

Article

Adaptive Weighting Feature Fusion Approach Based on Generative Adversarial Network for Hyperspectral Image Classification

Hongbo Liang ^{1,2} , Wenxing Bao ^{1,2,*}  and Xiangfei Shen ^{1,2} 

¹ School of Computer Science and Engineering, North Minzu University, Yinchuan 750021, China; 20187187@stu.nmu.edu.cn (H.L.); 20177197@stu.nmu.edu.cn (X.S.)

² The Key Laboratory of Images & Graphics Intelligent Processing of State Ethnic Affairs Commission, North Minzu University, Yinchuan 750021, China

* Correspondence: baowenxing@nun.edu.cn

Abstract: Recently, generative adversarial network (GAN)-based methods for hyperspectral image (HSI) classification have attracted research attention due to their ability to alleviate the challenges brought by having limited labeled samples. However, several studies have demonstrated that existing GAN-based HSI classification methods are limited in redundant spectral knowledge and cannot extract discriminative characteristics, thus affecting classification performance. In addition, GAN-based methods always suffer from the model collapse, which seriously hinders their development. In this study, we proposed a semi-supervised adaptive weighting feature fusion generative adversarial network (AWF²-GAN) to alleviate these problems. We introduced unlabeled data to address the issue of having a small number of samples. First, to build valid spectral-spatial feature engineering, the discriminator learns both the dense global spectrum and neighboring separable spatial context via well-designed extractors. Second, a lightweight adaptive feature weighting component is proposed for feature fusion; it considers four predictive fusion options, that is, adding or concatenating feature maps with similar or adaptive weights. Finally, for the mode collapse, the proposed AWF²-GAN combines supervised central loss and unsupervised mean minimization loss for optimization. Quantitative results on two HSI datasets show that our AWF²-GAN achieves superior performance over state-of-the-art GAN-based methods.

Keywords: generative adversarial networks; hyperspectral image classification; adaptive weighting feature fusion; semi-supervised deep learning



Citation: Liang, H.; Bao, W.; Shen, X. Adaptive Weighting Feature Fusion Approach Based on Generative Adversarial Network for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 198. <https://doi.org/10.3390/rs13020198>

Received: 12 December 2020

Accepted: 4 January 2021

Published: 8 January 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

Due to hyperspectral images (HSIs) contain hundreds of narrow and consecutive spectral bands, which enrich surface semantic information of remote sensing images [1–3], accurate interpretation of HSI ground materials has received significant attention from the machine learning and remote sensing communities [4]. With their abundant spectral signatures and high-resolution spatial context, hyperspectral image data provide powerful technical support for the applications of urban road monitoring [5], crop pest control [6], and environmental protection [7]. Classification is the cornerstone of these HSI applications.

Hyperspectral image classification aims at assigning a unique identifiable target to each pixel. Many recent studies have demonstrated that supervised deep learning methods can alleviate the challenges of high-dimensional non-linear characteristics in HSIs, and promote classification performance [8–11]. However, suffering from “the curse of the dimension,” if less labeled data are available, the Hughes phenomenon [12] will be observed with the increase in the number of trainable parameters. At present, three challenges still prevent deep learning methods from supplying precise and effective pixel-wise HSI classification maps. First, the redundant spectral characteristics of HSI pixels make

conventional optical imagery analysis unusable if we want to extract discriminative features for hyperspectral interpretation. Second, traditional deep learning methods employ a set of spectral–spatial filter banks to represent advanced features of HSIs. However, how to effectively encode spatial and spectral information is a challenging task. Third, the generalization of deep learning models is constrained by their shortage of labeled pixels, which gives rise to insufficient classification accuracy. In this study, we analyze these challenges and offer promising suggestions to mitigate them.

The first challenge, i.e., the redundant hyperspectral characteristic of HSI pixels, originates from abundant spectral signatures and the high-similarity neighboring spatial context. Traditional models aim at employing feature engineering to explain semantic features of spectral bands. For example, Hu et al. [13] directly use the convolutional neural network (CNN) to learn the spectral domain; Chen and Zhao et al. [14,15] adopt dimensionality reduction methods to extract the principal components for training deep CNNs. Nevertheless, these methods do not possess the potential feature representation ability of deep learning frameworks. Other scholars have introduced local neighboring spatial information, and achieved promising results. Mei et al. [16] incorporate the spatial context by band-wise means and standard deviations in a neighboring HSI cuboid. Li et al. [17] utilize pairs of neighboring pixels to extract spatial semantic features, and predict the final land-cover category with a majority voting strategy. Such research highlights the universality of HSI interpretation in the deep learning community. However, the aforementioned approaches focus only on statistics of spatial features, and overlook spectral purity characteristics.

The second challenge is derived from the complexity of HSI distribution, and can be interpreted as a “matter of the same spectrum in surface cover” and the “the same objects but different spectrum in surface cover” [18]. Multiple works adopt neighboring spatial prior knowledge to improve CNNs and enrich spectral semantic information. For example, Li et al. [19] applied three-dimensional (3D) CNNs for accurate HSI classification. The approach in [20] produced a rough segmentation map by extracting CNN-based spectral–spatial features from a 3D hyperspectral cuboid. Other recent studies have shown that there are two main ways to learn a spectral–spatial representation for HSI classification. The first way is to build an explicit engineering framework of the two constituent features. For instance, Liang et al. [21] proposed a superpixel based sparse auto-encoder to extract manifold feature for classification, while Liu et al. [22] designed a superpixel-guided layer-wise embedding CNN for remote sensing image classification. In the second paradigm, spectral–spatial features were implicitly learned from 3D homogeneous areas with annotations for the target pixels [23–26]. Zhong et al. [23] learned a set of spectral–spatial residual filter banks to extract continuous HSI features. Furthermore, Wang et al. [24] attempted to interpret HSI paths through dense convolutional filter banks, and obtained promising results. Zhu et al. [25] proposed an improved deep convolutional capsule neural network (Conv-CapsNet), which considered the pixel position attributes of HSIs. In addition, Cui et al. [26] utilized a multiscale pyramid to capture different spatial contexts of neighboring information for HSI classification. All the above methods improved the representation capacity of HSIs in learning spectral–spatial combinations. However, the joint spectral–spatial filter banks that introduced spatial context limit the contribution of each individual component to HSIs. Existing studies show that weighting fusion may be able to address this: Zhang et al. [27] applied a predictive weighting decision to highlight the validity of each component.

For the third challenge, obtaining a sufficient amount of labeled HSI data usually involves sustaining a higher cost, and can result in many technical issues. Deep learning models have become irreplaceable in mainstream HSI classification methods because of their representation ability. Several scholars suggest that these models require a large amount of data for generalization. For example, Chen and Li et al. [14,17] extended the training data by adding noise and pixel pairs. It is worth mentioning that in contrast with traditional visual images [28], which contain hundreds of categories, HSIs actually include

much fewer land-cover categories that need to classify. Furthermore, several scenes of HSIs are extremely lacking in training samples. Therefore, the theory that deep learning models require massive data for training is not suitable for HSIs. The required amount of unlabeled data for HSIs remains a topic of discussion. Several studies aimed at semi-supervised learning employed a small number of labeled HSI samples and a large number of unlabeled HSI samples for training. Mnih et al. [29] transmitted labels from annotated HSI pixels to non-annotated ones via multilayer neural networks. Fang et al. [30] adopted a semi-supervised double branch convolutional network with a resampling strategy for training sufficiently. Furthermore, Hu et al. [31] applied shape adaptive neighboring information to select valuable unlabeled data. Although such studies have obtained accurate classification results, these results may have been obtained from homogeneous areas with features with high spatial relatedness, rather than being achieved by the deep learning model.

