Multi-Complementary Generative Adversarial Networks with Contrastive Learning for Hyperspectral Image Classification

Jie Feng, *Senior Member, IEEE*, Zizuo Gao, Ronghua Shang, *Senior Member, IEEE*, Xiangrong Zhang, *Senior Member, IEEE*, and Licheng Jiao, *Fellow, IEEE*

***Abstract*—In the last decade, generative adversarial network (GAN) and its variants provide a powerful training mechanism for hyperspectral image (HSI) classification. In HSIs, the distribution of samples is more complicated due to the existence of abundant spatial-spectral information and multi-scale information. The single generation pattern of GANs is prone to modal collapse for the sample generation of HSIs. Moreover, the promotion of the generator only relies on adversarial learning with the discriminator, which limits the generator's performance. To address these problems, a multi-complementary GANs with contrastive learning (CMC-GAN) is proposed. CMC-GAN consists of two groups of GANs, where coarse-grained GAN adopts the structure in encoder-decoder form for hidden fine-scale and coarse-scale generation, and another fine-grained GAN is responsible for fine-scale generation. In fine-grained GAN, the discriminator is constructed to distinguish the fine-scale samples from different generators, which enforces the joint optimization of these two groups of GANs and makes GANs generate diverse multi-scale samples. Furthermore, a novel contrastive learning constraint is added into GANs, where a unidirectional contrastive loss guarantees the generators to extract intra-class invariant representation and a class-specific contrastive loss urges the discriminators to learn more discriminative features for classification. Finally, both discriminators are adaptively-fused to extract complementary multi-scale spatial-spectral features for classification under the guidance of diverse generated samples. The experimental results demonstrate CMC-GAN has superior classification performance, especially for small sample classification.**

***Index Terms*—Contrastive learning, generative adversarial networks (GANs), multi-grained GAN, hyperspectral image classification.**

I. INTRODUCTION

With the increasing demand for earth observation, many countries have built a large number of satellites to capture remote sensing data, so the acquisition of hyperspectral images (HSIs) becomes easier [1]. The HSI data is a three-dimensional cube by collecting hundreds of imaging wavelengths in the range of visible to infrared spectrum. In HSIs, each pixel represents the reflectance of different land-covers, which can reflect the difference in its interior structure and composition [2]. Due to the coexistence of rich spatial information and spectral information of HSIs, HSI processing technology has played an indispensable role in a wide range of fields, such as urbanization analysis, geological exploration, crop growth analysis and military reconnaissance [3-8]. As one of the most basic and important processing technologies, HSI classification uses spatial and spectral information to assign a label for each pixel through effective feature representation and classifier design.

The wide application of HSI classification has attracted the attention of academic communities. Originally, traditional machine learning methods were used to deal with HSI classification. As representatives of the traditional machine learning algorithms, decision trees [9], logistic regression [10] and support vector machines (SVMs) [11], have less memory cost and time consuming. However, these traditional methods rely on manual feature selection and expert knowledge heavily, so it is difficult to always maintain good classification performance in different scenarios.

After the 21st century, the explosive development of deep learning provides new solutions for plentiful subjects. The features obtained in a layer-by-layer training can mine invariant and discriminative deep information of data, which has made good progress in the field of image processing. Whereafter, some deep learning methods achieve good performance in HSI classification tasks. In 2014, Chen *et al*. first proposed an HSI classification method based on stacked autoencoders (SAE) [12], which uses SAE to obtain high-quality features and serve as a classifier simultaneously. The deep belief network (DBN) constructed by constrained Boltzmann machines in a layer-by-layer learning way was introduced for spectral-spatial classification of HSIs [13]. Later, diversified DBN [14] and hierarchical cascade DBN [15] were proposed. However, due to the usage of full connection layer, the above-mentioned algorithms have too many trainable parameters. Furthermore, these algorithms are established in the case of sufficient training samples. It is difficult to achieve satisfying classification results when training samples are scarce. Pan *et al*. [16] proposed a simplified deep learning method based on rolling guidance filter and vertex component analysis, which achieved good classification results when only a few HSI training samples

This work was supported in part by the National Natural Science Foundation of China under Grant 62271374, Grant 62176200, Grant 62077038, and Grant 62176196; in part by the State Key Program of National Natural Science of China under Grant 61836009; in part by the Natural Science Basic Research Program of Shaanxi under Grant 2022GY-065 and Grant 2022JC-45; in part by the Fundamental Research Funds for the Central Universities under Grant QTZX23047.

Jie Feng, Zizuo Gao, Ronghua Shang, Xiangrong Zhang, Licheng Jiao are with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an 710071, P.R. China (e-mail: jiefeng0109@163.com; gao_zz@qq.com; rhshang@mail.xidian.edu.cn; xrzhang@mail.xidian.edu.cn; lchjiao@mail.xidian.edu.cn).

were available.

Different from the above methods, convolutional neural network (CNN) can reduce the complexity of the model through local receptive fields and shared weights. It has certain invariance to rotation, translation and other deformations. With the continuous advancement of CNNs, the Unet architecture has demonstrated remarkable performance in certain tasks [59-60] by integrating pixel-level features with semantic-level features through its simple symmetric structure and skip connections. Deng et al. designed a two-layer capsule network for hyperspectral image classification by exploiting the properties of capsule networks, which overcomes the lack of an explicit notion of an entity and the loss of valuable information during max-pooling [57]. Furthermore, the rapid development of graph convolution and attention mechanism, GCN-based and transformer-based methods are also widely used for HSI classification tasks [63-64]. Hong et al. proposed a minibatch-based GCN approach, which allows to train large-scale GCNs in a minibatch fashion and can infer out-of-sample data without retraining [58]. Hong and Han et al. used transformers to design a cross-layer skip connection network to learn the local spectral sequence features of hyperspectral images from adjacent bands [56].

Recently, generative adversarial network (GAN) [33], as a rising star of generative models, has attracted extensive attention for its excellent generative ability. The generated samples can assist the training samples to alleviate the small sample problem of HSIs. GAN contains a generator and a discriminator to learn the distribution of real data while generating fake samples and identifying real or fake samples. The training of original GAN is prone to be unstable because of the difficulty in simultaneous convergence of the generator and discriminator. Later, Wasserstein GANs (WGAN) and WGAN with gradient penalty (WGAN-GP) alleviate this problem by using wassertein distance as the measure in the optimization [34, 35]. Then, Radford *et al.* [36] proposed a deep convolutional GAN, which provides an outstanding network topology for the training of GAN. It has been widely-used up to now. Subsequently, researchers tend to add the constraints to GAN to improve the generative capacity of GAN. Mirza [37] proposed a conditional GAN, which can feed additional auxiliary information into the generator with the noise to make the generated samples more instructive. Odena *et al.* [38] designed a GAN network with an auxiliary classifier, which could use both real and generated samples for multi-classification and true-false classification.

At present, the challenges in HSIs classification primarily revolve around the difficulty in feature extraction due to scarce training samples and the complex distribution of HSIs data. Although GAN-based methods can solve the above problems to some extent, these existing methods pay attention to generating single original scale of samples. Due to the limitation of original-scale resolution, it is difficult for these methods to generate rich detailed and boundary information simultaneously while generating obvious semantic information. In this case, these methods may cause the generated samples to be too similar and concentrated in a narrow data space. They may have difficulty in learning the overall distribution of HSI data. In the context of classification tasks, the integration of multi-complementary information becomes crucial. In addition, these methods mentioned above only update the generator with the adversarial learning from the discriminator. Due to the complex distribution of the concurrent spatial-spectral information in HSIs, it may be difficult to provide enough guidance information for the generator only through adversarial training.

To address these problems, a multi-complementary GAN model with contrastive learning (CMC-GAN) was proposed for HSI classification. In CMC-GAN, a network structure consisting of coarse-grained and fine-grained GANs is designed. The two generators produce the samples of different scales. The one from coarse-grained GAN generating coarse-scale samples focuses on salient semantic information, while the other from fine-grained GAN corresponding to fine-scale samples focuses on detailed and boundary information. Ingeniously, the coarse-grained GAN adopts an encoder-decoder structure that produces hidden fine-grained samples in the middle layer, which have the same size as the samples generated by the generator in fine-grained GAN. The two discriminators are responsible for distinguishing between real and generated samples of different scales. In the meantime, the fine-grained discriminator is designed to also distinguish the fine-scale samples from different generators, which enforces the joint optimization of two groups of GANs. By this way, the generator is urged to generate the diversity samples in the relaxed mapping space. In the optimization process, two kinds of contrastive losses are added as the regularization for the promotion of the generators and discriminators. In generators, a novel unidirectional contrastive loss is designed to ensure the in-class invariant representation by pulling the generated samples to the real samples in the embedding space. In discriminators, the class-specific contrastive loss is added to extract more discriminative features for classification. Finally, both discriminators are weighted adaptively to extract complementary multi-scale spatial-spectral features and obtain the ultimate HSI classification result.

We sum up three significant and novel contributions of this paper as follows:

1) In CMC-GAN, we propose a completely novel framework which includes multi-granularity generators and discriminators. This structure can make full use of the semantic information and boundary details of multi-scale samples, have a larger learning space, and ensure the diversity of multi-scale samples. Furthermore, the joint optimization of these two generators can guarantee the generation of diverse multi-scale samples, which alleviates the pattern collapse caused by the complex distribution in HSIs effectively.
2) In CMC-GAN, the fine-grained discriminator can differentiate between real and fake samples, as well as the samples from different generators. This motivates the fine-grained discriminator to have highly-discriminative fine-scale feature extraction ability. At the same time, the fusion of discriminators with different granularities can

extract the complementary discriminative features between different scales for classification.

3) Two kinds of contrastive losses brings the generated samples closer to the real samples unidirectionally in the projection space, and enforce the discriminators to improve their abilities to distinguish between different categories. Through the mechanism of adversarial learning and the constraint of contrastive learning, the classification performance of the discriminators can be continuously improved with the assistance of diverse generated samples.

The remainder of this paper contains five parts. The Section II introduces the related work based on CNN and GAN. The Section III introduces the relevant background of GAN and contrastive learning. The Section IV describes the architecture of CMC-GAN algorithm in detail. Then, the experimental setup and results of CMC-GAN are analyzed in Section V. Ultimately, some conclusions of CMC-GAN are given in Section VI, and some comments are made on the ideas that will be studied further in the future.

II. Related Work

A. CNN-based Methods

Nowadays, a series of CNN-based HSI classification methods emerged. According to the spectral, spatial, and spatial-spectral feature extraction of HSIs, the corresponding algorithms are developed separately [17, 18]. In [19], Hu *et al*. introduced one-dimensional CNN into the spectral domain for HSI classification. In [20], a sparse representation classification framework based on two-dimensional CNN was proposed to extract deep spatial features in HSIs. In [21], 1D-CNN and 2D-CNN were used to obtain spectral features and spatial features separately, and these features were cascaded for classification. A three-dimensional CNN (3D-CNN) was designed for joint spatial-spectral feature extraction of HSIs [22-24]. In addition, multi-scale spatial-spectral feature extraction has been developed. Inspired by inception module [25], Gong *et al*. introduced multi-scale filter banks into CNN and extracted multi-scale spatial-spectral features to promote the representation capability of CNN [26]. Wang *et al*. [27]. designed a strategy of extracting contextual information by using multi-scale blocks to enhance the spatial-spectral information of HSIs. Since then, deeper CNN models [28-30] have become a new mainstream trend. These networks can extract and obtain higher-order semantic features to further strengthen the classification ability. In HSIs, the collection of training samples requires field exploration, so it is not easy to obtain. For the scarce training samples of HSIs, the deeper network increases the computation cost, and also brings the performance degradation. Li *et al*. [31] proposed a novel pixel-pair method to reorganize the available training samples and ensure that the superiority of CNN can be fully utilized. Yan *et al*. [32] designed a spectral-spatial element attention network with simple element attention modules to ease the computing burden of the deep network, which enables the network to learn better in the case of small samples.

B. GAN-based Methods

With the rapid development of GAN, many improved GAN-based HSI classification methods have been paid sufficient attention in recent years [65]. The high-quality samples generated by the generator extend the training data for the HSI classification task. At the same time, the mechanism of adversarial training promotes the classification performance of the discriminator. In order to generate the spatial and spectral distribution of HSIs, Lin *et al*. [39] designed a 1-D GAN by using spectral features and a 3-D GAN by using spectral and spatial features for HSI generation and classification. In [40], multi-class spatial-spectral GAN (MSGAN) is devised. It has two generators to generate one-dimensional spectra and two-dimensional spatial patches. In MSGAN, one discriminator is designed to extract joint spatial–spectral features and get the probability of categories. Some researchers [41] rely on a large number of unlabeled samples to improve the generation ability of the generator. Zhan *et al*. [42] proposed a semi-supervised GAN, where the discriminator for true-false partitioning is first trained with unlabeled samples. Then, the discriminator is converted into a multi-class classifier and fine-tuned by limited labeled samples. Other researchers are trying to integrate sample augmentation into GAN to improve the generalization ability of the model. Wang *et al*. proposed a regularization method with adaptive shape DropBlock (ADGAN) for the generator and discriminator, which discards the masks with adaptive shapes [43]. Several methods employing multimodal and multitask learning have been proposed to enhance the classification accuracy of HSIs[61-62] in recent years. Furthermore, Roy *et al*. proposed a new GAN model, 3D-HyperGAMO [44]. It adopts an oversampling strategy with the help of the convex 3-D patch generator, which can automatically generate more minority class samples during the training and improve the classification performance.

