



Article

Variational Generative Adversarial Network with Crossed Spatial and Spectral Interactions for Hyperspectral Image Classification

Zhongwei Li ¹, Xue Zhu ² , Ziqi Xin ², Fangming Guo ¹, Xingshuai Cui ² and LeiQuan Wang ^{2,*}

¹ College of Oceanography and Space Informatics, China University of Petroleum (East China), Qingdao 266580, China; lizhongwei@upc.edu.cn (Z.L.); guofangming@s.upc.edu.cn (F.G.)

² College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China; S19070024@s.upc.edu.cn (X.Z.); S20070012@s.upc.edu.cn (Z.X.); cuixingshuai@s.upc.edu.cn (X.C.)

* Correspondence: 20060068@upc.edu.cn

Abstract: Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) have been widely used in hyperspectral image classification (HSIC) tasks. However, the generated HSI virtual samples by VAEs are often ambiguous, and GANs are prone to the mode collapse, which lead the poor generalization abilities ultimately. Moreover, most of these models only consider the extraction of spectral or spatial features. They fail to combine the two branches interactively and ignore the correlation between them. Consequently, the variational generative adversarial network with crossed spatial and spectral interactions (CSSVGAN) was proposed in this paper, which includes a dual-branch variational Encoder to map spectral and spatial information to different latent spaces, a crossed interactive Generator to improve the quality of generated virtual samples, and a Discriminator stuck with a classifier to enhance the classification performance. Combining these three subnetworks, the proposed CSSVGAN achieves excellent classification by ensuring the diversity and interacting spectral and spatial features in a crossed manner. The superior experimental results on three datasets verify the effectiveness of this method.

Keywords: hyperspectral image classification; variational autoencoder; generative adversarial network; crossed spatial and spectral interactions



Citation: Li, Z.; Zhu, X.; Xin, Z.; Guo, F.; Cui, X.; Wang, L. Variational Generative Adversarial Network with Crossed Spatial and Spectral Interactions for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 3131. <https://doi.org/10.3390/rs13163131>

Academic Editors: Fahimeh Farahnakian, Jukka Heikkonen and Pouya Jafarzadeh

Received: 12 July 2021

Accepted: 5 August 2021

Published: 7 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Hyperspectral images (HSI) contain hundreds of continuous and diverse bands rich in spectral and spatial information, which can distinguish land-cover types more efficiently compared with ordinary remote sensing images [1,2]. In recent years, Hyperspectral images classification (HSIC) has become one of the most important tasks in the field of remote sensing with wide application in scenarios such as urban planning, geological exploration, and agricultural monitoring [3–6].

Originally, models such as support vector machines (SVM) [7], logistic regression (LR) [8] and k-nearest neighbors algorithm (KNN) [9], have been widely used in HSI classification tasks for their intuitive outcomes. However, most of them only utilize handcrafted features, which fail to embody the distribution characteristics of different objects. To solve this problem, a series of deep discriminative models, such as convolutional neural networks (CNNs) [10–12], recurrent neural network (RNN) [13] and Deep Neural Networks (DNN) [14] have been proposed to optimize the classification results by fully utilizing and abstracting the limited data. Though having gained great progress, these methods only analyze the spectral characteristics through an end-to-end neural network without full consideration of special properties contained in HSI. Therefore, the extraction of high-level and abstract features in HSIC remains a challenging task. Meanwhile, the

jointed spectral-spatial features extraction methods [15,16] have aroused wide interest in Geosciences and Remote Sensing community [17]. Du proposed a jointed network to extract spectral and spatial features with dimensionality reduction [18]. Zhao et al. proposed a hybrid spectral CNN (HybridSN) to better extract double-way features [19], which combined spectral-spatial 3D-CNN with spatial 2D-CNN to improve the classification accuracy.

Although the methods above enhance the abilities of spectral and spatial features extraction, they are still based on the discriminative model in essence, which can neither calculate prior probability nor describe the unique features of HSI data. In addition, the access to acquire HSI data is very expensive and scarce, requiring huge human resources to label the samples by field investigation. These characteristics make it impractical to obtain enough markable samples for training. Therefore, the deep generative models have emerged at the call of the time. Variational auto encoder (VAE) [20] and generative adversarial network (GAN) [21] are the representative methods of generative models.

Liu [22] and Su [23] used VAEs to ensure the diversity of the generated data that were sampled from the latent space. However, the generated HSI virtual samples are often ambiguous, which cannot guarantee similarities with the real HSI data. Therefore, GANs have also been applied for HSI generation to improve the quality of generated virtual data. GANs strengthen the ability of discriminators to distinguish the true data sources from the false by introducing "Nash equilibrium" [24–29]. For example, Zhan [30] designed a 1-D GAN (HSGAN) to generate the virtual HSI pixels similar to the real ones, thus improving the performance of the classifier. Feng [31] devised two generators to generate 2D-spatial and 1D-spectral information respectively. Zhu [32] exploited 1D-GAN and 3D-GAN architectures to enhance the classification performance. However, GANs are prone to mode collapse, resulting in poor generalization ability of HSI classification.

To overcome the limitations of VAEs and GANs, VAE-GAN jointed framework has been proposed for HSIC. Wang proposed a conditional variational autoencoder with an adversarial training process for HSIC (CVA²E) [33]. In this work, GAN was spliced with VAE to realize high-quality restoration of the samples and achieve diversity. Tao et al. [34] proposed the semi-supervised variational generative adversarial networks with a collaborative relationship between the generation network and the classification network to produce meaningful samples that contribute to the final classification. To sum up, in VAE-GAN frameworks, VAE focuses on encoding the latent space, providing creativity of generated samples, while GAN concentrates on replicating the data, contributing to the high quality of virtual samples.

Spectral and spatial are two typical characteristics of HSI, both of which must be taken into account for HSIC. Nevertheless, the distributions of spectral and spatial features are not identical. Therefore, it is difficult to cope with such a complex situation for a single encoder in VAEs. Meanwhile, most of the existing generative methods use spectral and spatial features respectively for HSIC, which affects the generative model to generate realistic virtual samples. In fact, the spectral and spatial features are closely correlated, which cannot be treated separately. Interaction between spectral and spatial information should be established to refine the generated virtual samples for better classification performance.

In this paper, a variational generative adversarial network with crossed spatial and spectral interactions (CSSVGAN) was proposed for HSIC, which consists of a dual-branch variational Encoder, a crossed interactive Generator, and a Discriminator stuck together with a classifier. The dual-branch variational Encoder maps spectral and spatial information to different latent spaces. The crossed interactive Generator reconstructs the spatial and spectral samples from the latent spectral and spatial distribution in a crossed manner. Notably, the intersectional generation process promotes the consistency of learned spatial and spectral features and simulates the highly correlated spatial and spectral characteristics of true HSI. The Discriminator receives the samples from both generator and original training data to distinguish the authenticity of the data. To sum up, the variational Encoder ensures diversity, and the Generator guarantees authenticity. The two components place higher demands on the Discriminator to achieve better classification performance.

Compared with the existing literature, this paper is expected to make the following contributions:

- The dual-branch variational Encoder in the jointed VAE-GAN framework is developed to map spectral and spatial information into different latent spaces, provides discriminative spectral and spatial features, and ensures the diversity of generated virtual samples.
- The crossed interactive Generator is proposed to improve the quality of generated virtual samples, which exploits the consistency of learned spatial and spectral features to imitate the highly correlated spatial and spectral characteristics of HSI.
- The variational generative adversarial network with crossed spatial and spectral interactions is proposed for HSIC, where the diversity and authenticity of generated samples are enhanced simultaneously.
- Experimental results on the three public datasets demonstrate that the proposed CSSVGAN achieves better performance compared with other well-known models.

The remainder of this paper is arranged as follows. Section 2 introduces VAEs and GANs. Section 3 provides the details of the CSSVGAN framework and the crossed interactive module. Section 4 evaluates the performance of the proposed CSSVGAN through comparison with other methods. The results of the experiment are discussed in Section 5 and the conclusion is given in Section 6.

2. Related Work

2.1. Variational Autoencoder

Variational autoencoder is one variant of the standard AE, proposed by Kingma et al. for the first time [35]. The essence of VAE is to construct an exclusive distribution for each sample X and then sample it represented by Z . It brings Kullback–Leibler [36] divergence penalty method into the process of sampling and constrains it. Then the reconstructed data can be translated to generated simulation data through deep training. The above principle gives VAE a significant advantage in processing hyperspectral images with expensive and rare samples. VAE model adopts the posterior distribution method to verify that $\rho(Z|X)$ rather than $\rho(Z)$ obeys the normal distribution. Then it manages to find the mean μ and variance σ of $\rho(Z|X_k)$ corresponding to each X_k through the training of neural networks (where X_k represents the sample of the original data and $\rho(Z|X_k)$ represents the posterior distribution). Another particularity of VAE is that it makes all $\rho(Z|X)$ align with the standard normal distribution $N \sim (0, 1)$. Taking account of the complexity of HSI data, VAE has superiority over AE in terms of noise interference [37]. It can prevent the occurrence of zero noise, increase the diversity of samples, and further ensure the generation ability of the model.

A VAE model is consists of two parts: Encoder M and Decoder N . M is an approximator for the probability function $m_\tau(z|x)$, and N is to generate the posterior's approximate value $n\theta(x, z)$. τ and θ are the parameters of the deep neural network, aiming to optimize the following objective functions jointly.

$$V(P, Q) = -KL(m_\tau(z|x)||p_\theta(z|x)) + R(x), \quad (1)$$

Among them, R is to calculate the reconstruction loss of a given sample x in the VAE model. The framework of VAE is described in Figure 1, where e_i represents the sample of standard normal distribution, corresponding with X_k one to one.

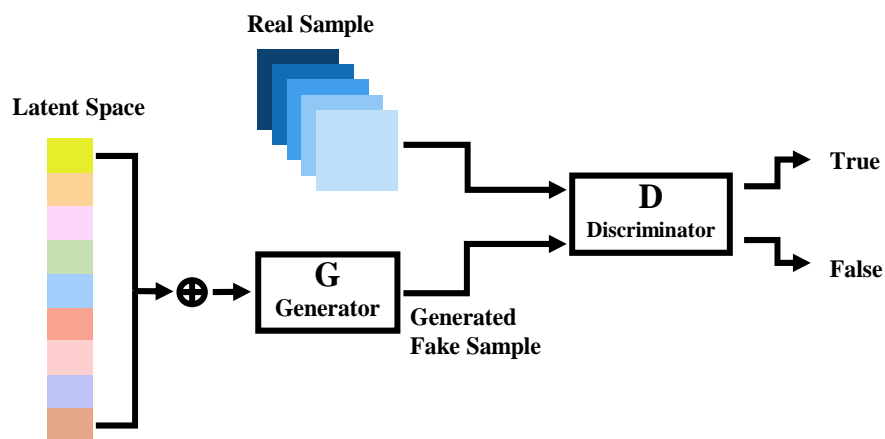


Figure 1. The framework of VAE.

