Pattern Recognition 142 (2023) 109701

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Features kept generative adversarial network data augmentation strategy for hyperspectral image classification

Mingyang Zhang[a], Zhaoyang Wang[a], Xiangyu Wang[a], Maoguo Gong[a,*], Yue Wu[b], Hao Li[a]

[a] School of Electronic Engineering of Xidian University, No. 2 South TaiBai Road, Xi'an 710071, China
[b] School of Computer Science and Technology of Xidian University, No. 2 South TaiBai Road, Xi'an 710071, China

ARTICLE INFO

Article history:
Received 7 February 2023
Revised 15 April 2023
Accepted 18 May 2023
Available online 23 May 2023

Keywords:
Hyperspectral images (HSIs)
Deep learning
Generative adversarial network (GAN)
Data augmentation

ABSTRACT

In recent years, significant breakthroughs have been achieved in hyperspectral image (HSI) processing using deep learning techniques, including classification, object detection, and anomaly detection. However, the practical application of deep learning in HSI processing is limited by challenges such as small-sample size and sample imbalance issues. To mitigate these limitations, we propose a novel data augmentation strategy called Feature-Preserving Generative Adversarial Network Data Augmentation (FPGANDA). What sets our data augmentation strategy apart from existing generative model-based approaches is that we preserve the main spectral bands of HSI data using a newly designed band selection method. Additionally, our proposed generative model generates synthetic spectral bands, which are combined with the real spectral bands using a mixture strategy to create augmented data. This approach ensures that the augmented data retain the main features of the original data while also incorporating diverse features from the generated data. We evaluate our method on three different HSI datasets, comparing it with state-of-the-art techniques. Experimental results demonstrate that our proposed method significantly improves classification performance in most scenes and exhibits remarkable compatibility.

1. Introduction

Hyperspectral images (HSIs) provide both spatial and spectral information. The difference between HSI data from traditional remote sensing images is that HSI data have hundreds of bands, which are useful in many applications, such as environmental monitoring, target detection, classification and mineral mining, and many more [1]. Among these applications, the joint spatial and spectral feature-oriented pixel-level classification of HSI is still an open challenge.

Substantial research on HSI classification has been done in the literature, and the majority of HSI classification frameworks seem to have been influenced by the approaches used in the computer vision area [2]. The majority of them employ handcrafted features to train the classifier in traditional machine learning-based HSI classification techniques. These techniques typically focus on applying engineering abilities and subject-matter knowledge to design several traits created by humans, like shape, texture, features of color, spectral content, and spatial distribution. Most of them are essential aspects of an image and can provide effective characteristics for image categorization, such as color histograms, histogram of oriented gradients (HOG) [3], scale-invariant feature transform (SIFT), global image scale-invariant transform global invariant scalable transform (GIST), local binary patterns (LBPs) [4] and kernel-based support vector machine (SVM) [5].

Most of these strategies are designed to artificially extract the features which can effectively represent the various attributes of an image. However, these features may be insubstantial in the practical application, making it challenging to fine-tune between robustness and discriminability, as significant differences exist between various data. Additionally, the design of the features with human input has a significant impact on the categorization procedure, as it demands a high level of domain knowledge.

In order to lessen the drawbacks of handcrafted feature designing, an approach for deep feature learning was proposed by Hinton and Salakhutdinov. Deep learning-based (DL-based) techniques can learn on their own, constructing the features from the data in a hierarchical fashion until a sufficient representation is reached. The behavior of any data can be learned using deep learning frameworks without any prior knowledge regarding the statistical distribution of the input data. Without any pre-specified knowledge, deep learning frameworks can extract both linear and nonlinear features from input data. Deep learning architectures of HSI clas-

[*] Corresponding author.
E-mail addresses: 957683213@qq.com (Z. Wang), 792926134@qq.com (X. Wang), gong@ieee.org (M. Gong), ywu@xidian.edu.cn (Y. Wu), haoli@xidian.edu.cn (H. Li).

https://doi.org/10.1016/j.patcog.2023.109701
0031-3203/© 2023 Elsevier Ltd. All rights reserved.

sification are mainly represented by convolutional neural networks (CNNs) [6], graph convolutional networks (GCN) [7], autoencoders (AEs), deep belief networks (DBNs) [8], and recurrent neural networks (RNNs) [9].

Among these deep models for HSI classification, the most commonly used is the CNN-based network. CNN is a hierarchical feature learning model, which is usually contained convolutional layers, pooling layers, and activation layers which can add nonlinear features to the model. The convolutional layer extracts the features from input data by convolving a learned kernel with it. By sharing parameters through convolutional kernels, CNNs can reduce the number of parameters a model needs to train and speed up training. CNN-based HSI classification frameworks can be divided into three main categories: spectral CNN frameworks, spatial CNN frameworks and spatial-spectral CNN frameworks. Spectral CNN models [10] only consider 1-D spectral information while spatial CNN models [11] only consider spatial information. Both lose a certain amount of feature information. To take full use of the spatial-spectral information of HSI data, this can be achieved on the one hand by fusing spatial and spectral feature information, and on the other hand by using 3D convolution to directly extract features from the HSI data cube, such as HamidaEtAl network [12], LiEtAL network [13], HeEtAl network [14] and many more [15]. Most recently, in Alkhatib et al. [16] a Tir-CNN was proposed to extract features in multiple dimensions for hyperspectral image classification and in Li et al. [17], the authors combined transformer with CNN to develop a convolution-transformer adaptive fusion network.

Despite the advantage of the DL-based approach described above, some challenges remain. The growing number of parameters makes processing such high-dimensional data a challenging task. This is known as the curse of dimensionality, which has a significant impact on classification performance. Since the size of the training data is not adequate, the training samples do not include any novel information for the model or they share similar patterns or structures, which makes it hard to correctly train the classifier. This could cause the model to overfit. This is known as the Hughes phenomena which occurs when the amount of labeled training data is much less than the total number of spectral bands contained in the data. A major issue in HSI classification is the lack of labeled HSI data because labeling HSI typically requires the employment of human experts or the investigation of real-time scenarios, which is time-consuming and expensive. In some cases, the available labeled samples may not be sufficient to train a robust classification model. In such scenarios, the model tends to overfit the training data, leading to poor generalization performance on the test data. Furthermore, the problem of imbalanced data categories is significant in HSI data and is a key issue limiting the improvement of classification accuracy. In many real-world applications, the number of samples in different classes may not be evenly distributed. Some classes may have significantly more samples than others, which can lead to bias in the classification model towards the majority classes. As a result, the model may perform well on the majority classes but poorly on the minority classes, which can be critical in applications where the minority classes are of high interest. In addition to this, more human and material resources will be consumed to address the category imbalance problem. Hyperspectral data is expensive to acquire in the first place, and some unique feature classes such as some priceless vegetation and minerals, are uncommon in an HSI.

To address these issues, various approaches have been proposed, such as data augmentation, transfer learning, and sample selection strategies. These methods aim to improve the robustness of the classification model against small-sample size and class imbalance, and to enhance the accuracy and reliability of hyperspectral classification in practical applications.

Data augmentation can be seen as implicit regularization to relieve these problems. Several data augmentation attempts have been conducted such as rotation, scaling, flipping, adding radiation noise or adding mixture noise, and elastic transforms of the image data [18]. These strategies are simple and effective but they have some limitations that can not add nonlinear and hierarchical features for classification. To alleviate this problem, deep learning methods like generative adversarial networks (GAN) [19] and variational autoencoder (VAE) [20] are utilized in data augmentation, which can generate new samples to expand the training set. These generative models can be trained to approximate the true distribution of the data, with some additional non-linear, hierarchical features added in the process of approximation. However, gradient explosion or gradient disappearance phenomena are common, and the generative model is difficult to train and converge. In this case, these generative models will destroy the original features of the data which are sensitive to our classification accuracy.

To overcome these limitations, in this study we propose a new data augmentation strategy based on GAN and a new band selection strategy. We combine the advantages of traditional methods and the advantages of generative model augmentation methods. We keep the main features of original data, which are selected by our band selection strategy. Then we fill the remaining part of the data with the ones generated by the GAN-based models. We call the whole data augmentation strategy ‘Feature-preserving generative adversarial network data augmentation’ (FPGANDA). The proposed FPGANDA model and all the attributes of the framework are illustrated in Fig. 1. The main contributions of this study can be summarized as follows.

1) Aiming at the problems of small-sample and imbalance problems in the classification of HSI, we propose a new ‘FPGANDA’ strategy to improve the generalization performance of hyperspectral classification networks.
2) We design a new band selection strategy considering both the data characteristics of the original data and the quality of the images generated by the generator. We also design a mixture strategy between the original data and the generated data, which can balance the original linear features of the synthesized data with the non-linear features introduced by GAN.
3) We incorporate feature band retention and fusion strategies into the GAN training process and also utilize the idea of sub-batch training. This approach improves the stability and diversity of GAN training.

The remainder of this study is organized as follows. Section 2 introduces some related works about the HSI data augmentation and the motivation of our strategy. Section 3 introduces the proposed “FPGANDA” in three parts: the overview of our framework, the band selection and mixture strategy, and Feature-preserving generative adversarial networks. Our experimental results are presented in Section 4 with a detailed discussion about the classification result and the classification maps. We also conduct the ablation study in this part. Section 5 makes a conclusion of our work and gives some recommendations for further research work.

2. Related works

2.1. GAN-based data augmentation strategy

To relieve the small samples and imbalance problems in HSI data, many data-generative strategies have been attempted. In [21], the authors simply mirror the training samples to generate new data for improved classification. Except for this simple image transformation, in Nalepa et al. [22], the authors propose a data augmentation strategy based on per spectral band standard deviation

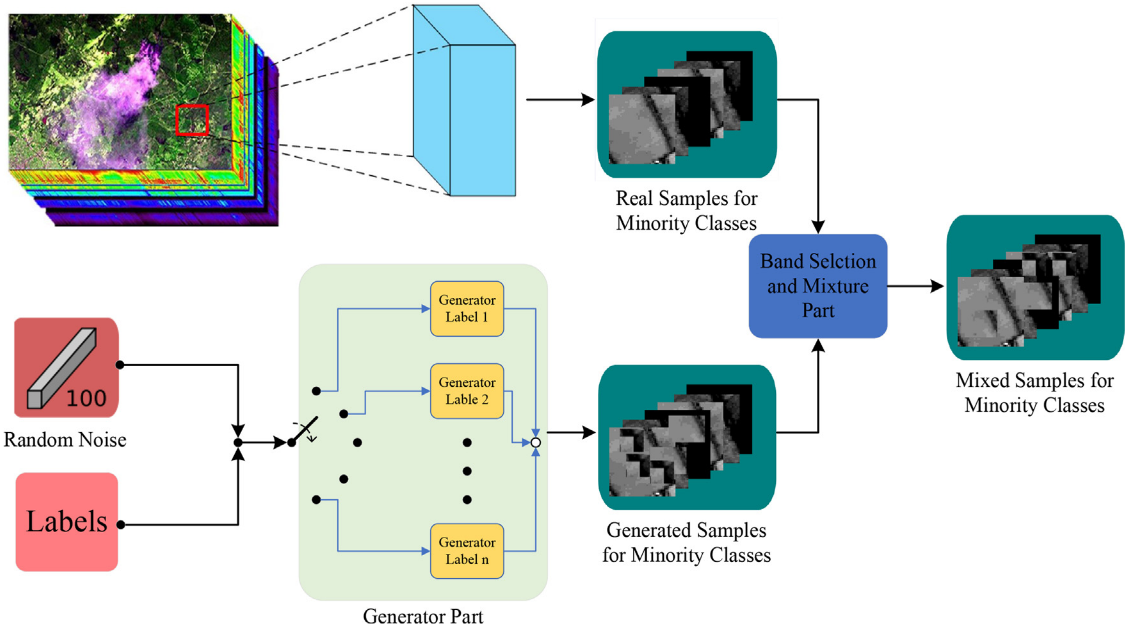

Real Samples for Minority Classes
Band Selction and Mixture Part
Mixed Samples for Minority Classes
Generator Label 1
Generator Label 2
Generator Label n
100
Random Noise
Labels
Generated Samples for Minority Classes
Generator Part

Fig. 1. Graphical overview of the proposed FPGANDA architecture.