III. Background

A. Generative Adversarial Network

In 2014, Goodfellow *et al.* proposed an avant-garde network based on the game idea: generative adversarial network (GAN) [33]. GAN contains a generative model (generator G) and a discriminant model (discriminator D), which is shown in Fig. 1. The generator and discriminator can be any form of network. The input of G is a noise vector z, and the goal of G is to obtain the generated samples $G(z)$ by learning the distribution of the real data x. For the discriminator D, its input is real data x and generated samples $G(z)$, and its corresponding outputs $D(x)$ and $D(G(x))$ represent the probability that D calculates the current input as the real data. The goal of D is to give x a high probability and $G(z)$ a low probability as much as possible. But G wants D to confuse the real data with the generated data, and give the generated data a result with a high value.

GAN hopes to achieve the Nash equilibrium of minimax game through alternate adversarial training, so as to achieve global optimization. Therefore, the objective function of GAN is designed as follow:

$$\begin{aligned}\max_{D}\min_{G} V(D,G) &= \mathbb{E}_{x\sim p_{data}(x)}\left[\log\left(D(x)\right)\right] \\ &+ \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1-D\left(G(z)\right)\right)\right]\end{aligned} \quad (1)$$

where p_{data} is the sampling in the distribution of real data x, and p_z is the distribution of the noise z. $\mathbb{E}(\bullet)$ represents the expectation operator.

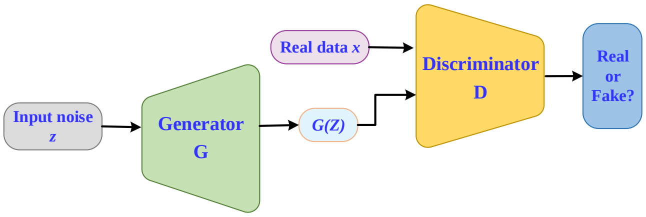


Fig. 1. Structure of generative adversarial network (GAN).

The optimization process of the network is realized by training G and D alternately. The specific steps are as follows: (1) G generates the sample $G(z)$ through the input noise, then G is fixed, and D is optimized by maximizing $V(D,G)$; (2) Fix D and optimize G by minimizing the $V(D,G)$. With the repeated steps (1) and (2), G can learn the regularities of real data's distribution and generate fake samples, and D is hard to distinguish whether the false sample comes from real data or G. While the distribution of generated data $p_g = p_{data}$, the network achieves the global optimization theoretically.

B. Contrastive Learning

Contrastive learning is an excellent representation learning method that uses the data itself to provide the supervision for algorithms. It constructs the representation by learning to code. This process makes two samples similar or different, which can provide valuable features for downstream tasks, such as classification, target detection and semantic segmentation [45, 46]. The purpose of contrastive learning is to maximize the feature similarity between positive data pairs and minimize the feature similarity between negative data pairs, so as to obtain excellent feature expression that is beneficial to the downstream tasks [47].

Data augmentation is an important step in contrastive learning [48]. Through different transformations or recoding operations on the original data, such as rotation, filtering, color distortion, the network can learn robust feature expression containing more universal information after comparative learning [49, 50]. We treat one sample and its augmented data as positive samples of each other, and all the other samples and their augmented data as its negative samples.

The general expression of the contrastive loss is calculated as follows:

$$\begin{aligned}L &= \sum_{x\in I} L_x \\ &= -\sum_{x\in I}\log\frac{\exp\left(f(x)\bullet f(x_+)/\tau\right)}{\exp\left(f(x)\bullet f(x_+)/\tau\right)+\sum\limits_{\alpha\in A(\mathrm{x})}\exp\left(f(x)\bullet f(a)/\tau\right)}\end{aligned} \quad (2)$$

where I is the set of any samples, $f(\cdot)$ represents the encoding operation on the samples, and x is called the anchor. x_+ is all the positive samples similar to x. Functional symbol $(\bullet)$ denotes the dot product of two vectors, describing the distance between them. $\alpha \in A(\mathrm{x})$ means the sample α in the set $A(\mathrm{x})$ of negative samples of all the x, and $\tau \in \Re^+$ is the variable temperature parameter. A smaller temperature parameter makes the contrastive loss more sensitive to differentiate positive and negative pairs, thus improving the model's discriminative power but may cause overfitting. A larger temperature parameter reduces the sensitivity of the contrastive loss to the similarity differences between different sample pairs, which improves the model's generalization ability but may sacrifice some discriminative power. Therefore, a reasonable temperature parameter needs to be adjusted according to the specific task and dataset to achieve the best training effect.

The contrastive loss can well express the matching degree between samples, and enhance the feature expression ability of samples in the network. In the case of rarely labeled or unsupervised tasks, the data itself can provide the supervision for the algorithm and can learn reusable knowledge for multiple downstream tasks rather than just for a specific task. Contrastive representation learning is favored by scholars because of its universality and good interpretability in different tasks. Nowadays, most existing contrastive learning methods have been developed on how to construct negative examples, define the loss functions for redundancy elimination, and design asymmetric contrastive networks [51-53].

IV. THE PROPOSED CMC-GAN METHOD

The proposed CMC-GAN method is shown in Fig. 2, which consists of coarse-grained and fine-grained generators and discriminators, and contrastive representation learning. The generators are designed to generate the samples of different scales. The discriminators are designed to classify these samples of different scales, and the contrastive constraint helps to improve the expression ability of the discriminators and generators. Coarse-grained generators can generate coarse-scale samples. Its overall structure is similar to encoder-decoder network, which encodes fine-scale samples in the hidden layer. The fine-grained generator incorporates original real data in the latent layer to guide the generation of fine-scale samples with detailed information. The coarse and fine-grained discriminators are responsible for the discrimination of the coarse and fine-scale generated samples respectively. In addition, the fine-grained discriminator is also responsible for distinguishing the fine-grained samples from the hidden layer generated by the coarse-grained generator. In this way, the two discriminators can extract complementary multi-scale information in different mapping spaces, while ensuring the diversity of samples from the two generators. In addition, the unidirectional contrastive loss and class-specific contrastive loss are added into the objective functions of generators and discriminators respectively, so that the generators can guarantee consistent expression of samples in the same category and the discriminators can learn discriminative features for classification. Finally, the confidence outputs of two discriminators are adaptively fused for classification.

A. The Coarse-grained and Fine-grained Generators for Multi-scale Sample Generation

In multi-class GAN, its classification performance depends on the ability of the discriminator. Due to unique adversarial learning mechanism, the performance improvement of the discriminator largely depends on the generator's ability. In the HSI classification task, it is difficult for the generator to learn the distribution of real samples well from a small number of training samples and complex spatial and spectral information of HSIs. Thus, it is challenging to generate a variety of high-quality samples. In some GAN models, resampling is common practice to enhance the training samples, but the samples with the same scale provide the information in a narrow range of data space. In CMC-GAN, the generators pay more attention to the complementary information contained in different scales of samples, and different coaching strategies, such as attention mechanism and real sample guidance, are imposed on the middle layers of the generators.

Fine-scale HSI data in high-resolution contains low-order, boundary and detailed information, which are often imperceptible but significant. In CMC-GAN, the generator generating such data is called fine-grained generator. Correspondingly, coarse-scale HSI data in low-resolution is generated by coarse-grained generator, which contains higher-order, salient and abstract semantic information. CMC-GAN constructs coarse-grained and fine-grained generators to generate the samples at different scales. The coarse-grained generator adopts encoder-decoder structure. The encoding part is the process of generating fine-scale samples from the hidden layer, and the decoding part refines the features on the basis of the above samples, then generates the final coarse-scale samples. In the process of generating from fine-scale hidden layer generated samples to coarse-scale samples, the spatial attention module is added to emphasize more discriminative features and suppress less important ones in the generation of spatial dimension. Thus, the generated coarse-grained samples can contain rich spatial information and their spatial distribution is closer to that of the real samples. The fine-grained generator fuses the real samples, which provide the guidance in the process of generation. Specifically, in the process of sample generation by fine-grained generator, the scale of the feature maps extracted from the network increases with the increase of network layers. At a certain layer, the generated feature map matches the size of the original real sample, and the sample and feature map are concatenated and

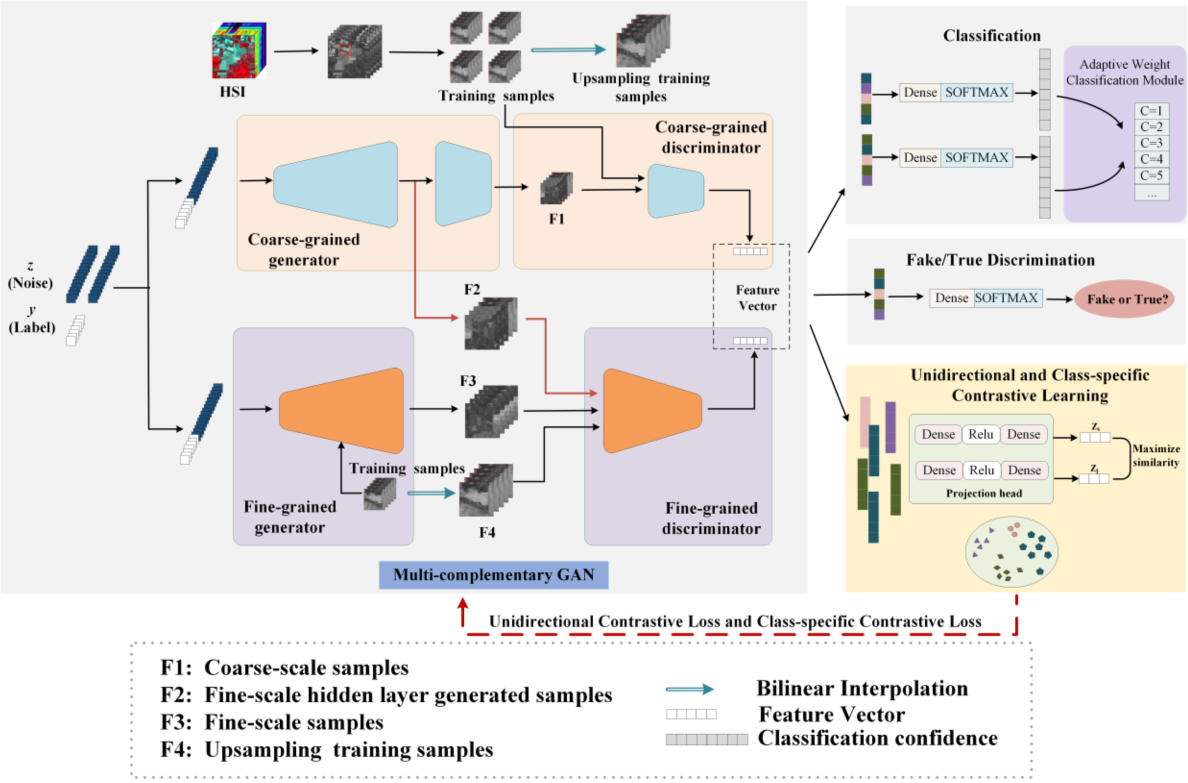


Fig. 2. Framework of proposed CMC-GAN method for HSI classification. The coarse-grained generator generates coarse-scale samples and fine-scale hidden layer samples, while the fine-grained generator generates fine-scale samples. The real samples and generated samples are separately inputted into the coarse-grained discriminator and fine-grained discriminator. Finally, the optimal classification results are obtained through the adaptive weight classification module.

fed as input to the next layer of the network. This operation is an effective way to ensure that fine-grained generator generates high-quality samples while maintaining the consistency and fidelity of the original samples. Multi-granularity generators in CMC-GAN not only learn complementary information from multi-scale samples, but also generate the samples that conform to the latent ontology structure of HSI data.

As shown in Fig. 2, the input of two generators includes the noise z and the category label y. It should be noted that the two generators use different random noises to prevent the generation from being too consistent. The generation process for the two generators is different. The input of coarse-grained generator passes through transposed convolution layers, batch normalization layers and Relu activation layers to form fine-scale hidden layer generated samples, and then coarse-scale samples are obtained through the convolution layers, batch normalization layers, Relu activation layers, and spatial attention module. The spatial attention module cascades the mean and maximum values of feature maps, passes through the convolution layer and sigmoid activation function layer, and then calculates the degree of importance of each element in the spatial feature maps. The fine-grained generator goes through the transposed convolution layers, batch normalization layers and Relu activation layers to generate fine-scale samples. The real samples are fused into the feature space of the same size. In addition, at the last layer of all the output samples, the activation function becomes the Tanh function.