To address these difficulties, two common semi-supervised frameworks—generative models and graph-based models—have been applied [32,33]. For instance, Wan et al. [32] constructed a graph neural network (GNN) with multiscale superpixels to reduce the calculation complexity accompanying HSIs. However, the local spatial regions built with superpixel methods cannot reconstruct the pixel-wise class boundaries. In addition, the utilization of unlabeled data only exists in homogeneous areas. If the patched neighboring HSI cuboid is small enough, the number of valuable unlabeled pixels is limited. Recently, generative adversarial networks (GANs) for HSI classification have attracted extensive attention in regards to small samples [33–39]. Specifically, Zhu et al. [36] built GANs with CNNs for HSI classification. Zhong et al. [37] combined graph models with a semi-supervised GAN to alleviate the challenge of limited labeled data, and refined HSI boundaries. However, this method applied two models for the solution, which is not a good end-to-end framework. Moreover, Wang et al. [40] proposed an adaptive dropblock and applied it to GANs to extract effective HSI pixels. Nevertheless, the model only took the first three principal component analysis channels of HSIs, and could not maximum spectral advantage.

Several reports have pointed out that high-quality generated samples are key to promoting the discriminator for HSI classification. However, the high-dimensional HSI spectral bands present a variety of non-linear characters in a highly spatial distribution, and thus it is difficult to reconstruct real hyperspectral cuboids through spectral–spatial engineering. Radford et al. [41] suggested to applying the transposed convolutions and convolutions without pooling layers instead of fully connected layers to construct generators and discriminators in GANs. Most GAN-based methods follow this principle, such as HS-GAN [35], and MS-GAN [42]. Although transpose convolutions generate shift-invariant local spectral–spatial information, the parameter sharing makes it unable to highlight the sensitive characteristics during the training process. Furthermore, regularization methods limit the two land-cover categories with similar spectral distributions to be regarded as a single category. Thus, it is necessary to select appropriate spectral and spatial features during the sample generation.

Inspired by [27] and [40], in this study, we build a semi-supervised GAN via an adaptive weighting feature fusion approach (AWF²-GAN) for HSI classification. Considering the limitation of labeled data, the discriminator is extended to double-branch networks to extract global spectral signatures and local spatial contexts. The generator contains fully connected layers for training so as to match the real HSI pixels. Moreover, we proposed an adaptive weighting feature fusion strategy for taking constraint conditions of spectral–spatial combinations in the discriminator into account; this is similar to employing an attention mechanism. By giving different fusion weights to each pixel, the fused spectral–spatial feature can be better expressed. Our fusion module can be configured through employing one of four options: by adding or concatenating spectral and spatial feature maps with similar or adaptive weights. Finally, we take the center loss and mean minimization loss into account for stabilizing GAN training. The main contributions of the paper are summarized as follows.

- (1) A novel GAN-based framework for HSI classification named AWF²-GAN is proposed; it considers an adaptive spectral–spatial combination pattern in the discriminator, and improves the efficiency of discriminative spectral–spatial feature extraction.
- (2) To explore the interdependence of spectral bands and neighboring pixels, the adaptive weighting feature fusion module provides four sets of weighting filter banks to improve performance.
- (3) To alleviate the mode collapse of AWF²-GANs, we jointly optimize the framework by considering both center loss and mean minimization loss.

The remainder of this paper is organized as follows. Section 2 introduces the GANs and the center loss. Section 3 presents the details of the proposed AWF²-GAN classification framework and adaptive weighting feature fusion module. Section 4 evaluates the performance of our method compared with those of other HSI methods. Section 5 provides a discussion of the results, and a conclusion is presented in Section 6.

2. Related Works

2.1. Generative Adversarial Networks

Recently, GANs have attracted significant attention from the visual imaging community, such as for image generation and translation. This stems from the fact that GANs can provide superior representation capacity to reconstruct real data distribution implicitly [43]. A GAN contains two subnets, the generator G and the discriminator D . G attempts to learn the latent mapping from the implicit distribution of the input, and synthesize the data subject to this mapping. D judges whether the input is from the real distribution or the fake one. Generally, G takes the random noise vectors z as the input, and transforms them to synthetic images $X_{fake} = G(z)$. D takes the real images or the output of G as input, and outputs the true probability distribution P . D and G are competitively trained to maximize the log-likelihoods they tend to their considered accurate sources. This is expressed as follows:

$$L_G = E_{z \sim P_z} [\log(1 - D(G(z)))], \quad (1)$$

$$L_D = E_{z \sim P_z} [\log(1 - D(G(z)))] + E_{x \sim P_{data}} [\log D(x)]. \quad (2)$$

Under the alternating optimization, the GAN is trained to balance D and G , and is guided to the Nash equilibrium. Specifically, we freeze the weights of D , and optimize G through the minimization of the Equation (1). Then we fix G , and optimize D through the maximization of the Equation (2). Each iteration of the model forms a confrontation training mode to promote the discriminator D and generator G mutually. When the Nash equilibrium is achieved, G explores the real data distribution and D has enhanced the advanced capacity to distinguish real/fake data and identify the categories labels.

Due to the output restrictions of the discriminator, GANs are not suitable for multilabel image classification. Odena et al. [44] proposed an auxiliary classifier GAN (ACGAN) to solve this limitation, and achieved accurate prediction for HSIs. The architecture of their network is shown in Figure 1.

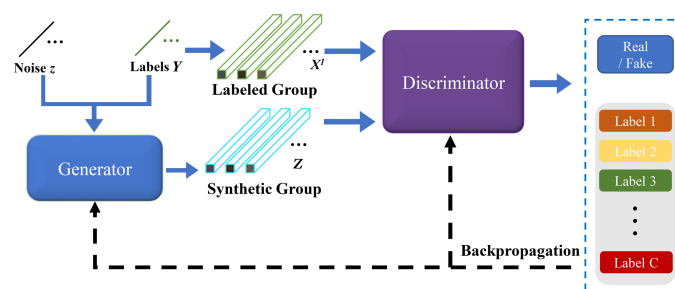


Figure 1. Architecture of ACGAN for HSI classification [44].

In the ACGAN, the generator G receives the embedding vectors from random noise z , and their assigned true labels Y are used as input. The synthetic data generated from

G and the real data with its corresponding annotations are fed into D . Then D predicts the input in terms of categories C with the softmax classifier, and outputs the probability to discriminate the real data from fake data. Therefore, the loss function of the ACGAN consists of the log-likelihood of its identified correct source L_S and the log-likelihood of the corresponding labels L_C , which can be calculated as follows:

$$L_S = E_{z \sim P_z} [\log(1 - D(G(z)))] [0] + E_{x \sim P_{data}} [\log D(x)] [1], \quad (3)$$

$$L_C = E_{x \sim P_{data}} [\log D(x)] [1 : C], \quad (4)$$

where $[0]$ and $[1]$ are the inputs of D derived from the real data and the fake data, respectively. During the training of the ACGAN, D is optimized to maximize $L_C + L_S$, and G is optimized to maximize $L_C - L_S$. Finally, G provides the samples of the desired category and D accurately predicts the classification map.

2.2. Center Loss for Local Spatial Context

In contrast to deep learning methods that learn a set of spectral-spatial filter banks to measure the similarity loss function, methods that measure the center loss function [45] pay more attention to local spatial feature cohesion and intra-class consistency. Cai et al. [46] first introduced the center loss into attention residual networks for HSI classification. They minimized the intra-class distribution while keeping features of different classes separate. Therefore, the similarity measurement of the class centers of deep features can be determined in terms of their corresponding annotations:

$$L_{Cen} = \frac{1}{2} \sum_{i=1}^m \|x_i - c_{y_i}\|_2^2, \quad (5)$$

where m is the size of mini-batch, x_i denotes the i th feature distribution of deep characteristics, and $c_{y_i} \in \mathbb{R}$ indicates the y_i th class center of deep features. Then, the updated equation of c_{y_i} and the gradients of L_{Cen} take the following form:

$$\Delta c_j = \frac{\sum_{i=1}^m \eta(y_i = j) \cdot (c_j - x_i)}{1 + \sum_{i=1}^m \eta(y_i = j)}, \quad (6)$$

$$\frac{\partial L_{Cen}}{\partial x_i} = x_i - c_{y_i}, \quad (7)$$

where $\eta(\cdot)$ denotes the execution condition, which is the updating constraint of c_{y_i} , and j is the current updating class label. If a value of 1 is returned, the condition is satisfied; if 0 is returned, the updating of c_{y_i} is stopped.

