel Core i7-10750H CPU, 16-GB RAM, and an NVIDIA GeForce RTX 2070 Super 8-GB GPU. The learning rate of the discriminator, generator, and classifier is set to 0.002. The epochs set on the three datasets of Houston 2013, Indian Pines 2010, and Xuzhou are 600. The Adam optimizer is used in HyperViTGAN with a minibatch size of 64 [59], and $\beta_1 = 0.0, \beta_2 = 0.99$.

C. Ablation Study

Ablation experiments on three HSI datasets are used to validate the contribution of the external classifier, as shown in Table IV. The GAN without classifier employs the loss design of D and G in [32], in which the architecture of D and G is identical to HyperViTGAN. Wang *et al.* [32] regard fake as a new class of category outputs for the discriminator; its discriminator has only a single classification output to perform both classification and discrimination tasks. First, the HyperViTGAN with an external classifier outperforms the GAN that only utilizes a discriminator to perform both classification and discrimination tasks on OA, AA, and k. Second, the external classifier design contributes the most to AA. The proposed HyperViTGAN is superior to the GAN with discriminators as classifiers by 0.12, 2.3, and 0.55 percentage points for AA on the Houston, Indian Pines, and Xuzhou datasets, respectively. The number of each class in the Houston 2013 training set is large and relatively uniform, but

TABLE IV
ABLATION ANALYSIS OF THE PROPOSED HYPERVITGAN, WITH AND WITHOUT EXTERNAL CLASSIFIER

Metric	Houston 2013		Indian Pines 2010		Xuzhou 2014	
	with	without	with	without	with	without
OA	**0.8904**	0.8888	**0.8950**	0.8841	**0.9346**	0.9305
AA	**0.8988**	0.8976	**0.8508**	0.8278	**0.9243**	0.9188
k	**0.8811**	0.8793	**0.8715**	0.8583	**0.9172**	0.9117

Note: Left value is evaluation metric of the proposed HyperViTGAN, followed by the evaluation metric of a GAN (same architecture as HyperViTGAN counterpart) that performs both discrimination and classification tasks using discriminator.

TABLE V
CLASSIFICATION ACCURACIES OBTAINED FROM THREE DATASETS USING DIFFERENT CLASSIFICATION METHODS WITH LIMITED TRAINING SAMPLES

Model	Houston 2013			Indian Pines 2010			Xuzhou 2014		
	OA	AA	k	OA	AA	k	OA	AA	k
KNN	0.6335	0.6568	0.6043	0.5855	0.6995	0.5255	0.7117	0.6807	0.6353
SAE	0.7180	0.7339	0.6955	0.7613	0.7943	0.7167	0.8413	0.8372	0.7999
CNN	0.5774	0.6041	0.5448	0.6828	0.6323	0.6194	0.8133	0.8103	0.7662
SSRN	0.7084	0.7401	0.6857	0.7861	0.7903	0.7438	0.9149	0.9205	0.8929
RNN	0.5812	0.6294	0.5491	0.6720	0.6902	0.6171	0.7174	0.7248	0.6462
ViT	0.6711	0.6884	0.6456	0.7243	0.7245	0.6723	0.8971	0.8973	0.8704
SpectralFormer	0.7148	0.7193	0.6919	0.7634	0.7814	0.7173	0.9181	0.9117	0.8967
EC-GAN	0.6642	0.6862	0.6384	0.7872	0.7441	0.7438	0.9216	0.9171	0.9010
ViTGAN	0.3746	0.3832	0.3268	0.4391	0.3191	0.3636	0.7124	0.6219	0.6274
HyperViTGAN	**0.7215**	**0.7430**	**0.6989**	**0.8156**	**0.8255**	**0.7787**	**0.9346**	**0.9243**	**0.9172**

they are slightly different from each other. The Xuzhou dataset has the same number of samples of each class in the training set. The number of each class in the Indian Pines 2010 training set is more unevenly distributed, with more severe class imbalance problems. The HyperViTGAN we designed still maintains a better AA, while the GAN with a discriminator as a classifier performs poorly. The reason for this phenomenon is that when using fake as the new class output of the discriminator, the discriminator is prone to associate minority class(es) with fake when the sample classes are unbalanced, thus jeopardizing the accuracy of the minority class(es). Moreover, since the external classifier designed in the HyperViTGAN is trained only on the high-quality fake HSI patches generated by the generator and all the real HSI patches, it avoids iterations of noisy HSI patches due to the incorporation of low-quality generated HSI patches into the training, which may damage the HSI classification performance of the classifier. Therefore, performing the classification task alone using external classifiers that do not share architecture enables better utilization of the GAN for the HSI classification task.