B. The Coarse-grained and Fine-grained Discriminators for Multi-scale Spatial-spectral Feature Extraction

Multi-scale samples generated by multi-granularity generators contain a large amount of low-order detail information and high-order semantic information, which often needs to be mined in different feature mapping spaces. It is difficult for a single discriminator to pay attention to the features of multi-scale samples at the same time. Therefore, we designed two 3D discriminators in CMC-GAN to extract spatial-spectral features of multi-scale samples: a coarse-grained discriminator and a fine-grained discriminator.

The input of the coarse-grained discriminator is real samples and coarse-scale samples produced by the coarse-grained generator. For the fine-grained discriminator, its input contains more challenging samples for adversarial training: up-sampling real samples and two kinds of fine-scale samples from the fine-grained and coarse-grained generators.

The purpose of the coarse-grained discriminator is to distinguish whether the input comes from the coarse-grained generator or real data, and classify the input simultaneously. The fine-grained discriminator aims to distinguish whether the input is from the coarse-grained generator, the fine-grained generator, or the real data and implement the classification task. Thus, new loss functions of discriminators in CMC-GAM are defined as follows:

$$L_{D\_coa} = L_{D\_coa}^{real} + L_{D\_coa}^{fake} \tag{3}$$

$$L_{D\_fin} = L_{D\_fin}^{real} + L_{D\_fin}^{fake} + L_{D\_fin}^{hid} \tag{4}$$

where L_{D_coa} and L_{D_fin} represent the losses of the coarse-gained discriminator and fine-gained discriminator, respectively.

The loss of the coarse-grained discriminator includes the discrimination between the coarse-grained generated samples and the original real samples, which are expressed as follows:

$$\begin{aligned} L_{D\_coa}^{real} = &-\mathbb{E}_{x\sim p_{data}(x)}\left[\log D_{coa}\left(real\middle|x\right)\right] \\ &-\sum_{j=1}^{k}\mathbb{E}_{x\sim p_{data}(x),y\in j}\left[\log D_{coa}\left(y\middle|x\right)\right] \end{aligned} \tag{5}$$

$$\begin{aligned} L_{D\_coa}^{fake} = &-\mathbb{E}_{z\sim p_z(z)}\left[\log D_{coa}\left(fake\middle|G_{coa}(z,y)\right)\right] \\ &-\sum_{j=1}^{k}\mathbb{E}_{z\sim p_z(z),y\in j}\left[\log D_{coa}\left(y\middle|G_{coa}(z,y)\right)\right] \end{aligned} \tag{6}$$

where $L_{D\_coa}^{real}$ and $L_{D\_coa}^{fake}$ represent the losses calculated by the input of real samples and coarse-scale generated samples for the coarse-grained discriminator. x indicates the sample from the real training sample set X in an $R^{n\times n\times d}$ space, $n\times n$ represents the size of neighborhood window of each sample, and d is the total number of spectral bands in HSIs. y is the class label of the training samples x cascaded onto the random noise z, and $y=\{1,2,...,k\}$, and k is the total number of classes.

The loss of the fine-grained discriminator needs to distinguish the up-sampling real samples and fine-grained generated samples from the fine-grained generator and the hidden layer of the coarse-grained generator, which can be expressed as follows:

$$\begin{aligned} L_{D\_fin}^{real} = &-\mathbb{E}_{x\sim p_{data}(x)}\left[\log D_{fin}\left(real\middle|x\right)\right] \\ &-\sum_{j=1}^{k}\mathbb{E}_{x\sim p_{data}(x),y\in j}\left[\log D_{fin}\left(y\middle|x\right)\right] \end{aligned} \tag{7}$$

$$\begin{aligned} L_{D\_fin}^{fake} = &-\mathbb{E}_{z\sim p_z(z)}\left[\log D_{fin}\left(fake\middle|G_{fin}(z,y)\right)\right] \\ &-\sum_{j=1}^{k}\mathbb{E}_{z\sim p_z(z),y\in j}\left[\log D_{fin}\left(y\middle|G_{fin}(z,y)\right)\right] \end{aligned} \tag{8}$$

$$\begin{aligned} L_{D\_fin}^{hid} = &-\mathbb{E}_{z\sim p_z(z)}\left[\log D_{fin}\left(fake\middle|G_{coa}(z,y)\right)\right] \\ &-\sum_{j=1}^{k}\mathbb{E}_{z\sim p_z(z),y\in j}\left[\log D_{fin}\left(y\middle|G_{coa}(z,y)\right)\right] \end{aligned} \tag{9}$$

where $L_{D\_fin}^{real}$, $L_{D\_fin}^{fake}$ and $L_{D\_fin}^{hid}$ represent the losses calculated by the input of the up-sampling real samples, the fine-scale samples generated by fine-grained generator and the hidden layer of coarse-grained generator for the fine-grained discriminator. $D_{coa}(\bullet)$ and $G_{coa}(\bullet)$ denotes the outputs of the discriminator and generator in coarse-grained GAN, and $D_{fin}(\bullet)$ and $G_{fin}(\bullet)$ correspond to fine-grained GAN.

For generators, the coarse-grained generator can generate coarse-scale samples and output fine-scale hidden-layer samples during the encoding. Fine-grained generators generate fine-scale samples in higher resolution. The generator minimizes the two loss functions (5) and (6), hoping to improve its generating ability by making the discriminator unable to recognize the generated samples and correctly classify the generated samples. Thus, the loss functions L_{G_coa} and L_{G_fin} of generators in CMC-GAN are defined as follows:

$$\begin{aligned} L_{G\_coa} = &-\mathbb{E}_{z\sim p_z(z)}\left[\log D_{coa}\left(real\middle|G_{coa}(z,y)\right)\right] \\ &-\sum_{j=1}^{k}\mathbb{E}_{z\sim p_z(z),y\in j}\left[\log D_{coa}\left(y\middle|G_{coa}(z,y)\right)\right] \end{aligned} \tag{10}$$

$$\begin{aligned} L_{G\_\mathrm{fin}} = &-\mathbb{E}_{z\sim p_z(z)}\left[\log D_{fin}\left(real\middle|G_{fin}(z,y)\right)\right] \\ &-\sum_{j=1}^{k}\mathbb{E}_{z\sim p_z(z),y\in Y_j}\left[\log D_{fin}\left(y\middle|G_{fin}(z,y)\right)\right] \end{aligned} \tag{11}$$

For two discriminators in CMC-GAN, the input sample passes through the convolutional layer, the batch normalization layer and the Relu activation function layer, two parallel classifiers. Then, a project head is connected. The project head will be introduced in detail in the next chapter. Both classifiers are composed of full connection layer and Softmax layer. One has an output dimension of 2, which represents the confidence of whether the sample is true or not; the other has an output dimension of k, which means the probability that the sample belongs to each category. It should be pointed out that, in order to better extract the spatial-spectral features of the samples, 3D convolution operation is adopted in the discriminator.

C. Optimization of CMC-GAN under Contrastive Learning Constraint

In GAN, the learning of the generator is only guided by the output probability of the discriminator. It is difficult to learn the spatial-spectral distribution and consistent expression of samples in HSIs. In addition, due to the height limitation of photography and the influence of natural conditions, the spectral features of different categories may be relatively similar. As a result, it is difficult for the discriminator to distinguish different categories, which is not conducive to the classification performance of HSIs.

In CMC-GAN, we add the unidirectional contrastive loss to ensure that the generators extract the in-class invariant representation and learn the feature distribution of the real sample, and add the class-specific contrastive loss to constrain the discriminators to learn the feature expression more conducive to classification.

The workflow of unidirectional contrastive loss and class-specific contrastive loss is shown in Fig. 3. For the unidirectional contrastive loss, it calculates the similarity between the generated samples and the real samples, and draws the feature expression of the generated sample closer to the real sample through the update of the generators. We need the generator to learn the distribution of real samples of different classes. The generated samples can only learn from the real samples, and the feature distribution of the generated samples also needs to be close to that of the real samples, so it is a process of directed learning. The selection of positive and negative sample pairs for unidirectional contrastive loss and class-specific contrastive loss is shown in Fig. 4. In addition, to provide better learning guide to the encoding part of the coarse-grained generator, the unidirectional contrastive loss is also used for the fine-scale samples in the fine-grained generator and the hidden layer from the coarse-grained generator.

In order to stabilize the training and make the discriminator learn the decoupling feature expression of samples with different classes, we restrict the discriminator to only learn

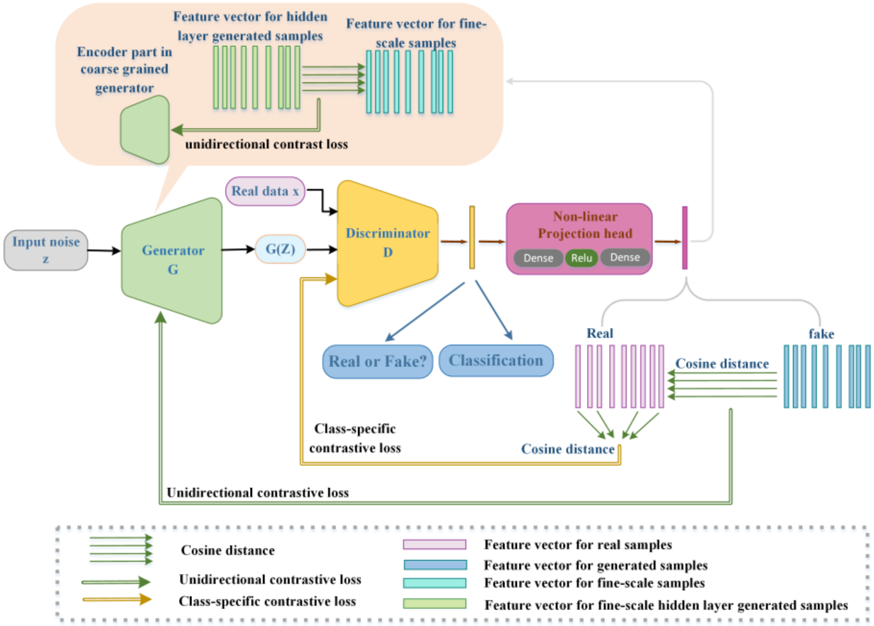

Feature vector for hidden layer generated samples
Feature vector for fine-scale samples
Encoder part in coarse grained generator
unidirectional contrast loss
Real data x
Input noise z
Generator G
G(Z)
Discriminator D
Non-linear Projection head
Dense
Relu
Dense
Real or Fake?
Classification
Real
fake
Cosine distance
Class-specific contrastive loss
Cosine distance
Unidirectional contrastive loss
Cosine distance
Unidirectional contrastive loss
Class-specific contrastive loss
Feature vector for real samples
Feature vector for generated samples
Feature vector for fine-scale samples
Feature vector for fine-scale hidden layer generated samples

Fig. 3. Workflow of unidirectional contrastive loss and class-specific contrastive loss. The class-specific contrastive loss acts between the feature vectors for the real sample, and the unidirectional contrastive loss acts between the feature vectors for generated samples and real samples. Furthermore, there is also a unidirectional contrastive loss between fine-scale hidden layer generated samples and the fine-scale samples generated by fine-grained generator.

from the contrastive loss by using the real images. The purpose in doing so is to prevent the discriminator from affecting the feature extraction of real samples when the generator is not stable enough. The class-specific contrastive loss is calculated as follows:

$$L_{con\_cla}=\sum_{i\in I}\frac{-1}{|P(i)|}\sum_{p\in P(i)}\ln\frac{\exp(z_i\bullet z_p/\tau)}{\sum_{a\in A(i)}\exp(z_i\bullet z_a/\tau)} \tag{12}$$

where z_i represents the feature expression of a real sample in all sample set I in a batch, while z_p represents feature expressions in real sample set $P(\cdot)$ of the same class as z_i, and z_a represents feature expressions in all the real sample set $A(\cdot)$ except z_i. The $\bullet$ is for similarity calculation, and we use cosine distance here. $|P(\cdot)|$ represents the number of feature vectors in set $P(\cdot)$.

The unidirectional contrastive loss function is:

$$L_{con\_uni}=\sum_{f\in F}\frac{-1}{|P(i)|}\sum_{\substack{p\in P(i)\\ i\in I}}\ln\frac{\exp(z_f\bullet z_p/\tau)}{\sum_{a\in A(i)}\exp(z_f\bullet z_a/\tau)} \tag{13}$$

where z_f represents the feature expression of a generated sample in all the sample set F in a batch, while z_p and z_a represent feature expressions in real sample set $P(\cdot)$ of the same class as z_f and all the real sample set $A(\cdot)$.