(for each class) in the training set. They use the rich spectral information to generate new samples, but the spatial information of the image is ignored. Later Li et al. utilize both spectral and spatial information to synthesize new samples [23], which can achieve better accuracy.

The generative adversarial networks [19] have already attracted research attention in several HSI tasks, due to its powerful data generation capabilities. In the scene of the HSI classification, a significant number of works have also been attempted using generative adversarial networks. For instance, Zhan et al.[24] have developed a framework for semi-supervised HSI classification using a 1-D GAN for HSI (HSGAN). The HSGAN classifies HSI data using discriminator features. For HSI classification, the GAN has also been investigated using CRFs [25]. The HSI classification map obtained is improved by the dense CRFs. In [26], by creating the samples with a 1-D structure triple generative adversarial network (TripleGAN) and categorizing the HSI data with a capsule network (CapsNet)[27], the Caps-TripleGAN model has been investigated. In [28], Lin et al. combined GAN and classifier together. The generator generates new samples from noises and the discriminator accepts both generated data and real data. Unlike the conventional discriminator, which only identifies the genuine and the fake, here the discriminator has to perform a classification task at the same time. Inspired by this, Mullick et al.[29] proposed generative adversarial minority oversampling to generate new samples, which uses a 1-D generator and a 1-D discriminator. Motivated by Mullick et al. [29], in Roy et al. [30], the authors upgraded the model by replacing the 1-D generator and the 1-D discriminator with 3-D, which can better utilize spatial information of HSI data. Most recently, in Sun et al. [31], the authors proposed a generative adversarial networks based on WGAN-GP[32] and ACGAN [33](AC-WGAN-GP) to generate new samples for better HSI classification.

2.2. Feature-preserving strategy

In [34], a data augmentation method that keeps the main features of the image data and replaces the rest with noise is proposed. Ghiasi et al. [35] demonstrate that a simple copy-and-paste strategy in instance partitioning works well. Both articles show that the key characteristics of the data are important for subsequent task performance. This motivated us to think about using this key feature retention strategy with hyperspectral data because of its high spectral complexity and information content. At the same time GAN has great research value and potential for data augmentation work, so we attempt to combine feature retention techniques with GAN networks.

In [34], although the main features can be retained, the noise filling of the rest is uncertain and the different noise in the rest can have different effects on the network classification, which can cause network underfitting problems if not chosen properly. The randomness of the noise contradicts the stability of the strategy. In the scenario where GAN is used to augment the data, it faces two problems: firstly, GAN generative models are very difficult to train, difficult to converge, and will often encounter gradient explosion or gradient disappearance during the training process [36], which can lead to the generated images being very far from the real data, misleading the classification network in the already sparse data situation and causing a certain underfitting problem. The second problem is that in the HSI classification scenario, we use GAN to solve small sample problems, which means that there is very little data for training GAN. The GAN will be more challenging to train and converge with too few training samples, and the produced samples won't be diverse enough to yield adequate feature information for classification.

2.3. Motivation

To address the above limitations, we propose an amplification method for data generation based on feature retention called “Feature-preserving generative adversarial network data augmentation” (FPGANDA). Hyperspectral data has rich spectral information, and the amount of information in different channels of the spectral information varies. Firstly, the informative ones will be retained, and then we fill the remaining part of the data with the ones generated by the GAN-based models, which does not destroy the most realistic and dominant features of the image data. Secondly, replacing the rest with GAN-generated data does not introduce too much noise. The retention of the main features ensures that the classification network does not suffer from a degree of underfitting, while the replacement of the rest with GAN-generated

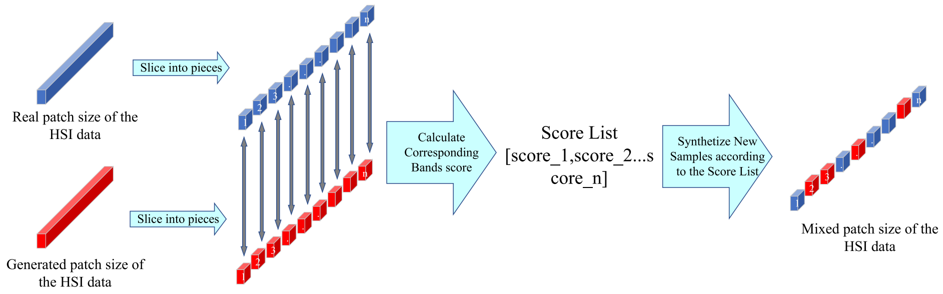

Real patch size of the HSI data
Slice into pieces
Generated patch size of the HSI data
Slice into pieces
1
2
3
n
Calculate Corresponding Bands score
Score List [score_1,score_2...score_n]
Synthetize New Samples according to the Score List
Mixed patch size of the HSI data

Fig. 2. Graphical overview and the detailed information of the band selection and mixture part.

data ensures that the classification network does not suffer from a degree of overfitting. Converting noise-substituted parts to GAN-generated ones can introduce certain non-linear features and improve the overall quality of the composite image, because GAN-generated data are more representative and predictable than noisy data, which helps to improve classification accuracy and stability. Furthermore, in this case, the aim of our GAN is not to generate complete new data end-to-end, but to generate partial bands of the data. Therefore the GAN will be easier to train and converge, as the main features are retained which is equivalent to having the main features already trained. This effectively reduces the impact of gradient disappearance, gradient explosion and the Hughes phenomena on GAN generation results during the training process.

3. Methodology

3.1. Overview of our framework

The overview of our framework can be seen in Fig. 1. First, we enter the noise and the labels of the samples to be generated into the generation part. The generator is pre-trained according to the augmentation sample labels. Detailed information on the training process of the generator is available in the sub-section C. The generation part will output data of the same size as the real data, both of which will be fed into the band selection and mixture part at the same time and then output feature-preserving hybrid composite data. The synthesized data will be used as augmented data and added to the classification network along with the real data for training. Detailed information on the individual parts is described in the sub-section that follows.

3.2. Band selection and mixture strategy

The HSI data have rich spectral information. In order to select the most typical bands, we design a new band selection strategy. In our strategy, we keep the informative bands and use the data generated by the generator to replace the rest of the bands. The graphical overview of our proposed strategy can be seen in Fig. 2. The biggest difference between our strategy and the traditional band selection strategy is that we consider both the original features of the raw HSI bands and the quality of the data generated by our generator. Kullback-Leibler Divergence (KL Divergence) is generally used to measure the “distance” between two probability distribution functions, whose definition is given by:

$$\boldsymbol{KL}[P(X)||Q(X)] = \sum_{\mathbf{x}\epsilon X}[P(\mathbf{x})log\frac{P(\mathbf{x})}{Q(\mathbf{x})}] \tag{1}$$

where $P(\mathbf{x})$ and $Q(\mathbf{x})$ represent two different data distributions.

We use the KL Divergence between the original data and the generated data to measure the quality of the generated images. A large value means that our generator is generating poor quality and a small value means that our generator works well. We use the variance of the data to characterize the original features of the data. A large value means the band contains more information. First, we slice the original data and the generated data, which has the same size, by each band.

$$P(X) = \{P(X_1), P(X_2), P(X_3)\dots P(X_n)\} \tag{2}$$

$$G(Z) = \{G(Z_1), G(Z_2), G(Z_3)\dots G(Z_n)\} \tag{3}$$

where $P(X_i)$ represents the distribution of the band from the original data, and $G(Z_i)$ represents the distribution of the band from the generated data. Our score function is given by:

$$Score_i = \boldsymbol{KL}[P(X_i)||G(Z_i)] + \sigma(P(X_i)) \tag{4}$$

$$= \sum_{\mathbf{x}\epsilon X_i,\mathbf{z}\epsilon Z_i}[P(\mathbf{x})log\frac{P(\mathbf{x})}{G(\mathbf{z})}] + \sqrt{\frac{\sum_{i=1}^{n}(\mathbf{x}-\mu)^2}{n}} \tag{5}$$

The $\sigma(x)$ is a function to calculate the variance. When the value of the first term is high, it means that the quality of the band generated by the generator is poor and we tend not to apply the band generated by the generator but to retain the original band. When the value of the second term is high, it means that the original band contains more information and we prefer to keep this band. After calculating the scores for each band, they are sorted in descending order and the top bands are selected for retention. We found it reasonable to set the retention percentage at 60%, due to our experimental of the hyperparameter. Then we update the mask bands $M(S)$,where $M(S) = [M(S_i)]_i$ is the binary mask for S and generate new mixed data. Our mixing function is given by:

$$\hat{x} = (1 - M(S)) \odot G(Z) + M(S) \odot P(X) \tag{6}$$

where $\odot$ denotes the element-wise multiplication of two vectors in corresponding positions and $(1 - M(S))$ represents a binary matrix obtained by performing a position-by-position 01 inversion of matrix $M(S)$. The general workflow of our proposed band selection and mixture strategy is presented in Algorithm 1 Algorithm 2.

Algorithm 1 Feature-preserving generative adversarial networks data augmentation strategy.

INPUT: training data: $P(X)$, augmentation labels: $i, j\dots.k$, random noise: z
1. Input the training data $P(X)$, augmentation labels $i, j\dots.k$ and random noise z into the generative adversarial networks part.
2. Train the corresponding generative adversarial networks.
3. Input the random noise z and augmentation labels $i, j\dots.k$ into the generative adversarial networks part to generate new data: $G(Z)$.
4. Input the training data $P(X)$ and the generated data $G(Z)$ to the band selection and mixture part to synthesize mixed data: $P(\hat{X})$.
Return the mixed data: $P(\hat{X})$

Algorithm 2 Band selection and mixture strategy.

```
INPUT: original data: $P(X)$, generated data: $G(Z)$,
1. Slice the original data $P(X)$ and the generated data $G(Z)$ into each band.
    $P(X) = \{P(X_1), P(X_2), P(X_3) \ldots P(X_n)\}$
    $G(Z) = \{G(Z_1), G(Z_2), G(Z_3) \ldots G(Z_n)\}$
2. Calculate the score for each corresponding band between $P(X)$ and $G(Z)$.
    $Score(i) = \boldsymbol{KL}[P(X_i)||G(Z_i)] + \sigma(P(X_i))$
3. Sort the scores for each band in descending order and update the mask band $M(S)$.
4. Mix to generate new data.
    $\hat{x} = (1 - M(S)) \odot G(Z) + M(S) \odot P(X)$
Return $\hat{x}$
```

3.3. Feature-preserving generative adversarial networks

The generative adversarial networks contain two main parts: the generator and the discriminator. Our generative network consists of the following layers.

1) Three ConvTranspose3D layers with ten filters which have kernel size of (3,3,3), followed by ten filters which have kernel size of (3,1,1) and followed by one filter which has a kernel size of (3,3,3).
2) Three BatchNorm3D layers which can accelerate training speed and improve the accuracy.
3) Three LeakyReLU layers followed by each BatchNorm3D layer and one Tanh layer.
4) One flattened layer which can spread all features into one dimension, three Linear layers reshaping the size of the feature to the target size and one Reshape layer produces the final output, which has the same size as the real data ($Patchs \times Patchs \times Bands$).

We symmetrically design our discriminator. The differences between the generator network and the discriminator network are as follows.

1) We replace the ConvTranspose3D layers with Conv3D layers.
2) We replace the BatchNorm3D layers with the LayerNorm layers.
3) We remove the final Tanh layer and the Reshape layer. The final Linear layer outputs the probability of being classified as a real patch.

The detailed information on the generative network is given in Fig. 3.

Our model employs 3D convolution and 3D deconvolution to better extract and recover spatial and spectral features of hyperspectral data, given its high spectral dimensionality. In particular, we set the convolution kernel and step size of the second 3D convolution layer of the discriminator to (3,1,1) and (2,1,1), respectively, to efficiently compress information in the spectral dimension and achieve fast dimensionality reduction. To avoid gradient disappearance, we use LeakyReLU as the activation function in both the discriminator and the generator. Additionally, we use batch normalization to speed up training and prevent overfitting in the generator, while in the discriminator, we use layer normalization instead as batch normalization contradicts the principle of the loss function we use [32].