2.2. Generative Adversarial Network

Generative adversarial network is put forward by Goodfellow et al. [24], which trains the generation model with a minimax game based on the game theory. The GAN has gained remarkable results in representing the distribution of latent variables for its special structure, which has attracted more attention from the field of visual image processing. A GAN model includes two subnets: the generator G , denoted as $G(z; \theta_g)$ and the discriminator D , denoted as $G(x; \theta_d)$, and θ_g and θ_d are defined as parameters of the deep neural networks. G shows a prominent capacity in learning the mapping of latent variables and synthesizing new similar data from mapping represented by $G(z)$. The function of D is to take the original HSI or the fake image generated by G as input and then distinguish its authenticity. The architecture of GAN is shown in Figure 2.

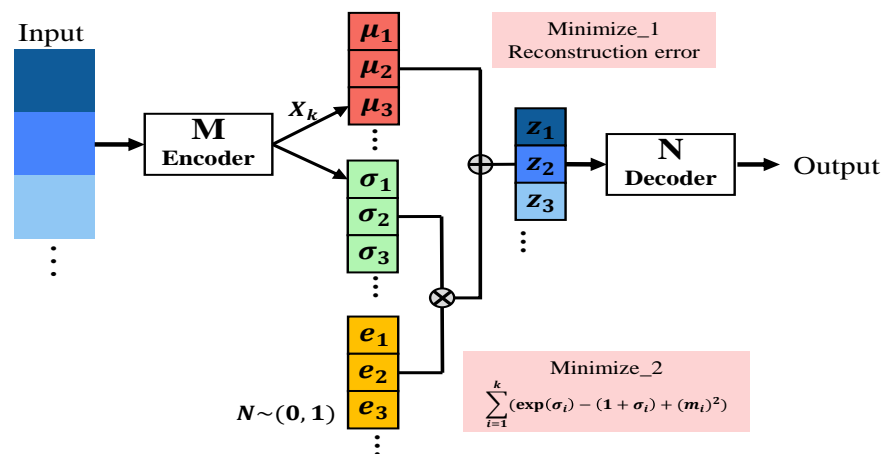


Figure 2. The architecture of GAN.

After the game training, G and D would maximize log-likelihood respectively and achieve the best generation effect by competing with each other. The expression of the above process is as follows:

$$\min_G \max_D V(G, D) = E_{x \sim P(x)} [\log_D(x)] + E_{x \sim P_{g(z)}} [\log(1 - D(G(z)))] \tag{2}$$

where $P(x)$ represents the real data distribution and $P_{g(z)}$ means the samples' distribution generated by G . The game would reach a global equilibrium situation between the two players when $P(x)$ equaling to $P_{g(z)}$ happened. In this case, the best performance of $D(x)$ can be expressed as:

$$D(x)_{max} = P_{(x)+P_{g(x)}} \tag{3}$$

X_1 into the Discriminator for adversarial training to get the predicted classification results $\hat{Y} = \hat{y}_i$ by the classifier.

3.2. The Dual-Branch Variational Encoder in CSSVGAN

In the CSSVGAN model mentioned above, the Encoder (Figure 4) is composed of a dual-branch spatial feature extraction E_1 and a spectral feature extraction E_1 to generate more diverse samples. In the E_1 module, the size of the 3D convolution kernel is $(1 \times 1 \times 2)$, the stride is $(2, 2, 2)$ and the spectral features are marked as Z_1 . The implementation details are described in Table 1. Identically, in the E_2 module, the 3D convolution kernels, the strides and the spatial features are presented by $(5 \times 5 \times 1)$, $(2, 2, 2)$ and Z_2 respectively, as described in Table 2.

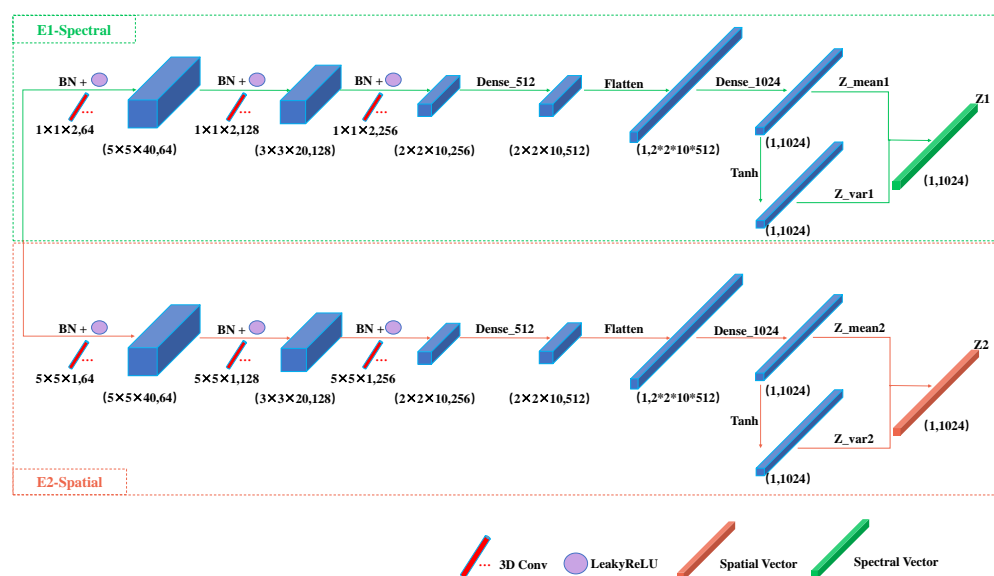


Figure 4. The dual-branch Encoder in CSSVGAN.

Table 1. The implementation details of the Spectral feature extraction E_1 .

Input Size	Layer Operations	Output Size
$(9 \times 9 \times 80, 1)$	Conv3D $(1 \times 1 \times 2, 64) - BN - LeakyReLU$	$(5 \times 5 \times 40, 64)$
$(5 \times 5 \times 40, 64)$	Conv3D $(1 \times 1 \times 2, 128) - BN - LeakyReLU$	$(3 \times 3 \times 20, 128)$
$(3 \times 3 \times 20, 128)$	Conv3D $(1 \times 1 \times 2, 256) - BN - LeakyReLU$	$(2 \times 2 \times 10, 256)$
$(2 \times 2 \times 10, 256)$	Dense (512) - BN - LeakyReLU	$(2 \times 2 \times 10, 512)$
$(2 \times 2 \times 10, 512)$	Flatten	$(, 20, 480)$
$(, 20, 480)$	Dense (1024)	$(, 1024)$
$(, 1024)$	Dense (1024) - Tanh	$(, 1024)$
$(, 1024)$	Lambda (Sampling)	$(, 1024)$

Table 2. The implementation details of the Spatial feature extraction E_2 .

Input Size	Layer Operations	Output Size
$(9 \times 9 \times 80, 1)$	Conv3D $(5 \times 5 \times 1, 64) - BN - LeakyReLU$	$(5 \times 5 \times 40, 64)$
$(5 \times 5 \times 40, 64)$	Conv3D $(5 \times 5 \times 1, 128) - BN - LeakyReLU$	$(3 \times 3 \times 20, 128)$
$(3 \times 3 \times 20, 128)$	Conv3D $(5 \times 5 \times 1, 256) - BN - LeakyReLU$	$(2 \times 2 \times 10, 256)$
$(2 \times 2 \times 10, 256)$	Dense (512) - BN - LeakyReLU	$(2 \times 2 \times 10, 512)$
$(2 \times 2 \times 10, 512)$	Flatten	$(, 20, 480)$
$(, 20, 480)$	Dense (1024)	$(, 1024)$
$(, 1024)$	Dense (1024) - Tanh	$(, 1024)$
$(, 1024)$	Lambda (Sampling)	$(, 1024)$

Meanwhile, to ensure the consistent distribution of samples and original data, KL divergence principle is utilized to constrain Z_1 and Z_2 separately. Assuming that the mean and variance of Z_i are expressed as Z_{mean_i} and $Z_{vari}(i = 1, 2)$, the loss function in the training process is as follows:

$$L_i(\theta, \varphi) = -KL(q_\varphi(z_i|x)||p_\theta(z_i|x)), \quad (5)$$

where $p(z_i|x)$ is the posterior distribution of potential eigenvectors in the Encoder module, and its calculation is based on the Bayesian formula as shown below. But when the dimension of Z is too high, the calculation of $P(x)$ is not feasible. At this time, a known distribution $q(z_i|x)$ is required to approximate $p(z_i|x)$, which is given by KL divergence. By minimizing KL divergence, the approximate $p(z_i|x)$ can be obtained. θ and φ represent the parameters of distribution function p and q separately.

$$L_i(\theta, \varphi) = E_{q_\varphi(z_i,x)}[\log_{\frac{p_\theta(x,z_i)}{q_\varphi(z_i,x)}}] - E_q(x)[\log_q(x)], \quad (6)$$

Formula (6) in the back is provided with a constant term $\log N$, the entropy of empirical distribution $q(x)$. The advantage of it is that the optimization objective function is more explicit, that is, when $p_\theta(z_i, x)$ is equal to $q_\varphi(z_i, x)$, KL dispersion can be minimized.

3.3. The Crossed Interactive Generator in CSSVGAN

In CSSVGAN, the crossed interactive Generator module plays a role in data restoration of VAE and data expansion of GAN, which includes the spectral Generator G_1 and the spatial Generator G_2 in the crossed manner. G_1 accepts the spatial latent variables Z_2 to generate spectral virtual data F_1 , and G_2 accepts the spectral latent variables Z_1 to generate spatial virtual data F_2 .

As shown in Figure 5, the 3D convolution of spectral Generator G_1 is $(1 \times 1 \times 2)$ that uses $(2, 2, 2)$ strides to convert the spatial latent variables Z_2 to the generated samples. Similarly, the spatial Generator G_2 with $(5 \times 5 \times 1)$ convolution uses $(2, 2, 2)$ strides to transform the spectral latent variables Z_1 into generated samples. Therefore, the correlation between spectral and spatial features in HSI can be fully considered to further improve the quality and authenticity of the generated samples. The implementation details of G_1 and G_2 are described in Tables 3 and 4.