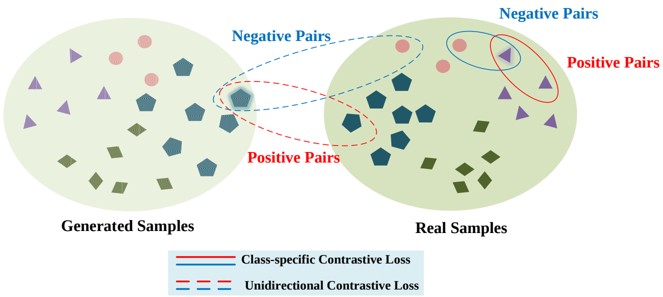


Fig. 4. Selection of positive and negative sample pairs for unidirectional contrastive loss and class-specific contrastive loss.

At the end of each discriminator, a nonlinear projection head is set, which is designed as a combination of two fully connected layers, and a nonlinear Relu activation function is directly added to these two layers. The feature expression extracted by the discriminator is input into the project head. The feature vector *z* is obtained, which is used to calculate the contrastive loss in (12) and (13).

In general, the total loss functions of generators and discriminators are shown in (14)-(17), where λ_1 and λ_2 are controllable coefficients, which are uniformly set as 0.03.

In addition, since the feature distribution of different HSI datasets varies greatly, we set an adaptive weight classification module to comprehensively consider their classification preferences and utilize their respective advantages to improve the overall classification performance. It is composed of a simple fully connected layer, which can learn the fusion weight, so as to adaptively fuse the classification results of the two discriminators.

TABLE I
DETAILED TRAINING PROCESS OF CMC-GAN

```
The workflow of CMC-GAN
INPUT: the training samples $X$, the labels $Y$, the upsampling of training samples $\overline{X}$, batch size $b$, the number of training epochs $E$, the updating times of the generators and discriminators $m$ and $n$, The temperature parameter $\tau$, the updating times of adaptive weight classification module $w$
OUTPUT: Output
Initialize: The parameters of whole network $\theta=\{\theta^{W},\theta^{D1},\theta^{D2},\theta^{G1},\theta^{G2}\}$, $\theta^{G1}=\{\theta^{G1\_decoder},\theta^{G1\_encoder}\}$.
for $E$ epochs do:
    for n times do:
        $G_1$ generates coarse-scale samples and fine-scale hidden layer generated samples, $G_2$ generates fine-scale samples.
        Update $\theta^{D1}$ by minimizing $L'_{D\_coa}$ in (16)
        $\theta^{D1}\xleftarrow{+}-\nabla_{\theta^{D1}}(L'_{D\_coa})$
        Update $\theta^{D2}$ by minimizing $L'_{D\_fin}$ in (17)
        $\theta^{D2}\xleftarrow{+}-\nabla_{\theta^{D2}}(L'_{D\_fin})$
    end for
    for m times do:
        Update $\theta^{G1}$ by minimizing $L'_{G\_coa}$ in (14)
        $\theta^{G1}\xleftarrow{+}-\nabla_{\theta^{G1}}(L'_{G\_coa})$
        Update $\theta^{G2}$ by minimizing $L'_{G\_fin}$ in (15)
        $\theta^{G2}\xleftarrow{+}-\nabla_{\theta^{G2}}(L'_{G\_fin})$
        Update $\theta^{G1\_decoder}$ by minimizing $(L^{hid}_{D\_fin}+L_{con\_uni})$ in (9) and (13)    $\theta^{G1\_decoder}\xleftarrow{+}-\nabla_{\theta^{G1\_decoder}}(L^{hid}_{D\_fin}+L_{con\_uni})$
    end for
    for w times:
        Update $\theta^{W}$ by minimizing cross entropy
    end for
end for
```

$$L'_{G\_coa}=L_{G\_coa}+\lambda_1 L_{con\_uni} \tag{14}$$

$$L'_{G\_fin}=L_{G\_fin}+\lambda_1 L_{con\_uni} \tag{15}$$

$$L'_{D\_coa}=L_{D\_coa}+\lambda_2 L_{con\_cla} \tag{16}$$

$$L'_{D\_fin}=L_{D\_fin}+\lambda_2 L_{con\_cla} \tag{17}$$

The training process of the CMC-GAN method is shown in Table I. G_1 and D_1 represent the generator and discriminator in coarse-grained GAN, G_2 and D_2 represent the generator and discriminator in fine-grained GAN, and W represents adaptive weight classification module. In the HSI classification task, the parameters of generators and discriminators are updated alternately, and the parameters between generators are updated simultaneously, as are parameters between discriminators. Specifically, the discriminators' parameters are fixed when the generator updates. Similarly, the discriminator updates the generator's parameters. We fix discriminators' parameters when generators' parameters are updated. Similarly, the parameters of discriminators are updated to fix the generators' parameters. In order to make the training process stable, for every optimization of discriminators, the generators are optimized m times. In addition, the training of the network can be finely tuned. When the generators are fixed, the training times of the discriminators can also be adjusted. Thus, the CMC-GAN network can be set up more flexibly. Due to the differences in features learned by coarse-grained GAN and fine-grained GAN, the training ratio between G_1 and G_2 and the training ratio between D_1 and D_2 can also be adjusted appropriately. After the two discriminators are trained, the features obtained from D_1 and D_2 are input into W for training and classification, and the best weighted results are obtained.

V. Experimental Results

In this section, the effectiveness of the proposed CMC-GAN algorithm is demonstrated on three publicly available hyperspectral remote sensing datasets. At the same time, some state-of-art hyperspectral classification algorithms are tested and compared.

A. Description of Datasets

1) The Indian Pines dataset.

The Indian Pines dataset was acquired in 1992 by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) over the Indian Pine Proving Ground in northwest Indiana, USA. As the first test dataset used for hyperspectral image classification, it contains 145×145 pixels. The original dataset contains 220 bands. After removing 20 bands which cannot be reflected by water, 200 spectral bands with a wavelength of 0.2 ~ 0.4 μm are retained. The scene includes 16 unevenly distributed categories of different crops, so they have relatively similar spectral curves. Its spatial resolution is about 20 meters, which brings difficulty to classification. The false-color composite image and ground truth of the Indian Pines dataset are shown in Fig. 5.

2) The Pavia University dataset.

The data of Pavia University was taken by the German Reflective Optics Spectrographic Imaging System (ROSIS-03). It mainly covers part of the urban scene of Pavia in Italy. There are 115 bands with a wavelength range of 0.43 to 0.86μm, and 102 bands with a spatial resolution of 1.3m are left after filtering out the noisy bands. The size of the Pavia University data is 610×340, with a total of 42,776 pixels and nine categories for classification except background. Therefore, its false-color composite image and ground truth are shown in Fig. 6(a) and (b), respectively.

3) The Washington dataset.

The Washington dataset was collected by the Virginia Center for Spectral Information Technology Applications. This is the scene above the Washington DC mall in 1995. It consists of 750x307 pixels with a spatial resolution of 2.8 meters and a total of 210 bands. After removing the atmospheric opaque bands, the remaining 191 bands with wavelengths ranging from 0.4 to 2.4μm are usually used for the experiment. The scene consists of seven distinct categories, whose false-color composite image and ground truth are shown in Fig. 7.

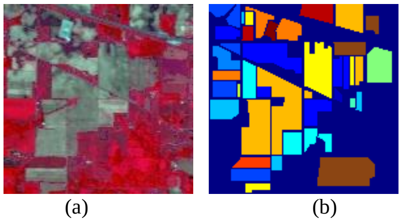


Fig. 5. Indian Pines dataset. (a) False-color composite image. (b) Ground truth.

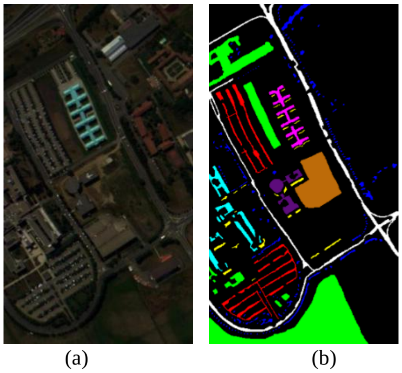


Fig. 6. Pavia University dataset. (a) False-color composite image. (b) Ground truth.

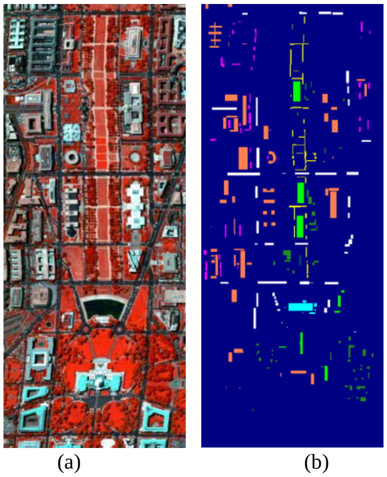


Fig. 7. Washington dataset. (a) False-color composite image. (b) Ground truth.

B. Experimental Setting

In order to verify the effectiveness of CMC-GAN algorithm persuasively, we configure different algorithms for comparison, and set up elaborate and fair experimental details.

We first set up several classical methods, including RBF-SVM [11], PPF-CNN [31], 3D-CNN [18], and HS-GAN [42]. In addition, several GAN-based algorithms with excellent performance in HSI classification, 3D-GAN [39], AD-GAN [43] and 3D-HyperGAMO [44], are set to prove the superiority of the proposed algorithm. In the experiments, all the comparison algorithms use the optimal parameters of the open-source code or the original paper. Then, the above methods are roughly introduced as follows:

RBF-SVM: It's adopted five-fold cross-validation to determine the best penalty parameter and gamma parameter, and RBF-SVM takes advantage of one-against-all strategy for multi-classification.

PPF-CNN: The architecture and the size of spatial neighborhood window in PPF-CNN can be seen in [31].

3D-CNN: The spatial window size is set to the suggested value of 27×27 in [18].

HS-GAN: The architecture of the HS-GAN is proposed in [42], which uses 1×3 and 1×5 convolutional kernels for convolution.

3D-GAN: The 3-D input of 3D-GAN is a window with the side length 64 and 3 channel dimensions.

AD-GAN: For the AD-GAN, the block size is set to 7 and the percent k of discarded elements is set to 40, and the spatial window size is set to 27×27.

3D-HyperGAMO: The input of the network is a spatial window of size 11 x 11 extracted from the dataset, and the original number of bands is retained.

In CMC-GAN, The input random noise z is a 100-dimensional vector conforming to the Gaussian distribution. The label y is converted into a one-hot encoding form and cascaded with noise z. The training epoch is set to 100, and the batch size is set to 16. The RMSprop optimizer is applied throughout the GAN network, the learning rates of the generator and discriminator are set to 0.01 and 0.008, respectively, and the decay of the learning rate is 0.95 every 50 epochs. The input window size of the coarse-grained discriminator is 27×27, and that of the fine-grained discriminator is 108×108. In the contrastive learning, the temperature parameter is set to 0.5, and all the weights of losses are set to 0.3. In the adaptive weighted module, the number of training epoch is 1000, the initial learning rate is 0.005, and the optimizer adopts SGD.

The classification performance of the proposed CMC-GAN method is verified by several extensively used indexes, such as the overall accuracy (OA), the average accuracy(AA), and the kappa coefficient [54]. OA is defined as the proportion of correctly classified samples to total test samples. AA is defined as the average of the classification accuracy of each category.Kappa coefficient indicates that there is strong mutual consistency between the classification map generated by the model and the ground truth. The experimental results presented in this paper are the average of 20 independent experiments. The computer equipment used for the experiment is an Intel i7-7820X processor and an NVIDIA 2080Ti graphics card. All the source code for the framework is implemented using the Python language and the Pytorch library.

C. Classification Results of CMC-GAN

For the Indian Pines dataset, 5% samples from each class are randomly selected for training CMC-GAN during each experiment, and the remaining 95% samples are used for testing. Table II shows the detailed number of training and tests for all the categories. The quantitative results of CMC-GAN as well as the results of the other seven comparison algorithms on the Indian Pines dataset are shown in Table III. In addition to OA, AA and Kappa, Table VI also shows the accuracy of each category. In the experimental results, the highest values are highlighted in a gray text background.

TABLE II
16 Classes of Indian Pines Scene and The Numbers of Training Samples and Test Samples for Each Class.

Class		Numbers of samples	
No	Name	Training	Test
1	Alfalfa	2	44
2	Corn-notill	71	1357
3	Corn-mintill	42	788
4	Corn	12	225
5	Grass-pasture	24	459
6	Grass-trees	36	694
7	Grass-pasture-mowed	1	27
8	Hay-windrowed	24	454
9	Oats	1	19
10	Soybean-notill	49	923
11	Soybean-mintill	123	2332
12	Soybean-clean	30	563
13	Wheat	10	195
14	Woods	63	1202
15	Buildings-Grass-Trees-Drives	19	367
16	Stone-Steel-Towers	5	88
Total		512	9737

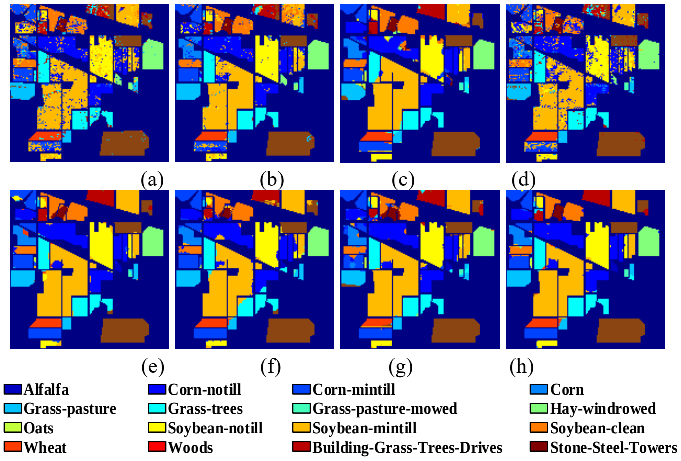


Fig. 8. Classification visual maps of the Indian Pines dataset. (a)RBF-SVM, (b) PPF-CNN, (c) 3DCNN, (d) HSGAN, (e) 3D-GAN, (f) AD-GAN, (g) 3D-HyperGAMO, (h)CMC-GAN.