The core idea of the traditional generative adversarial networks [19] is given by:

$$\min_G \max_D \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_r} \log[D(\mathbf{x})] + \mathbb{E}_{\hat{\mathbf{x}} \sim \mathbb{P}_g}[log(1 - D(\hat{\mathbf{x}}))] \quad (7)$$

where $\mathbb{P}_r$ is the real data distribution and $\mathbb{P}_g$ is the fake data distribution, which is defined by $\hat{x} = G(z), z \sim p(z)$, where z is sampled from simple noise distribution, such as the uniform distribution or a spherical Gaussian distribution. In [36], the authors utilize the $Earth - Mover$ (also called Wasserstein-1) distance $W(q, p)$, which is informally defined as the minimum cost of transporting mass in order to transform the distribution q into the distribution p. The WGAN value function is given by:

$$\min_G \max_D \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_r} D(\mathbf{x}) - \mathbb{E}_{\hat{\mathbf{x}} \sim \mathbb{P}_g} D(\hat{\mathbf{x}}) \quad (8)$$

In the WGAN model, Lipschitz constraint on the critic must be enforced. In [36], the authors propose to clip the weights of the critic to lie within a compact space $[-c, c]$, which is simple and effective to implement but also leads to optimization difficulties and makes the model fail to converge. In [32], to overcome these limitations, the authors propose $gradient - penalty$, which is applied in our model. Based on (8), [32], they add a new penalty item and the new loss function is given by:

$$\begin{aligned} Loss = \quad & \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_r} D(\mathbf{x}) - \mathbb{E}_{\hat{\mathbf{x}} \sim \mathbb{P}_g} D(\hat{\mathbf{x}}) \\ & + \lambda * \mathbb{E}_{\hat{\mathbf{x}} \sim \mathbb{P}_{\hat{\mathbf{x}}}}[(|| \bigtriangledown_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})||_2 - 1)^2] \end{aligned} \quad (9)$$

where λ is the penalty coefficient, which is equal to 10 according to [32].

In [37], the authors propose a data augmentation generative adversarial network (DAGAN), in which the data of the same class are divided into two batch sizes. When training the generator, only one of them is fed into the generator, while both are fed when training the discriminator. Adversarial training leads the network to generate new images from old ones that appear to be within the same class (whatever that class is), but look different enough to be a different sample.

We add this training strategy into our model and combine it with our band selection and mixture strategy. The detailed information on our training strategy can be seen in Fig. 4 and the training process is shown in Algorithm 3. First, the data of the same class are divided into two batch sizes ($(P(X_i)_1$ and $P(X_i)_2$). We send noise z into the generator to generate the fake data $G(Z)$. Next, one batch size ($P(X_i)_1$) and the generated data $G(Z)$ are fed into the band selection and mixture part to synthesize the mixed data $P(\hat{X})$. It is important to note that we also utilize the band selection and mixture module during the training of our GAN, which has the same components as the module in our overall FPGANDA but serves a different purpose. In the overall module, the band selection and mixture module is used to create the final hybrid data. In the GAN training process, the ultimate aim of our GAN network is to generate the remaining bands after band selection. Adding the band selection and mixture module helps to clarify the training ob-

Algorithm 3 The training process of the generative adversarial network.

```
INPUT: the random noise: $z$, the augmentation label: $i$, the training dataset: $P(X)$, training epochs: $n$.
1. Extract all the data of the category from the training dataset according to the selected labels: $P(X_i)$
2. Splitting $P(X_i)$ into two training sets of equal number: $P(X_i)_1$ and $P(X_i)_2$
3. while the training epoch $< n$:
    1. Input $z$ to the generator to generate the fake data: $G(Z)$.
    2. Input $G(Z)$ and $P(X_i)_1$ to the band selection and mixture part to synthesize the mixed data: $P(\hat{X})$.
    3. Input $G(Z)$ and $P(\hat{X})$ to the discriminator as the fake data and input $P(X_i)_1$ and $P(X_i)_2$ to the discriminator as the true data.
    4. Update the weights of the generator and the discriminator.(Eqs. (10), (11))
Return $Generator(), Discriminator()$
```

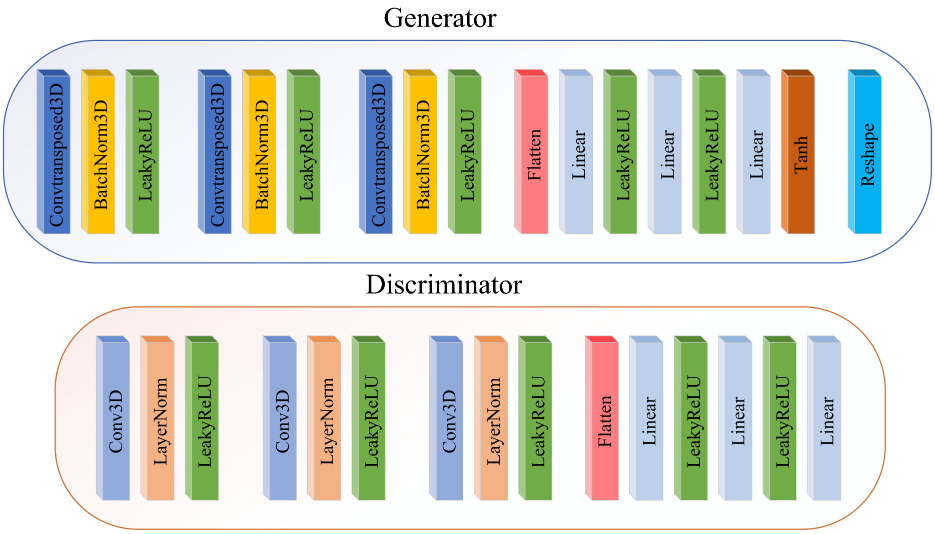


Fig. 3. Detailed information for our proposed generative adversarial network.

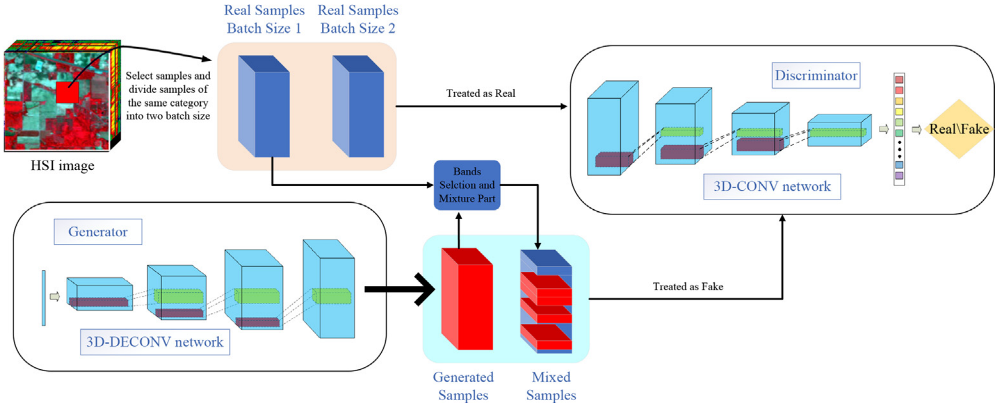

Real Samples Batch Size 1
Real Samples Batch Size 2
Select samples and divide samples of the same category into two batch size
HSI image
Treated as Real
Discriminator
Real\Fake
3D-CONV network
Bands Selection and Mixture Part
Generator
3D-DECONV network
Treated as Fake
Generated Samples
Mixed Samples

Fig. 4. Graphical overview of the training process.

jectives of the network and improve the stability and speed of the training process.

During the training, the score of each band varies as the quality of the data generated by the generator changes, which results in a potentially different band being selected each time. As the number of training iterations increases, the final selected bands are determined. The generated data $G(Z)$ and the mixed data $P(\hat{X})$ are seen as fake data distribution for the discriminator, while both real batch sizes $P(X_i)_1$ and $P(X_i)_2$ are treated as real data distribution. (see in Fig. 4) When synthesizing the mixed data, we only add $P(X_i)_1$ so that the mixed data $P(\hat{X})$ do not contain the features from $P(X_i)_2$, but all the true features are fed into the discriminator, which can lead the generative network to synthesize samples that look as different as possible from the real ones. Based on (9), our discriminator loss function is given by:

$$\begin{aligned}D\_Loss = & \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_1} D(\mathbf{x}) + \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_2} D(\mathbf{x}) \\ & - \mathbb{E}_{\hat{\mathbf{x}}\sim\mathbb{P}_g} D(\hat{\mathbf{x}}) - \mathbb{E}_{\hat{\mathbf{x}}\sim\mathbb{P}_{\hat{X}}} D(\hat{\mathbf{x}}) \\ & + \lambda * \mathbb{E}_{\hat{\mathbf{x}}\sim\mathbb{P}_{\hat{x}}}[(|| \nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})||_2 - 1)^2]\end{aligned} \tag{10}$$

where $\mathbb{P}_1$ and $\mathbb{P}_2$ represent the distribution of the two batches belonging to the uniform category split respectively. The $\mathbb{P}_g$ represents the distribution of the generated data and the $\mathbb{P}_{\hat{X}}$ represents the distribution of the mixed data. The corresponding generator loss function is given by:

$$G\_Loss = \mathbb{E}_{\hat{\mathbf{x}}\sim\mathbb{P}_g} D(\hat{\mathbf{x}}) + \mathbb{E}_{\hat{\mathbf{x}}\sim\mathbb{P}_{\hat{X}}} D(\hat{\mathbf{x}}) \tag{11}$$

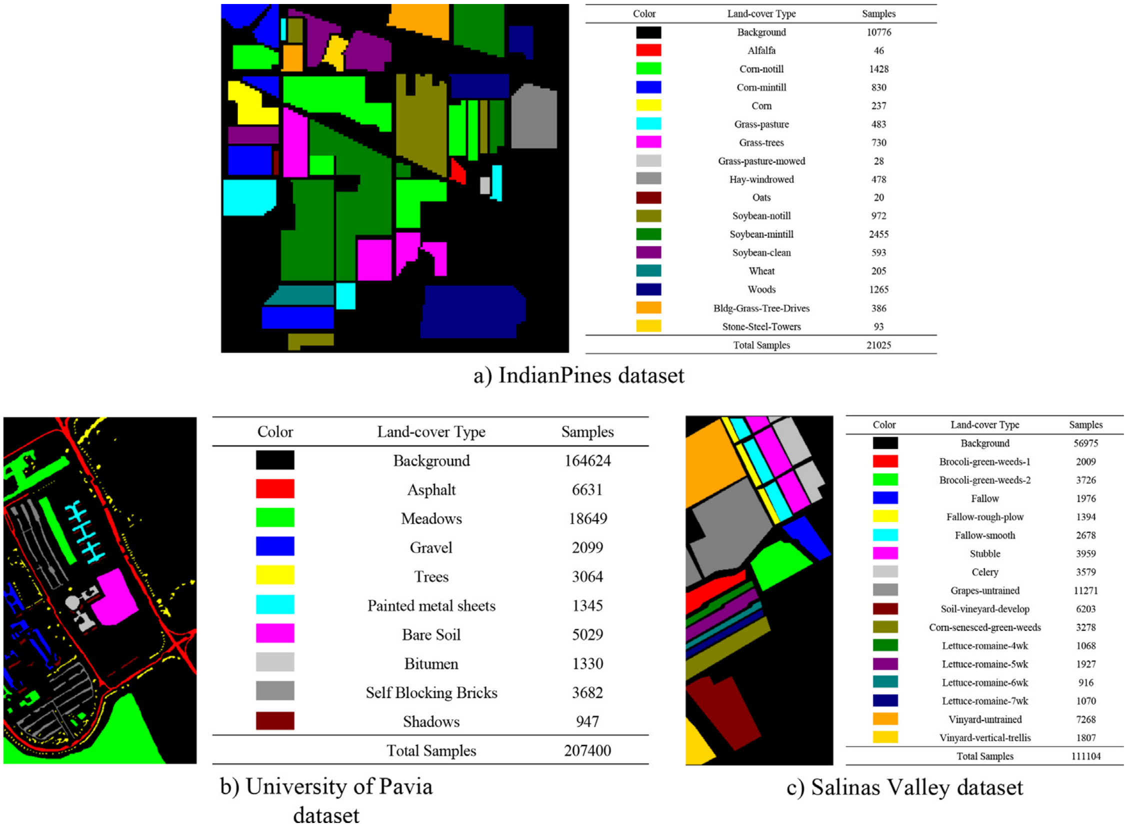
The loss function of the generator consists of two parts, one for the generator-generated distribution and the other for the mixture-generated distribution, both of which together serve as the distribution of fake data. A major advantage of adding the mixture-generated distribution is that it contains more true features of the real data and its value is usually higher than the first term, which can be seen as a regularization part to accelerate training and improve the stability of the GAN.