Table 3. The implementation details of spectral Generator G_1 .

Input Size	Layer Operations	Output Size
(, 1024)	Dense $(2 * 2 * 10 * 256)$	(10, 240)
(, 10, 240)	Reshape $(2 \times 2 \times 10 \times 256)BN - LeakyReLU$	(2, 2, 10, 256)
(2, 2, 10, 256)	Conv3DTranspose $(1 \times 1 \times 2, 128)BN - LeakyReLU$	(4, 4, 20, 128)
(4, 4, 20, 128)	Conv3DTranspose $(1 \times 1 \times 2, 64)BN - LeakyReLU$	(8, 8, 40, 64)
(8, 8, 40, 64)	Conv3DTranspose $(1 \times 1 \times 2, 1)LeakyReLU - Tanh$	(9, 9, 80, 1)

Table 4. The implementation details of spatial Generator G_2 .

Input Size	Layer Operations	Output Size
(, 1024)	Dense $(2 * 2 * 10 * 256)$	(, 10, 240)
(, 10, 240)	Reshape $(2 \times 2 \times 10 \times 256)BN - LeakyReLU$	(2, 2, 10, 256)
(2, 2, 10, 256)	Conv3DTranspose $(5 \times 5 \times 1, 128)BN - LeakyReLU$	(4, 4, 20, 128)
(4, 4, 20, 128)	Conv3DTranspose $(5 \times 5 \times 1, 64)BN - LeakyReLU$	(8, 8, 40, 64)
(8, 8, 40, 64)	Conv3DTranspose $(5 \times 5 \times 1, 1)LeakyReLU - Tanh$	(9, 9, 80, 1)

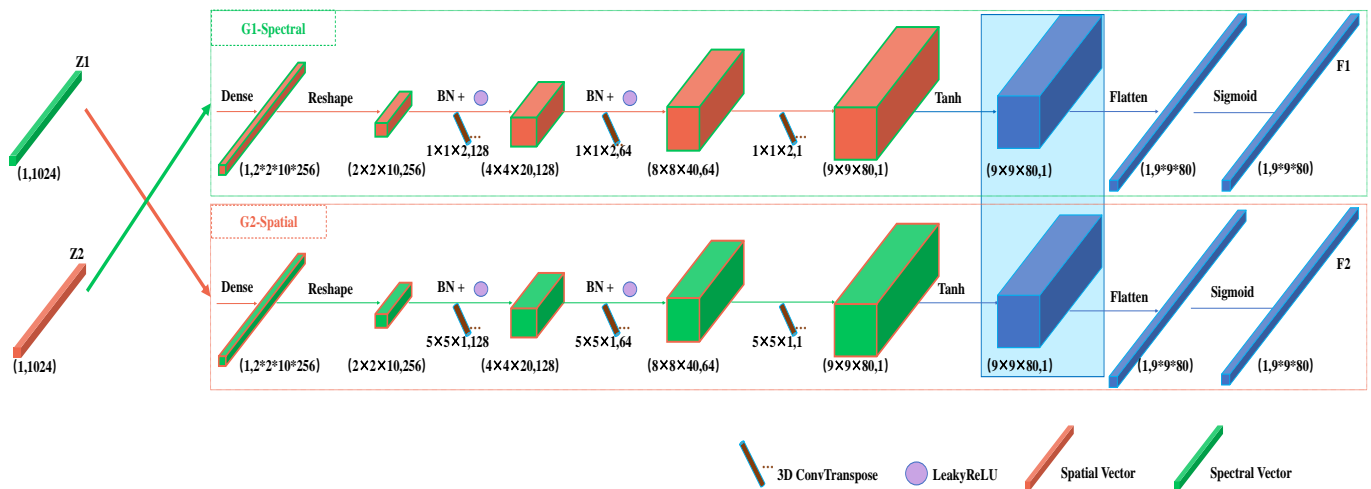


Figure 5. The Crossed Interactive Generator in CSSVGAN.

Because the mechanism of GAN is that the Generator and Discriminator are against each other before reaching the Nash equilibrium, the Generator has two target functions, as shown below.

$$MSE_{Loss_i} = \frac{1}{n} \sum (y_{ij} - \bar{y}_{ij})^2, \quad (7)$$

where n is the number of samples, $i = 1, 2$, y_j means the label of virtual samples, and \bar{y}_j represents the label of the original data corresponding to y_j . The above formula makes the virtual samples generated by crossed interactive Generator as similar as possible to the original data.

$$Binary_{Loss_i} = -\frac{1}{N} \sum_{j=1}^N y_{ij} \cdot \log(p(y_{ij})) + (1 - y_{ij}) \cdot (1 - p(y_{ij})), \quad (8)$$

$Binary_{Loss}$ is a logarithmic loss function and can be applied to the binary classification task. Where y is the label (either true or false), and $p(y)$ is the probability that N sample points belonging to the real label. Only if y_j equals to $p(y_i)$, the total loss would be zero.

3.4. The Discriminator Stuck with a Classifier in CSSVGAN

As shown in Figure 6, the Discriminator needs to specifically identify the generated data as false and the real HSI data as true. This process can be regarded as a two-category task using one-sided label smoothing: defining the real HSI data as 0.9 and the false as zero. The loss function of it marked with $Binary_{(Loss_D)}$ is the same as the Formula (10) enumerated above. Moreover, the classifier is stuck as an interface to the output of Discriminator and the classification results are calculated directly through the SoftMax layer, where C represents the total number of labels in training data. As mentioned above, the Encoder ensures diversity and the Generator guarantees authenticity. All these contributions place higher demands on Discriminator to achieve better classification performance. Thus, the CSSVGAN framework yields a better classification result.

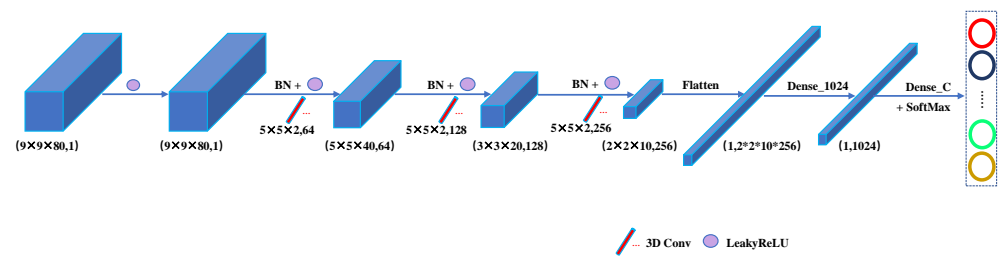


Figure 6. The Discriminator stuck with a classifier in CSSVGAN.

The implementation details of the Discriminator in CSSVGAN are described in Table 5 with the 3D convolution of $(5 \times 5 \times 2)$ and strides of $(2, 2, 2)$. Identifying C categories belongs to a multi-classification assignment. The SoftMax method is taken as the standard for HSIC. As shown below, the CSSVGAN method should allocate the sample x of each class c to the most likely one of the C classes to get the predicted classification results. The specific formula is as follows:

$$y_i = S(x_i) = \frac{e^{x_i}}{\sum_{j=1}^C e^{x_j}}, \quad (9)$$

Then the category of X can be expressed as the formula below:

$$class(c) = \arg \max_i (y_i = S(x_i)), \quad (10)$$

where S , C , X , Y_i signify the SoftMax function, the total number of categories, the input of SoftMax, and the probability that the prediction object belongs to class C , respectively. X_i similar with X_j is a sample of one certain category. Therefore, the following formula can be used for the loss function of objective constraint.

$$C_{Loss} = - \sum_{i=1}^n p(y_{i1}) \cdot \log y_{i1} + p(y_{i2}) \cdot \log(y_{i2}) + \dots + p(y_{ic}) \cdot \log(y_{ic}), \quad (11)$$

where n means the total number of samples, C represents the total number of categories, and y denotes the single label (either true or false) with the same description as above.

Table 5. The implementation details in Discriminator.

Input Size	Layer Operations	Output Size
$(9 \times 9 \times 80, 1)$	<i>BN – LeakyReLU</i>	$(9 \times 9 \times 80, 1)$
$(9 \times 9 \times 80, 1)$	Conv3D $(5 \times 5 \times 2, 64)$ – <i>BN – LeakyReLU</i>	$(5 \times 5 \times 40, 64)$
$(5 \times 5 \times 40, 64)$	Conv3D $(5 \times 5 \times 2, 128)$ – <i>BN – LeakyReLU</i>	$(3 \times 3 \times 20, 128)$
$(3 \times 3 \times 20, 128)$	Conv3D $(5 \times 5 \times 2, 256)$ – <i>BN – LeakyReLU</i>	$(2 \times 2 \times 10, 256)$
$(2 \times 2 \times 10, 256)$	Flatten	$(, 10, 240)$
$(, 10, 240)$	Dense (16)	$(, 16)$

3.5. The Total Loss of CSSVGAN

As illustrated in Figure 3, up till now, the final goal of the total loss of the CSSVGAN model can be divided into four parts: two KL divergence constraint losses and a mean-square error loss from the Encoder, two binary losses from the Generator, one binary loss from the Discriminator and one multi-classification loss from the multi classifier. The ensemble formula can be expressed as:

$$\begin{aligned}
 L_{Total} = & \underbrace{\sigma_1 L_1(\theta, \varphi) + \sigma_2 L_2(\theta, \varphi) + \sigma_3 MSE_{Loss_{1,2}}}_{Encoder_Loss} \\
 & + \underbrace{\sigma_4 Binary_{Loss_1} + \sigma_5 Binary_{Loss_2}}_{Generator_Loss} + \underbrace{Binary_{Loss_D}}_{Discriminator_Loss} + \underbrace{C_{Loss}}_{Classifier_Loss}, \quad (12)
 \end{aligned}$$

where L_1 and L_2 represent the loss between Z_1 or Z_2 and the standard normal distribution respectively in Section 3.2. MSE_{Loss_1} and MSE_{Loss_2} signify the mean square error of y_1 and y_2 in Section 3.3 separately. $MSE_{Loss_{1,2}}$ calculates the mean square error between y_1 and y_2 . The purpose of $Binary_{Loss_1}$ and $Binary_{Loss_2}$ is to assume that the virtual data F_1 and F_2 (in Section 3.3) are true with a value of one. $Binary_{Loss_D}$ denotes that the Discriminator identifies F_1 and F_2 as false data with a value of zero. Finally, the C_{Loss} is the loss of multi classes of the classifier.