As shown in Table III, the proposed CMC-GAN has a good performance in most categories. Different from RBF-SVM, as deep learning algorithms, PPF-CNN, 3D-CNN, 3DGAN, AD-GAN, 3DHyperGAMO and CMC-GAN, have excellent feature extraction ability through nonlinear mapping and can achieve higher accuracy. The extended sample strategy improves the and classification performance of PPF-CNN, and its OA index is 9.9% higher than that of RBF-SVM. For HS-GAN, although a large number of unlabeled samples are used, the network only performs feature extraction in the spectral dimension, and the results are not ideal. Both 3D-CNN and 3D-GAN make use of the combined spatial and spectral information, and the OA index is increased by 4.9% and 5.6% compared with PPF-CNN. Among them, 3D-GAN exercises the classifier's feature extraction ability by generating new samples, and the effect is more ideal. AD-GAN adopts adaptive regularization method, the effect is improved by 0.5% compared with 3D-GAN. CMC-GAN achieved the highest results in most categories. In CMC-GAN, the multi-granularity generator and the process of joint

This article has been accepted for publication in IEEE Transactions on Geoscience and Remote Sensing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TGRS.2023.3304836

TABLE III
CLASSIFICATION RESULTS OF RBF-SVM, PPF-CNN, 3D-CNN, HS-GAN, 3D-GAN, AD-GAN, 3D-HYPERGAMO AND CMC-GAN ON THE INDIAN PINES DATASET

Class	RBF-SVM[11]	PPF-CNN[31]	3D-CNN[18]	HS-GAN[42]	3D-GAN[39]	AD-GAN[43]	3D-HyperGAMO [44]	CMC-GAN
1	6.1±11.2	50.4±8.4	83.8±13.4	17.7±5.2	90.9±5.2	47.7±0.7	61.7±17.9	97.1±3.8
2	72.9±3.6	89.2±2.1	92.7±3.5	66.3±1.1	91.0±1.7	91.1±0.2	82.5±7.9	95.3±1.2
3	58.0±3.6	77.1±2.7	87.2±10.4	60.2±2.9	90.4±2.1	98.9±0.2	67.4±23.8	96.0±2.4
4	39.0±15.0	87.7±3.7	83.4±8.3	57.8±4.7	93.7±4.3	91.6±0.6	97.9±2.9	90.3±5.6
5	87.0±4.5	92.7±1.0	84.0±5.7	82.0±6.1	93.2±4.5	95.2±0.2	99.7±0.6	90.6±3.0
6	92.4±2.0	93.1±1.9	93.4±2.5	94.3±2.2	95.4±0.7	92.7±0.3	83.3±13.4	98.3±0.6
7	0±0	0±0	97.2±4.8	23.8±12.2	94.9±0.1	29.6±1.1	88.9±4.6	98.4±4.0
8	98.1±1.4	99.6±0.3	97.4±2.8	98.8±0.3	99.9±0.2	99.8±0.1	92.5±4.2	100±0.2
9	0±0	0±0	77.0±11.1	13.7±12.1	53.5±1.4	21.0±0.7	99.8±0.2	68.4±10.8
10	65.8±3.7	85.6±2.8	93.3±5.0	68.5±3.6	94.2±0.3	87.1±0.3	82.8±10.4	95.1±1.8
11	85.3±2.9	83.8±1.6	94.9±2.7	79.7±0.5	94.7±1.5	97.8±0.1	98.6±1.4	98.6±0.6
12	69.6±6.5	90.4±3.1	89.8±4.3	48.8±4.5	92.1±2.3	90.9±0.5	90.4±4.4	93.3±1.2
13	92.3±4.1	97.8±0.9	92.8±5.9	89.2±2.7	95.5±0.2	100.0±0.1	90.0±4.1	99.2±1.4
14	96.6±1.0	95.5±1.1	98.3±1.3	96.0±1.1	95.6±0.3	97.5±0.1	97.3±1.5	98.4±1.0
15	41.7±7.0	78.0±2.4	77.8±13.4	37.9±11.4	87.7±2.1	94.8±0.7	90.8±1.6	98.5±1.8
16	75.2±9.0	97.3±1.3	88.4±5.3	73.0±5.3	92.6±2.3	67.0±0.5	95.2±2.1	96.1±2.0
OA(%)	77.8±0.8	87.9±0.8	92.8±0.8	74.0±0.9	93.5±0.3	94.0±0.5	93.0±1.9	96.7±0.3
AA(%)	61.3±1.4	76.5±0.6	89.4±1.4	60.2±2.6	84.8±2.7	81.4±1.0	88.7±4.0	95.8±0.3
Kappa(%)	74.5±1.0	86.3±0.9	91.9±0.9	70.0±1.0	93.1±1.2	93.1±0.6	91.9±2.2	95.0±1.0

TABLE IV
9 CLASS OF PAVIA UNIVERSITY SCENE AND THE NUMBERS OF TRAINING SAMPLES AND TEST SAMPLES FOR EACH CLASS.

Class		Numbers of samples	
No	Name	Training	Test
1	Asphalt	199	6432
2	Meadows	559	18090
3	Gravel	63	2036
4	Trees	92	2972
5	Painted metal sheets	40	1305
6	Bare Soil	151	4878
7	Bitumen	40	1290
8	Self-Blocking Bricks	110	3572
9	Shadows	28	919
Total		1282	41494

optimization greatly expand the number of samples for the discriminator. In addition, finer-grained feature learning enables the features of samples to be fully mined. Among the overall evaluation indicators, CMC-GAN achieved the highest OA, AA and Kappa, which were increased by at least 2.7%, 6.4% and 1.9%compared with other methods, respectively.

Fig. 8 shows the visualized classification results of different methods on the Indian Pines dataset. As shown in Fig. 8, RBF-SVM, PPF-CNN and HS-GAN have a large number of misclassified noisy points in some areas of most classes, such as Corn-notill, Soybean-notill and Soybean-mintill. AD-GAN does not perform well in categories such as Alfalfa, Grass-pasture-mowed and Oats. Compared with above methods, 3D-GAN, 3D-HyperGAMO, and CMC-GAN have better visual effects. CMC-GAN has a relatively smooth segmentation edge and improves the uniformity of the segmentation region. Specifically, CMC-GAN is best visually in Grass-pasture-mowed.

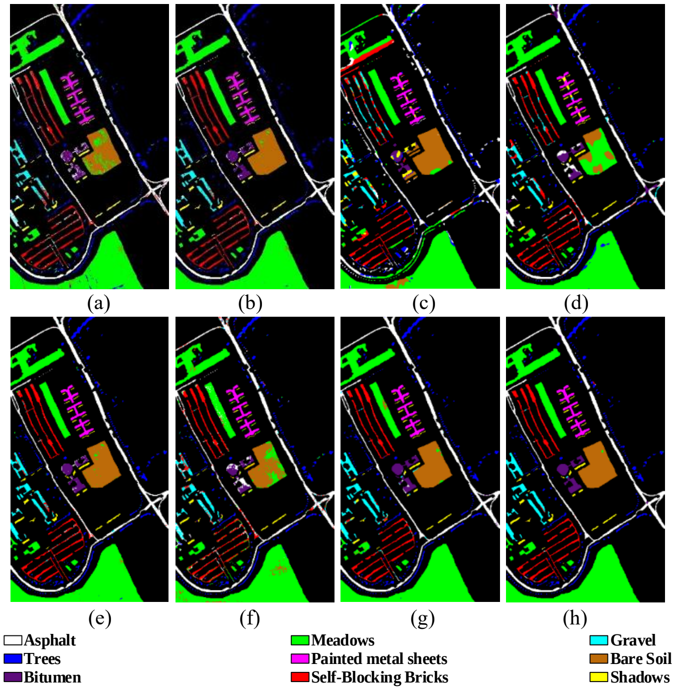


Fig. 9. Classification visual maps of the Pavia University dataset. (a)RBF-SVM, (b) PPF-CNN, (c) 3DCNN, (d) HSGAN, (e) 3D-GAN, (f) AD-GAN, (g) 3D-HyperGAMO, (h)CMC-GAN.

For the Pavia University dataset, we randomly selected 3% of each class for training CMC-GAN and the remaining 97% for testing. Table IV shows the number of detailed training and tests for all the categories. The quantitative results of CMC-GAN and other 7 comparison algorithms on the Pavia University dataset are shown in Table V.

TABLE V
CLASSIFICATION RESULT OF RBF-SVM, PPF-CNN, 3D-CNN, H-SGAN, 3D-GAN, AD-GAN, 3D-HYPERGAMO AND CMC-GAN ON THE PAVIA UNIVERSITY DATASET

Class	RBF-SVM[11]	PPF-CNN[31]	3D-CNN[18]	HS-GAN[42]	3D-GAN[39]	AD-GAN[43]	3D-HyperGAMO[44]	CMC-GAN
1	90.7±1.1	98.0±0.1	95.5±1.2	87.4±0.9	99.3±0.4	99.3±0.1	99.4±0.6	98.7±0.8
2	96.8±0.7	99.2±0.2	99.4±0.3	93.0±1.9	99.7±0.3	99.7±0.0	100±0.0	100.0±0.0
3	60.2±5.4	84.9±1.8	92.6±5.4	63.6±7.2	92.5±2.3	94.1±0.1	93.5±2/4	94.6±2.0
4	90.8±2.0	95.8±0.8	75.2±4.9	89.6±4.0	93.2±2.3	97.3±0.1	93.1±1.7	98.1±1.3
5	98.8±0.4	99.8±0.1	95.4±4.3	99.6±0.2	99.8±0.3	99.9±0.2	98.3±0.9	99.6±1.1
6	79.5±4.9	96.4±0.3	99.4±0.6	70.5±7.8	98.7±0.8	94.6±0.1	97.4±1.5	99.7±0.6
7	74.3±5.1	89.2±0.8	91.5±3.4	69.7±6.0	92.2±3.3	94.7±0.7	98.8±0.7	95.9±2.6
8	88.8±2.2	93.7±1.2	94.8±1.4	73.6±4.3	98.0±1.2	98.4±0.2	98.4±0.9	98.5±0.8
9	99.8±0.1	98.5±0.7	77.4±2.8	97.8±1.3	94.5±3.7	96.8±0.6	99.5±0.3	98.3±1.8
OA(%)	90.3±0.6	96.9±0.2	95.2±0.7	85.7±0.2	98.2±0.1	98.3±0.0	98.6±0.3	99.0±0.3
AA(%)	86.6±0.9	95.1±0.2	91.2±1.1	82.8±0.8	96.4±0.5	97.2±0.1	97.7±0.4	98.7±0.3
Kappa(%)	87.1±0.8	96.0±0.2	93.8±0.9	81.0±0.3	97.6±0.2	97.7±0.0	98.1±0.4	98.7±0.4

TABLE VI
7 CLASSES OF WASHINGTON SCENE AND THE NUMBERS OF TRAINING SAMPLES AND TEST SAMPLES FOR EACH CLASS.

Class		Numbers of samples	
No	Name	Training	Test
1	Roads	86	2787
2	Grass	51	1663
3	Water	19	611
4	Roofs	31	1005
5	Trails	38	1240
6	Trees	35	1118
7	Shadows	168	5443
Total		428	13867

As shown in Table V, all the OA, AA and Kappa coefficient of HSGAN are below 90.0%. The classification result of RBF-SVM is slightly better than that of HSGAN. For other comparison algorithms, all the indexes of PPF-CNN, 3D-CNN, 3D-GAN, ADGAN and 3D-HyperGAMO reach more than 90.0%. Their OA indicators are 6.6%, 4.9%, 7.9%, 8.0% and 8.3% higher than RBF-SVM, respectively. In the GAN-based comparison methods, the result of 3D-HyperGAMO is the best among HS-GAN, 3D-GAN and AD-GAN. Among the eight methods, the proposed CMC-GAN achieves optimal results in most of the classes, and its OA, AA and Kappa coefficient increase by at least 0.4%, 1.0% and 0.6%. The visualization results of the above algorithms in Pavia University dataset are shown in Fig. 9. Gravel and Bitumen categories are difficult to classify for RBF-SVM and HS-GAN. 3D-CNN does not classify well on the Shadows class. Compared with these methods, the visualization results of CMC-GAN have more neat edges, fewer misclassified noisy points and better regional consistency.