D. Classification Results With Limited Samples

In HSI classification tasks, the number of labeled samples is not sufficient because the acquisition of labels is time-consuming and costly. Therefore, models that can obtain superior classification results with a limited number of labeled training samples are more suitable for HSI classification tasks. To this end, we perform experiments with limited training samples (only ten samples per class are selected to constitute the training set) to verify the classification performance of the proposed HyperViTGAN and comparison models. For the Houston 2013 and Indian Pines 2010 datasets, ten samples were randomly selected from each class in the standard partitioned training set to form the new training set, and the rest were used as the test set; for the Xuzhou dataset, ten samples were randomly selected from each class to form the training set, and the rest were used as the test set. The classification accuracies obtained by the HyperViTGAN and other comparison models on the three datasets are shown in Table V. From Table V, we can observe that compared to the classification results on the standard partitioned Houston dataset, KNN, SAE, CNN, SSRN, RNN, ViT, SpectralFormer, EC-GAN, ViTGAN, and HyperViTGAN decreased by 16.87, 7.34, 18.40, 14.99, 23.96, 13.08, 10.66, 15.39, 42.40, and 16.89 percentage points in OA, respectively. Subsequently, compared to the classification results on the standard partitioned Indian Pines dataset, KNN, SAE, CNN, SSRN, RNN, ViT, SpectralFormer, EC-GAN, ViTGAN, and HyperViTGAN decreased in OA by 27.30,12.94, 16.58, 10.24, 18.38, 11.73, 10.60, 9.25, 45.41, and 7.94 percentage points, respectively. Both traditional algorithms (e.g., KNN) and deep models show a very substantial decrease in classification performance when the number of training samples is reduced. When the HyperViTGAN is compared with other GAN-based models (e.g., ViTGAN and EC-GAN), the OA of the HyperViTGAN and EC-GAN using the same external classifier with limited training samples decreases by 15.39 and 16.89 percentage points on the Houston dataset and 9.25 and 7.94 percentage points on the Indian Pines dataset, respectively. Comparing ViTGAN trained with the standard partitioned training set with ViTGAN trained with limited training samples, the OA of the ViTGAN decreases by 42.40 and 45.41 percentage points on the Houston and Indian Pines datasets, respectively. In particular, in the Xuzhou dataset with only 90 training samples, HyperViTGAN and EC-GAN achieved the first and second OA, respectively, compared to other models. This indicates that the design of the external classifier is well adapted to the case of

TABLE VI
CLASSIFICATION ACCURACIES OBTAINED FROM THE HOUSTON 2013 DATASET USING DIFFERENT CLASSIFICATION METHODS

Class No.	KNN	SAE	CNN	SSRN	RNN	ViT	SpectralFormer	EC-GAN	ViTGAN	HyperViTGAN
1	0.8281	0.8234	0.8251	0.8393	**0.8568**	0.8243	0.8201	0.8173	0.8180	0.8310
2	0.9568	0.9654	**0.9944**	0.9861	0.9214	0.9197	0.9801	0.9656	0.9624	0.9827
3	0.9960	0.9960	0.9489	0.9398	**0.9988**	0.6907	0.8400	0.7046	0.7865	0.9838
4	0.9498	0.9860	0.9581	0.9689	0.8839	0.9705	0.9652	0.9619	0.9419	**0.9934**
5	0.9725	0.9498	0.9968	**0.9992**	0.9621	0.9837	0.9949	0.9860	0.9869	0.9985
6	**0.9860**	0.7692	0.7385	0.9329	0.9301	0.8545	0.9021	0.8713	0.9021	0.9706
7	**0.8638**	0.8340	0.7959	0.8360	0.7442	0.8364	0.8310	0.8341	0.8424	0.8407
8	0.5204	0.4490	0.5007	**0.9231**	0.5434	0.7221	0.6923	0.8027	0.6860	0.8756
9	0.7705	0.7362	0.7141	0.8104	**0.8008**	0.7726	0.7739	0.7620	0.7409	0.7577
10	0.6506	0.5948	0.4429	0.4703	0.7515	0.6060	0.5193	0.6662	0.5983	**0.8104**
11	0.8719	0.7844	0.7457	0.8410	0.8133	0.6423	0.8182	0.7793	0.6620	**0.8729**
12	0.5024	0.6292	0.4784	0.7585	0.7383	0.7262	0.7191	0.6691	0.6742	**0.8680**
13	0.3789	0.5740	0.5368	**0.8786**	0.7340	0.7025	0.7032	0.6372	0.7432	0.7326
14	0.9838	0.9700	0.8996	0.9951	0.9903	0.9466	0.9741	0.9223	0.9668	**0.9984**
15	0.9810	0.9708	0.9641	0.9797	**0.9844**	0.9074	0.9721	0.8300	0.8710	0.9662
OA	0.8022	0.7914	0.7614	0.8583	0.8208	0.8019	0.8214	0.8181	0.7986	**0.8904**
AA	0.8142	0.8133	0.7693	0.8773	0.8436	0.8070	0.8337	0.8138	0.8122	**0.8988**
k	0.7855	0.7740	0.7410	0.8462	0.8058	0.7850	0.8061	0.8025	0.7814	**0.8811**
Time(s)	2.90	410.45	50.25	141.65	101.09	458.45	615.08	12831.53	5192.00	8457.88