4. Experiments

4.1. Dataset Description

In this paper, three representative hyperspectral datasets recognized by the remote sensing community (i.e., Indian Pines, Pavia University and Salinas) are accepted as benchmark datasets. The details of them are as follows:

(1) Indian pines (IP): The first dataset was accepted for HSI classification imaged by Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) in Northwestern Indiana in the USA. It includes 16 categories with a spatial resolution of approximately 20 m per pixel. Samples are shown in Figure 7. The spectral of AVIRIS coverage ranges from 0.4 to 2.5 μm and includes 200 bands for continuous imaging of ground objects (20 bands are influenced by noise or steam, so only 200 bands are left for research), bring about the total image size of $145 \times 145 \times 200$. However, since it contains a complex sample distribution, the category samples of training labels were very imbalanced. As some classes have more than 2000 samples while some have less than 30 merely, it is relatively difficult to achieve a high-precision classification of IP HSI.

(2) Pavia University (PU): The second dataset was a part of the hyperspectral image data of the Pavia city in Italy, photographed by the German airborne reflective optics spectral imaging system (Rosis-03) in 2003, containing 9 categories (see Figure 8). The resolution of this spectral imager is 1.3 m, including continuously 115 wavebands in the range of 0.43–0.86 μm . Among these bands, 12 bands were eliminated due to the influence of noise. Therefore, the images with the remaining 103 spectral bands in size 610×340 are normally used.

(3) Salinas (SA): The third dataset recorded the image of Salinas Valley in California, USA, which was also captured by AVIRIS. Unlike the IP dataset, it has a spatial resolution of 3.7 m and consists of 224 bands. However, researchers generally utilize the image of 204 bands after excluding 20 bands affected by water reflection. Thus, the size of the Salinas is 512×217 , and Figure 9 depicts the color composite of the image as well as the ground truth map.

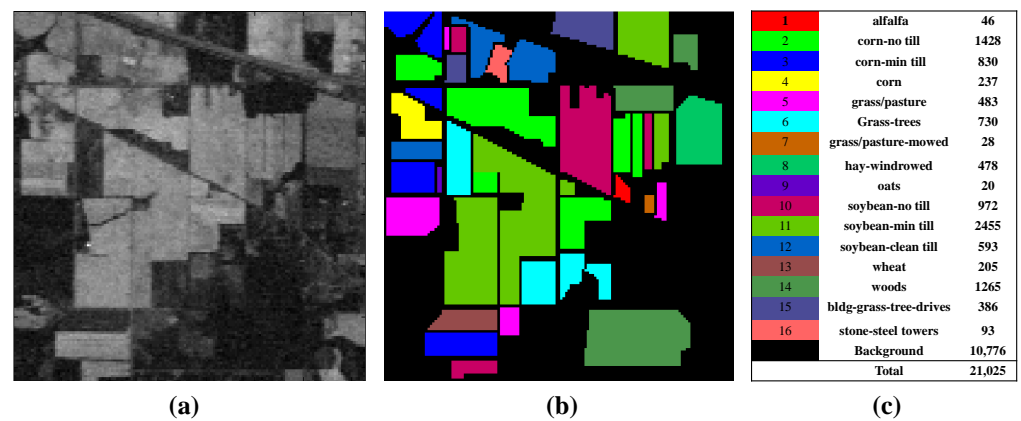


Figure 7. Indian Pines imagery: (a) color composite with RGB, (b) ground truth, and (c) category names with labeled samples.

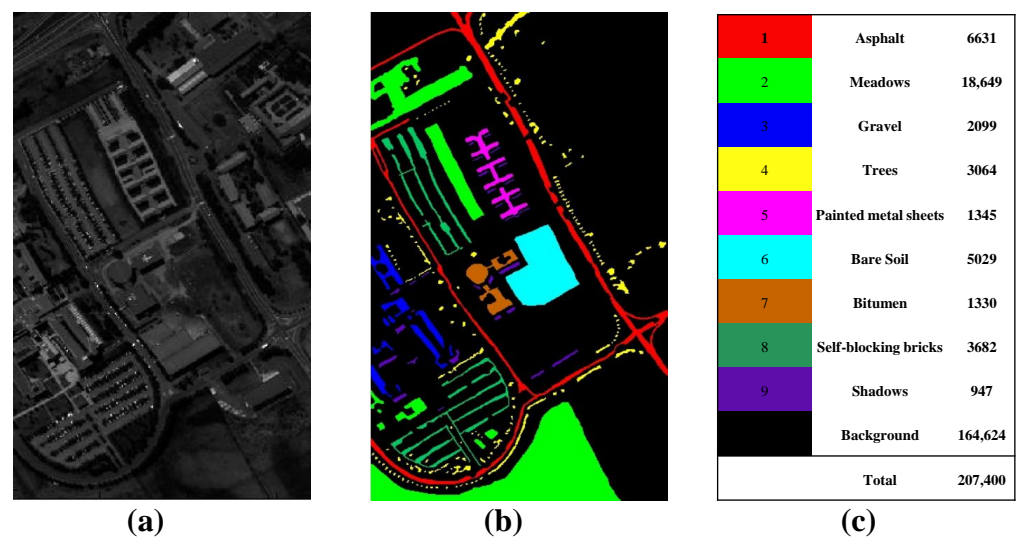


Figure 8. Pavia University imagery: (a) color composite with RGB, (b) ground truth, and (c) class names with available samples.

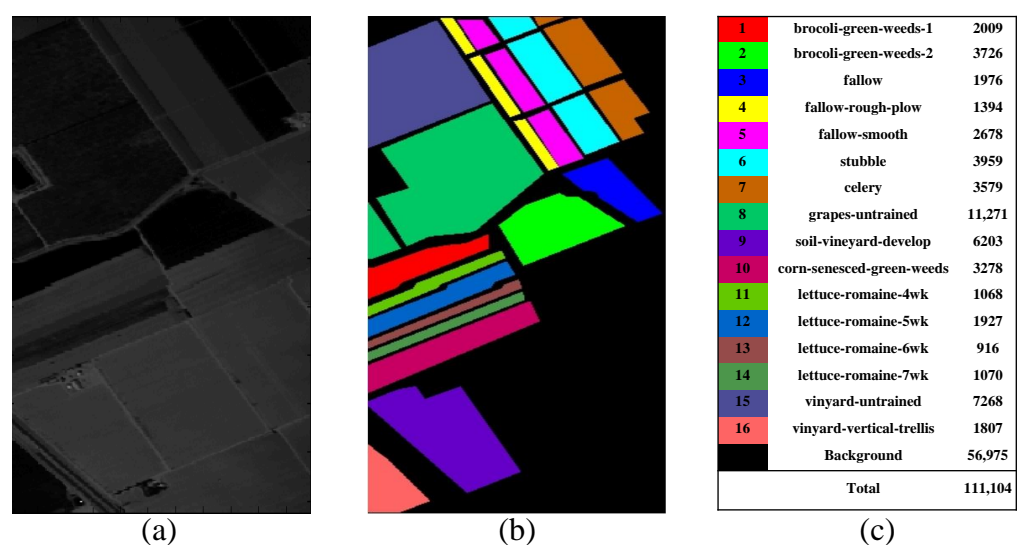


Figure 9. Salinas imagery: (a) color composite with RGB, (b) ground truth, and (c) class names with available samples.

4.2. Evaluation Measures

In the experiments, the available data of these datasets were randomly divided into two parts, a small part for training and the rest for testing. Whether the training samples or the testing samples were arranged according to the pixels, whose size was in $1 \times p_x$ (p_x is selected as 80 in this paper). Each pixel can be treated as a feature of a certain class, corresponding to a unique label and classified by the classifier stuck to the Discriminator. Tables 6–8 list the sample numbers for the training and testing of three datasets.

Table 6. The samples for each category of training and testing for the Indian Pines dataset.

Number	Class	Train	Test	Total
1	Alfalfa	3	43	46
2	Corn-notill	71	1357	1428
3	Corn-mintill	41	789	830
4	Corn	11	226	237
5	Grass-pasture	24	459	483
6	Grass-trees	36	694	730
7	Grass-pasture-mowed	3	25	28
8	Hay-windrowed	23	455	478
9	Oats	3	17	20
10	Soybean-notill	48	924	972
11	Soybean-mintill	122	2333	2455
12	Soybean-clean	29	564	593
13	Wheat	10	195	205
14	Woods	63	1202	1265
15	Buildings-Grass-Trees-Drives	19	367	386
16	Stone-Steel-Towers	4	89	93
Total		510	9739	10,249

Table 7. The samples for each category of training and testing for the Pavia University dataset.

Number	Class	Train	Test	Total
1	Asphalt	66	6565	6631
2	Meadows	186	18,463	18,649
3	Gravel	20	2079	2099
4	Trees	30	3034	3064
5	Painted metal sheets	13	1333	1345
6	Bare Soil	50	4979	5029
7	Bitumen	13	1317	1330
8	Self-Blocking Bricks	36	3646	3682
9	Shadows	9	938	947
Total		423	42,353	42,776

Table 8. The samples for each category of training and testing for the Salinas dataset.

Number	Class	Train	Test	Total
1	Broccoli_green_weeds_1	20	1989	2009
2	Broccoli_green_weeds_2	37	3689	3726
3	Fallow	19	1960	1976
4	Fallow_rough_plow	13	1381	1394
5	Fallow_smooth	26	2652	2678
6	Stubble	39	3920	3959
7	Celery	35	3544	3579
8	Grapes_untrained	112	11,159	11,271
9	Soil_vineyard_develop	62	6141	6203
10	Corn_senesced_green_weeds	32	3236	3278
11	Lettuce_romaine_4wk	10	1058	1068
12	Lettuce_romaine_5wk	19	1908	1927
13	Lettuce_romaine_6wk	9	909	916
14	Lettuce_romaine_7wk	10	1060	1070
15	Vineyard_untrained	72	7196	7268
16	Vineyard_vertical_trellis	18	1789	1807
Total		533	53,596	54,129

Taking the phenomenon of “foreign matter of the same spectrum in surface cover” [15,43] into consideration, the average accuracy was reported to evaluate the experiment results quantitatively. Meanwhile, the proposed method was contrasted with the comparative method by three famous indexes, i.e., overall accuracy (OA), average accuracy (AA) and kappa coefficient (KA) [44], which can be denoted as below:

$$OA = \frac{\text{sum}(\text{diag}(M))}{\text{sum}(M)}, \quad (13)$$

$$AA = \frac{\text{mean}(\text{diag}(M) ./ (\text{sum}(M, 2)))}{\text{sum}(M, 2)}, \quad (14)$$

$$\text{Kappa} = \frac{OA - \frac{\text{sum}(M, 1) \times \text{sum}(M, 2)}{(\text{sum}(M))^2}}{1 - \frac{\text{sum}(M, 2)}{(\text{sum}(M))^2}}, \quad (15)$$

where m represents the number of land cover categories and $M \in R^{(m \times n)}$ symbolizes the confusion matrix of the classification results. Then, $\text{diag}(M) \in R^{m \times 1}$ comes to be a vector of diagonal elements in M , $\text{sum}() \in R^1$ proves to be the sum of all elements of matrices, where $(, 1)$ means each column and $(, 2)$ means each row. Finally, the $\text{mean}() \in R^1$ describes the mean value of all elements along with the $./$, which implies the element-wise division.