The visualization results of the above algorithms on the Pavia University dataset are shown in Fig. 9. Obviously, the classification results of RBF-SVM, HS-GAN and ADGAN have a large number of noisy points and poor classification consistency, while the proposed CMC-GAN has more intuitive and excellent classification results.

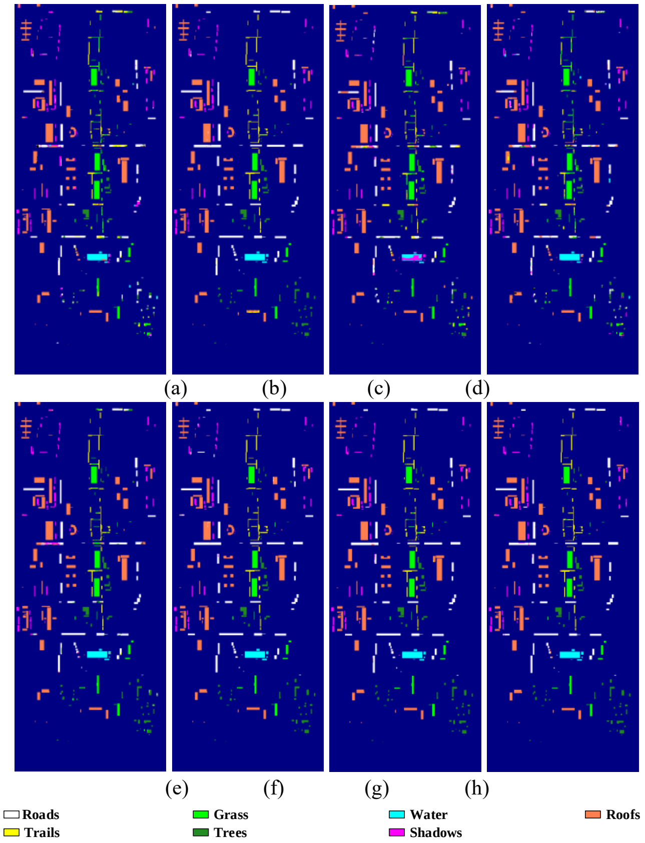


Fig. 10. Classification visual maps of the Washington dataset. (a)RBF-SVM, (b) PPF-CNN, (c) 3DCNN, (d) HSGAN, (e) 3D-GAN, (f) AD-GAN, (g) 3D-HyperGAMO, (h)CMC-GAN

For the Washington dataset, we randomly selected 3% of each class for training CMC-GAN and the remaining 97% for testing. Table VI shows the number of training and test in each category. The classification results of all the algorithms on the Washington dataset are shown in Table VII.

As can be seen from Table VII, the OA indicators of all the methods are above 90%. However, RBF-SVM does not perform

TABLE VII
CLASSIFICATION RESULTS OF RBF-SVM, PPF-CNN, 3D-CNN, HS-GAN, 3D-GAN, AD-GAN, 3D-HYPERGAMO AND CMC-GAN ON THE WASHINGTON DATASET

Class	RBF-SVM[11]	PPF-CNN[31]	3D-CNN[18]	HS-GAN[42]	3D-GAN[39]	AD-GAN[43]	3D-HyperGAMO [44]	CMC-GAN
1	94.1±3.1	97.9±0.6	92.5±2.2	92.8±3.1	96.1±0.1	95.5±1.4	99.7±0.3	98.8±0.6
2	93.4±0.6	97.6±0.1	92.8±7.0	94.9±0.1	95.4±3.8	99.2±0.6	94.9±3.0	99.9±0.2
3	98.3±0.1	100.0±0.0	91.6±5.8	95.8±0.3	99.6±0.0	99.6±0.6	95.7±2.4	99.7±0.4
4	88.2±3.9	95.6±3.1	91.5±3.2	90.8±3.5	99.0±1.1	94.4±3.8	99.9±0.1	96.9±2.0
5	95.6±0.4	99.9±0.1	88.0±5.2	90.0±0.0	99.5±0.3	98.2±0.5	99.3±0.6	99.9±0.2
6	91.6±3.5	97.5±1.2	90.4±2.4	91.6±1.1	97.0±1.0	89.8±3.5	98.7±1.1	95.2±1.7
7	98.2±1.5	94.9±0.1	99.0±0.4	94.7±1.3	98.2±0.7	99.8±0.3	99.6±0.3	99.9±0.1
OA(%)	93.7±0.4	95.7±0.3	94.4±0.6	92.5±1.6	97.2±0.3	97.5±0.5	98.8±0.5	99.0±0.1
AA(%)	92.3±0.8	95.9±0.5	92.3±0.8	90.8±1.8	97.0±0.5	96.7±0.6	98.3±0.7	98.5±0.2
Kappa(%)	93.7±0.6	95.5±0.3	92.8±0.7	90.3±1.6	96.7±0.4	96.8±0.6	98.4±0.6	98.8±0.3

TABLE VIII
RUNNING TIME OF RBF-SVM, PPF-CNN, 3DCNN, HSGAN, 3D-GAN, AD-GAN, 3D-HYPERGAMO AND CMC-GAN ON THE INDIAN PINES DATASET

Dataset	Method	Training Time(s)	Test Time(s)
Indian Pines	RBF-SVM	0.4±0.1	1.2±0.1
	PPF-CNN	2056.0±36.7	5.3±0.3
	3D-CNN	2690.2±57.9	16.0±0.1
	HS-GAN	444.7±73.1	0.3±0.0
	3D-GAN	597.67±60.8	0.3±0.0
	AD-GAN	327.4±4.2	0.4±0.0
	3D-HyperGAMO	3194.6±41.6	6.9±0.3
	CMC-GAN	322.9±30.3	0.2±0.0

TABLE IX
RUNNING TIME OF RBF-SVM, PPF-CNN, 3DCNN, HSGAN, 3D-GAN, AD-GAN, 3D-HYPERGAMO AND CMC-GAN ON THE PAVIA DATASET

Dataset	Method	Training Time(s)	Test Time(s)
Pavia University	RBF-SVM	0.5±0.1	1.4±0.2
	PPF-CNN	2414.0±374.0	19.8±6.2
	3D-CNN	1979.0±12.6	31.4±5.5
	HS-GAN	580.2±20.5	0.5±0.1
	3D-GAN	724.4±50.7	0.6±0.1
	AD-GAN	894.6±26.8	2.9±0.0
	3D-HyperGAMO	2083.9±437.1	17.3±0.4
	CMC-GAN	1177.95±30.7	1.26±0.1

TABLE X
RUNNING TIME OF RBF-SVM, PPF-CNN, 3DCNN, HSGAN, 3D-GAN, AD-GAN, 3D-HYPERGAMO AND CMC-GAN ON THE WASHINGTON DATASET

Dataset	Method	Training Time(s)	Test Time(s)
Washington	RBF-SVM	0.3±0.0	0.2±0.0
	PPF-CNN	926.8±29.5	5.2±0.5
	3D-CNN	1588.8±18.6	7.8±0.2
	HS-GAN	493.4±73.8	0.2±0.1
	3D-GAN	673.3±23.7	0.3±0.1
	AD-GAN	301.3±5.3	1.9±0.0
	3D-HyperGAMO	2023.7±12.3	9.6±0.4
	CMC-GAN	248.2±6.6	0.4±0.1

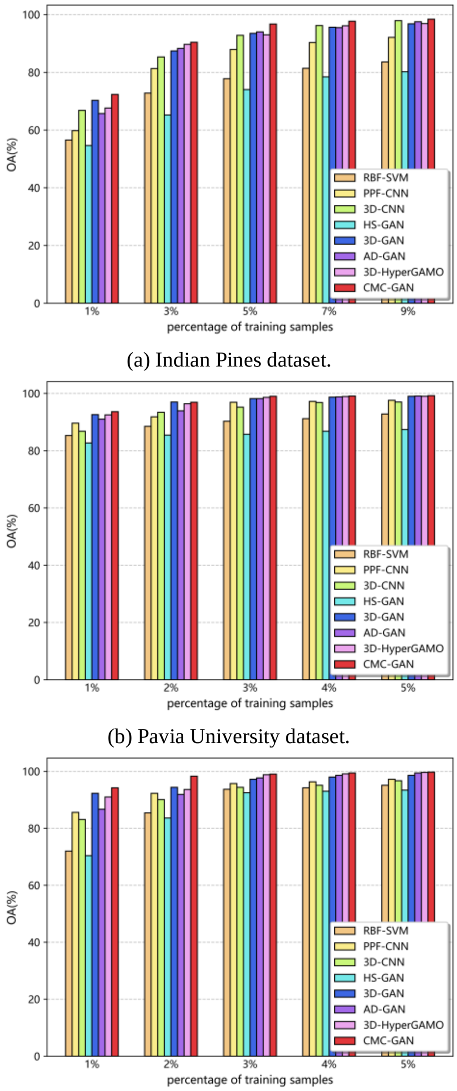
well in Roofs classification. The classification results of PPF-CNN in the Water reach the optimal, while 3D-CNN does not learn well in the Trails class. Among the GAN-based comparison methods, the classification of HS-GAN is not stable compared with other methods, and AD-GAN has poor classification performance in Trees category. The OA indicator of 3D-HyperGAMO is 5.5%, 1.6% and 1.5% higher than that of HS-GAN, 3D-GAN and AD-GAN, respectively. Compared with the above seven methods, the classification effect of the proposed CMC-GAN is the best, with OA, AA and Kappa increasing by at least 0.2%, 0.3% and 0.4%, respectively.

A visualization of the classification results on the Washington dataset is shown in Fig. 10. RBF-SVM, 3D-CNN and HS-GAN have poor visual performance with many misclassified areas. CMC-GAN has the best visualization results, with better regional consistency and fewer sample misclassified points within the region.

D. Analysis on Running Time

The training time and test time of above eight methods on the three HSI datasets are shown in Tables VIII-X. In the training stage of the three datasets, the input of RBF-SVM is one-dimensional spectral vector, and there is no iterative learning process, which has the lowest time consumption. PPF-CNN adopts the extended sample training strategy, which takes longer time than 3D-CNN. In the GAN-based methods, the input size of 3D-GAN is relatively large and the time consumption is relatively high. 3D-HyperGAMO focuses on the generation and feature extraction of a small number of categories, and the number of training iterations is larger. Therefore, the training time of 3D-HyperGAMO is longer than that of HS-GAN, 3D-GAN, AD-GAN and CMC-GAN. CMC-GAN takes some time to train the two generators and two discriminators, but in most cases CMC-GAN has a high time efficiency, and ranks second only behind RBF-SVM on the Indian Pines and Washington datasets.

In the test stage, PPF-CNN applies the voting strategy, 3D-GAN and 3D-Hypergamo apply 3D convolution, and these methods consume more time. The test time of HS-GAN is slightly faster than that of AD-GAN. CMC-GAN has less time consumption than AD-GAN and 3D-HyperGAMO.



(a) Indian Pines dataset.

(b) Pavia University dataset.

(c) Washington dataset.

Fig. 11. OA results with different percentages of training samples.

E. Sensitivity to the Proportion of Training Samples

In this part, we discuss the effectiveness of eight algorithms under different proportions of training samples. Fig. 11 indicates the results of the experiment. In the Indian Pines dataset, 1%, 3%, 5%, 7% and 9% of each category were randomly selected as training data. For the University of Pavia and Washington datasets, we selected 1%, 2%, 3%, 4%, and 5% of each category as the training sets.

As can be seen from Fig. 11, with the increase of training samples, the OA indexes of all the methods are improved, but the increase is gradually slow. The number of categories in Indian Pines dataset is not evenly distributed. In addition, the number of training samples of the Indian Pines dataset is small. As a result, the feature distribution in Indian Pines dataset is very complicated, and it is difficult for the network to extract features efficiently. When the data increases, the network can obtain more data for learning the feature distribution, the classification results greatly improved. In the three datasets, 3D-GAN performs better than 3D-CNN, PPF-CNN and HS-GAN when the training samples are relatively small. When the data volume is only 1%, the sample generation of 3D-GAN alleviates the problem of few training samples, and its performance ranks second after the proposed CMC-GAN. With the increase of training data, the performance gap between different models gradually narrows. However, the proposed method is obviously superior to other methods, which can achieve higher accuracy with fewer samples and save more labor cost and calculation power. Overall, the proposed CMC-GAN method is significantly better than other methods, especially under the condition of 1% training data, which proves its ability to deal with small sample problems.