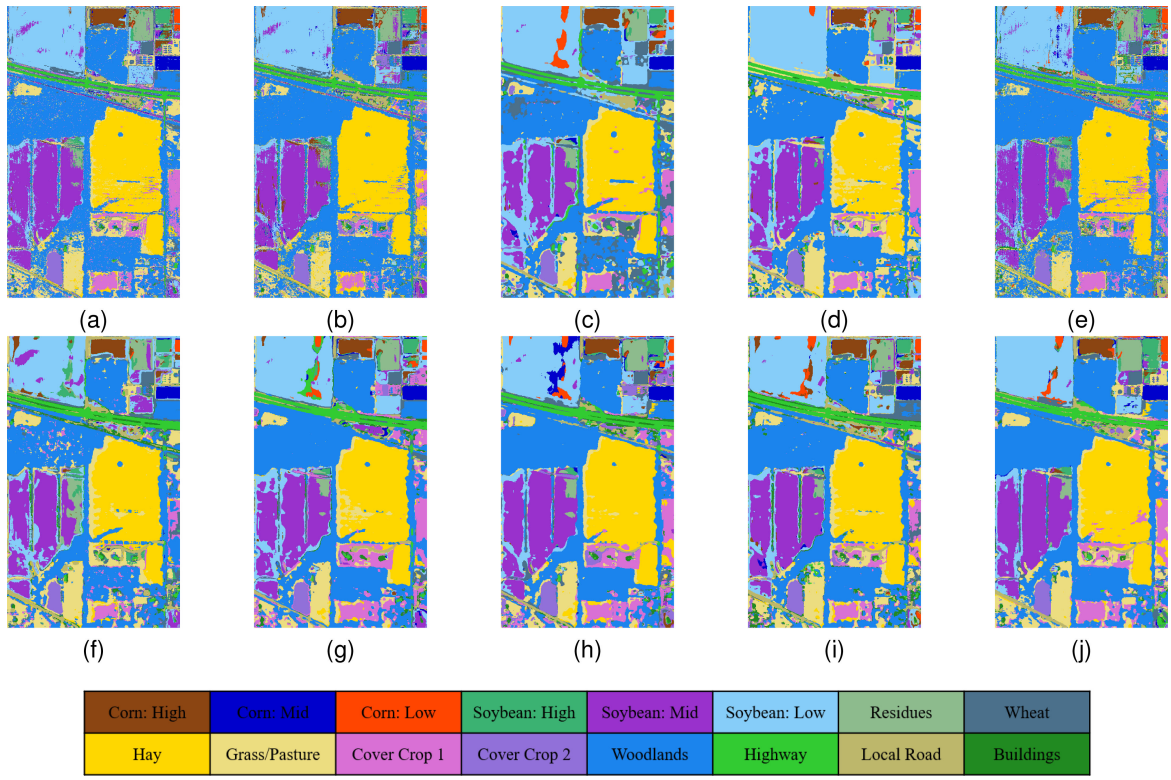

Fig. 10. Classification maps obtained on Indian Pines 2010 dataset. (a) KNN. (b) SAE. (c) CNN. (d) SSRN. (e) RNN. (f) ViT. (g) SpectralFormer. (h) EC-GAN. (i) ViTGAN. (j) HyperViTGAN.

limited training samples. It can be concluded that the GAN with external classifier design is able to maintain the performance of the GAN-based model with limited training samples.

E. Quantitative Results and Analysis

Tables VI–VIII show the running time in seconds and HSI classification performance of all the compared algorithms in OA, AA, k, and each land-cover class for the Houston 2013, Indian Pines 2010, and Xuzhou datasets, respectively. The best results are shown in bold in Tables VI–VIII. For the Houston 2013 data classification results listed in Table VI, the proposed HyperViTGAN leads all traditional and deep models with absolute OA (89.04%). Moreover, HyperViTGAN achieves the highest accuracy on five land-cover classes, while KNN, SAE, CNN, SSRN, RNN, ViT, SpectralFormer, EC-GAN, and ViTGAN achieve the optimal accuracy on only 2, 0, 1, 3, 4, 0, 0, 0, and 0 land-cover classes, respectively. HyperViTGAN greatly