3. Methodology

3.1. The Proposed AWF²-GAN Framework

To solve the challenges associated with applying deep learning to HSI classification, we proposed a semi-supervised AWF²-GAN framework. As shown in Figure 2, the discriminator D of the AWF²-GAN is comprised of three parts: spectral filter banks, spectral-spatial filter banks, and an adaptive weighting fusion module. The generator G is built with fully connected filter banks that explore the real distribution and reconstruct the synthetic cuboids implicitly.

Here, we use the Pavia University dataset to illustrate the framework. We suppose an HSI cuboid X contains n pixels $X \in \mathbb{R}^{l_x \times n}$ (l_x denotes the spectral bands of each pixel). Then, the neighboring cuboids are extracted from X in the form of image patches: the labeled group $X^1 = \{x_i^1\} \in \mathbb{R}^{l_x \times s \times s \times n_l}$ and unlabeled group $X^2 = \{x_i^2\} \in \mathbb{R}^{l_x \times s \times s \times n_u}$ are the real inputs of D , where s , n_l , and n_u represent the spatial sizes of neighboring HSI cuboids, and the numbers of labeled and unlabeled samples, respectively. Since each pixel

belongs to HSI cuboid groups in the $\{X_i^1, X_i^2\}$ set, $n = n_l + n_u$. Meanwhile, the synthetic group $Z = \{Z_i\}$ is the fake input of D for training, and is acquired from the generator G by feeding labels y and random noise vectors z into the fully connected filter banks. Actually, D is a dual-branch fusion networks that employs the spectral filter banks to learn the advanced features of spectra, and utilizes the spectral–spatial filter banks for the local spatial semantic representation.

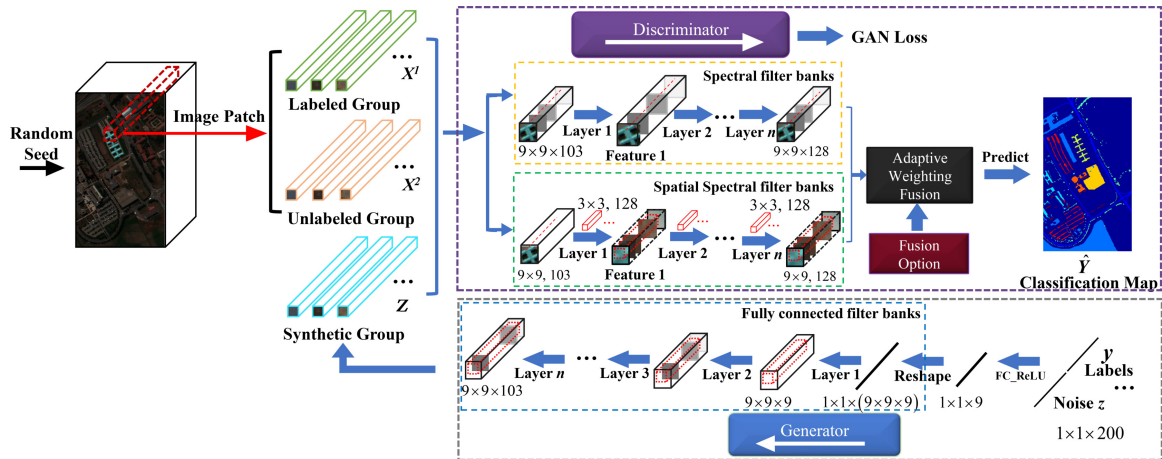


Figure 2. Framework of the AWF²-GAN for HSI classification.

It is worth mentioning that HSI cuboids X have $W \times H$ pixels, and retain redundant spectra l_x . In contrast to image-based classification frameworks, those of patch-based methods select training samples and their neighboring pixels to explore effective local spatial characteristics. Therefore, constructing 3D spectral–spatial cuboids with $s \times s$ image patches increases the potential to generalize models. Even with a small number of HSI samples, such a sampling strategy still retains sufficient trainable parameters. In the following, we utilize $9 \times 9 \times l_x$ neighboring cuboids as inputs in each filter bank, and take tensor volumes to represent outputs, embeddings, and variables in each layer of the AWF²-GAN.

The extracted spectral signatures X_{spc} and spatial contexts X_{spa} are sent into an adaptive weighting feature fusion module, and discriminative fusion features are output. This lightweight module involves feature summing and feature concatenation, and four agile options for feature fusion.

Finally, the combined feature is passed through the softmax operation, and the classification map $\hat{Y} = \{\hat{y}_i\}$ is predicted.

3.2. Adaptive Weighting Feature Fusion Module

In the proposed AWF²-GANs, the generator G and the discriminator D are composed of different sets of filter banks. In D , spectral and spatial filter banks learn the spectral distribution and spatial correlation of the central pixels in neighboring cuboids of HSIs. In this section, to fuse the extracted spectral and spatial features. We commence with the basic fusion methods, i.e., element-wise addition and channel concatenation, to check their effectiveness. Furthermore, we propose a novel weighting fusion method with an adaptive feature weighting strategy, and obtain the relationship between the spectral and spatial feature with a lightweight neural network.

3.2.1. Basic Feature Fusion Modules

Baseline fusion models are employed to combine the spectral and spatial features of HSIs via feature addition and channel concatenation. The straightforward methods generally perform the above operations on the outputs X_{spc} and X_{spa} of the feature extractors. However, if X_{spc} and X_{spa} terms from complex feature engineering, their tensor dimensions are not aligned, and therefore they cannot be fused directly.

Instead, the simple baseline fusion models are built for tensor alignment, and transform the input X_{spc} and X_{spa} into output features G_{spc} and G_{spa} , respectively. For the feature dimensions, the intermediate layer between the inputs and outputs contains a Conv2D-BN-ReLU block. Specifically, we adopt a convolutional operation (Conv2D) to align feature tensors of both X_{spc} and X_{spa} with the same kernel of size 1×1 . Batch normalization [47] and a rectified linear unit (BN-ReLU) are at every convolutional layer of each branch. Then, we acquire the normalization results G_{spc} and G_{spa} , and use these to standardize the training process. Finally, the coarse fusion feature F_{ss} is formed with G_{spc} and G_{spa} via element-wise addition or concatenating along the channel dimension. Figure 3a,b illustrate the architectures of these two basic fusion models.

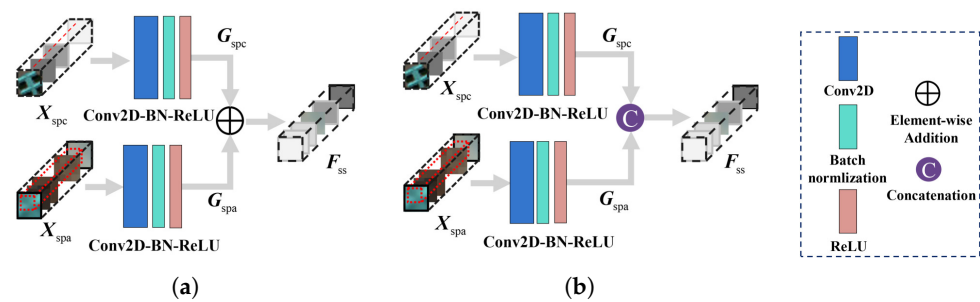


Figure 3. Basic fusion models with (a) element-wise addition and (b) feature concatenation.