4.3. Experimental Setting

In this section, for the sake of verifying the effectiveness of CSSVGAN, several classical hyperspectral classification methods such as SVM [45], Multit-3DCNN [46], SS3DCNN [47], SSRN [15] and certain deep generative algorithms like VAE, GAN and some jointed VAE-GAN models like the CVA²E [33] and the semisupervised variational generative adversarial networks (SSVGAN) [34] were used for comparison.

To ensure the fairness of the comparative experiments, the best hyperparameter settings were adopted for each method based on their papers. All experiments were executed on the NVIDIA GeForce GTX 2070 SUPER GPU with a memory of 32 GB. Moreover, Adam [48] was used as the optimizer with an initial learning rate of 1×10^{-3} for Generator and 1×10^{-4} for Discriminator, and the training epoch was set to 200.

4.4. Experiments Results

All experiments in this paper were randomly selected train samples from the labeled pixels, and the accuracies of three datasets were reported to two decimal places in this chapter.

4.4.1. Experiments on the IP Dataset

The experimental test on IP Dataset was performed to evaluate the proposed CSSV-GAN model quantitatively with other methods for HSIC. For the labeled samples, 5% of each class was randomly selected for training. The quantitative evaluation of various methods is shown in Table 9, which describes the classification accuracy of different categories in detail, as well as the indicators including OA, AA and kappa for different methods. The best value is marked in dark gray.

Table 9. The classification results for the IP dataset with 5% training samples.

Num/IP	ClassName	SVM	M3DCNN	SS3DCNN	SSRN	VAE	GAN	CVA ² E	SSVGAN	CSSVGAN
1	Alfalfa	58.33	0.00	0.00	100.00	100.00	60.29	67.35	90.00	50.00
2	Corn-notill	65.52	34.35	39.61	89.94	73.86	90.61	90.61	90.81	90.61
3	Corn-mintill	73.85	17.83	33.75	93.36	97.66	92.97	93.56	94.77	92.30
4	Corn	58.72	9.40	10.41	82.56	100.00	93.48	98.91	98.47	95.29
5	Grass-pasture	85.75	33.46	32.33	100.00	82.00	98.03	96.48	97.72	87.27
6	Grass-trees	83.04	90.68	82.10	95.93	91.98	93.69	95.69	90.49	97.60
7	Grass-pasture-mowed	88.00	0.00	0.00	94.73	0.00	0.00	100.00	82.76	93.33
8	Hay-windrowed	90.51	87.70	85.29	95.68	100.00	97.22	98.70	99.34	91.71
9	Oats	66.67	0.00	0.00	39.29	100.00	50.00	100.00	100.00	100.00
10	Soybean-notill	69.84	37.46	51.53	79.08	92.88	80.04	94.77	86.52	94.74
11	Soybean-mintill	67.23	57.98	64.71	88.80	92.42	94.40	88.56	98.51	95.75
12	Soybean-clean	46.11	21.08	21.26	94.43	84.48	80.84	81.30	84.03	84.48
13	Wheat	87.56	83.33	41.18	99.45	100.00	77.63	98.99	94.20	100.00
14	Woods	85.95	83.00	85.04	95.26	98.38	97.62	98.19	87.67	98.04
15	Buildings-GT-Drives	73.56	34.16	31.43	97.18	100.00	91.35	95.63	83.49	97.08
16	Stone-Steel-Towers	100.00	0.00	0.00	93.10	98.21	96.55	98.72	90.14	91.30
	OA(%)	72.82	53.54	56.23	91.04	90.07	91.01	92.48	91.99	93.61
	AA(%)	75.02	34.48	33.57	89.92	73.82	82.47	85.69	89.49	91.16
	Kappa(%)	68.57	45.73	49.46	89.75	88.61	89.77	91.40	90.91	93.58

First of all, although SVM achieves good exactitude, there is still a certain gap from the exact classification because of the IP dataset containing high texture spatial information, which leads to bad performance. Secondly, some conventional deep learning methods (such as M3DCNN, SS3DCNN) does not perform well in some categories due to the limitation of the number of training samples. Thirdly, the algorithms with jointed spectral-spatial feature extraction (like SSRN, etc.) show a better performance, which indicate a necessity to combine spectral information and spatial information for HSIC. Moreover, it is obvious that the generated virtual samples by VAE tend to be fuzzy and cannot guarantee similarities with the real data. While GAN lacks sampling constraints, leading to the low quality of the generated samples. Contrasted with these two deep generative models, CSSVGAN overcomes their shortcomings. Finally, compared with CVA²E and SSVGAN, the two latest jointed models published in IEEE, CSSVGAN uses dual-branch feature extractions and crossed interactive method, which proves that these manners are more suitable for HSIC works. It can increase the diversity of samples and promote the generated data more similar to the original.

Among these comparative methods, CSSVGAN acquires the best accuracy in OA, AA and kappa, which improves by 2.57%, 1.24% and 3.81% respectively, at least. In addition, although all the methods have different degrees of misclassification, CSSVGAN achieves perfect accuracy in “Oats” “Wheat” and so on. The classification visualizations on the Indian Pines of comparative experiments are shown in Figure 10.

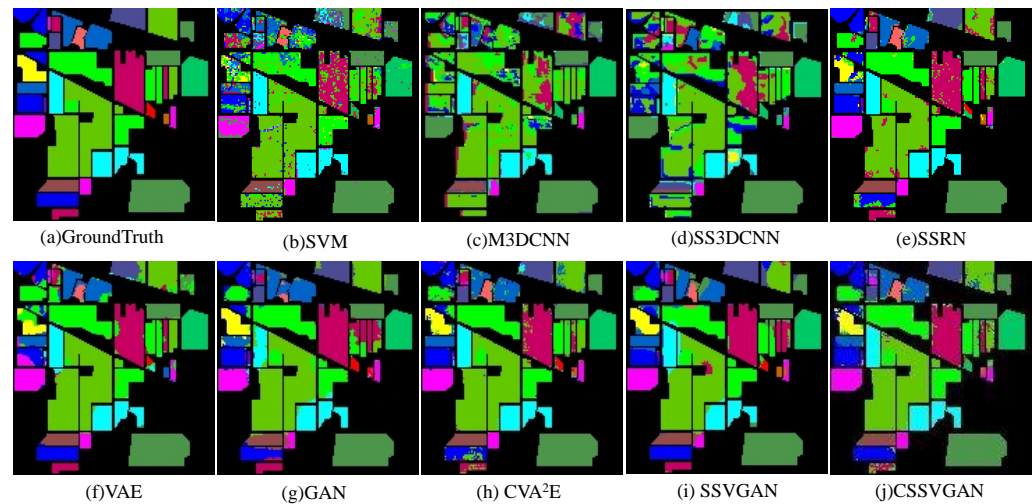


Figure 10. Classification maps for the IP dataset with 5% labeled training samples: (a) GroundTruth (b) SVM (c) M3DCNN (d) SS3DCNN (e) SSRN (f) VAE (g) GAN (h) CVA²E (i) SSVGAN (j) CSSVGAN.

From Figure 10, it can be seen that CSSVGAN reduces the noisy scattering points and effectively improves the regional uniformity. That is because CSSVGAN can generate more realistic images from diverse samples.

4.4.2. Experiments on the PU Dataset

Differ from the IP dataset experiments, 1% labeled samples were selected for training and the rest for testing. Table 10 shows the quantitative evaluation of each class in comparative experiments. The best accurate value is marked in dark gray to emphasize, and the classification visualizations on the Pavia university are shown in Figure 11.

Table 10. The classification results for the PU dataset with 1% training samples.

Num/PU	ClassName	SVM	M3DCNN	SS3DCNN	SSRN	VAE	GAN	CVA ² E	SSVGAN	CSSVGAN
1	Asphalt	86.21	71.39	80.28	97.24	87.96	97.13	86.99	90.18	98.78
2	Meadows	90.79	82.38	86.38	83.38	86.39	96.32	96.91	94.90	99.89
3	Gravel	67.56	17.85	33.76	93.70	93.46	58.95	87.91	78.30	97.70
4	Trees	92.41	80.24	87.04	99.51	93.04	78.38	97.86	95.11	98.91
5	Painted metal sheets	95.34	99.09	99.67	99.55	99.92	93.50	96.86	96.70	99.70
6	Bare Soil	84.57	25.37	51.71	96.70	98.15	99.64	98.48	98.00	99.42
7	Bitumen	60.87	47.14	49.60	98.72	75.06	52.11	75.25	86.92	99.47
8	Self-Blocking Bricks	75.36	44.69	68.81	86.33	62.53	84.06	72.50	91.17	96.03
9	Shadows	100.00	88.35	97.80	100.00	82.86	42.57	97.13	82.53	99.14
	OA(%)	86.36	68.43	76.59	89.27	85.08	87.58	91.97	92.93	99.11
	AA(%)	83.68	53.00	64.14	95.01	73.45	83.58	89.32	87.83	98.47
	Kappa(%)	81.76	56.60	68.80	85.21	79.58	83.67	85.64	90.53	98.83

Table 10 shows that, as a non-deep learning algorithm, SVM has been able to improve the classification result to 86.36%, which is wonderful to some extent. VAE shows good performance in the training of the “Painted metal sheets” class but low accuracy in the “Self-blocking bricks” class, which leads to the “fuzzy” phenomenon of a single VAE network

in the training of individual classes. SSRN achieves a completely correct classification in “shadows,” but it lost to the CSSVGAN overall. In the index of OA results, CSSVGAN improved 12.75%, 30.68%, 22.52%, 9.83%, 14.03%, 11.53%, 7.14% and 6.18% respectively and in the index of Kappa results, CSSVGAN improved 17.07%, 42.23%, 30.03%, 13.62%, 19.25%, 15.16%, 13.19% and 8.3% respectively compared with the other eight algorithms.