TABLE VII
CLASSIFICATION ACCURACIES OBTAINED FROM THE INDIAN PINES 2010 DATASET USING DIFFERENT CLASSIFICATION METHODS

Class No.	KNN	SAE	CNN	SSRN	RNN	ViT	SpectralFormer	EC-GAN	ViTGAN	HyperViTGAN
1	0.8869	0.8988	**0.9378**	0.9073	0.8277	0.9067	0.8750	0.8947	0.8798	0.8985
2	0.9733	0.9922	0.8000	0.9842	0.9864	0.9777	0.9895	0.9987	0.9994	**1**
3	0.9724	0.9724	0.7545	0.9717	0.9862	0.8483	0.9076	0.8041	0.8379	**0.9993**
4	0.8184	0.7222	0.8711	0.8140	0.7585	0.8282	0.8402	**0.8895**	0.8394	0.8448
5	0.8058	0.8367	0.8117	0.8379	0.7538	0.7505	**0.8398**	0.8026	0.8100	0.8177
6	0.9313	0.9403	0.8821	0.9658	0.9206	0.8932	0.9024	0.9211	**0.9717**	0.9318
7	0.6287	0.6234	0.6212	0.6736	0.6017	0.5800	0.5966	0.6447	0.6322	**0.7208**
8	0.2839	0.3030	0.1561	0.2403	0.3085	0.2442	0.2331	0.2695	**0.6017**	0.3089
9	0.8178	**0.8990**	0.8407	0.8442	0.8364	0.8416	0.8474	0.8569	0.8652	0.8813
10	0.8263	0.7634	0.8116	0.8462	0.7359	0.7424	0.7770	0.8382	0.8326	**0.8565**
11	0.7886	0.7132	0.6904	0.8769	0.8315	0.6216	0.7020	0.8113	0.7011	**0.8980**
12	**1**	0.9997	0.3997	**1**	0.9982	0.9997	0.9907	**1**	0.9987	**1**
13	0.9568	0.9821	0.9801	0.9859	0.9873	0.9525	0.9810	0.9952	0.9928	**0.9959**
14	0.9266	0.9382	0.7957	0.9934	0.9086	0.9640	0.9256	0.9951	0.9943	**0.9957**
15	0.8333	0.9471	0.4947	0.9742	0.9582	0.8533	0.8116	0.9929	0.9742	**1**
16	0.4466	0.4304	0.3126	0.5237	0.3577	0.2407	0.2731	**0.5360**	0.4636	0.4640
OA	0.8585	0.8907	0.8486	0.8885	0.8558	0.8416	0.8694	0.8797	0.8932	**0.8950**
AA	0.8061	0.8101	0.6975	0.8400	0.7973	0.7653	0.7808	0.8281	0.8372	**0.8508**
k	0.8275	0.8659	0.8154	0.8636	0.8234	0.8063	0.8402	0.8530	0.8695	**0.8715**
Time(s)	182.35	4858.07	389.84	842.13	623.04	3058.16	3516.43	57682.11	30102.30	40591.68

TABLE VIII
CLASSIFICATION ACCURACIES OBTAINED FROM THE XUZHOU 2014 DATASET USING DIFFERENT CLASSIFICATION METHODS

Class No.	KNN	SAE	CNN	SSRN	RNN	ViT	SpectralFormer	EC-GAN	ViTGAN	HyperViTGAN
1	0.8571	0.8518	0.8719	0.9183	0.8310	0.9087	0.9164	0.9226	0.9186	**0.9560**
2	0.9683	0.9683	0.9448	0.9497	**0.9738**	0.9440	0.9296	0.9318	0.8847	0.9718
3	0.7589	0.8631	0.9256	**0.9781**	0.7981	0.9556	0.9359	0.9569	0.7043	0.9022
4	0.4654	0.8372	0.6663	0.8397	0.6954	0.7624	**0.8399**	0.7361	0.2792	0.8352
5	0.5577	0.8632	0.6934	0.9015	0.4898	0.8868	0.9522	**0.9556**	0.6413	0.9355
6	0.6434	0.7104	0.6182	0.9004	0.6049	0.8215	0.8490	**0.9173**	0.6418	0.9137
7	0.5652	0.6900	0.7576	0.9020	0.5300	0.8852	0.9061	**0.9388**	0.4434	0.9179
8	0.6214	0.8694	0.9568	0.9665	0.6783	0.9459	0.9394	**0.9695**	0.8154	0.9318
9	0.6892	0.8814	0.8582	0.9281	0.9220	**0.9660**	0.9366	0.9253	0.2683	0.9551
OA	0.7117	0.8413	0.8133	0.9149	0.7174	0.8971	0.9181	0.9216	0.7124	**0.9346**
AA	0.6807	0.8372	0.8103	0.9205	0.7248	0.8973	0.9117	0.9171	0.6219	**0.9243**
k	0.6353	0.7999	0.7662	0.8929	0.6462	0.8704	0.8967	0.9010	0.6274	**0.9172**
Time(s)	3.72	1690.48	90.65	152.19	42.54	607.48	754.18	5433.80	4730.92	4840.37

improves the performance of the ViTGAN on HSI classification tasks by 9.18 percentage points on OA compared to ViTGAN. In addition, HyperViTGAN is 6.9 percentage points higher in OA compared to SpectralFormer due to the excellent ability of the GAN to solve the dilemma of limited training data in deep models. For the Indian Pines 2010 dataset listed in Table VII, HyperViTGAN outperformed the conventional KNN algorithm by 3.65 percentage points in OA, respectively. Compared to the deep models, the HyperViTGAN outperforms ViTGAN, EC-GAN, SpectralFormer, ViT, RNN, SSRN, CNN, and SAE by 0.18, 1.53, 2.56, 5.34, 3.92, 0.65, 4.64, and 0.43 percentage points in OA, respectively. HyperViTGAN not only performs best in OA compared to the other models but also achieves the best accuracy on nine land-cover classes: it has the highest number of classes and achieves the highest accuracy among all comparison models. In addition, unlike the Houston and Xuzhou datasets, the number of individual categories in the training set of Indian Pines 2010 is unbalanced. Among the six least categories in the Indian Pines 2010 training set (class No. 3, 8, 12, 14, 15, and 16), the HyperViTGAN achieves the best accuracy in class No. 3, 12, 14, and 15 compared to other comparison models. We can notice that the HyperViTGAN can achieve superior accuracy even when the training classes are not balanced. This is due to the fact that the HyperViTGAN uses external classifiers alone for the classification task, avoiding the situation that minority-class samples are likely to be identified as fakes when the discriminator does both classification and discrimination tasks. At the same time, the HyperViTGAN continuously generates fake HSI patches by adversarial learning between the generator and discriminator and selects confident HSI patches to expand the training set of classifiers. Thus, the HyperViTGAN can well solve the dilemma when the training classes are not balanced. For the randomly divided training and testing Xuzhou dataset (see Table VIII), the HyperViTGAN achieved the highest OA, AA, and k compared to all the comparison models.