4. Experimental results

4.1. Datasets

The performance of our proposed model is evaluated on three open HSI datasets: Indian Pines, University of Pavia and Salinas Valley. Fig. 5 show a detailed summary of these datasets and their corresponding ground-truth information with the number of samples per class. In the following, we summarize the characteristics of each dataset.

1) The Indian Pines is an image gathered by the airborne visible/infrared imaging spectrometer (AVIRIS) in 1992 from the Agricultural and Forestry Hyperspectral. Experiment site in northwestern Indiana. This dataset is the size of 145 × 145, and there are 16 mutually exclusive land cover classes with a highly imbalanced number of samples. It has 224 reflectance bands, but 200 bands were available owing to water absorption.



Color	Land-cover Type	Samples
■	Background	10776
■	Alfalfa	46
■	Corn-notill	1428
■	Corn-mintill	830
■	Corn	237
■	Grass-pasture	483
■	Grass-trees	730
■	Grass-pasture-mowed	28
■	Hay-windrowed	478
■	Oats	20
■	Soybean-notill	972
■	Soybean-mintill	2455
■	Soybean-clean	593
■	Wheat	205
■	Woods	1265
■	Bldg-Grass-Tree-Drives	386
■	Stone-Steel-Towers	93
	Total Samples	21025

a) IndianPines dataset

Color	Land-cover Type	Samples
■	Background	164624
■	Asphalt	6631
■	Meadows	18649
■	Gravel	2099
■	Trees	3064
■	Painted metal sheets	1345
■	Bare Soil	5029
■	Bitumen	1330
■	Self Blocking Bricks	3682
■	Shadows	947
	Total Samples	207400

b) University of Pavia dataset

Color	Land-cover Type	Samples
■	Background	56975
■	Brocoli-green-weeds-1	2009
■	Brocoli-green-weeds-2	3726
■	Fallow	1976
■	Fallow-rough-plow	1394
■	Fallow-smooth	2678
■	Stubble	3959
■	Celery	3579
■	Grapes-untrained	11271
■	Soil-vineyard-develop	6203
■	Corn-senesced-green-weeds	3278
■	Lettuce-romaine-4wk	1068
■	Lettuce-romaine-5wk	1927
■	Lettuce-romaine-6wk	916
■	Lettuce-romaine-7wk	1070
■	Vinyard-untrained	7268
■	Vinyard-vertical-trellis	1807
	Total Samples	111104

c) Salinas Valley dataset

Fig. 5. Ground truth and details of the Indian Pines dataset, the University of Pavia dataset and the Salinas Valley dataset.

2) The University of Pavia dataset was collected by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor during a flight campaign over Pavia, Northern Italy. It has a size of 610 × 340 with 9 land-cover classes from urban areas. This scene contains 103 spectral bands after 12 noisy bands were discarded.
3) The Salinas Valley was captured by the AVIRIS sensor over Salinas Valley, CA, USA. The size of the data is 512 × 217 × 224. A total of 16 in different land cover classes are included in this dataset. In our experiments, 200 bands remained after the removal of water absorption bands and noisy bands.

4.2. Experimental settings

Small-sample and imbalance problems are obvious in IP and Salinas datasets. To highlight the data imbalance problem in the dataset, we created one artificially unbalanced versions of the dataset based on the existing ones, such as the unbalanced PU dataset. Table 1 shows the number of training samples (TRS), synthesizing samples (SYN) and the test samples (TES) used for the IP, Salinas, and unbalanced PU,respectively.

The performance of our model is tested with three deep learning models: fully-connected neural network (FNN) model, HamidaEtAl model [12] and LiEtAL model[13]. For each dataset, we first conducted comparative experiments using the simple deep learning network FNN as a backbone. To further demonstrate the generalizability of our model, we chose two more advanced convolutional neural networks (HamidaEtAl, LiEtAL) for hyperspectral data classification as backbones. The LiEtAL network, which comprises two 3D convolutional layers and one fully connected layer, is used as the backbone for the IP dataset and the unbalanced PU dataset. On the other hand, for the Salinas dataset, the HamidaEtAL network, which includes four 3D convolutional layers, two pooling layers, and one fully connected layer, is adopted as the backbone.

We start with two aspects of optimizer selection and hyperparameter settings for each network. Firstly, the parameter settings of each model are based on the recommended parameters of the paper [12,13]. Secondly, during our experimental tests, we observed that these parameter settings produced better results. In the FNN model which contains four linear layers, we utilize Adam optimizer whose learning rate is set to 0.0005 to optimize the parameters. The parameters of the HamidaEtAl model are optimized with the SGD optimizer, using a learning rate of 0.001 and a weights decay of 0.0005. The parameters of the LiEtAL model are optimized with the Adam optimizer, using a learning rate of 0.001 for the IP dataset and with the SGD optimizer, using a learning rate of 0.0005, a weights decay of 0.0005 and a momentum of 0.9 for the unbalanced PU dataset.

The performance of the proposed model is evaluated in terms of several widely used quantitative measurements, such as OA, AA and statistical kappa coefficient (κ) [38]. The ratio of correctly classified samples among the total test samples is determined as OA. The mean of classwise accuracy is determined as AA. κ represents a strong mutual agreement between the generated classification maps of one network model and the provided ground truth. The parameters of our generative adversarial network are optimized with the Adam optimizer, using a learning rate of 0.0002 for all the datasets. The spatial dimension of the noise vector is set to 100. To train our generative adversarial networks, each experiment was run for 20,000 iterations until the generalization ability of the generated training samples in the minority classes is stabilized. In the comparison experiments, in addition to the traditional data amplification methods, we compare two traditional generative network models: WGAN [36] and DCGAN [39], and three modified generative network model for hyperspectral characteristics: HSGAN [24],

Table 1
Detailed number of training samples (TRS), synthesizing samples (SYN) and test samples (TES).

Class	IndianPines			Unbalanced University of Pavia			Salinas Valley		
	TRS	SYN	TES	TRS	SYN	TES	TRS	SYN	TES
1	4	4	42	265	0	6365	20	0	1989
2	142	0	1286	745	0	17,903	37	0	3689
3	83	83	747	10	10	2089	19	0	1957
4	23	0	214	122	0	2941	13	0	1381
5	48	0	435	7	7	1338	26	0	2652
6	73	0	657	25	25	5004	39	0	3920
7	2	2	26	7	7	1323	35	0	3544
8	47	0	431	147	0	3534	112	112	11,159
9	2	2	19	37	0	910	62	0	6141
10	97	0	875				32	32	3246
11	245	0	2210				10	0	1058
12	59	0	534				19	0	1908
13	20	0	185				9	0	907
14	126	0	1139				10	0	1060
15	38	38	348				72	72	7196
16	9	0	84				18	18	1789

3DGAMO [30] and AC-WGAN-GP [31] with our proposed model. To ensure fairness in the comparison, for each of the above generative models, we keep the inputs of the same dimension, the optimizers the same, the learning rates the same, and the number of training iterations the same. The only difference is the architecture of the generative model.

In our experiments, a spatial window of size $7 \times 7 \times B$ is extracted from the original dataset, where B is related to the number of bands used for each dataset. If the number of augmentations is much larger than the number of small samples, it will make it difficult for the network to focus on the features of the real data, while if the number of augmentations is too small, it will not be able to fully utilize the feature representation of the available data. Therefore, to take full use of the features of the small sample datasets, we aligned the number of augmentation with the number of small samples. For each sample, its features are retained and the rest is replaced with the generated data. The number of samples augmented was kept consistent when we used a comparison of other data augmentation methods. The synthesized labels are chosen based on the baseline accuracy, which varies depending on the classification network and dataset. Some samples are facing small-sample problems but are very easily distinguishable, and a high degree of accuracy can already be achieved without augmentation. In this case, it is more effective to target the augmentation to the sample that is needed.

4.3. 4.3 Classification results

Fig. 5 shows 16 land-cover classes and the number of associated samples per class for the IP dataset. Table 1 shows the specific number of training samples and the number of test samples. The class-specific samples are highly imbalanced, and the lowest and highest numbers of training samples are observed for the seventh and eleventh classes of the IP dataset (2 and 245, respectively). Table 2 tabulates the classification performance of all the compared methods, including the OA, AA, and kappa coefficients, in addition to the class-specific accuracies for all the classes of the IP dataset. The 16 rows in Table 2 represent the accuracies for each land-cover class, whereas the last three rows represent the OA, AA, and kappa values for all the compared methods and all the experimental models. The highest classification results are represented in bold across all the compared methods. The synthesized labels are chosen based on the Baseline accuracy. In the IndianPines dataset for the LiEtAL model, we choose 1,3,7,9,15 labels to augment which can be observed in Table 1, whose baseline accuracies are 58.5%, 76.8%, 64.0% 38.9% and 69.2%, which are much lower than the AA (82.638%). The numbers of these four class samples are doubled through the proposed method. The numbers of each class we synthesize, according to Table 1, are 4, 83, 2, 2 and 38. It can be observed that the proposed method achieves the best performance than other competitors as shown in Table 2. The proposed model achieves the highest OA (91.122%) and kappa coefficient (89.9), while the Flip achieves the best AA (90.731%) performance in the IndianPines dataset for the LiEtAL model. It is worth noting that WGAN and AC-WGAN-GP both outperform the other GANs, whereas the WGAN achieves the best OA (90.179%), AA (88.196%) and kappa (88.8) among these GANs. To the best of our knowledge, this is probably due to the power of the $Earth - Mover$ distance, which is used in these two models. The $Earth - Mover$ distance is proved to be more stable when the training samples are very small. Among the traditional data augmentation strategies, Flip and Radiation, the best performance achieved are 90.493% (OA), 90.731% (AA) and 89.2 (kappa). It is worth noting that traditional data augmentation strategies can sometimes achieve decent accuracy, for example, the Flip strategy has achieved the highest AA accuracy in the IP scene, which has inspired us to apply certain traditional data augmentation ideas to our model, which is also responsible for the higher accuracy our model can achieve.

To determine the generalization power of the considered HSI data augmentation strategy, the artificially imbalanced PU train-test set is also adopted and the detailed information is shown in Fig. 5. Table 1 shows the types of samples selected for augmentation and their numbers during the comparison experiments. The Painted Metal Sheets class and the Bitumen class have the fewest samples, respectively, whereas the Meadows class has the most samples. Data imbalance problems and small sample problems are significant. The OA, AA, and kappa coefficients are shown in Table 4 along with the class-specific accuracies for all the land cover classes of the unbalanced PU dataset. Tables 4 and 5 show the superior performance of the proposed strategy in terms of OA (89.618%) and kappa (85.8) in the LiEtAL model and terms of OA (83.473%), AA(79.000%) and kappa (77.9) in FNN model. It is worth noting that the best AA (86.178%) accuracy is still achieved by the Flip strategy. The proposed strategy achieves significant performance improvement of at least 1.77%, 1.57% and 1.90% in terms of OA, AA and kappa compared with the baseline reported in Table 4. It can be observed that in both the LiEtAL model backbone and the FNN model backbone traditional data augmentation strategies outperform GAN-based data augmentation strategies overall. This phenomenon can be related to the dataset, as in the PU image whose pixels can be easily classified owing to its comparatively

Table 2
Classification results obtained by LiEtAL model using 10% randomly selected training samples from the IP dataset.