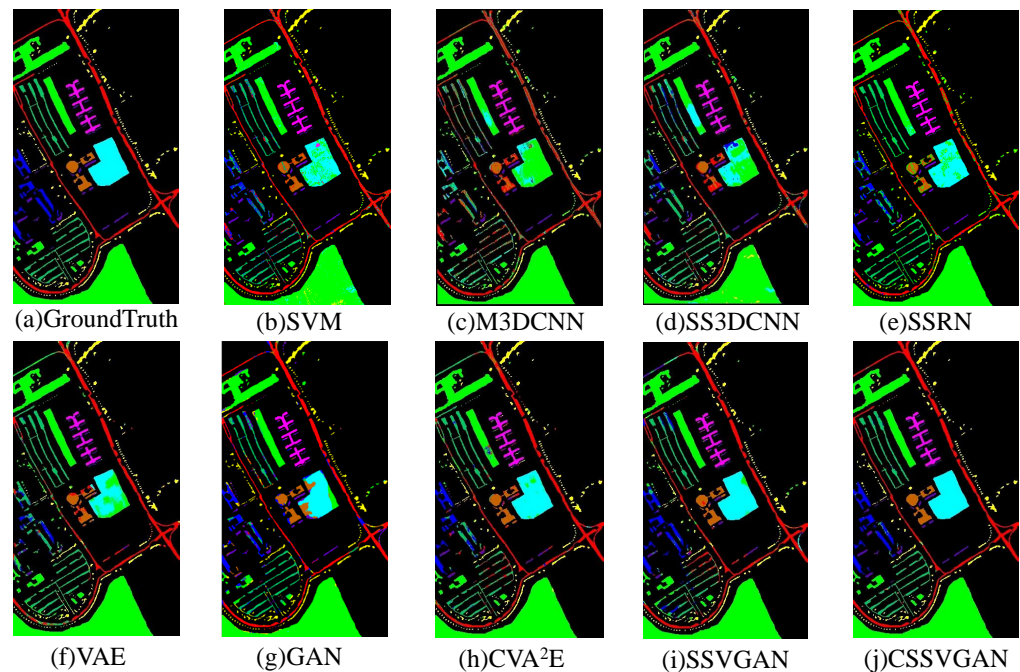


Figure 11. Classification maps for the PU dataset with 1% labeled training samples: (a) GroundTruth (b) SVM (c) M3DCNN (d) SS3DCNN (e) SSRN (f) VAE (g) GAN (h) CVA²E (i) SSVGAN (j) CSSVGAN.

In Figure 11, the proposed CSSVGAN has better boundary integrity and better classification accuracy in most of the classes because the Encoder can ensure the diversity of samples, the Generator can promote the authenticity of the generated virtual data, and the Discriminator can adjust the overall framework to obtain the optimal results.

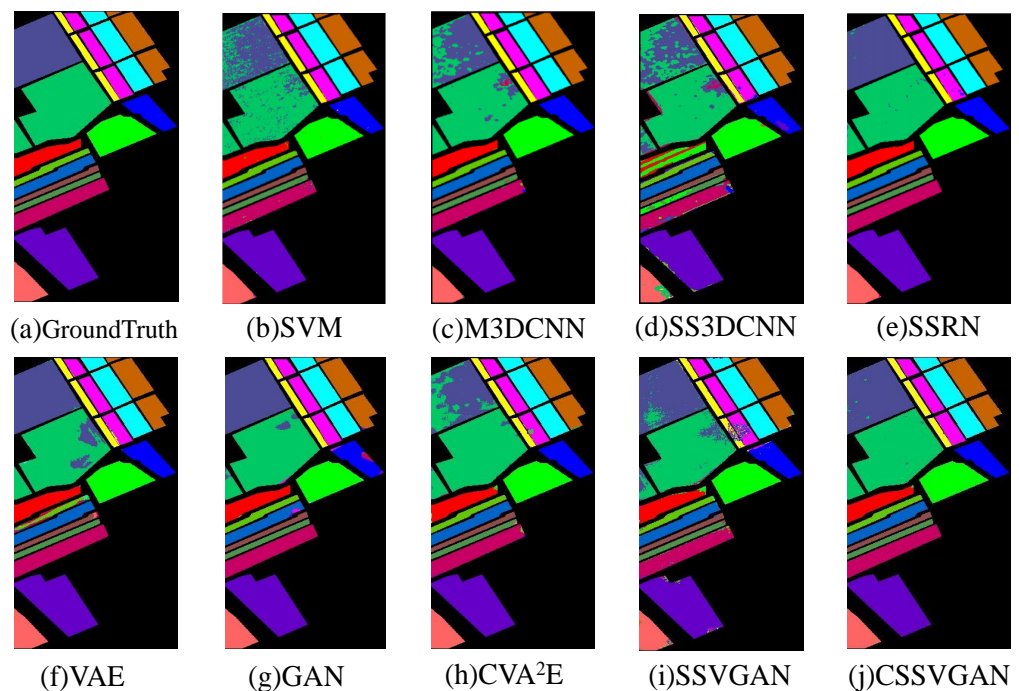
4.4.3. Experiments on the SA Dataset

The experimental setting on the Salinas dataset is the same as PU. Table 11 shows the quantitative evaluation of each class in various methods with dark gray to emphasize the best results. The classification visualization of the comparative experiments on Salinas is shown in Figure 12.

Table 11 shows that in the index of OA, AA and Kappa, CSSVGAN improved 0.57%, 1.27% and 0.62% at least compared with others. Moreover, it has a better performance in the “brocoli-green-weeds-1” and “stubble” class with a test accuracy of 100%. For the precisions of other classes, although SSRN, VAE or SSRN prevails, CSSVGAN is almost equal to them. It can be seen that CSSVGAN has smoother edges and the minimum misclassification in Figure 12, which further proves that the proposed CSSVGAN can generate more realistic virtual data according to the diversity of extracted features of samples.

Table 11. The classification results for the SA dataset with 1% training samples.

Num/SA	ClassName	SVM	M3DCNN	SS3DCNN	SSRN	VAE	GAN	CVA ² E	SSVGAN	CSSVGAN
1	Broccoli_green_weeds_1	99.95	94.85	56.23	100.00	97.10	100.00	100.00	100.00	100.00
2	Broccoli_green_weeds_2	98.03	65.16	81.56	98.86	97.13	62.32	99.34	97.51	99.92
3	Fallow	88.58	40.61	92.40	99.40	100.00	99.78	100.00	93.74	98.99
4	Fallow_rough_plow	99.16	97.04	95.63	96.00	98.68	93.91	99.76	91.88	99.35
5	Fallow_smooth	90.38	89.31	95.08	95.11	99.26	97.67	99.30	94.08	99.08
6	Stubble	99.64	95.64	98.78	99.69	99.24	94.36	90.53	99.31	100.00
7	Celery	98.58	75.75	98.90	99.32	97.98	98.93	99.39	99.54	99.66
8	Grapes_untrained	77.58	65.28	81.87	89.16	96.55	96.87	89.36	93.57	92.79
9	Soil_vineyard_develop	99.50	96.04	96.20	98.33	99.74	89.66	89.85	98.53	99.56
10	Corn_sg_weeds	95.01	44.82	84.13	97.67	96.79	91.71	95.71	92.44	97.81
11	Lettuce_romaine_4wk	94.00	44.66	79.64	96.02	100.00	87.95	96.82	91.62	97.76
12	Lettuce_romaine_5wk	97.40	36.69	96.19	98.45	90.89	98.73	100.00	99.42	99.32
13	Lettuce_romaine_6wk	95.93	12.17	91.50	99.76	99.87	100.00	91.97	96.78	99.67
14	Lettuce_romaine_7wk	94.86	79.53	66.83	97.72	95.83	94.14	100.00	95.85	99.71
15	Vineyard_untrained	79.87	40.93	69.11	83.74	88.09	57.33	85.41	85.17	91.75
16	Vineyard_vertical_trellis	98.76	57.78	85.09	97.07	99.61	97.32	97.00	99.11	99.66
	OA(%)	90.54	66.90	85.14	94.40	96.43	86.97	95.06	94.60	97.00
	AA(%)	94.20	56.78	78.89	96.65	95.87	92.17	97.08	95.50	98.35
	Kappa(%)	89.44	62.94	83.41	93.76	96.03	85.50	94.48	94.00	96.65

**Figure 12.** Classification maps for the SA dataset with 1% labeled training samples: (a) GroundTruth (b) SVM (c) M3DCNN (d) SS3DCNN (e) SSRN (f) VAE (g) GAN (h) CVA²E (i) SSVGAN (j) CSSVGAN.

5. Discussions

5.1. The Ablation Experiment in CSSVGAN

Taking IP, PU and SA datasets as examples, the frameworks of ablation experiments are shown in Figure 13, including NSSNCSG, SSNCSCG and SSNCDCG.

As shown in Table 12, compared with NSSNCSG, the OA of CSSVGAN on IP, PU and SA datasets increased by 1.02%, 6.90% and 4.63%, respectively.

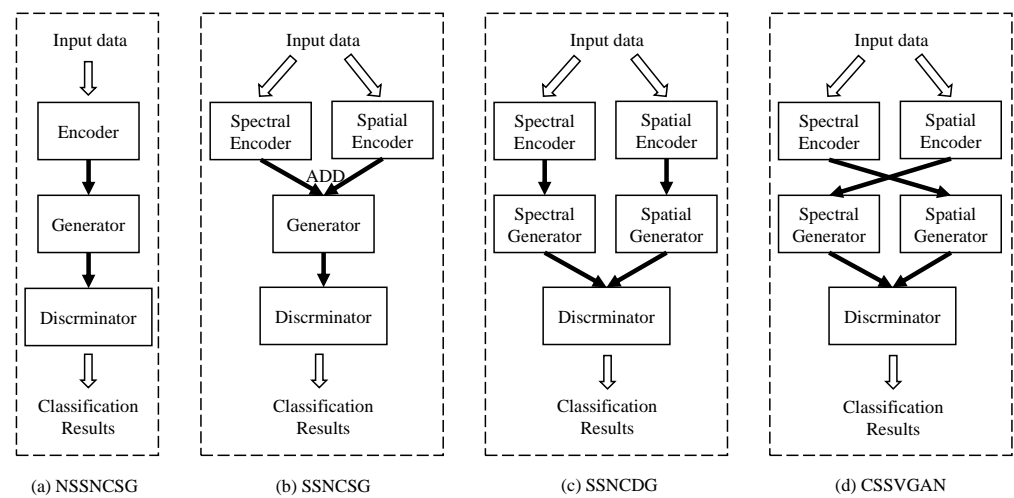


Figure 13. The frameworks of ablation experiments: (a) NSSNCSG (b) SSNCSG (c) SSNCDG (d) CSSVGAN.