By analyzing the quantitative experiments on the three datasets, we can draw the following conclusions. First, the proposed HyperViTGAN always achieves the highest OA, AA, and k on all three datasets when compared to KNN, SAE, CNN, SSRN, RNN, ViT, SpectralFormer, EC-GAN, and ViTGAN. The HyperViTGAN also obtains the highest number of optimal

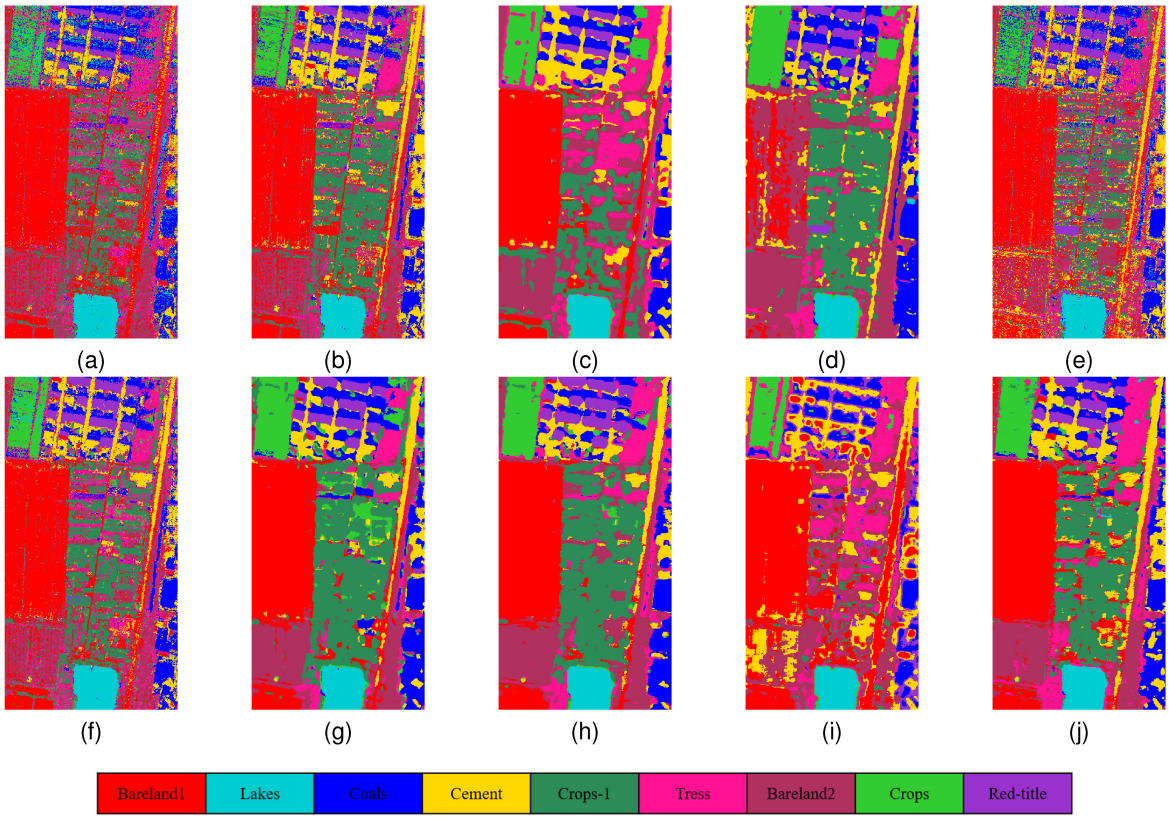

Fig. 11. Classification maps obtained on Xuzhou 2014 dataset. (a) KNN. (b) SAE. (c) CNN. (d) SSRN. (e) RNN. (f) ViT. (g) SpectralFormer. (h) EC-GAN. (i) ViTGAN. (j) HyperViTGAN.