F. Structural Ablation Analysis of CMC-GAN

To prove the validity of the proposed CMC-GAN, we have conducted ablation experiments on each part of the network separately. Nine experiments are set up for comparison. Specifically, the experiments are set for the respective effectiveness of multi-granularity adaptive fusion, unidirectional contrastive loss and class-specific contrastive

TABLE XI
CLASSIFICATION RESULTS (OA) OF THE EFFECTIVENESS OF DIFFERENT MODULES ON THE THREE DATASETS

Name	Coarse-grained GAN	Coarse-grained GAN	Unidirectional contrastive loss	Class-specific contrastive loss	Indian Pines dataset(%)	Pavia University(%)	Washington(%)
cgGAN	√				91.7±0.6	95.9±0.9	97.2±0.5
u-cgGAN	√		√		93.6±0.5	97.2±0.6	97.9±0.6
c-cgGAN	√			√	94.1±0.4	97.8±0.4	98.1±0.5
cgGAN-con	√		√	√	94.8±0.6	98.3±0.1	98.6±0.5
fgGAN		√			92.0±0.7	97.1±0.7	97.5±0.3
coGAN	√	√			93.0±0.6	97.8±0.5	98.1±0.3
u-coGAN	√	√	√		95.2±0.5	98.4±0.4	98.5±0.2
c-coGAN	√	√		√	96.0±0.3	98.6±0.2	98.7±0.1
CMC-GAN	√	√	√	√	**96.7±0.3**	**99.0±0.3**	**99.0±0.1**

TABLE XII
CLASSIFICATION RESULTS OF RBF-SVM, PPF-CNN, 3D-CNN, HS-GAN, 3D-GAN, AD-GAN, 3D-HYPERGAMO AND CMC-GAN ON THE HOUSTON 2018 DATASET

Class	RBF-SVM[11]	PPF-CNN[31]	3D-CNN[18]	HS-GAN[42]	3D-GAN[39]	AD-GAN[43]	3D-HyperGAMO[44]	CMC-GAN
OA(%)	55.3±0.2	71.1±0.2	74.4±0.1	62.1±0.3	73.4±0.1	76.9±0.1	78.6±0.1	83.4±0.2
AA(%)	26.6±0.4	49.4±0.4	52.7±0.2	35.1±0.7	58.4±0.3	64.5±0.1	70.2±0.1	76.1±0.3
Kappa(%)	40.6±0.5	62.8±0.2	66.9±0.1	48.6±0.4	66.0±0.1	70.7±0.1	72.9±0.1	79.5±0.1

TABLE XIII
CLASSIFICATION RESULTS OF MDGCN, SPECTRALFORMER AND CMC-GAN ON THE THREE DATASETS

Dataset	Index	MDGCN[53]	Spectral Former[54]	CMC-GAN
Indian Pines	OA(%)	94.8±0.1	80.0±1.3	96.7±0.3
	AA(%)	79.5±0.2	67.8±2.3	95.8±0.3
	Kappa(%)	95.3±0.8	77.2±1.5	95.0±1.0
Pavia University	OA(%)	97.2±0.1	92.8±0.7	99.0±0.3
	AA(%)	91.7±0.3	90.5±1.0	98.7±0.3
	Kappa(%)	96.3±0.1	90.4±1.0	98.7±0.4
Washington	OA(%)	97.4±0.1	96.7±0.6	99.0±0.1
	AA(%)	96.7±0.1	95.6±1.1	98.5±0.2
	Kappa(%)	96.6±0.1	95.7±0.8	98.8±0.3

TABLE XIV
RUNNING TIME OF MDGCN, SPECTRALFORMER AND CMC-GAN ON THE THREE DATASETS

Dataset	Method	Training Time(s)	Test Time(s)
Indian Pines	MDGCN[53]	665.9±4.7	0.6±0.0
	SpectralFormer[54]	167.7±1.5	1.6±0.0
	CMC-GAN	322.9±30.3	0.2±0.0
Pavia University	MDGCN[53]	2701±67.7	3.7±0.1
	SpectralFormer[54]	453.4±16.2	4.9±0.2
	CMC-GAN	1177.95±30.7	1.26±0.1
Washington	MDGCN[53]	545.2±32.2	3.1±0.1
	SpectralFormer[54]	253.2±2.5	2.3±0.0
	CMC-GAN	248.2±6.6	0.4±0.1

loss. Based on the original coarse-grained GAN (cgGAN), we have added unidirectional contrastive loss (u-cgGAN), and class-specific contrastive loss (c-cgGAN), and have added two kinds of contrastive losses (cgGAN-con). Secondly, the coarse-grained GAN and fine-grained GAN (fgGAN) are adaptively weighted (coGAN) for classification, then, unidirectional contrastive loss (u-coGAN), class-specific contrastive loss (c-coGAN) and two kinds of contrastive losses are added(proposed CMC-GAN) to coGAN for comparison.

The statistical results of the above methods in the three datasets are demonstrated in Table XI. The fgGAN behaves better than cgGAN because fgGAN extracts and learns more deeply abstract features for classification. Without adding the contrastlive losses, coGAN improves by at least 0.6% in the three datasets, indicating that the discriminators of different granularity can extract features from different perspectives. For the Indian Pines dataset, the training data is limited, and the category distribution is unbalanced, which increases the difficulty of classification. In the circumstances, coGAN has a relatively obvious improvement. The cgGAN holds significance as it demonstrates the potential to enhance performance in challenging data distribution scenarios. The performance of only adding class-specific contrastive loss is slightly better than adding unidirectional contrastive loss, which is because class-specific contrastive loss directly acts on the discriminators and improves the training of classification tasks more directly. After adding two kinds of contrastive losses, CMC-GAN significantly increases by 3.7% in the Indian Pines, 1.2% in Pavia University and 0.9% in Washington dataset, which indicates that the contrastive constraint in CMC-GAN can further improve the network performance.

G. Classification Results on the Houston 2018 dataset

The Houston 2018 dataset, utilized in the 2018 IEEE GRSS Data Fusion competition, comprises 48 bands and 20 classes, with the image size of 2384×601. In our experiments, we employed the upper half of the image as the training dataset and the lower half as the test set to validate the effectiveness of our proposed CMC-GAN on the big hyperspectral dataset. The experimental results for CMC-GAN and seven other comparative algorithms on the Houston 2018 dataset are are shown in Table XII.

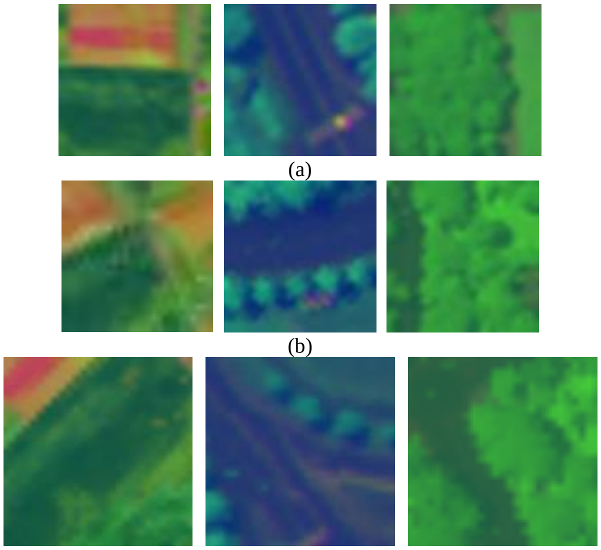
As shown in Table XII, the imbalanced and extensive number of classes in the large dataset pose a significant challenge to the discriminative capability of the discriminator. As a result, the majority of algorithms achieve an overall accuracy (OA) below 75%. Among the 8 algorithms, the classification results of RBF-SVM are lower than those of other algorithms. In the CNN-based methods, the OA of 3D-CNN is 3.3% higher than PPF-CNN. For GAN-based methods, HS-GAN performs poorly, and the OA values of 3D-GAN, AD-GAN, 3D-HyperGAMO and CMC-GAN are 11.3%, 14.8%, 16.5%, and 21.3% higher than HS-GAN, respectively. Among them, the three indexes of the proposed CMC-GAN are the highest, which proves the superiority of the proposed algorithm under large datasets.

H. Comparison With Other Deep Learning Algorithms

We add a novel GCN based approach[53] and a novel Transformer based approach [54] for comparison, which are recent state-of-the-art methods for HSI classification. The experiment is set up according to the original paper, using 5% training data for Indian Pines dataset and 3% training data for

TABLE XV
OVERALL ACCURACY (%) OF DIFFERENT TEMPERATURE COEFFICIENTS IN CMC-GAN UNDER INDIAN PINES, PAVIA UNIVERSITY AND WASHINGTON DATASET

τ dataset	0.1	0.3	0.5	0.7	0.9	1
Indian Pines	96.1±0.2	96.4±0.2	96.7±0.3	96.3±0.3	95.5±0.2	95.1±0.4
Pavia University	98.7±0.3	98.9±0.2	99.0±0.3	98.8±0.3	98.7±0.2	98.5±0.5
Washington	98.1±0.1	98.7±0.1	99.0±0.1	98.7±0.2	98.3±0.1	98.0±0.2


(a)

(b)

(c)

Figure. 12. Visualization of multi-scale samples generated by CMC-GAN. From left to right: Indian Pines dataset: Grass-pasture, Pavia University dataset: Meadows, Washington dataset: Trails. (a) real samples. (b) coarse-scale samples generated by CMC-GAN. (c) fine-scale samples generated by CMC-GAN.

Washington and Pavia University datasets. The specific experimental results are shown in the Table XIII.

As shown in Table XIII, MDGCN achieves better classification performance than SpectralFormer. MDGCN leverages graph convolution to effectively consider the spatial and spectral relationships and handle irregular data. In contrast, although SpectralFormer focuses on the local sequential features of the spectra, its network contains a large number of parameters, requiring a greater number of labeled samples for training. The OA index of MDGCN is 14.8%, 4.4% and 0.7% higher than SpectralFormer in terms of OA on three datasets respectively. Compared with MDGCN and SpectralFormer, the proposed CMC-GAN method achieves better classification performance on three HSI datasets.

The training time and test time of above eight methods on the three HSI datasets are shown in the Table XIV. In the training stage of the three datasets, it can be seen that the training time of SpectralFormer is slightly better than that of MDGCN, indicating that SpectralFormer has better time efficiency, second only to RBF-SVM. In the test phase, SpectralFormer is similar to the time consumed by MDGCN and the proposed CMC-GAN.

I. Analysis of the Temperature Parameter τ

In this part, we analyze the temperature parameters of the unidirectional contrastive loss and class-specific contrastive loss in the proposed CMC-GAN method. The experimental results are presented in Table XV.

It can be seen from Table XV that when the temperature parameter is less than 0.1, the classification result is the worst. When the temperature parameter is between 0.3 and 0.7, CMC-GAN has better classification results under the three datasets. When the temperature coefficient gradually tends to 0.5, the classification accuracy reaches the highest. A smaller temperature parameter will make the contrastive loss more sensitive, while a larger temperature parameter will reduce the sensitivity of the contrastive loss to the similarity difference between different sample pairs and improve the generalization ability of the model, but it may sacrifice part of the differentiation ability. Therefore, it is very important to choose the appropriate temperature parameter according to the network and dataset.

J. Visualization of Generated Samples by CMC-GAN

As shown in Figure. 12., the multi-scale samples generated by CMC-GAN under the three datasets are selected for display. The figures from the first row to the last row are the real samples, the samples generated by the coarse-grained generator, and the samples generated by the fine-grained generator in CMC-GAN. Among them, the size of the real samples and the coarse-scale samples obtained by the coarse-grained generator are 27×27, and the size of the fine-scale samples generated by the fine-grained generator is 108× 108.

It can be observed that the multi-scale samples generated by CMC-GAN exhibit favorable visual effects, demonstrating a certain degree of similarity to real samples while maintaining rich diversity. The multi-scale samples contain both the macroscopic and detailed information concerning the categories, which provides greater diversity and richer perspectives for the network's learning. The proposed unidirectional contrastive loss and class-specific contrastive loss introduce additional constraints and supervision, enabling joint optimization of two generators. This approach imposes greater restrictions and guidance on the generators, facilitating the generation of more diverse and mutually distinguishable samples. The above details show that multi-granularity complementary generators can effectively mitigate the phenomenon of mode collapse, which is prone to occur in GANs due to the complexity of the underlying distribution, of HSIs.

VI. CONCLUSION AND FUTURE WORK

This paper presents a unique CMC-GAN algorithm for HSI classification. The design of the GAN with complementary granularity is beneficial to the extraction of detailed boundary information and abstract semantic information for the samples.

The jointly optimized design improves the performance of the generators and discriminators, and alleviates the common pattern collapse in GANs. The addition of unidirectional contrastive loss and class-specific contrastive loss further improves the sample quality of generators and the feature extraction ability of discriminators. The experimental results on several HSI datasets show that the proposed CMC-GAN outperforms several state-of-the-art HSI classification methods. In the future, we will conduct more in-depth studies along the lines of processing fewer and more unbalanced samples, hoping to further improve the classification performance. In addition, we will explore the design of multiple branch networks to enable the GAN to learn and extract richer multi-granularity features, thereby generating more diverse high-quality samples.