The construction of the baseline fusion strategy is motivated by several points. The Conv2D-BN-ReLU block consists of three parts, Conv2D employs a 2D kernel of size 1×1 to construct feature maps through element-wise addition, and we can enhance the correlation between different channels. Furthermore, the block introduces much fewer order of magnitude parameters, and Conv2D share their non-linear weighting coefficients in each feature map. This will not greatly impact feature dimension reduction. BN and ReLU are important for normalization because they standardize the output to avoid either feature becoming dominant, thus encouraging contribution from both of them. Furthermore, BN and ReLU can accelerate the backpropagation of the gradients so as to improve generalization on testing sets.

3.2.2. Fusion Models with Adaptive Feature Weighting

The aforementioned basic fusion models regard both features as having the same tensor shape, and assumes that spectral and spatial features account equally form the fusion feature. In effect, the contribution of different features to each pixel in neighboring cuboids is different. This may be due to the semantic information or spectral content in highly textured areas or homogeneous regions. In particular, the proportion of spectral signatures depends on the abundance of surface materials, and that of spatial contexts stems from the material composition within the local neighborhood. Considering this, we propose an adaptive weighting mechanism to measure the importance of each feature to neighboring pixels. The novelty of this mechanism is that different features are predictable and assignable in all cases.

(1) *Feature Adaptive Weighting*: Our adaptive feature weighting strategy is inspired by the multiview fusion mechanism for 3D object monitoring and image interpretation. In this paper, we employ the same fundamental to machining the spectral and spatial features, and consider hyperspectral interpretation from the 3D perspective.

Figure 4 presents the adaptive feature weighting built with multibranch neural networks. In each branch, each layer/operator contains a fully-connected (FC)-ReLU block, which maps the distributed feature representation from coarse fusion to the label space of samples. This initializes the weights of each feature in each pixel. For different branch networks, the feature map of each element contains a different weight matrix, that is, the parameter variation is provided by FC operators that are not shared.

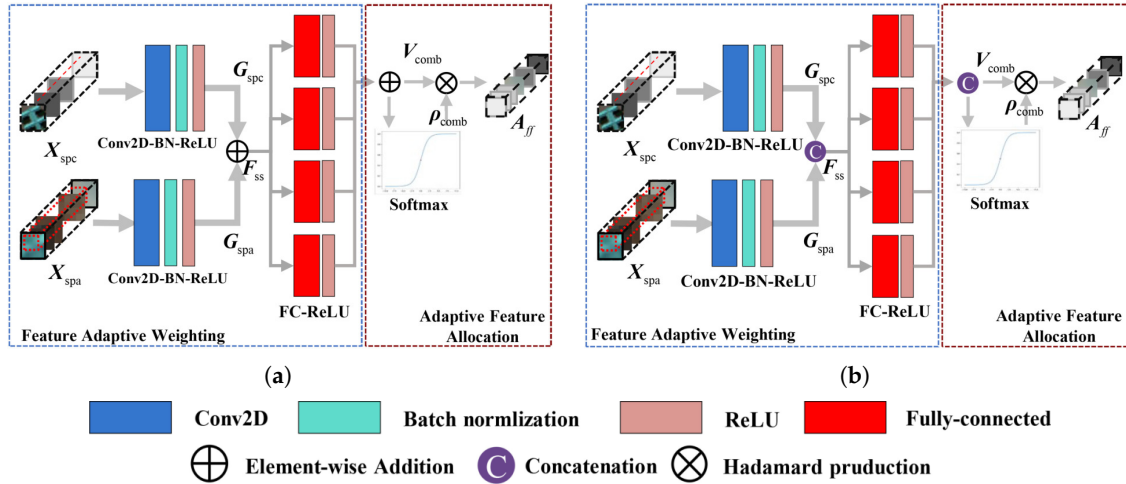


Figure 4. Adaptive weighting fusion models with (a) element-wise addition and (b) feature concatenation.

We suppose that T_f and b_f denote the transform (weight) matrix and the bias of the non-linear layer that directly takes the coarse fusion feature tensors F_{ss} as the input. Subsequently, F_{ss} is transformed into projection vectors v through the FC-ReLU blocks of different branches. In this way, the output of the $k + 1$ th layer in the i th branch can be represented as

$$v_i^{k+1} = R\left(F_{ss}^k T_{f_i}^{k+1} + b_{f_i}^{k+1}\right), \quad i \in N^+, \quad (8)$$

where $R(\cdot)$ indicates a non-linear activation function (which is chosen as the ReLU function in this paper).

We adopt two measures to integrate all feature tensors from various branches into n branches, as shown in Figure 4. Therefore, the weighted feature V_{comb} after branch fusion is formulated by a weighted sum as

$$V_{comb}^{k+2} = \sum_{i=1}^n \left(v_i^{k+1}\right), \quad i \in N^+, \quad (9)$$

or by channel concatenation as

$$V_{comb}^{k+2} = \left[v_1^{k+1}, v_2^{k+1}, \dots, v_i^{k+1}\right], \quad i \in N^+, \quad (10)$$

where “,” denotes the channel concatenation notation.

Finally, the adaptive probabilistic predictive feature transformed with the softmax activation function is as follows

$$\rho_{comb} = E(V_{comb}). \quad (11)$$

Thus, each element of the fusion feature maps have a unique weight.

(2) *Adaptive Feature Allocation*: The fusion architectures that adopt adaptive feature weighting are described in Figure 4. Through these designs, with the predictive feature given by Equation (11), the hybrid predictive matrix can be allocated to branch fusion features by a multiplication to refine the fusion weights as follows:

$$A_{ff} = \rho_{comb} \otimes V_{comb}. \quad (12)$$

Whether through weighted summation or channel concatenation, each feature of each element is assigned a unique score. Thus, whether in the highly textured neighborhood or in hyperspectrally homogeneous regions, the proposed adaptive feature weighting

mechanism always enhances discriminability, and thus improves the capacity for HSI interpretation.

3.3. Details of the AWF²-GAN Architecture

In deep learning methods, the main means to significantly improve HSI classification is to extract discriminative spectral–spatial characteristics. Thus, learning a set of efficient spectral–spatial filter banks is common in HSI feature extraction. Nevertheless, HSIs consist of complex hyperspectral channels and limited labeled trainable data, which constrains the representation of spectral–spatial filter banks. Furthermore, Ref. [27] pointed out that the effects of hyperspectral contents is obviously distinct from those of spatial contexts in highly textured local neighborhoods or homogeneous regions.

We learned a set of global spectral filter banks and neighboring spectral–spatial filter banks for HSI feature extraction, and extended them into the discriminator D of the AWF²-GAN. In addition, a generator G was employed in the AWF²-GAN to synthesize fake data. This added interference factors to D to improve its robustness. Furthermore, the fake data and the real data were passed to D to balance training data through our AWF²-GAN. The architecture of D and G in the AWF²-GAN is illustrated in Figure 5. For the input of D , it takes the raw 3D neighboring cubes of HSIs as input data without feature preprocessing. The sampling strategy of neighboring cubes are randomly selected which centered at pixels in labeled group X^1 . Also, D contains two branches, they treat the input cubes as the independent component for feature extraction from each other. For the input of G , it consists of the random noise vectors with corresponding labels y to generate synthetic data cubes Z .