accuracies on each individual land-cover class on the standard partitioned Houston 2013 and Indian Pines 2010 datasets, achieving 8 and 9 optimal classification accuracies on the Houston 2013 and Indian Pines 2010, respectively. This finding indicates that HyperViTGAN designed for the HSI classification task is well suited to HSI data, and learns hyperspectral feature information by considering HSI data as a sequence. Second, the skip connection between different blocks of SpectralFormer and HyperViTGAN always outperforms ViT and ViTGAN without skip connections between different blocks in terms of OA, since the design of establishing connections between different blocks of the transformer effectively prevents the loss of critical information, and thus, improve the model classification performance. By establishing skip connections between different blocks to preserve the shallow local low-level features and enhance the deep high-level features, spurious gradient explosion and gradient disappearance problems are avoided. Third, further comparing ViTGAN and HyperViTGAN, which are both based on GAN and transformer frameworks, the HyperViTGAN outperforms the ViTGAN on all three datasets owing to its strong generalization and stability. Fourth, comparing the EC-GAN with the same extra classifier with HyperViTGAN, the HyperViTGAN benefits from the transformer and outperforms the EC-GAN in OA, OA, and k in three HSI datasets. Finally, by observing the running time in seconds in Tables VI–VIII, we can find that GAN-based models commonly have longer running time compared to other deep models, and the traditional KNN has the shortest running time.

F. Visual Evaluation

Classification maps for the Houston 2013, Indian Pines 2010, and Xuzhou datasets acquired by KNN, SAE, CNN, SSRN, RNN, ViT, SpectralFormer, EC-GAN, ViTGAN, and HyperViTGAN are shown in Figs. 9–11, respectively. First, it can be observed that the classification map of the deep models is smoother and has fewer noise points compared to conventional classifiers, such as KNN. Second, Fig. 9(i) and (j) illustrates that the HyperViTGAN reduces the information loss well during the learning process and displays more realistic details. By looking at the classification graphs of the ViT, ViTGAN, SpectralFormer, and HyperViTGAN, it can be concluded that the transformer-based framework does not perform as well as the transformer-based model with the CAF module in terms of edge details and textures without skipping connections between blocks. The model with the CAF module design is able to better distinguish complex feature classes while having sharper edges and fewer noise points in detail features.

V. Conclusion

The GAN can address the limitations of small training samples when deep models are applied for HSI classification. At the

same time, as a new convolution-free novel network skeleton transformer is yielding unusually brilliant results in the field of image classification. For this purpose, then, we propose a combination of the transformer and GAN in a novel semisupervised network called HyperViTGAN for the HSI classification task. Three well-designed cascaded elements, a hyperspectral generator, discriminator, and classifier, are used for HSI patch generation, discrimination, and classification tasks, respectively. A skip connection component is employed to deliver memory-like components and prevent key information. Data augmentation is also used to strengthen model generalization and stability, in consideration of the instability of the GAN combined with self-attentive mechanisms. Experimental results on three well-known HSI datasets show that the HyperViTGAN exhibits state-of-the-art classification performance through adversarial learning and semisupervised learning on HSI classification tasks compared to the best models available at present.

ACKNOWLEDGMENT

The authors would like to thank Prof. M. Crawford for providing the Indian Pines 2010 data. In addition, the authors would like to thank the National Center for Airborne Laser Mapping, University of Houston, Texas, for providing the CASI Houston dataset and the IEEE Geoscience and Remote Sensing Society Image Analysis and Data Fusion Technical Committee for organizing the 2013 Data Fusion Contest.

REFERENCES

[1] 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.

[2] P. Ghamisi et al., "New frontiers in spectral-spatial hyperspectral image classification: The latest advances based on mathematical morphology, Markov random fields, segmentation, sparse representation, and deep learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 6, no. 3, pp. 10–43, Sep. 2018.

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

[4] J. A. Gualtieri and R. F. Cromp, "Support vector machines for hyperspectral remote sensing classification," in *Proc. 27th AIPR Workshop, Adv. Comput.-Assisted Recognit.*, 1999, vol. 3584, pp. 221–232.

[5] 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.

[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] L. Ma, M. M. Crawford, and J. Tian, "Local manifold learning-based k-nearest-neighbor for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4099–4109, Nov. 2010.

[8] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and markov random fields," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 809–823, Mar. 2012.

[9] H. Yang, "A back-propagation neural network for mineralogical mapping from AVIRIS data," *Int. J. Remote Sens.*, vol. 20, no. 1, pp. 97–110, 1999.

[10] Y. Zhang, G. Cao, A. Shafique, and P. Fu, "Label propagation ensemble for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 9, pp. 3623–3636, Sep. 2019.

[11] X. Shen, H. Yu, C. Yu, Y. Wang, and M. Song, "Global spatial and local spectral similarity based sample augment and extended subspace projection for hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2021, pp. 3637–3640.

[12] F. Gao, Q. Wang, J. Dong, and Q. Xu, "Spectral and spatial classification of hyperspectral images based on random multi-graphs," *Remote Sens.*, vol. 10, no. 8, 2018, Art. no. 1271.

[13] Z. He, K. Xia, T. Li, B. Zu, Z. Yin, and J. Zhang, "A constrained graph-based semi-supervised algorithm combined with particle cooperation and competition for hyperspectral image classification," *Remote Sens.*, vol. 13, no. 2, 2021, Art. no. 193.