Class	LiEtAL model								
	Baseline	Flip	Radiation	WGAN	DCGAN	HSGAN	3DGAMO	AC-WGAN-GP	Proposed
1	58.5	**85.4**	68.3	53.7	39.0	34.1	39.0	65.9	61.0
2	89.3	**95.0**	93.4	91.9	89.7	86.5	90.9	90.0	92.6
3	76.8	**87.8**	80.7	80.5	72.3	76.3	77.2	77.9	82.1
4	**90.1**	82.2	86.9	78.4	73.7	82.2	75.1	77.0	85.4
5	86.4	79.5	80.9	85.3	84.1	84.1	84.8	88.5	**92.2**
6	**99.8**	99.1	98.3	99.2	98.9	97.3	99.4	98.5	98.3
7	64.0	100	92.0	100	44.0	48.0	44.0	88.0	**100**
8	99.8	100	100	100	100	100	100	100	**100**
9	38.9	**100**	88.9	100	88.9	100	66.7	83.3	27.8
10	83.9	88.9	**95.0**	83.8	79.8	84.6	82.4	83.5	86.4
11	91.0	88.6	87.7	**93.1**	91.8	88.0	92.8	92.8	91.4
12	86.5	85.4	93.4	86.0	81.8	82.8	85.4	84.5	**91.4**
13	97.8	97.8	**99.5**	95.7	88.6	82.7	83.2	88.6	94.1
14	97.3	94.0	94.6	96.9	96.7	94.9	96.0	97.4	**97.5**
15	69.2	**78.7**	66.9	69.2	61.1	70.9	67.4	69.5	75.5
16	92.9	89.3	86.9	**97.6**	88.1	90.5	89.3	90.5	94.0
OA	89.279	90.493	89.255	90.179	87.295	86.905	88.683	89.442	**91.122**
AA	82.638	**90.731**	88.337	88.196	79.912	81.422	79.611	85.989	85.606
K(×100)	87.8	89.2	87.8	88.8	85.2	85.1	87.1	88.0	**89.9**

Table 3
Classification results obtained by FNN model using 10% randomly selected training samples from the IP dataset.

Class	FNN model								
	Baseline	Flip	Radiation	WGAN	DCGAN	HSGAN	3DGAMO	AC-WGAN-GP	Proposed
1	43.9	61.0	**75.6**	43.9	34.1	24.4	56.1	56.1	65.9
2	84.8	81.4	72.5	72.1	82.6	82.3	**86.1**	85.6	85.7
3	67.2	61.3	55.4	62.7	64.1	64.0	64.0	63.6	**69.3**
4	62.9	**80.3**	74.2	61.5	66.2	53.1	55.9	58.2	62.4
5	85.3	85.1	74.0	75.9	83.0	85.3	86.0	**87.6**	84.6
6	98.3	97.3	89.2	95.3	97.9	97.0	98.2	98.5	**98.6**
7	**84.0**	72.0	36.0	56.0	72.0	64.0	72.0	60.0	80.0
8	99.8	97.9	92.1	98.8	99.3	97.7	99.8	98.8	**99.8**
9	50.0	**66.7**	55.6	50.0	44.4	38.9	50.0	50.0	55.6
10	80.1	**84.3**	81.1	83.2	76.6	74.1	79.8	79.2	79.1
11	89.6	**92.9**	88.1	85.1	90.2	88.1	91.0	90.7	90.8
12	71.3	67.4	39.9	53.4	64.8	63.5	**77.2**	76.4	73.8
13	84.3	83.2	77.3	78.9	83.8	83.8	83.8	84.3	**84.9**
14	94.1	91.9	94.6	94.5	94.6	94.6	94.3	96.5	**94.8**
15	60.8	62.0	61.4	58.2	59.1	**62.0**	59.7	54.8	59.1
16	92.9	96.4	94.0	79.8	97.6	97.6	96.4	94.0	**97.6**
OA	84.542	84.650	78.493	79.480	83.263	82.092	84.997	84.856	**85.312**
AA	78.086	80.068	72.562	71.824	75.642	73.134	78.134	77.144	**80.125**
K(×100)	82.4	82.5	75.4	76.6	80.9	79.6	82.9	82.7	**83.2**

pure spectral signatures that result from the high spatial resolution and the abundance of spatial information. Therefore, in this case, traditional data augmentation can better improve the accuracy performance. It also shows that the linear characteristics of the original data are important for the synthesis of new data. Our proposed method of data synthesis that retains some of the original features of the data is therefore one of the reasons why our strategy can achieve higher accuracy. Although not performing as well as classical data augmentation methods on PU datasets, overall, the GAN-based augmentation strategy on minority classes still can improve the accuracy of the classification network by adding more non-linear relevant features.

In the last part of the experiment, we tested our model on the Salinas dataset. Fig. 5 shows the description of the land-cover classes and the number of available samples per class. To validate the performance of the proposed strategy, the Salinas dataset is randomly divided into training and test sets, where 1% of the available samples are utilized for training and the remaining 99% are used for testing. Table 1 shows the number of synthesized samples for the Salinas dataset, which selects poorly classified categories in the baseline. Table 6 reports the achieved classwise accuracy and the statistical analysis of parameters: OA, AA, and kappa for all the compared methods. The proposed strategy achieves the highest classification performances of OA (90.696%, 88.891%), AA (92.912%, 91.173%), and kappa (89.7, 87.7) in both backbones. On the Salinas dataset, AC-WGAN-GP can achieve the second highest accuracy on both FNN and Hamida backbone networks, which can achieve an OA of 90.046% and 88.482%, an AA of 92.649% and 90.696% and a kappa of 88.9 and 87.2 during the strategy evaluation. This result reaffirms the efficacy of $Earth - Mover$ distances and architectural networks processed for hyperspectral data in data augmentation strategies. Overall, the proposed strategy achieves an average performance gain of at least 2.06% in OA, 1.06% in AA and 2.30% in kappa compared to the baseline listed in Table 6.

Finally, we combine the classification results on the three datasets. Focusing on the hyperspectral-processed GAN such as HSGAN, 3DGAMO and AC-WGAN-GP, these models usually reach higher accuracy results than traditional GAN such as WGAN and DCGAN in some datasets, such as PU and Salinas scenes. Due to the rich spectral information in hyperspectral data, a simple fully-connected architecture or a 2D convolutional architecture can not capture the feature information of the data well, and a 3D convo-

Table 4
Classification results obtained by LiEtAL model and the selection ratio of the unbalanced PU dataset is set to 0.005 for data labels 3,5,6,7 and 0.04 for the rest.

Class	LiEtAL model								
	Baseline	Flip	Radiation	WGAN	DCGAN	HSGAN	3DGAMO	AC-WGAN-GP	Proposed
1	94.9	91.0	94.5	94.4	**95.1**	94.2	92.8	93.6	94.1
2	92.4	92.3	92.4	92.2	92.2	92.4	92.2	**92.4**	91.7
3	60.4	47.6	60.0	60.7	45.3	**62.5**	61.1	60.9	58.9
4	93.9	94.0	93.7	93.2	94.5	93.1	94.0	93.6	**94.9**
5	93.0	**99.2**	93.3	94.3	97.8	86.9	97.1	92.6	90.1
6	66.6	79.2	72.4	70.2	71.0	62.5	67.8	64.0	**80.9**
7	59.5	**74.1**	63.4	63.4	30.9	58.1	56.0	61.5	67.1
8	97.2	**99.0**	97.6	97.1	96.2	97.0	97.5	97.4	96.7
9	99.2	99.2	99.0	98.0	98.3	97.9	**99.2**	97.4	96.8
OA	87.844	88.849	88.599	88.216	86.748	87.028	87.690	87.371	**89.618**
AA	84.122	**86.178**	85.167	84.833	80.146	82.737	84.207	83.710	85.687
K(×100)	83.9	85.4	85.0	84.4	82.5	82.8	83.7	83.3	**85.8**

Table 5
Classification results obtained by FNN model and the selection ratio of the unbalanced PU dataset is set to 0.005 for data labels 3,5,6,7 and 0.04 for the rest.

Class	FNN model								
	Baseline	Flip	Radiation	WGAN	DCGAN	HSGAN	3DGAMO	AC-WGAN-GP	Proposed
1	91.2	89.0	93.2	**93.4**	92.3	93.2	92.7	92.3	92.2
2	92.0	89.0	91.2	92.1	92.0	92.1	92.1	92.1	**92.1**
3	42.0	39.2	39.7	**44.5**	44.4	42.6	39.4	40.7	43.2
4	94.9	94.5	**95.5**	93.4	95.1	94.9	94.2	94.7	94.7
5	69.2	**99.3**	75.5	55.4	84.8	88.8	83.4	97.9	91.7
6	41.9	36.0	**54.9**	43.8	39.0	37.7	41.2	42.4	43.7
7	49.6	**66.2**	27.1	52.9	53.0	49.5	50.4	53.9	62.7
8	**93.0**	89.9	93.2	92.2	91.0	90.8	92.6	90.1	92.6
9	89.5	92.4	96.8	97.5	97.6	97.7	95.6	95.7	**98.1**
OA	81.727	80.480	82.856	82.109	82.316	82.226	82.309	82.869	**83.473**
AA	73.700	77.278	74.122	73.911	76.586	76.373	75.734	77.760	**79.000**
K(×100)	75.6	73.9	77.2	76.1	76.3	76.1	76.3	77.1	**77.9**

Table 6
Classification results obtained by HamidaEtAl model using 1% randomly selected training samples from the Salinas dataset.

Class	HamidaEtAL model								
	Baseline	Flip	Radiation	WGAN	DCGAN	HSGAN	3DGAMO	AC-WGAN-GP	Proposed
1	**95.3**	95.0	95.1	94.8	94.6	95.0	95.1	94.8	93.3
2	**99.8**	97.8	97.6	97.3	97.5	97.9	99.6	99.3	99.0
3	97.2	99.2	98.6	**99.9**	99.6	96.9	96.0	99.6	98.5
4	77.5	95.2	83.5	87.8	86.2	94.4	90.3	92.9	**95.5**
5	95.8	93.5	98.0	96.6	98.3	95.7	93.6	95.8	**98.4**
6	95.9	93.4	**98.4**	93.8	94.5	93.8	97.7	96.7	97.6
7	97.0	98.5	**98.6**	94.9	95.0	96.6	96.6	97.0	97.7
8	82.6	91.2	91.3	81.5	84.4	83.0	**91.7**	86.8	83.9
9	99.5	99.1	98.9	99.3	98.8	98.2	99.5	99.4	**99.5**
10	87.5	90.6	**91.3**	90.3	88.4	89.5	89.3	89.9	91.0
11	92.0	93.0	89.8	92.2	89.2	87.0	90.7	**94.1**	93.9
12	97.2	97.2	97.2	97.2	97.2	97.2	97.2	97.2	**97.2**
13	96.4	90.1	**97.6**	94.6	85.6	85.1	95.0	96.0	96.8
14	95.0	89.3	95.3	96.4	94.8	95.1	95.3	**96.9**	94.9
15	64.4	57.3	53.6	67.5	64.3	65.8	46.5	67.5	**74.5**
16	81.6	79.9	81.1	81.3	**81.6**	78.2	81.1	78.5	80.1
OA	88.163	89.076	89.098	88.430	88.264	88.113	87.973	90.046	**90.696**
AA	90.904	91.270	91.607	91.591	90.609	90.588	90.947	92.649	**92.912**
K(×100)	86.8	87.8	87.8	87.1	86.9	86.8	86.6	88.9	**89.7**

lutional or 3D deconvolutional network designed for hyperspectral data can better approximate the true distribution of the data. It can be noticed that the traditional strategies can not produce better results when the hyperspectral data is complex, such as the Salinas scene, where the spatial information is restricted to the boundaries of the various fields and the spectral signatures are heavily jumbled. There is another point worth noting that traditional data augmentation strategies tend to underperform most GAN-based data augmentation strategies when the baseline classification network of our application is less robust and prone to overfitting problems, such as in the FNN model of the PU scene the AA (80.408%) accuracy achieved through the Flip strategy is lower than the baseline (81.727%). We believe that the reason for this may be that the traditional augmentation strategies, such as Flip, do not introduce sufficient new feature information, which makes it difficult to further improve accuracy in some networks where the classification network itself is poor at feature extraction. In these cases, more elaborate data augmentation strategies should be conducted. The proposed method can outperform these strategies in this context. Furthermore, it is noteworthy that the Radiation strategy sometimes shows a negative impact compared to the benchmark in Tables 3 and 7. This phenomenon may arise due to the weak fea-

Table 7
Classification results obtained by FNN model using 1% randomly selected training samples from the Salinas dataset.