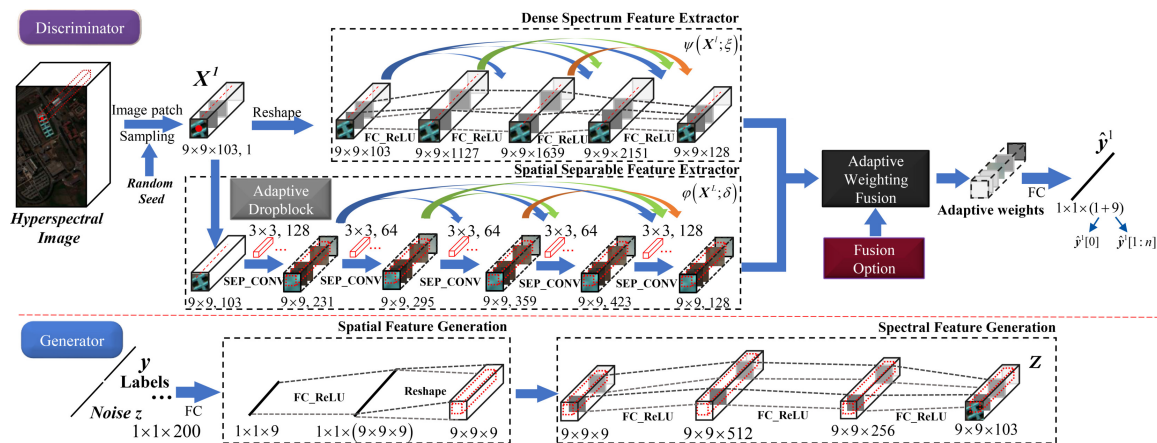


Figure 5. Adaptive weighting feature fusion discriminator (**upper**), consisting of a dense spectrum and spatially separable feature extractors. Their resulting features are fed into an adaptive weighting fusion model, which outputs a vector that indicates whether the data is fake or real and contains categorical probabilities. A generator (**lower**) contains consecutive spatial and spectral feature generation blocks to generate synthetic HSI cuboid Z .

3.3.1. Adaptive Weighting Feature Fusion Discriminator

We built two extractors to consider the composition of spectral and spatial filter banks. One is the dense spectrum feature extractor, and the other is the spatially separable feature extractor. They train the networks on 3D HSI cuboids with dimensions of $9 \times 9 \times l_b$ (where l_b are the bands of the spectra; we take 103 from the Pavia University dataset for illustrative purposes). The spectral signature and spatial contexts are obtained from the above two extractors, and we feed them into the adaptive weighting fusion module to predict fusion features. Then we map the fusion features to sample labels of each neighboring cuboid (or central pixel) through the FC layer with a softmax activation function. The feature extractors are described below.

(1) *Dense Spectrum Feature Extractor*: The main task of the spectral extractor is to capture the global spectral differences of the input neighboring cuboids, and purify the salient features of the central pixel within redundant spectral bands. Here, the obtained feature considers the overall bands of hyperspectral pixels centered at neighboring cuboids. Details of dense spectrum feature extraction are depicted above the discriminator in Figure 5.

To obtain differences between spectra, the dense spectrum feature extractor is designed as a denseNet. We suppose that the network includes l layers/operators, each of which is calculated as an FC-ReLU block with weight matrix H and bias vector b . Then the l th input feature is a dense connection mapping from the 0th to the $l - 1$ th layer. Thus, the l th operation of the extractor $\psi(X^l; \xi)$ and the output feature X_{spc} take the form

$$X^l = [X^0, \psi(X^0; \xi^1), \dots, \psi(X^{l-1}; \xi^l)], \quad (13)$$

$$\psi(X^{l-1}; \xi^l) = R(X^{l-1})H^l + b^l, \quad l \in N^+, \quad (14)$$

$$X_{spc} = \psi(X^l; \xi^{l+1}). \quad (15)$$

This dense feature extractor combines the advantages of spectral channels to maintain an abundance of feature maps. In addition, each FC-ReLU block contains various neural units to delay the gradient descent, which results in a deeper monitoring effect.

(2) *Spatially Separable Feature Extractor*: We also explore the local spatial representation, i.e., we encode the spatial relationship into the input neighboring HSI cuboids. It can be regarded as an auxiliary discriminative factor to the global dense spectral feature. The spatially separable feature extractor has two core components: an adaptive dropblock, and a set of continuous separable feature convolutional filter banks (Sep-Conv blocks). The architecture of this extractor is illustrated below the discriminator in Figure 5.

Specifically, the adaptive dropblock is utilized to regularize the neighboring texture information. This has been verified in [40] to alleviate overparameterization and standardize neurons.

Meanwhile, Sep-Conv considers the local spatial similarity between spectral channels, and begins with a depth convolution with kernels of size $k \times k$. The output channel M of kernels is consistent with that of input image channel N . Then, the point convolutional filter banks of size 1×1 are applied to integrate the local spatial semantics. Again, we combine these semantics with a dense connection. If Sep-Conv blocks have L layers with h filter banks and biases b , the spatial feature extractor architecture $\varphi(X^{L-1}; \delta^L)$ can be represented as follows:

$$X^L = [X^0, \varphi(X^0; \delta^1), \dots, \varphi(X^{L-1}; \delta^L)], \quad (16)$$

$$\varphi(X^{L-1}; \delta^L) = R(X^{L-1}) * H^L + b^L, \quad L \in N^+, \quad (17)$$

$$X_{spa} = \varphi(X^L; \delta^{L+1}), \quad (18)$$

where $*$ is the convolution operation, and δ denotes the trainable parameters from various separable convolution operations. δ can be calculated as

$$\delta = k \times k \times M + M \times N. \quad (19)$$

The specific configurations of D are provided in Table 1. In order to guarantee the generalization of spectral features and avoid “the curse of the dimension,” four FC-ReLU blocks are employed as subcomponents. All the fully connected layers are initialized with “The normal distribution initialization method” [48] and zero bias vectors. Furthermore, we reduce the number of neurons of each subcomponent from 1024 to 128 in turn, and gather significant features from deep spectral mapping and output spectral feature cuboids

with dimensions of $9 \times 9 \times 128$. It is worth mentioning that this shallow structure is sufficient for the neighboring cuboid sizes utilized in our experiments, which is not more than 9×9 . An ablation study to assess the classification performance with respect to the subcomponents architecture is given in Section 5.2.

Table 1. Implementation details of the proposed generator G of the AWF²-GAN.

Block	Input Dimension	Generator G		
		Layer/Operator ¹	Reshape ¹	Number of Neurons ¹
Spatial Feature Generation	$1 \times 1 \times 200$	FC-ReLU	No	9
	$1 \times 1 \times 9$	FC-ReLU	No	$9 \times 9 \times 9$
	$1 \times 1 \times (9 \times 9 \times 9)$	+	Yes	+
Spectral Feature Generation	$9 \times 9 \times 9$	FC-ReLU	No	512
	$9 \times 9 \times 512$	FC-ReLU	No	256
	$9 \times 9 \times 256$	FC-ReLU	No	103
Synthetic HSI cuboid Z	$9 \times 9 \times 103$	Output	+	+

Note: ¹ “+” denotes no operation.

Table 2 lists the generic layer specifications of the spatially separable feature extractor. All the convolution layers in the network include a kernel size of 3×3 , a stride with 1, and padding mode of “same,” and obtain shift-invariant features from patches. ReLU, as the activation function, is executed at each Sep-Conv block. Although deep feature mapping can be achieved by increasing the number of kernel, we found that more kernels may not necessarily improve the classification accuracy. A detailed ablation study on the influence of the number of kernels in Sep-Conv blocks on accuracy is provided in Section 5.2. We also discuss the selection of optimal depths of Sep-Conv blocks in Section 5.3.

Table 2. Specific configurations of the proposed discriminator D of the AWF²-GAN.