Table 12. The OA(%) of Ablation experiments.

Name	Dual Branch	Crossed Interaction	Single Generator	Double Generator	IP	PU	SA
NSSNCSG	×	×	✓	×	92.59	92.21	92.07
SSNCSG	✓	×	✓	×	92.62	98.54	96.61
SSNCDG	✓	×	×	✓	92.36	98.67	96.26
CSSVGAN	✓	✓	×	✓	93.61	99.11	97.00

It shows that the effect of using dual-branch special-spatial feature extraction is better than not using it because the distributions of spectral and spatial features are not identical, and a single Encoder cannot handle this complex situation. Consequently, using the dual-branch variational Encoder can increase the diversity of samples. Under the constraint of KL divergence, the distribution of latent variables is more consistent with the distribution of real data.

Contrasted with SSNCSG, the OA index on IP, PU and SA datasets increase by 0.99%, 1.07% and 0.39% respectively, which means that the result of utilizing the crossed interactive method is more effective, and further influences that the crossed interactive double Generator can fully learn the spectral and spatial information and generate spatial and spectral virtual samples in higher qualities.

Finally, a comparison is made between SSNCDG and CSSVGAN, where the latter can better improve the authenticity of virtual samples by crossed manner. All these contributions of both the Encoder and the Generator put forward higher requirements to the Discriminator, optimizing Discriminator's ability to identify the true or false data and further achieve the final classification results more accurately.

5.2. Sensitivity to the Proportion of Training Samples

To verify the effectiveness of the proposed CSSVGAN, three datasets were taken as examples. The percentage of training samples was changed for each class from 1% to 9% at 4% intervals and added 10%. Figures 14–16 shows the OAs of all the comparative algorithms with various percentages of training samples.

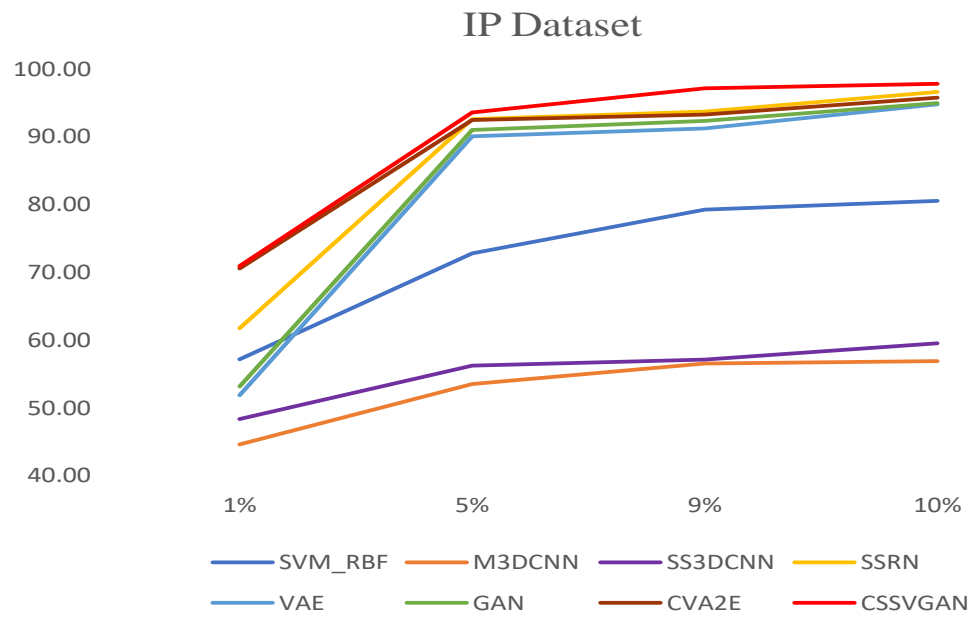


Figure 14. Sensitivity to the Proportion of Training Samples in IP dataset.

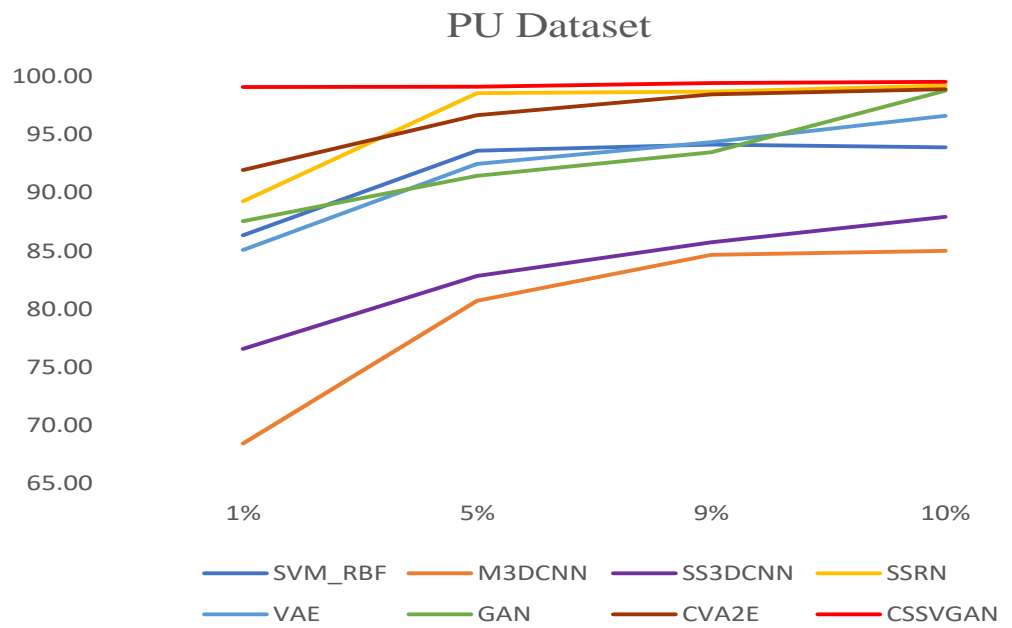


Figure 15. Sensitivity to the Proportion of Training Samples in PU dataset.

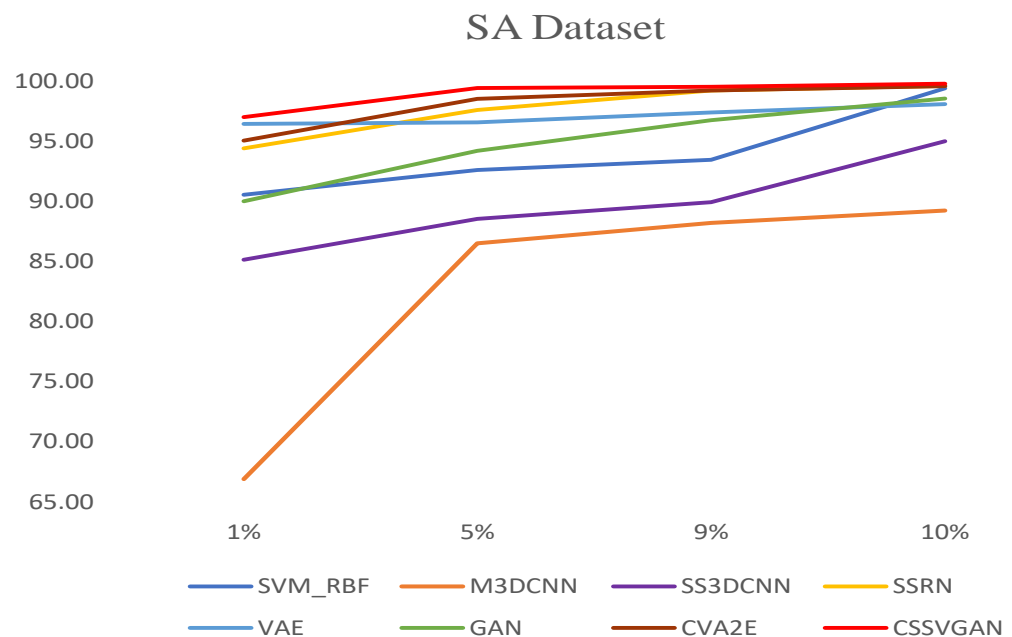


Figure 16. Sensitivity to the Proportion of Training Samples in SA dataset.

It can be seen that the CSSVGAN has the optimal effect in each proportion of training samples in three datasets because CSSVGAN can learn the extracted features interactively, ensure diverse samples and improve the quality of generated images.

5.3. Investigation of the Proportion of Loss Function

Taking the IP dataset as an example, the proportion σ_i ($i = 1, 2, \dots, 5$) of loss functions and other super parameters of each module are adjusted to observe their impact on classification accuracy and the results are recorded in Table 13 (the best results are marked in dark gray). Moreover, the learning rate is also an important factor, which will not be repeated here. It can be obtained by experiments that using 1×10^{-3} for Generator and 1×10^{-4} for Discriminator are the best assignments.

Table 13. Investigation of the proportion σ_i of loss functions in IP dataset with 5% training samples.

σ_1	σ_2	σ_3	σ_4	σ_5	IP_Result
0.25	0.25	0.15	0.15	0.2	91.88
0.3	0.3	0.15	0.15	0.1	91.23
0.3	0.3	0.1	0.1	0.2	92.87
0.35	0.35	0.05	0.05	0.2	92.75
0.35	0.35	0.1	0.1	0.1	93.61

Analyzing Table 13 reveals that when $\sigma_1 \sim \sigma_5$ are set as 0.35, 0.35, 0.1, 0.1 and 0.1 respectively, the CSSVGAN model achieves the best performance. Under this condition, the Encoder can acquire the maximum diversity of samples. The Discriminator is able to realize the most accurate classification, and the Generator is capable of generating the images most like the original data. Moreover, the best parameter combination $\sigma_1 \sim \sigma_5$ on the SA dataset is similar to IP, while in the PU dataset, they are set as 0.3, 0.3, 0.1, 0.1 and 0.2.