Class	FNN model								
	Baseline	Flip	Radiation	WGAN	DCGAN	HSGAN	3DGAMO	AC-WGAN-GP	Proposed
1	93.4	93.1	93.3	93.9	93.1	92.9	93.7	89.2	**94.7**
2	94.6	92.6	95.9	71.9	94.5	95.9	94.4	**96.0**	95.9
3	89.6	38.8	08.3	86.1	58.2	95.3	76.8	90.4	**94.7**
4	86.6	79.3	39.9	86.4	94.5	59.2	92.2	94.9	**97.0**
5	98.9	**99.8**	96.8	94.5	98.2	98.2	98.2	98.3	98.6
6	96.5	94.0	96.2	**96.5**	93.1	94.2	95.3	94.2	94.4
7	96.5	07.5	91.2	**98.3**	95.7	92.0	94.0	93.3	95.1
8	64.6	**93.2**	86.1	76.2	87.8	88.4	73.4	86.5	82.4
9	99.3	**99.5**	97.9	99.2	99.0	99.4	98.9	99.5	98.8
10	91.6	78.4	89.7	**94.1**	91.1	87.1	88.0	89.5	89.7
11	88.8	84.4	00.1	70.3	75.7	89.1	87.6	**91.7**	87.6
12	**96.6**	79.8	95.9	96.4	90.7	96.2	95.2	95.9	90.7
13	95.0	87.4	77.4	**97.1**	92.8	91.5	96.7	88.4	90.1
14	96.5	96.3	29.9	96.1	94.8	87.7	94.0	**96.7**	96.5
15	**87.0**	46.7	55.5	74.4	65.6	67.4	81.4	64.9	72.8
16	81.2	79.6	69.3	81.7	81.7	75.9	77.9	**81.7**	79.7
OA	87.296	77.973	78.917	86.036	87.374	88.007	87.294	88.482	**88.891**
AA	91.052	78.149	70.211	88.315	87.912	88.162	89.860	90.696	**91.173**
K(×100)	86.0	75.4	76.4	84.5	86.0	86.6	85.9	87.2	**87.7**

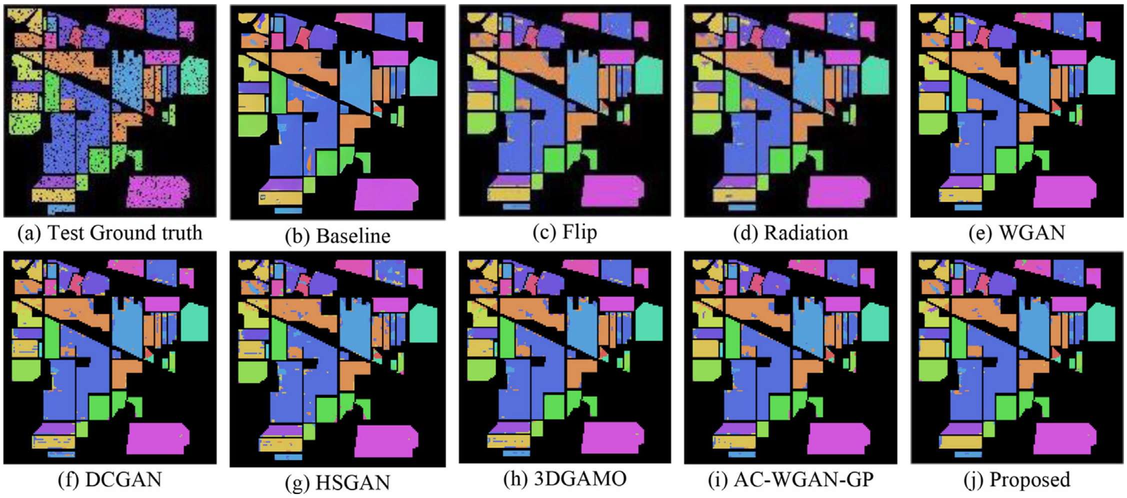

(a) Test Ground truth (b) Baseline (c) Flip (d) Radiation (e) WGAN
(f) DCGAN (g) HSGAN (h) 3DGAMO (i) AC-WGAN-GP (j) Proposed

Fig. 6. (a) Ground truth and classification maps obtained from IndianPines dataset through LiEtAL model by (b) Flip, (c) Radiation, (d) WGAN, (e) DCGAN, (f) HSGAN, (g) 3DGAMO, (h) AC-WGAN-GP and (i) Proposed method, using 10% randomly selected training samples.

ture extraction capability and low baseline accuracy of the FNN used as the baseline network. The Radiation strategy introduces excessive noise, hindering the model from learning the distinctive features of the samples, resulting in reduced accuracy. This suggests that the Radiation strategy may not be suitable for datasets with weak baseline models, and our proposed method with feature band retention and fusion strategies may be more effective in such cases.

4.4. Discussion of the obtained classification maps

We also test the robustness of our proposed strategy through an evaluation of the generated classification maps. Figs. 6–8 show the classification maps obtained by the nine compared methods on the IP, PU and Salinas datasets through the LiEtAL model and HamidaEtAL model. As shown in Fig. 6, the maps for Baseline, Radiation, DCGAN, HSGAN and 3DGAMO contain salt and pepper noise in the classification results due to the misclassification of many points. It is visible in the land-cover classes, such as Soybean-mintill and Soybean-clean in the generated map of the IP dataset. Compared to the previous methods, Flip, WGAN, and AC-WGAN-GP obtain classification maps with higher quality, whereas the classification maps are significantly improved by using the proposed strategy. Additionally, it also helps to enhance the consistency of the land-cover regions, as shown in Fig. 6(c), (e), (i), (g). It can be observed that the greatest variation between methods in the Bare Soil category can be seen in baseline, DCGAN and HSGAN, where a large number of misclassifications can be seen in the central and edge regions of the category, which is shown in Fig. 7(b), (f), (g). In our proposed method, the categories can be distinguished more clearly and with fewer misclassifications than in other methods. The specific information of classification maps of the Salinas dataset can be seen in Fig. 8. Compared to other methods, our method continues to improve firstly in the determination of feature edge categories such as Corn-senesced-green-weeds, and secondly in the classification of Vinyard-untrained categories, which is also relatively more accurate, as shown in Fig. 8. In summary, our proposed strategy can help generate higher-quality classification maps.

4.5. Ablation study

In this subsection, we perform a component-wise ablation analysis to understand the contribution of the major components of our strategy and for the selection of the hyperparameter mixing

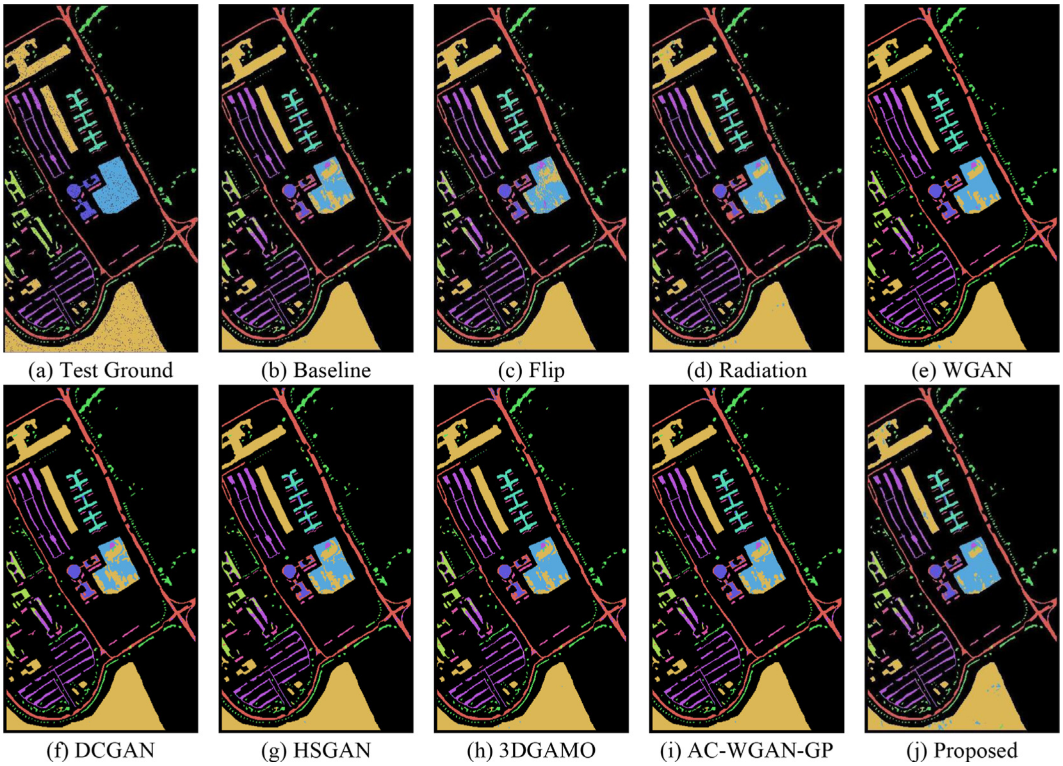

(a) Test Ground
(b) Baseline
(c) Flip
(d) Radiation
(e) WGAN
(f) DCGAN
(g) HSGAN
(h) 3DGAMO
(i) AC-WGAN-GP
(j) Proposed

Fig. 7. (a) Ground truth and classification maps obtained from PaviaU dataset through LiEtAL model by (b) Flip, (c) Radiation, (d) WGAN, (e) DCGAN, (f) HSGAN, (g) 3DGAMO, (h) AC-WGAN-GP and (i) Proposed method, using the fixed training set.