REFERENCES

[1] B. Tu, X. Zhang, X. Kang *et al.*, “Spatial Density Peak Clustering for Hyperspectral Image Classification With Noisy Labels,” *IEEE Transactions on Geoscience and Remote Sensing,* vol. 57, no. 7, pp. 5085-5097, 2019.

[2] C. Chang, *Hyperspectral Data Exploitation: Theory and Applications*: Hyperspectral Data Exploitation: Theory and Applications, 2007.

[3] K. C. Tiwari, M. K. Arora, and D. Singh, “An assessment of independent component analysis for detection of military targets from hyperspectral images,” *International Journal of Applied Earth Observations & Geoinformation,* vol. 13, no. 5, pp. 730-740, 2011.

[4] C. M. Gevaert, J. Suomalainen, J. Tang *et al.*, “Generation of spectral–temporal response surfaces by combining multispectral satellite and hyperspectral UAV imagery for precision agriculture applications,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 8, no. 6, pp. 3140-3146, 2015.

[5] T. A. Carrino, A. Crósta, C. Toledo *et al.*, “Hyperspectral remote sensing applied to mineral exploration in southern Peru: A multiple data integration approach in the Chapi Chiara gold prospect,” *International Journal of Applied Earth Observation & Geoinformation*, pp. S0303243417301071, 2017.

[6] M. Moroni, E. Lupo, E. Marra *et al.*, “Hyperspectral Image Analysis in Environmental Monitoring: Setup of a New Tunable Filter Platform,” *Procedia Environmental Sciences,* vol. 19, pp. 885-894, 2013.

[7] J. Feng, J. Chen, Q. Sun *et al.*, “Convolutional neural network based on bandwise-independent convolution and hard thresholding for hyperspectral band selection,” *IEEE Transactions on Cybernetics*, vol. 51, no. 9, pp. 4414-4428, 2020.

[8] J. Feng, G. Bai, D. Li *et al*., “MR-Selection: A Meta Reinforcement Learning Approach for Zero-Shot Hyperspectral Band Selection,” *IEEE Transactions on Geoscience and Remote Sensing*, 2022.

[9] S. Xu, S. Liu, H. Wang *et al.*, “A hyperspectral image classification approach based on feature fusion and multi-layered gradient boosting decision trees,” *Entropy,* vol. 23, no. 1, pp. 20, 2020.

[10] J. Li, J. M. Bioucas-Dias, and A. Plaza, “Spectral–spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields,” *IEEE Transactions on Geoscience and Remote Sensing,* vol. 50, no. 3, pp. 809-823, 2011.

[11] F. Melgani, and L. Bruzzone, “Classification of hyperspectral remote sensing images with support vector machines,” *IEEE Transactions on geoscience and remote sensing,* vol. 42, no. 8, pp. 1778-1790, 2004.

[12] Y. Chen, Z. Lin, X. Zhao *et al.*, “Deep learning-based classification of hyperspectral data,” *IEEE Journal of Selected topics in applied earth observations and remote sensing,* vol. 7, no. 6, pp. 2094-2107, 2014.

[13] Y. Chen, X. Zhao, and X. Jia, “Spectral–spatial classification of hyperspectral data based on deep belief network,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 8, no. 6, pp. 2381-2392, 2015.

[14] P. Zhong, Z. Gong, S. Li *et al.*, “Learning to diversify deep belief networks for hyperspectral image classification,” *IEEE Transactions on Geoscience and Remote Sensing,* vol. 55, no. 6, pp. 3516-3530, 2017.

[15] C.-Y. Wang, P.-C. Chang, J.-J. Ding *et al.*, “Spectral-temporal receptive field-based descriptors and hierarchical cascade deep belief network for guitar playing technique classification,” *IEEE Transactions on Cybernetics*, 2020.

[16] B. Pan, Z. Shi, and X. Xu, “R-VCANet: A new deep-learning-based hyperspectral image classification method,” *IEEE Journal of selected topics in applied earth observations and remote sensing,* vol. 10, no. 5, pp. 1975-1986, 2017.

[17] S. Li, W. Song, L. Fang *et al.*, “Deep learning for hyperspectral image classification: An overview,” *IEEE Transactions on Geoscience and Remote Sensing,* vol. 57, no. 9, pp. 6690-6709, 2019.

[18] Y. Chen, H. Jiang, C. Li *et al.*, “Deep feature extraction and classification of hyperspectral images based on convolutional neural networks,” *IEEE Transactions on Geoscience and Remote Sensing,* vol. 54, no. 10, pp. 6232-6251, 2016.

[19] W. Hu, Y. Huang, L. Wei *et al.*, “Deep convolutional neural networks for hyperspectral image classification,” *Journal of Sensors,* vol. 2015, 2015.

[20] H. Liang, and Q. Li, “Hyperspectral imagery classification using sparse representations of convolutional neural network features,” *Remote Sensing,* vol. 8, no. 2, pp. 99, 2016.

[21] H. Zhang, Y. Li, Y. Zhang *et al.*, “Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network,” *Remote sensing letters,* vol. 8, no. 5, pp. 438-447, 2017.

[22] A. B. Hamida, A. Benoit, P. Lambert *et al.*, “3-D deep learning approach for remote sensing image classification,” *IEEE Transactions on geoscience and remote sensing,* vol. 56, no. 8, pp. 4420-4434, 2018.

[23] M. He, B. Li, and H. Chen, "Multi-scale 3D deep convolutional neural network for hyperspectral image classification." pp. 3904-3908.

[24] Z. Zhong, J. Li, Z. Luo *et al.*, “Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework,” *IEEE Transactions on Geoscience and Remote Sensing,* vol. 56, no. 2, pp. 847-858, 2017.

[25] C. Szegedy, W. Liu, Y. Jia *et al.*, "Going deeper with convolutions." in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1-9, 2015.

[26] Z. Gong, P. Zhong, Y. Yu *et al.*, “A CNN with multiscale convolution and diversified metric for hyperspectral image classification,” *IEEE Transactions on Geoscience and Remote Sensing,* vol. 57, no. 6, pp. 3599-3618, 2019.

[27] X. Wang, K. Tan, P. Du *et al.*, “A Unified Multiscale Learning Framework for Hyperspectral Image Classification,” *IEEE Transactions on Geoscience and Remote Sensing,* vol. 60, pp. 1-19, 2022.

[28] D. Wang, B. Du, L. Zhang *et al.*, “Adaptive spectral–spatial multiscale contextual feature extraction for hyperspectral image classification,” *IEEE Transactions on Geoscience and Remote Sensing,* vol. 59, no. 3, pp. 2461-2477, 2020.

[29] W. Song, S. Li, L. Fang *et al.*, “Hyperspectral image classification with deep feature fusion network,” *IEEE Transactions on Geoscience and Remote Sensing,* vol. 56, no. 6, pp. 3173-3184, 2018.

[30] H. Lee, and H. Kwon, “Going deeper with contextual CNN for hyperspectral image classification,” *IEEE Transactions on Image Processing,* vol. 26, no. 10, pp. 4843-4855, 2017.

[31] W. Li, G. Wu, F. Zhang *et al.*, “Hyperspectral image classification using deep pixel-pair features,” *IEEE Transactions on Geoscience and Remote Sensing,* vol. 55, no. 2, pp. 844-853, 2016.

[32] H. Yan, J. Wang, L. Tang *et al.*, “A 3D cascaded spectral–spatial element attention network for hyperspectral image classification,” *Remote Sensing,* vol. 13, no. 13, pp. 2451, 2021.

[33] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza *et al.*, "Generative Adversarial Nets." in *NIPS*. 2014.

[34] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks." in *International conference on machine learning*, pp. 214-223, 2017.

[35] I. Gulrajani, F. Ahmed, M. Arjovsky *et al.*, “Improved training of wasserstein gans,” *Advances in neural information processing systems,* vol. 30, 2017.

[36] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434*, 2015.

[37] M. Mirza, and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014.

[38] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans." in *International conference on machine learning*, pp. 2642-2651, 2017.

[39] L. Zhu, Y. Chen, P. Ghamisi *et al.*, “Generative adversarial networks for hyperspectral image classification,” *IEEE Transactions on Geoscience and Remote Sensing,* vol. 56, no. 9, pp. 5046-5063, 2018.

[40] J. Feng, H. Yu, L. Wang *et al.*, "Classification of hyperspectral images based on multiclass spatial–spectral generative adversarial networks," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 57, no. 8, pp. 5329-5343, 2019.

[41] H. Wang, C. Tao, J. Qi *et al.*, "Semi-supervised variational generative adversarial networks for hyperspectral image classification.". in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 9792-9794, 2019.

[42] Y. Zhan, D. Hu, Y. Wang *et al.*, "Semisupervised hyperspectral image classification based on generative adversarial networks," *IEEE Geoscience and Remote Sensing Letters,* vol. 15, no. 2, pp. 212-216, 2017.

[43] J. Wang, F. Gao, J. Dong *et al.*, "Adaptive DropBlock-Enhanced Generative Adversarial Networks for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 59, no. 6, pp. 5040-5053, 2021.

[44] S. K. Roy, J. M. Haut, M. E. Paoletti *et al.*, "Generative Adversarial Minority Oversampling for Spectral–Spatial Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 60, pp. 3052048, 2022.

[45] T. Chen, S. Kornblith, M. Norouzi *et al.*, "A simple framework for contrastive learning of visual representations." in *International conference on machine learning*. 2020. pp. 1597-1607.

[46] K. He, H. Fan, Y. Wu *et al.*, "Momentum contrast for unsupervised visual representation learning." in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 9729-9738, 2020.

[47] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon *et al.*, "Learning deep representations by mutual information estimation and maximization," *arXiv preprint arXiv:1808.06670*, 2018.

[48] W. Wang, T. Zhou, F. Yu *et al.*, "Exploring cross-image pixel contrast for semantic segmentation." in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7303-7313, 2021.

[49] Z. Wu, Y. Xiong, S. X. Yu *et al.*, "Unsupervised feature learning via non-parametric instance discrimination." in *Proc. IEEE conference on computer vision and pattern recognition*, pp. 3733-3742, 2018.

[50] M. Caron, I. Misra, J. Mairal *et al.*, "Unsupervised learning of visual features by contrasting cluster assignments," *Advances in Neural Information Processing Systems,* vol. 33, pp. 9912-9924, 2020.

[51] P. Khosla, P. Teterwak, C. Wang *et al.*, "Supervised contrastive learning," *Advances in Neural Information Processing Systems,* vol. 33, pp. 18661-18673, 2020.

[52] J. Robinson, C.-Y. Chuang, S. Sra *et al.*, "Contrastive learning with hard negative samples," *arXiv preprint arXiv:2010.04592*, 2020.

[53] Y. Kalantidis, M. B. Sariyildiz, N. Pion *et al.*, "Hard negative mixing for contrastive learning," *Advances in Neural Information Processing Systems,* vol. 33, pp. 21798-21809, 2020.

[54] G. M. Foody, "Status of land cover classification accuracy assessment," *Remote Sensing of Environment,* vol. 80, no. 1, pp. 185-201, 2002.

[55] S. Wan, C. Gong, P. Zhong, B. Du, L. Zhang, and J. Yang, "Multiscale dynamic graph convolutional network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens*., vol. 58, no. 5, pp. 3162–3177, 1088 May 2020.

[56] D. Hong, Z. Han, J. Yao et al., "SpectralFormer: Rethinking hyperspectral image classification with transformers," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-15, 2021.

[57] F. Deng, S. Pu, X. Chen et al., "Hyperspectral image classification with capsule network using limited training samples," *Sensors,* vol. 18, no. 9, pp. 3153, 2018.

[58] D. Hong, L. Gao, J. Yao et al., "Graph convolutional networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 7, pp. 5966-5978, 2020.

[59] A. Paul, and S. Bhoumik, "Classification of hyperspectral imagery using spectrally partitioned HyperUnet," *Neural Computing and Applications*, pp. 1-10, 2022.

[60] X. Wu, D. Hong, and J. Chanussot, "UIU-Net: U-Net in U-Net for infrared small object detection," *IEEE Transactions on Image Processing*, vol. 32, pp. 364-376, 2022.

[61] R. Hang, X. Qian, and Q. Liu, "Cross-Modality Contrastive Learning for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-12, 2022.

[62] R. Hang, F. Zhou, Q. Liu et al., "Classification of hyperspectral images via multitask generative adversarial networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 2, pp. 1424-1436, 2020.

[63] Z. Zhang, Y. Ding, X. Zhao et al., "Multireceptive field: An adaptive path aggregation graph neural framework for hyperspectral image classification," *Expert Systems with Applications,* vol. 217, pp. 119508, 2023.

[64] Y. Ding, Z. Zhang, X. Zhao et al., "Multi-scale receptive fields: Graph attention neural network for hyperspectral image classification," *Expert Systems with Applications*, vol. 223, pp. 119858, 2023.

[65] J. Feng, N. Zhao, R. Shang et al., "Self-supervised divide-and-conquer generative adversarial network for classification of hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-17, 2022.