6. Conclusions

In this paper, variational generative adversarial network with crossed spatial and spectral interactions (CSSVGAN) is proposed for HSIC. It mainly consists of three modules: a dual-branch variational Encoder, a crossed interactive Generator, and a Discriminator

stuck with a classifier. From the experiment results of these three datasets, it showed that CSSVGAN can outperform the other methods in the index of OA, AA and Kappa in its abilities because of the dual-branch and the crossed interactive manners. Moreover, using the dual-branch Encoder can ensure the diversity of generated samples by mapping spectral and spatial information into different latent spaces, and utilizing the crossed interactive Generator can imitate the highly correlated spatial and spectral characteristics of HSI by exploiting the consistency of learned spectral and spatial features. All these contributions made the proposed CSSVGAN give the best performance in three datasets. In the future, we will develop towards to realize lightweight generative models and explore the application of the jointed “Transformer and GAN” model for HSIC.

Author Contributions: Conceptualization, Z.L. and X.Z.; methodology, Z.L., X.Z. and L.W.; software, Z.L., X.Z., L.W. and Z.X.; validation, Z.L., F.G. and X.C.; writing—original draft preparation, L.W. and X.Z.; writing—review and editing, Z.L., Z.X. and F.G.; project administration, Z.L. and L.W.; funding acquisition, Z.L. and L.W. All authors read and agreed to the published version of the manuscript.

Funding: This research was funded by the Joint Funds of the General Program of the National Natural Science Foundation of China, Grant Number 62071491, the National Natural Science Foundation of China, Grant Number U1906217, and the Fundamental Research Funds for the Central Universities, Grant No. 19CX05003A-11.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Publicly available datasets were analyzed in this study, which can be found here: http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes, latest accessed on 29 July 2021.

Acknowledgments: The authors are grateful for the positive and constructive comments of editor and reviewers, which have significantly improved this work.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; nor in the decision to publish the results.

References

- Chen, P.; Jiao, L.; Liu, F.; Zhao, J.; Zhao, Z. Dimensionality reduction for hyperspectral image classification based on multiview graphs ensemble. *J. Appl. Remote Sens.* **2016**, *10*, 030501. [CrossRef]
- Shi, G.; Luo, F.; Tang, Y.; Li, Y. Dimensionality Reduction of Hyperspectral Image Based on Local Constrained Manifold Structure Collaborative Preserving Embedding. *Remote Sens.* **2021**, *13*, 1363. [CrossRef]
- Atzberger, C. Advances in remote sensing of agriculture: Context description, existing operational monitoring systems and major information needs. *Remote Sens.* **2013**, *5*, 949–981. [CrossRef]
- Sun, Y.; Wang, S.; Liu, Q.; Hang, R.; Liu, G. Hypergraph embedding for spatial-spectral joint feature extraction in hyperspectral images. *Remote Sens.* **2017**, *9*, 506. [CrossRef]
- Abbate, G.; Fiumi, L.; De Lorenzo, C.; Vintila, R. Evaluation of remote sensing data for urban planning. Applicative examples by means of multispectral and hyperspectral data. In Proceedings of the 2003 2nd GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas, Berlin, Germany, 22–23 May 2003; pp. 201–205.
- Yuen, P.W.; Richardson, M. An introduction to hyperspectral imaging and its application for security, surveillance and target acquisition. *Imaging Sci. J.* **2010**, *58*, 241–253. [CrossRef]
- Tan, K.; Zhang, J.; Du, Q.; Wang, X. GPU parallel implementation of support vector machines for hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 4647–4656. [CrossRef]
- Li, J.; Bioucas-Dias, J.M.; Plaza, A. Semisupervised hyperspectral image classification using soft sparse multinomial logistic regression. *IEEE Geosci. Remote Sens. Lett.* **2012**, *10*, 318–322.
- Tan, K.; Hu, J.; Li, J.; Du, P. A novel semi-supervised hyperspectral image classification approach based on spatial neighborhood information and classifier combination. *ISPRS J. Photogramm. Remote Sens.* **2015**, *105*, 19–29. [CrossRef]
- Gao, Q.; Lim, S.; Jia, X. Hyperspectral image classification using convolutional neural networks and multiple feature learning. *Remote Sens.* **2018**, *10*, 299. [CrossRef]
- Chen, Y.; Jiang, H.; Li, C.; Jia, X.; Ghamisi, P. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6232–6251. [CrossRef]
- Zhang, B.; Zhao, L.; Zhang, X. Three-dimensional convolutional neural network model for tree species classification using airborne hyperspectral images. *Remote Sens. Environ.* **2020**, *247*, 111938. [CrossRef]

13. Chen, Y.C.; Lei, T.C.; Yao, S.; Wang, H.P. PM2.5 Prediction Model Based on Combinational Hammerstein Recurrent Neural Networks. *Mathematics* **2020**, *8*, 2178. [[CrossRef](#)]
14. Nezami, S.; Khoramshahi, E.; Nevalainen, O.; Pölonen, I.; Honkavaara, E. Tree species classification of drone hyperspectral and rgb imagery with deep learning convolutional neural networks. *Remote Sens.* **2020**, *12*, 1070. [[CrossRef](#)]
15. Zhong, Z.; Li, J.; Luo, Z.; Chapman, M. Spectral-spatial residual network for hyperspectral image classification: A 3-D deep learning framework. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 847–858. [[CrossRef](#)]
16. Xu, Y.; Zhang, L.; Du, B.; Zhang, F. Spectral-spatial unified networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5893–5909. [[CrossRef](#)]
17. Liu, G.; Gao, L.; Qi, L. Hyperspectral Image Classification via Multiattribute Correlation Adaptive Representation. *Remote Sens.* **2021**, *13*, 1253. [[CrossRef](#)]
18. Zhao, W.; Du, S. Spectral-spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4544–4554. [[CrossRef](#)]
19. Roy, S.K.; Krishna, G.; Dubey, S.R.; Chaudhuri, B.B. HybridSN: Exploring 3-D–2-D CNN feature hierarchy for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2019**, *17*, 277–281. [[CrossRef](#)]
20. Belwalkar, A.; Nath, A.; Dikshit, O. Spectral-Spatial Classification of Hyperspectral Remote Sensing Images Using Variational Autoencoder and Convolution Neural Network. In Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Dehradun, India, 20–23 November 2018.
21. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *arXiv* **2014**, arXiv:1406.2661v1.
22. Liu, X.; Gherbi, A.; Wei, Z.; Li, W.; Cheriet, M. Multispectral image reconstruction from color images using enhanced variational autoencoder and generative adversarial network. *IEEE Access* **2020**, *9*, 1666–1679. [[CrossRef](#)]
23. Su, Y.; Li, J.; Plaza, A.; Marinoni, A.; Gamba, P.; Chakravorty, S. DAEN: Deep autoencoder networks for hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 4309–4321. [[CrossRef](#)]
24. Makhzani, A.; Shlens, J.; Jaitly, N.; Goodfellow, I.; Frey, B. Adversarial autoencoders. *arXiv* **2015**, arXiv:1511.05644.
25. Bao, J.; Chen, D.; Wen, F.; Li, H.; Hua, G. CVAE-GAN: Fine-grained image generation through asymmetric training. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2745–2754.
26. He, Z.; Liu, H.; Wang, Y.; Hu, J. Generative adversarial networks-based semi-supervised learning for hyperspectral image classification. *Remote Sens.* **2017**, *9*, 1042. [[CrossRef](#)]
27. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
28. Chen, X.; Duan, Y.; Houthoofd, R.; Schulman, J.; Sutskever, I.; Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Kyoto, Japan, 16–21 October, 2016; pp. 2180–2188.
29. Feng, J.; Feng, X.; Chen, J.; Cao, X.; Zhang, X.; Jiao, L.; Yu, T. Generative adversarial networks based on collaborative learning and attention mechanism for hyperspectral image classification. *Remote Sens.* **2020**, *12*, 1149. [[CrossRef](#)]
30. Zhan, Y.; Hu, D.; Wang, Y.; Yu, X. Semisupervised hyperspectral image classification based on generative adversarial networks. *IEEE Geosci. Remote Sens. Lett.* **2017**, *15*, 212–216. [[CrossRef](#)]
31. Feng, J.; Yu, H.; Wang, L.; Cao, X.; Zhang, X.; Jiao, L. Classification of hyperspectral images based on multiclass spatial-spectral generative adversarial networks. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5329–5343. [[CrossRef](#)]
32. Zhu, L.; Chen, Y.; Ghamisi, P.; Benediktsson, J.A. Generative adversarial networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5046–5063. [[CrossRef](#)]
33. Wang, X.; Tan, K.; Du, Q.; Chen, Y.; Du, P. CVA2E: A conditional variational autoencoder with an adversarial training process for hyperspectral imagery classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 5676–5692. [[CrossRef](#)]
34. Wang, H.; Tao, C.; Qi, J.; Li, H.; Tang, Y. Semi-supervised variational generative adversarial networks for hyperspectral image classification. In Proceedings of the IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 9792–9794.
35. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
36. Kullback, S.; Leibler, R.A. On information and sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86. [[CrossRef](#)]
37. Wu, C.; Wu, F.; Wu, S.; Yuan, Z.; Liu, J.; Huang, Y. Semi-supervised dimensional sentiment analysis with variational autoencoder. *Knowl. Based Syst.* **2019**, *165*, 30–39. [[CrossRef](#)]
38. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein GAN. *arXiv* **2017**, arXiv:1701.07875.
39. Mao, X.; Li, Q.; Xie, H.; Lau, R.Y.; Wang, Z.; Paul Smolley, S. Least squares generative adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2794–2802.
40. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2223–2232.
41. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved techniques for training gans. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 2234–2242.
42. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434.

43. Imani, M.; Ghassemian, H. An overview on spectral and spatial information fusion for hyperspectral image classification: Current trends and challenges. *Inf. Fusion* **2020**, *59*, 59–83. [[CrossRef](#)]
44. Sun, H.; Zheng, X.; Lu, X.; Wu, S. Spectral–spatial attention network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 3232–3245. [[CrossRef](#)]
45. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [[CrossRef](#)]
46. He, M.; Li, B.; Chen, H. Multi-scale 3D deep convolutional neural network for hyperspectral image classification. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3904–3908.
47. Li, Y.; Zhang, H.; Shen, Q. Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sens.* **2017**, *9*, 67. [[CrossRef](#)]
48. Loshchilov, I.; Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv* **2016**, arXiv:1608.03983.