Component	Input Dimension ¹	Discriminator D			
		Layer /Operator	Neural Units ²	Kernel ²	Concatenation ²
Dense Spectrum Feature Extractor	$9 \times 9 \times 103$	FC-ReLU	1024	+	No
	$9 \times 9 \times 1127$	FC-ReLU	512	+	Yes
	$9 \times 9 \times 1639$	FC-ReLU	512	+	Yes
	$9 \times 9 \times 2151$	FC-ReLU	128	+	No
Spectral Feature X_{spc}	$9 \times 9 \times 128$	Output	+	+	+
Spatially Separable Feature Extractor	$9 \times 9 \times 103$	Adaptive Dropblock	+	+	No
	$9 \times 9 \times 103$	Sep-CONV	+	$3 \times 3 \times 128$	No
	$9 \times 9 \times 231$	Sep-CONV	+	$3 \times 3 \times 64$	Yes
	$9 \times 9 \times 295$	Sep-CONV	+	$3 \times 3 \times 64$	Yes
	$9 \times 9 \times 359$	Sep-CONV	+	$3 \times 3 \times 64$	Yes
	$9 \times 9 \times 423$	Sep-CONV	+	$3 \times 3 \times 128$	No
Spatial Feature X_{spa}	$9 \times 9 \times 128$	Output	+	+	+
Adaptive Weight Matrix	$(X_{spc} + X_{spa}) / (X_{spc} \cdot X_{spa})$	Adaptive Weighting Feature Fusion	+	+	+
Classification Vectors	$1 \times 1 \times (1 + 9)$	Fully-connection	+	+	+

Note: ¹ We used the neighboring spatial cuboids from the Pavia University dataset for this example ($9 \times 9 \times 103$). ² “+” denotes no operation.

Since the features are extracted from two paths with different physical meanings, a fusion module (discussed in Section 3.2) is added after these two extractors. Four groups

of feature weighting schemes are used to generate adaptive weight matrix A_{ff} . The resulting prediction $\hat{y}_i^1[1:n_y]$ is passed through the softmax layer, and a vector outputs that shows the probabilities of an HSI cuboid belonging to the n_y categories). Again, $\hat{y}_i^1[0]$ indicates the genuineness of the training cuboid. To meet the conditions of softmax, the average pooling and flatten operation are generally executed on the adaptive weight matrix A_{ff} .

3.3.2. Generator with Fully Connected Components

In GANs, generated samples are employed to improve the classification performance of the discriminator. Ref. [37] suggested that the discriminator allows a bad generator as the regularizer to enhance latent representation of hyperspectral cuboids. This is in contradiction with what was proposed by [39]. Actually, the structure of generators has different physical meanings in various stages of GANs. Furthermore, rough noises hinder sample generation, and result in the model collapse.

To this end, we proposed spatial and spectral feature generation blocks built by groups of FC-ReLU blocks, as shown in Figure 5. The reason for this choice is that the fully connected layer provides more trainable parameters, and allocates an unique trainable weight for each element of the feature map. During the first few iterations (nb_epoch), G acts as a regularizer of D , forcing D to approach the real distribution of HSI cuboids. With the optimization of the AWF²-GAN, the fitting ability of FC-ReLU blocks provide more promising results for G . The data generation is divided into two steps: spatial feature generation, and spectral feature generation. The specific configuration of each layer of the AWF²-GAN is given in Table 1.

We suppose a set of Gaussian noise vectors z with their homologous labels y . Then, z is sent into the spatial feature generator to generate the feature tensor with neighborhood space. Next, we use three FC-ReLU blocks to design the spectral feature generator, which gradually approaches the real sample distribution in different neurons, and generates pseudo hyperspectral cube Z .

3.4. Training Loss Functions

The semi-supervised GAN focuses on addressing the limited number of labeled HSI samples. Thus, generator G 's unlabeled set and synthetic set are important regularizers of the discriminator D to improve HSI classification. The objective loss function of the semi-supervised GAN takes the form

$$\begin{aligned} L_{SEMI}(\Omega_D, \Omega_G) &= L_{SUP}(\Omega_D, \Omega_G) + L_{UNSUP}(\Omega_D, \Omega_G) \\ &= L_{SUP}(\Omega_D) + L_G(\Omega_D, \Omega_G) \\ &\quad + L_{D1}(\Omega_D) + L_{D2}(\Omega_D, \Omega_G) \end{aligned} \quad (20)$$

where Ω_D and Ω_G are the hyperparameters of D and G , respectively. L_{SUP} , L_{D1} , and L_{D2} are the supervised and unsupervised entries of D , and the unsupervised entry of G , respectively.

Given the labeled HSI cuboid $X^1 = \{x_i^1\} \in \mathbb{R}^{9 \times 9 \times 103}$ with its corresponding annotation $Y^1 = \{y_i^1\} \in \mathbb{R}^{1 \times (1+n_y)}$, the prediction of D can be formed as $\hat{Y}^1 = D(X^1; \Omega_D)$. Therefore, each entry of L_{SEMI} is formulated as

$$\begin{aligned} L_{SUP}(\Omega_D) &= -E_{X^1 \sim p_{data}} \log D(X^1; \Omega_D)[1:n] \\ &= -E_{X^1 \sim p_{data}} \log \hat{Y}^1[1:n] \end{aligned} \quad (21)$$

$$\begin{aligned} L_{D1}(\Omega_D) &= -E_{X^1 \sim p_{data}} \left(1 - \log D(X^1; \Omega_D)[0] \right) \\ &= -E_{X^1 \sim p_{data}} \log \left(1 - \hat{Y}^1[0] \right) \end{aligned} \quad (22)$$

$$\begin{aligned}
L_{D2}(\Omega_D, \Omega_G) &= -E_{z \sim p_z} \log D(G(z; \Omega_D))[0] \\
&= -E_{z \sim p_z} \log D(Z; \Omega_D)[0] \\
&= -E_{z \sim p_z} \log \hat{Y}^1[0]
\end{aligned} \quad (23)$$

where $\Omega_D = \{\xi_D, \delta_D\}$ indicates the parameters of D , which can be updated from Equations (14) and (17). $\Omega_G = \{\xi_G\}$ denotes the parameters of G . Due to the G consists of FC-ReLU blocks, the parameters of Ω_G can be updated by the Equation (14). $\hat{Y}^1[0]$ is the authenticity of the pixel $x_i \in X^1$, and $\hat{Y}^1[1:n]$ denotes the output vectors of softmax, which is the probability of each category that y_i^1 belongs to.

To stabilize the training of GANs, we introduced the mean minimization loss into the unsupervised entry. This decreases the value and variance in high-dimensional features from the second to last layers of D , and inhibits overfitting. The mean minimization loss takes the form

$$\Omega^* = \operatorname{argmin}_{\Omega} \left(\frac{1}{N} \sum_{i=1}^N \operatorname{average}(f(x_i; \Omega)) \right), \quad (24)$$

where N is the total number of batch samples, x_i is the training sample, and $f(x_i; \Omega)$ indicates the high-dimensional output of the network, which in this paper is the output before the fully connected layer.

For the supervised item, to constrain the intra-class feature distribution, we employed the center loss to guide the feature distribution to shift to the central classes. It can be applied as a regular term before the softmax operation, and can be formulated as

$$n_y^* = \operatorname{argmin}_{n_y} \left(\frac{1}{2} \sum_{i=1}^m \left(F(x_i; y_i^1[1:n_y]) \right) \right), \quad (25)$$

where $F(x_i; y_i^1[1:n_y])$ denotes the central distribution before the softmax operation.