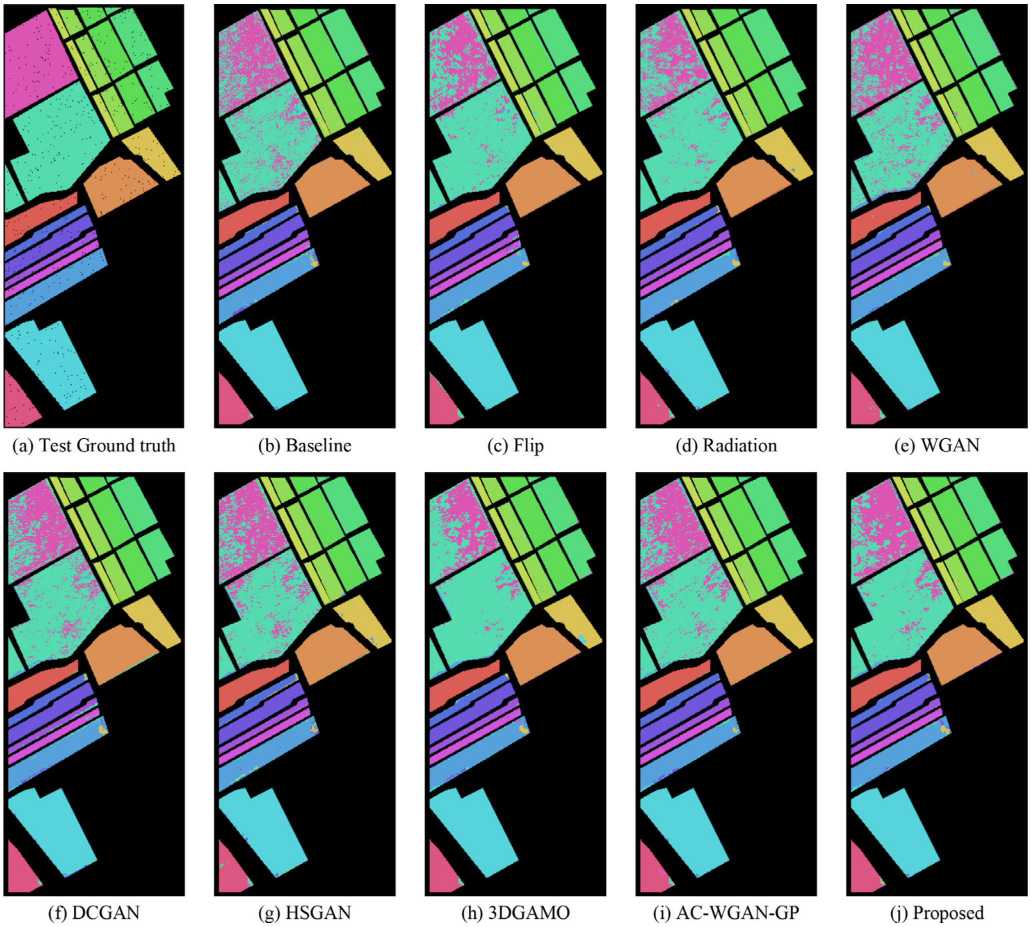

(a) Test Ground truth
(b) Baseline
(c) Flip
(d) Radiation
(e) WGAN
(f) DCGAN
(g) HSGAN
(h) 3DGAMO
(i) AC-WGAN-GP
(j) Proposed

Fig. 8. (a) Ground truth and classification maps obtained from Salinas dataset through HamidaEtAl model by (b) Flip, (c) Radiation, (d) WGAN, (e) DCGAN, (f) HSGAN, (g) 3DGAMO, (h) AC-WGAN-GP and (i) Proposed method, using 1% randomly selected training samples.

Table 8
Classification results obtained by LiEtAL model, using different mixing percent (0.1 intervals from 0 to 1) and using 10% randomly selected training samples from the IP dataset.

Class	Baseline	Mixing percent										
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
1	58.5	78.0	41.5	43.9	43.9	78.9	36.6	61.0	48.8	41.5	39.0	41.5
2	89.3	90.6	90.1	91.1	90.4	91.6	92.9	92.6	91.8	90.4	88.6	90.6
3	76.8	80.1	82.2	80.5	81.9	82.4	79.4	82.1	78.0	81.3	77.6	80.2
4	90.1	76.1	71.4	78.9	74.2	81.0	77.5	85.4	85.0	77.5	75.6	76.1
5	86.4	83.7	85.7	84.1	88.7	91.2	86.2	92.2	85.7	84.8	84.1	83.9
6	99.8	99.2	99.1	99.4	99.2	99.0	99.7	98.3	98.6	99.2	98.8	99.2
7	64.0	100	92.0	56.0	76.0	71.8	96.0	100	100	80.0	84.0	76.0
8	99.8	100	100	100	100	97.6	100	100	100	100	100	100
9	38.9	77.8	94.4	61.1	100	83.7	100	27.8	100	61.1	88.9	55.6
10	83.9	82.4	83.4	83.0	83.4	87.2	83.7	86.4	85.0	82.5	81.1	82.9
11	91.0	92.9	92.2	93.3	93.2	91.1	93.8	91.4	93.3	93.2	92.3	93.4
12	86.5	86.5	90.8	87.5	87.5	88.1	84.1	91.4	86.3	86.3	84.3	85.8
13	97.8	93.5	89.2	97.3	92.4	93.8	98.9	94.1	95.7	91.9	83.8	85.9
14	97.3	97.5	97.9	97.9	98.2	97.7	97.6	97.5	97.6	97.5	97.4	97.5
15	69.2	66.3	68.9	65.7	68.6	78.0	74.6	75.5	74.6	66.0	68.0	67.1
16	92.9	77.4	98.8	96.4	96.4	98.2	94.0	94.0	94.0	88.1	96.4	95.2
OA	89.279	89.496	89.864	89.875	90.168	89.843	90.569	91.122	90.537	89.528	88.423	89.398
AA	82.638	86.373	86.103	82.251	85.874	85.943	87.187	85.606	88.416	82.583	83.743	81.925
K(×100)	87.8	88.0	88.5	88.5	88.8	88.4	89.2	89.9	89.2	88.1	86.8	87.9

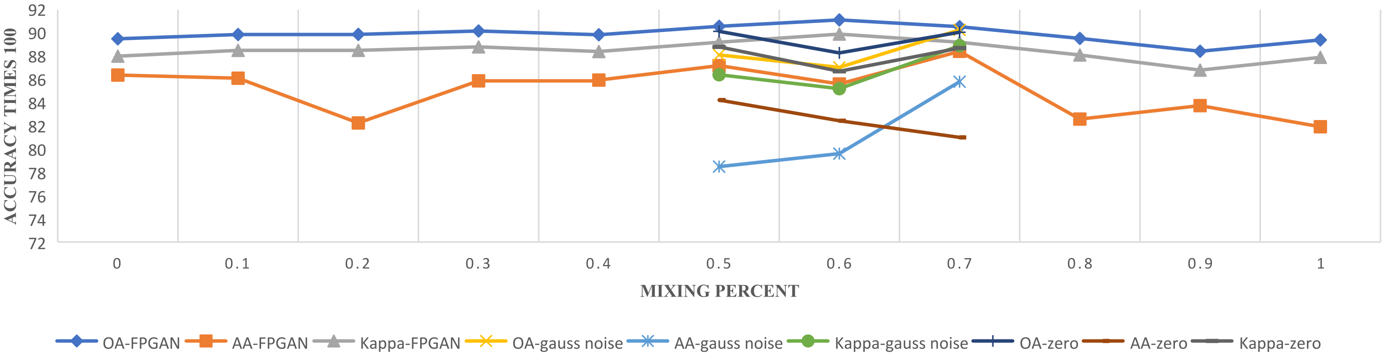


Fig. 9. Trend graph from IP dataset with AA, OA and Kappa indicators, with data mixed at 0.1 intervals from 0 to 1 and two remaining band substitution comparisons for Gaussian noise and null band (0).

ratio in our network, we conduct an experimental and analytical discussion. In the ablation experiment, we use the IP dataset with significant imbalance problems and small sample problems, using the LiEtAL network as the backbone, which has the same specific parameter settings as set in the previous section, with the only change being the mixing ratio, which is spaced from 0 to 1 at 0.1 intervals. The specific classification information is shown in Table 8, which contains the classification accuracy for each category and the three overall evaluation indicators of OA, AA and Kappa coefficient. The trend graph for the three overall evaluation indicators is shown in Fig. 9. When the mixing ratio is 0, it means that the original data features are not preserved and all the data generated by the generator is used, without using the band selection and mixing module of our strategy, and vice versa.

As can be seen in Table 8, when the mixing ratio is 0, there is almost no improvement in the accuracy of the OA (89.496%) and Kappa coefficient (88.0) compared to Baseline (OA: 89.279%, Kappa: 87.8), while AA has an improvement of 3.74%. This is probably due to the difficulty of training the GAN itself to converge when the samples are too small, resulting in the introduction of too many non-linear features in the generated samples, which misleads the classifier when it performs feature extraction for classification. However, the introduction of nonlinear features is not useless, which can improve the accuracy of AA. It can be observed that when the mixing ratio is 1, on top of no improvement in the OA and kappa coefficients, AA even shows a negative gain (0.713%). To the best of our knowledge, this may be due to the overfitting problem caused by retaining the linear features of the original data completely, not introducing new features and simply replicating to increase the number of small samples. When gradually increasing the mixing ratio, the accuracies of OA, AA and Kappa coefficients appear to rise and then fall. We can observe from Fig. 9 that satisfactory accuracy can be achieved when the cross ratios are around 0.5 and 0.7. Therefore, we recommend setting the mixing ratio around 0.6, where the linear features of the original data and the non-linear features introduced by the generator can be balanced.

Furthermore, to provide additional evidence of the superiority of our proposed model, we conducted a comparison experiment where we replaced the remaining bands after feature retention with Gaussian noise and empty band (0), respectively. The scale of the comparison experiment was set within the range where our model performed best, which was 0.5-0.7 for the crossover scale, and we conducted the comparison experiments at intervals of 0.1. The evaluation metrics including AA, OA, and Kappa coefficients were recorded separately for each method. The detailed information is shown in Table 9. It is worth noting that our proposed method consistently provides a stable performance improvement compared to the benchmark method, while the other two methods, replacement with Gaussian noise and empty band, result in negative performance improvements in most cases. However, in some cases, these methods do show slight improvements in the classification results. Based on our understanding, this phenomenon can be attributed to the effectiveness of our feature band selection strategy. When we perform band selection, we take into account

Table 9
Classification results for remaining band substitution comparison for GAN generated, Gaussian noise and null band (0) respectively .

Evaluation indicators	Baseline	FPGAN generated			Gaussian noise			null band (0)		
		0.5	0.6	0.7	0.5	0.6	0.7	0.5	0.6	0.7
OA	89.279	90.596	91.122	90.537	88.108	87.035	90.266	90.136	88.282	90.060
AA	82.638	87.187	85.606	88.416	78.490	79.610	85.804	84.211	82.441	81.009
K($\times$100)	87.8	89.2	89.9	89.2	86.4	85.2	88.9	88.8	86.7	88.7

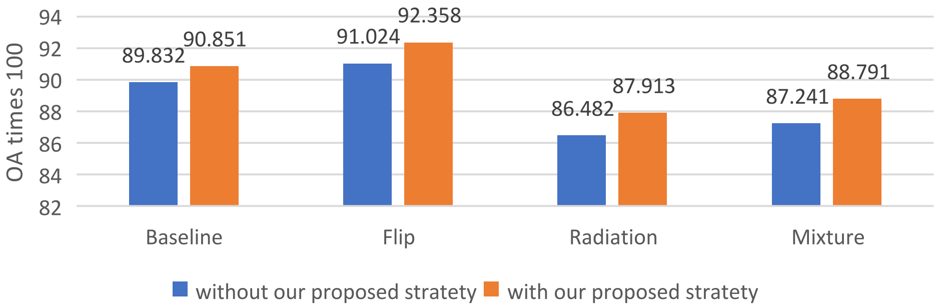


Fig. 10. Bar charts from the IP dataset, using the HeEtAl model, containing the four base cases and the four base cases after joining our method.

the similarity between the original and generated bands. As a result, important bands that are difficult to generate or fit and have high band scores are more likely to be retained. These retained feature bands can be particularly useful in the subsequent classification task, leading to a stable boost in performance compared to the benchmark method.

Finally, we verified the compatibility of our strategy. We simultaneously utilize two data augmentation strategies. One is a traditional data augmentation strategy and the other is our proposed strategy. The experiment is conducted in the IP dataset, using the HeEtAl model [14] to further confirm the efficacy of our strategy. The training ratio is equal to 10% and we utilize the Adagrad optimizer with a learning rate of 0.01. The hyperparameters of our generative adversarial network and the mixing ratio are not changed. The detailed result can be seen in Fig. 10. It can be observed adding our strategy has a significant improvement in the classification results. Using both Flip and our proposed strategy can achieve the highest OA of 92.358%. Although when utilizing the Mixture strategy and Radiation strategy the accuracy is lower than the baseline, after adding our proposed strategy the accuracy can still be improved by about 1.43% and 1.55%. The results show our proposed strategy does not conflict with other data augmentation strategies.