[14] Z. He, K. Xia, Y. Hu, Z. Yin, S. Wang, and J. Zhang, "Semi-supervised anchor graph ensemble for large-scale hyperspectral image classification," *Int. J. Remote Sens.*, vol. 43, no. 5, pp. 1894–1918, 2022.

[15] R. Liu and X. Zhu, "Endmember bundle extraction based on multiobjective optimization," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 10, pp. 8630–8645, Oct. 2021.

[16] 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.

[17] S. Jia et al., "A semisupervised siamese network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–17, 2022, Art no. 5516417.

[18] K. Tan, F. Wu, Q. Du, P. Du, and Y. Chen, "A parallel Gaussian–Bernoulli restricted Boltzmann machine for mining area classification with hyperspectral imagery," *IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens.*, vol. 12, no. 2, pp. 627–636, Feb. 2019.

[19] Q. Shi, X. Tang, T. Yang, R. Liu, and L. Zhang, "Hyperspectral image denoising using a 3-D attention denoising network," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 12, pp. 10348–10363, Dec. 2021.

[20] S. K. Roy, S. Manna, T. Song, and L. Bruzzone, "Attention-based adaptive spectral–spatial kernel ResNet for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7831–7843, Sep. 2021.

[21] R. Hang, Q. Liu, D. Hong, and P. Ghamisi, "Cascaded recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5384–5394, Aug. 2019.

[22] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5966–5978, Jul. 2021.

[23] H. Lee and H. Kwon, "Going deeper with contextual CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4843–4855, Oct. 2017.

[24] 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.

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

[26] B. Rasti et al., "Feature extraction for hyperspectral imagery: The evolution from shallow to deep: Overview and toolbox," *IEEE Trans. Geosci. Remote Sens.*, vol. 8, no. 4, pp. 60–88, Dec. 2020.

[27] 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.

[28] I. Goodfellow et al., "Generative adversarial nets," *Adv. Neural Inf. Process. Syst.*, vol. 3, pp. 2672–2680, 2014.

[29] 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.

[30] X. Wang, K. Tan, Q. Du, Y. Chen, and P. Du, "Caps-TripleGAN: GAN-assisted CapsNet for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 7232–7245, Sep. 2019.

[31] W. Y. Wang, H. C. Li, Y. J. Deng, L. Y. Shao, X. Q. Lu, and Q. Du, "Generative adversarial capsule network with ConvLSTM for hyperspectral image classification," *IEEE Geosci. Remote. Sens. Lett.*, vol. 18, no. 3, pp. 523–527, Mar. 2021.

[32] J. Wang, F. Gao, J. Dong, and Q. Du, "Adaptive dropblock-enhanced generative adversarial networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 5040–5053, Jun. 2021.

[33] S. K. Roy, J. M. Haut, M. E. Paoletti, S. R. Dubey, and A. Plaza, "Generative adversarial minority oversampling for spectral–spatial hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022, Art no. 5500615.

[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[35] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," in *Proc. 8th Workshop Syntax, Semantics Struct. Statistical Transl., SSST*, 2014, pp. 103–111, *arXiv:1409.1259*.

[36] A. Vaswani et al., "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 5999–6009, 2017.

[37] A. Kolesnikov et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Lear. Representations*, 2021, *arXiv:2010.11929*.
[38] D. Hong et al., "Spectralformer: Rethinking hyperspectral image classification with transformers," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, Nov. 2021.
[39] Y. Jiang, S. Chang, and Z. Wang, "TransGAN: Two pure transformers can make one strong GAN, and that can scale up," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 14745–14758, 2021.
[40] K. Lee, H. Chang, L. Jiang, H. Zhang, Z. Tu, and C. Liu, "ViTGAN: Training GANs with vision transformers," in *Proc. Int. Conf. Learn. Representations*, 2022, *arXiv:2107.04589*.
[41] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of styleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8110–8119.
[42] S. Nowozin, B. Cseke, and R. Tomioka, "f-GAN: Training generative neural samplers using variational divergence minimization," *Adv. Neural Inf. Process. Syst.*, vol. 29, pp. 271–279, 2016.
[43] A. Müller, "Integral probability metrics and their generating classes of functions," *Adv. Appl. Probability*, vol. 29, no. 2, pp. 429–443, 1997.
[44] J. Song and S. Ermon, "Bridging the gap between f-GANs and wasserstein GANs," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 9078–9087.
[45] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214–223.
[46] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.
[47] A. Odena, "Semi-supervised learning with generative adversarial networks," in *Proc. Data Efficient Mach. Learn. workshop ICML*, 2016, pp. 2234–2242, *arXiv:1606.01583*.
[48] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," *Adv. Neural Inf. Process. Syst.*, vol. 29, pp. 2234–2242, 2016.
[49] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2642–2651.
[50] J. D. Kenton, M.-W. Chang, and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186, *arXiv:1810.04805*.
[51] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Representations*, 2018.
[52] A. Haque, "EC-GAN: Low-sample classification using semi-supervised algorithms and GANs," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 18, 2021, pp. 15797–15798, *arXiv:2012.15864*.
[53] D.-H. Lee et al., "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Workshop Challenges Representation Learn.*, 2013, vol. 3, no. 2, Art. no. 896.
[54] C. Debes et al., "Hyperspectral and lidar data fusion: Outcome of the 2013 GRSS data fusion contest," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2405–2418, Jun. 2014.
[55] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdiscipl. Rev., Comput. Statist.*, vol. 2, no. 4, pp. 433–459, 2010.
[56] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, 2010.
[57] S. Yu, S. Jia, and C. Xu, "Convolutional neural networks for hyperspectral image classification," *Neurocomputing*, vol. 219, pp. 88–98, 2017.
[58] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. Int. Conf. Learn. Representations*, 2016, *arXiv:1511.06434*.
[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2014, *arXiv:1412.6980*.