$L_{D1} + L_{D2}$ is also part of the GAN loss for training the generator of the AWF²-GAN, whose corresponding loss function L_G can be formulated as

$$\begin{aligned}
L_G(\Omega_D, \Omega_G) &= -E_{z \sim p_z} \log(1 - D(G(z; \Omega_D))[0]) \\
&= -E_{z \sim p_z} \log(1 - D(Z; \Omega_D)[0]) \\
&= -E_{z \sim p_z} \log(1 - \hat{Y}^1[0])
\end{aligned} \quad (26)$$

The training of the AWF²-GAN involves two alternating steps through RMS or adjacent optimization fashions at each iteration. First, the gradients of D , $-\nabla_{\Omega_D} L_{SEMI}$ are employed to update Ω_D to learn discriminative characteristics of HSIs. Second, the gradients of $-\nabla_{\Omega_D} L_G$ are applied to update Ω_G to improve the adversarial training of the AWF²-GAN.

4. Experimental Results

In this section, two challenging hyperspectral datasets are adopted for classification with the proposed AWF²-GAN. To verify the effectiveness of the AWF²-GAN, several advanced GAN-based HSI algorithms, HS-GAN [35], 3D-GAN [36], SS-GAN [37], and AD-GAN [40] are employed for comparison. Furthermore, to demonstrate the feasibility of the spectral-spatial feature fusion architecture, we also compared AWF²-GANs with various fusion options: F²-Concat (with basic concatenation), F²-Add (with basic addition), AWF²-Concat (with adaptive weighting concatenation) and AWF²-Add (with adaptive weighting addition).

4.1. Experiment Setup

In this paper, two hyperspectral image datasets were employed to evaluate the performance of the proposed model: Indian Pines and Pavia University.

(1) Indian Pines (IN) contains scenes from India acquired by the AVIRIS (Airborne Visible/Infrared Imaging Spectrometer) sensor. It includes 16 categories and images are 145×145 pixels with 20 m ground sample distance. Samples are shown in Figure 6. Since 20 bands were discarded due to atmospheric absorption, there are 200 spectral bands in the range of 400–2500 nm.

(2) Pavia University (UP) includes imagery of Northern Italy obtained by the ROSIS (Reflective Optics System Imaging Spectrometer) sensor. The images are 610×340 pixels with 9 urban land-cover classes, and 1.3 m spatial resolution per pixel (see Figure 7). After abandoning 20 noisy bands, the remaining 103 spectral bands in the range of 430–860 nm are employed for evaluation.

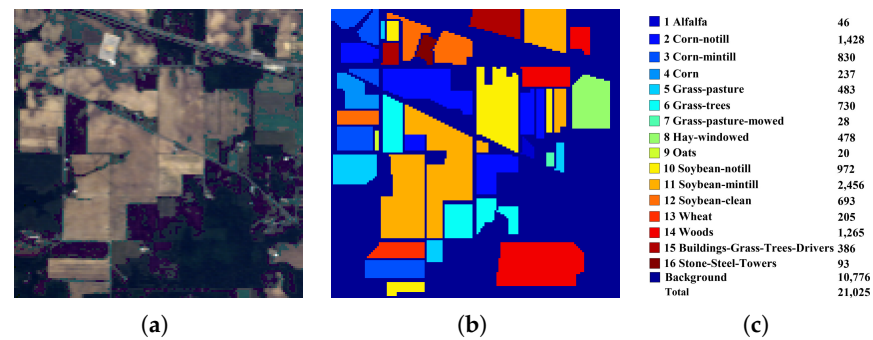


Figure 6. Indian Pines data with (a) color composite with RGB bands (29,19,9), (b) ground truth, and (c) category names with labeled samples.

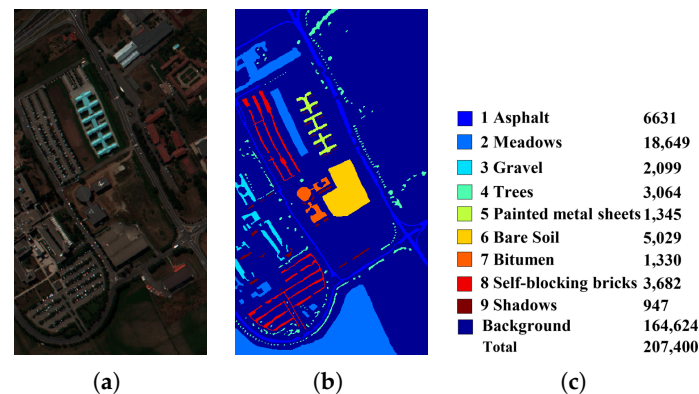


Figure 7. Pavia University imagery: (a) color composite with RGB bands (61,25,13), (b) ground truth, and (c) class names with available samples.

To demonstrate the effectiveness of the AWF²-GAN algorithm, we use five representative HSI classification methods for comparison. Due to the high accuracy and robust performance in terms of classical HSI classification of SVM, the SVM-based HSIs classifier was adopted for the comparison. The EMAP(Extended Multi-Attributes Profiles) with SVM [49] was employed for the spectral-spatial classification, which considered four attributes: (1) a , the area of the regions; (2) d , the length of the diagonal of the box bounding the region; (3) i , the moment of inertia; (4) s , the standard deviation. These extended attribute profiles (EAPs) are obtained by applying thickening and thinning operations to extract spatial information on the first three components of HSI, which were computed by PCA, and retain 20 principal components. For the penalty λ and gamma γ parameters of SVM, the grid-search and 10-fold cross-validation are employed to finetune them. In this experiment, the search range was exponentially growing sequences of λ and γ ($\lambda = 10^{-5}, 10^{-4}, \dots, 10^5$; $\gamma = 10^{-5}, 10^{-4}, \dots, 10^5$). For fair comparison, all GAN-based methods used their optimal parameters. For HS-GAN, the kernel size was set as 1×3 , and the number of training epochs was set to 200. For 3D-GAN, the convolutional kernel

sizes were set according to the literature [36]. For SS-GAN, the spatial size of 3D input was set as 9×9 , and other parameters were suggested in the literature [37]. For AD-GAN, the adaptive dropblock was executed one time in the discriminator.

We also compared the proposed AWF²-GANs with the four feature fusion options. All the parameters of AWF²-GANs are initialized with “He normal distribution initialization method” [48]. For convergence speed and accuracy, the framework is optimized using a RMSProp optimizer with the hyper parameters $\gamma = 0.9$ and learning rate $lr = 0.0005$. Taking account of the model collapse of GANs, lr decay is 0.9 per 50 steps and the total iterations are $nb_epoch = 200$. Each batch size is set to 16 within 9×9 input patches for local spatial contexts. All experiments are implemented with TensorFlow deep learning framework CUDA 9.0, an Intel Xeon Gold 6154 CPU, 256 GB of RAM, and an NVIDIA TITAN V 12GB GPU.

4.2. Experiments on the IN Dataset

The IN dataset contains a complex sample distribution in which trainable labeled samples of categories are unbalanced. Specifically, there are some categories with no more than 50 labeled samples, such as “Alfalfa”, “Grass-pasture-mowed” and “Oats.” Conversely, “Soybean-mintill” has more than 2000 labeled samples. Furthermore, the partial spectral bands between these classes are approximate, like those between “Corn-mintill” and “Soybean-notill”, which have the similar spectrum in range from 100 to 150 spectral bands. This is caused by the phenomenon of “foreign matter of the same spectrum in surface cover” [18], which also reported this viewpoint in literatures [6,23,39]. To this end, the IN dataset can evaluate the stability of the AWF²-GANs. For fair comparison, all compared algorithms use their optimal parameters as suggested in the literature. Besides, to address the model collapse of GAN-based methods, the Monte Carlo sampling strategy is employed to marginalize noise during training.