5. Conclusion

In this study, we proposed a new ‘FPGANDA’ framework for a data augmentation strategy to alleviate the small sample problem and data imbalance problem encountered in hyperspectral data classification. It consists of a 3D-generator network and a 3D discriminator, where we use 3D convolution and 3D deconvolution to better capture the feature information of hyperspectral data and to better approximate the true distribution of hyperspectral data. It also contains a band selection strategy for balancing the generation quality of the generated data with the original characteristics of the original data. A cross-synthesis strategy is devised, whereby the bands with the most feature information are selected and retained, and the other bands are replaced with the bands generated by the generator. In this way, we generate new samples that contain some linear features similar to those in traditional data augmentation strategies, while introducing some non-linear features through GAN-based data augmentation strategies. We conducted extensive experiments in which we used three classification networks as backbones on three hyperspectral datasets against eight methods, containing two traditional data augmentation methods, two traditional GAN-based data augmentation methods and three hyperspectral-processed GAN-based data augmentation methods. The proposed strategy outperforms existing strategies in most of these cases and can improve the quality of the classification map. We also perform a component-wise ablation study and make an experiment to prove that our proposed strategy does not conflict with others. Thus, from an experimental viewpoint, we can conclude that the proposed ‘FPGANDA’ strategy is more effective with small training sets with high imbalance, which is the most common scenario in real remote sensing applications.

However, our model still has some weaknesses. It requires significant time and computing power to train the generative model for effective performance. Additionally, retraining the model is necessary when encountering new data in applications. While our model is designed for hyperspectral image classification, this data augmentation method can be adapted for other remote sensing image processing tasks, such as super-resolution, denoising, and change detection. The migration of this method to such tasks can be attempted in scenarios where data imbalance and shortage are present. In future work, we plan to investigate the feasibility of applying this strategy to other task scenarios. Moreover, we will further explore the potential of the generative model and assess its effectiveness in other tasks.

Several questions still remain to be answered. Firstly, further works could be done to explore a better architecture for generative networks. Secondly, further investigations and experimentations into the combination of the traditional data augmentation strategy and the GAN-based data augmentation are strongly recommended. Finally, further studies need to be carried out in order to validate the application of the proposed strategy to other data types such as multispectral images, SAR images.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

The authors wish to thank the editors and anonymous reviewers for their valuable comments and helpful suggestions which greatly improved the paper's quality. This work was supported in part by the Key-Area Research and Development Program of Guangdong Province (Grant 2020B090921001), in part by the Natural Science Foundation of China (Grant no. 62036006 and Grand no. 62276200), in part by the Alibaba Group through Alibaba Innovative Research Program and the China Postdoctoral Science Foundation (Grant no. 2021T140528).

References

[1] R. Qureshi, M. Uzair, K. Khurshid, H. Yan, Hyperspectral document image processing: applications, challenges and future prospects, Pattern Recognit. 90 (2019) 12–22.

[2] O. Okwuashi, C.E. Ndehedehe, Deep support vector machine for hyperspectral image classification, Pattern Recognit. 103 (2020) 107298.

[3] O. Déniz, G. Bueno, J. Salido, F. De la Torre, Face recognition using histograms of oriented gradients, Pattern Recognit. Lett. 32 (12) (2011) 1598–1603.

[4] M. Heikkilä, M. Pietikäinen, C. Schmid, Description of interest regions with local binary patterns, Pattern Recognit. 42 (3) (2009) 425–436.

[5] S. Ozer, C.H. Chen, H.A. Cirpan, A set of new Chebyshev kernel functions for support vector machine pattern classification, Pattern Recognit. 44 (7) (2011) 1435–1447.

[6] F. Yuan, Z. Zhang, Z. Fang, An effective CNN and transformer complementary network for medical image segmentation, Pattern Recognit. 136 (2023) 109228.

[7] X. Liu, Z. You, Y. He, S. Bi, J. Wang, Symmetry-driven hyper feature GCN for skeleton-based gait recognition, Pattern Recognit. 125 (2022) 108520.

[8] N.-N. Ji, J.-S. Zhang, C.-X. Zhang, A sparse-response deep belief network based on rate distortion theory, Pattern Recognit. 47 (9) (2014) 3179–3191.

[9] F.A. Van-Horenbeke, A. Peer, Nilrnn: a neocortex-inspired locally recurrent neural network for unsupervised feature learning in sequential data, Cogn. Comput. (2023) 1–17.

[10] S. Yu, S. Jia, C. Xu, Convolutional neural networks for hyperspectral image classification, Neural Comput. 219 (2017) 88–98.

[11] J.M. Haut, M.E. Paoletti, J. Plaza, A. Plaza, J. Li, Hyperspectral image classification using random occlusion data augmentation, IEEE Geosci. Remote Sens. Lett. 16 (11) (2019) 1751–1755.

[12] A.B. Hamida, A. Benoit, P. Lambert, C.B. Amar, 3-D deep learning approach for remote sensing image classification, IEEE Trans. Geosci. Remote Sens. 56 (8) (2018) 4420–4434.

[13] Y. Li, H. Zhang, Q. Shen, Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network, Remote Sens. 9 (1) (2017) 67.

[14] M. He, B. Li, H. Chen, Multi-scale 3D deep convolutional neural network for hyperspectral image classification, in: 2017 IEEE Int. Conf. Image Process. (ICIP), IEEE, 2017, pp. 3904–3908.

[15] A. Sellami, A.B. Abbes, V. Barra, I.R. Farah, Fused 3-D spectral-spatial deep neural networks and spectral clustering for hyperspectral image classification, Pattern Recognit. Lett. 138 (2020) 594–600.

[16] M.Q. Alkhatib, M. Al-Saad, N. Aburaed, S. Almansoori, J. Zabalza, S. Marshall, H. Al-Ahmad, Tri-CNN: a three branch model for hyperspectral image classification, Remote Sens. 15 (2) (2023) 316.

[17] J. Li, H. Xing, Z. Ao, H. Wang, W. Liu, A. Zhang, Convolution-transformer adaptive fusion network for hyperspectral image classification, Appl. Sci. 13 (1) (2023) 492.

[18] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Commun. ACM 63 (11) (2020) 139–144.

[20] D.P. Kingma, M. Welling, et al., An introduction to variational autoencoders, Found. Trends® Mach. Learn. 12 (4) (2019) 307–392.

[21] H. Lee, H. Kwon, Going deeper with contextual CNN for hyperspectral image classification, IEEE Trans. Image Process. 26 (10) (2017) 4843–4855.

[22] J. Nalepa, M. Myller, M. Kawulok, Hyperspectral data augmentation, (2019). arXiv preprint arXiv:1903.05580.

[23] X. Yang, W. Yin, L. Li, L. Zhang, Dense people counting using IR-UWB radar with a hybrid feature extraction method, IEEE Trans. Geosci. Remote Sens. Lett. 16 (1) (2018) 30–34.

[24] Y. Zhan, D. Hu, Y. Wang, X. Yu, Semisupervised hyperspectral image classification based on generative adversarial networks, IEEE Trans. Geosci. Remote Sens. Lett. 15 (2) (2017) 212–216.

[25] Z. Zhong, J. Li, D.A. Clausi, A. Wong, Generative adversarial networks and conditional random fields for hyperspectral image classification, IEEE Trans. Cybern. 50 (7) (2019) 3318–3329.

[26] X. Wang, K. Tan, Q. Du, Y. Chen, P. Du, Caps-tripleGAN: GAN-assisted capsnet for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 57 (9) (2019) 7232–7245.

[27] M.E. Paoletti, J.M. Haut, R. Fernandez-Beltran, J. Plaza, A. Plaza, J. Li, F. Pla, Capsule networks for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 57 (4) (2018) 2145–2160.

[28] L. Zhu, Y. Chen, P. Ghamisi, J.A. Benediktsson, Generative adversarial networks for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 56 (9) (2018) 5046–5063.

[29] S.S. Mullick, S. Datta, S. Das, Generative adversarial minority oversampling, in: Proc. IEEE/CVF Conf. Comput. Vis., 2019, pp. 1695–1704.

[30] S.K. Roy, J.M. Haut, M.E. Paoletti, S.R. Dubey, A. Plaza, Generative adversarial minority oversampling for spectral–spatial hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 60 (2021) 1–15.

[31] C. Sun, X. Zhang, H. Meng, X. Cao, J. Zhang, AC-WGAN-GP: generating labeled samples for improving hyperspectral image classification with small-samples, Remote Sens. 14 (19) (2022) 4910.

[32] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A.C. Courville, Improved training of Wasserstein gans, Proc. NIPS 30 (2017).

[33] A. Odena, C. Olah, J. Shlens, Conditional image synthesis with auxiliary classifier GANs, in: Int. Conf. Machine Learning, PMLR, 2017, pp. 2642–2651.

[34] C. Gong, D. Wang, M. Li, V. Chandra, Q. Liu, Keepaugment: a simple information-preserving data augmentation approach, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit, 2021, pp. 1055–1064.

[35] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E.D. Cubuk, Q.V. Le, B. Zoph, Simple copy-paste is a strong data augmentation method for instance segmentation, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit, 2021, pp. 2918–2928.

[36] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: Int. Conf. Machine Learning, PMLR, 2017, pp. 214–223.

[37] A. Antoniou, A. Storkey, H. Edwards, Data augmentation generative adversarial networks, (2017). arXiv preprint arXiv:1711.04340.

[38] G.M. Foody, Status of land cover classification accuracy assessment, Remote Sens. Environ. 80 (1) (2002) 185–201.

[39] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, (2015). arXiv preprint arXiv:1511.06434.

Mingyang Zhang received the B.S. degree in automation and the Ph.D. degree in pattern recognition and intelligent system from Xidian University, Xi'an, China, in 2012 and 2018, respectively. Since 2018, he has been a Lecturer with the School of Electronic Engineering, Xidian University. His research interests include computational intelligence and remote sensing image understanding.

Zhaoyang Wang is currently pursuing the master's degree in Electronic Science and Technology at the School of Electronic Engineering, Xidian University, Xi'an, China. He received the B.S. degree in Electronic Science and Technology at the School of Electronic Engineering, Xidian University, Xi'an, China. His research interests include remote sensing image processing and deep learning.

Xiangyu Wang is currently pursuing the master's degree in Electronic Science and Technology at the School of Electronic Engineering, Xidian University, Xi'an, China. He received the B.S. degree in Intelligence Science and Technology at the School of Artificial Intelligence, Xidian University, Xi'an, China. His research interests include remote sensing image processing and deep learning.

Maoguo Gong received the B. Eng degree and Ph.D. degree from Xidian University. Since 2006, he has been a teacher of Xidian University. He was promoted to associate professor and full professor in 2008 and 2010, respectively, both with exceptive admission. Gong's research interests are broadly in the area of computational intelligence, with applications to optimization, learning, data mining and image understanding. He has published over one hundred papers in journals and conferences, and holds over twenty granted patents as the first inventor. He is leading or has completed over twenty projects as the Principle Investigator, funded by the National Natural Science Foundation of China, the National Key Research and Development Program of China, and others. He was the recipient of the prestigious National Program for Support of the Leading Innovative Talents from the Central Organization Department of China, the Leading Innovative Talent in the Science and Technology from the Ministry of Science and Technology of China, the Excellent Young Scientist Foundation from the National Natural Science Foundation of China, the New Century Excellent Talent from the Ministry of Education of China, and the National Natural Science Award of China. He is the Director of Chinese Association for Artificial Intelligence-Youth Branch, Senior Member of IEEE and Chinese Computer Federation, Associate Editor or Editorial Board Member for over five journals including the IEEE Transactions on Evolutionary Computation, the IEEE Transactions on Neural Networks and Learning Systems, and the IEEE Transactions on Emerging Topics in Computational Intelligence.

Yue Wu received the B. Eng. and Ph.D. degrees from Xidian University, Xi'an, China, in 2011 and 2016, respectively. Since 2016, he has been a Teacher with Xidian University. He is currently an Associate Professor with Xidian University. He has authored or co-authored more than 40 papers in refereed journals and proceedings. His research interests include computer vision and computational intelligence.

Hao Li received the B.S. degree in electronic engineering and the Ph.D. degree in pattern recognition and intelligent systems from Xidian University, Xi'an, China, in 2013 and 2018, respectively. He is currently an Associate Professor with the School of Electronic Engineering, Xidian University. His current research interests include computational intelligence and machine learning.