Ziping He is currently working toward the Ph.D. degree with the School of Electronics and Information Engineering, Hebei University of Technology, Tianjin, China.

From October 2021 to October 2023, she is a visiting Ph.D. student with the Machine Learning Group, Helmholtz-Zentrum and Dresden-Rossendorf, Freiberg, Germany, supported by the China Scholarship Council. Her research interests include semisupervised learning, machine learning, deep learning, and hyperspectral image classification.

Kewen Xia received the Ph.D. degree in electronics from Xi'an Jiaotong University (XJTU), Xi'an, China, in 2003.

His postdoctoral research was completed with the Computer Department, XJTU, in 2006. From 2010 to 2011, he worked as a Research Scholar in electronics with the University of Illinois at Urbana-Champaign, Champaign, IL, USA. He is currently a Professor and Ph.D. candidate supervisor with the School of Electronics and Information Engineering, Hebei University of Technology, Tianjin, China. His research interests include computational intelligence and wireless communication technology.

Pedram Ghamisi (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Iceland, Reykjavik, Iceland, in 2015.

He works as the Head with the Machine Learning Group, Helmholtz-Zentrum Dresden-Rossendorf (HZDR), Freiberg, Germany, and a Visiting Professor and Group Leader with Artificial Intelligence for Remote Sensing (AI4RS) Institute, Advanced Research in Artificial Intelligence (IARAI), Vienna, Austria. He is a cofounder of VasoGnosis Inc. with two branches in San Jose and Milwaukee, in the USA. His research interests include interdisciplinary research on machine (deep) learning, image and signal processing, and multisensor data fusion.

Dr. Ghamisi was the co-chair of IEEE Image Analysis and Data Fusion Committee (IEEE IADF) between 2019 and 2021. He was the recipient of the IEEE Mikio Takagi Prize for winning the Student Paper Competition at IEEE International Geoscience and Remote Sensing Symposium in 2013, the first prize of the data fusion contest organized by the IEEE IADF in 2017, the Best Reviewer Prize of IEEE Geoscience and Remote Sensing Letters in 2017, and the IEEE Geoscience and Remote Sensing Society 2020 Highest-Impact Paper Award. He is an Associate Editor for IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING and IEEE GEOSCIENCE AND REMOTE SENSING LETTERS.

Yuhen Hu (Life Fellow, IEEE) received the B.S.E.E. degree from National Taiwan University, Taipei, Taiwan, in 1976, and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 1980 and 1982, respectively.

He was on the faculty with the Electrical Engineering Department, Southern Methodist University, Dallas, TX, USA, and is currently a Professor with the Electrical and Computer Engineering Department, University of Wisconsin-Madison, Madison, WI, USA. He has authored more than 300 journals and conference papers and edited and coauthored several books. His current research interests include multimedia signal processing and communication, design methodology, implementation of embedded algorithms and systems, and nano-scale IC design methodologies.

Dr. Hu served as the Secretary for the IEEE Signal Processing Society and the Board of Governors of the IEEE Neural Networks Council. He served as the Chair for the IEEE Multimedia Signal Processing Technical Committee and the IEEE Neural Network Signal Processing Technical Committee. He also served as an Associate Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE SIGNAL PROCESSING LETTERS, *Journal of Very Large Scale Integration Systems Signal Processing*, IEEE MULTIMEDIA MAGAZINE, and *European Journal of Applied Signal Processing*.

Shurui Fan received the Ph.D. degree in control theory and control engineering from the Hebei University of Technology, Tianjin, China, in 2011 .

He is currently a Professor with the School of Electronics and Information Engineering, Hebei University of Technology. He was a Postdoctoral student with the School of Information Technology and Electrical Engineering, The University of Queensland, from March 2015 to September 2016. His research interests include embedded system artificial intelligence, wireless sensor networks, gas sensor array, autonomous search, and data visualization analysis method.

Baokai Zu received the Ph.D. in electronic science and technology degree from the School of Electronic and Information Engineering, Hebei University of Technology, Tianjin, China, in 2019.

She is currently an Assistant Professor with the Faculty of Information Technology, Beijing University of Technology, Beijing, China. Her current research interests include machine learning, deep learning, data mining, and image processing.