The first test was a quantitative experiment to evaluate the proposed model and other state-of-the-art methods. In this test, we randomly selected 525 labeled samples to constitute the labeled group X^1 mentioned in Section 3.1; this is a small size that only accounts for approximately 5% of the total labeled samples. Only 3 samples in this class were randomly selected for training if the number of samples of one class was less than 40; meanwhile, only 5 samples were randomly selected for training when the number of samples of one class was more than 40 but no more than 100. For the unlabeled group X^2 , the sampling ratio was equal to X^1 . To assess the performance of various methods, we adopted three evaluation indices: the overall accuracy (OA), average accuracy (AA), and kappa coefficient (κ). Table 3 reports the individual classification performance of various methods, and Figure 8 presents the produced classification maps.

Firstly, although the SVM showed considerable accuracy through EMAPs, that contain high texture spatial information, it still had the worst performance. Secondly, GAN-based classifiers (e.g., HS-GAN, 3D-GAN, SS-GAN, and AD-GAN) provided higher classification accuracy than the SVM. This illustrates that GAN-based deep learning algorithms have a framework suitable for HSI classification. Thirdly, F²-GANs with basic fusion options (F²-Concat and F²-Add) achieved superior accuracies compared to 3D-GAN, proving that spectral-spatial combinations can improve classification performance. Lastly, AWF²-GANs with adaptive feature weighting options (AWF²-Concat and AWF²-Add) achieved the best results, thereby demonstrating that well-designed spectral-spatial networks combining weighting fusion features are suitable for HSI classification. Furthermore, it achieves perfect accuracy on the “Alfalfa,” “Grass pasture,” “Grass-pasture-mowed,” “Hay-windowed,” “Oats,” and “Wheat” categories, three of which contain few labeled training samples. As shown in Figure 8, AWF²-GANs obtained more uniform regions for ground objects in contrast to other algorithms. Moreover, AWF²-Add preserved the most accurate boundary of the “Soybean-mintill” class.

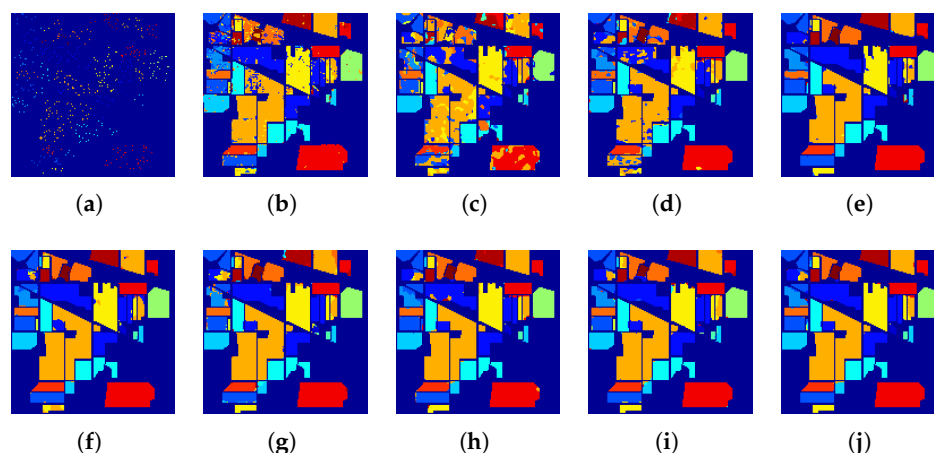


Figure 8. Classification maps for the IN dataset with 525 labeled training samples: (a) training samples (b) SVM (EMAPs) (c) HS-GAN (d) 3D-GAN (e) SS-GAN (f) AD-GAN (g) F²-Concat. (h) F²-Add. (i) AWF²-Concat. and (j) AWF²-Add.

In the second test using the Indian Pines dataset, we verified the sensitivity of the models to different percentages of labeled training samples. We randomly selected 1%, 3%, 5%, 7%, and 10% of labeled samples per class. Table 4 details the overall accuracy for these classification methods with these training samples. Similar to in our first experiment, SVM (EMAPs)'s generalization performance for HSI improved by increasing the percentage of randomly selected training samples. However, the SVM displayed the slowest convergence of all methods. Secondly, GAN-based methods achieved advanced classification accuracies compared to traditional methods. Thirdly, F²-GANs had higher classification accuracies than previous GAN-based algorithms in all tested cases, which proved that spectral-spatial combinations extracted more effective and discriminative features than spectral-spatial representations. Besides, AWF²-GANs with adaptive feature weighting options provided the best performance in comparison to the other methods in all tested cases, which indicated that the method is reliable and robust for HSI classification.

4.3. Experiments Using the UP Dataset

Images from Pavia University (UP) consist of evenly distributed surface materials of 9 urban land-cover classes with highly textured characteristics. In particular, the "Painted metal sheets" and "Gravel" classes contain abundant texture information; it is difficult to predict the complete boundaries of these classes with traditional methods. Furthermore, in contrast to the IN dataset, UP images contain more uniform regions such as in the "Meadows," "Bare Soil," and "Bitumen" classes. Again, the concept of a "different body with the same spectrum" is observed in UP images like those belonging to the "Meadows" and "Bare Soil" classes. Therefore, we compare the performance of different algorithms with high-resolution images. Firstly, 350 samples were randomly selected as training data, which is approximately 0.8% of the labeled pixels. Table 5 displays the quantitative experimental evaluations with four metrics: the individual accuracy, overall accuracy, average accuracy, and κ coefficient. Figure 9 shows the classification results.

Table 3. The overall accuracy (OA), average accuracy (AA), kappa coefficient (κ), and individual class accuracies for the IN dataset with 525 labeled and unlabeled samples for training. The best results are highlighted in bold typeface.

Class	Train (Test)	SVM	HS-GAN	3D-GAN	SS-GAN	AD-GAN	AWF ² -GANs			
							F ² -Con.	F ² -Add.	AWF ² -Con.	AWF ² -Add.
1	5	64.49	20.39	58.48	93.04	100	100	100	100	100
	(41)	± 0.81	± 7.35	± 0.26	± 0.99	± 0.00	± 0.00	± 0.00	± 0.00	± 0.00
2	72	81.06	68.29	90.49	96.1	96.24	96.95	95.53	95.97	98.00
	(1356)	± 3.04	± 0.75	± 0.57	± 0.32	± 0.27	± 0.16	± 0.19	± 0.09	± 0.09
3	42	73.55	62.25	81.14	92.72	93.45	97.03	93.71	95.07	95.14
	(788)	± 0.98	± 1.79	± 0.90	± 0.39	± 0.81	± 0.06	± 0.09	± 0.10	± 0.14
4	12	40.02	60.45	90.72	92.51	94.09	92.81	100	98.75	98.84
	(225)	± 0.13	± 3.57	± 0.87	± 0.59	± 0.15	± 0.30	± 0.00	± 0.18	± 0.25
5	25	81.79	72.34	72.34	99.07	99.44	94.02	99.16	99.45	100
	(458)	± 0.50	± 7.89	± 0.07	± 0.79	± 0.18	± 0.09	± 0.13	± 0.25	± 0.00
6	37	87.88	90.78	90.78	98.95	99.75	96.31	99.34	95.97	98.56
	(693)	± 0.41	± 1.06	± 1.06	± 0.74	± 0.24	± 0.13	± 0.14	± 0.08	± 0.81
7	3	59.13	34.79	94.56	96.27	100	78.51	100	80.09	100
	(25)	± 0.18	± 5.75	± 0.67	± 0.83	± 0.00	± 0.30	± 0.00	± 0.34	± 0.00
8	24	91.13	94.56	99.09	99.58	99.76	98.89	99.73	98.95	100
	(454)	± 0.30	± 1.67	± 0.14	± 0.60	± 0.00	± 0.27	± 0.22	± 0.11	± 0.00
9	3	40.00	34.63	34.63	98.09	100	66.53	84.35	79.77	100
	(18)	± 0.25	± 3.85	± 0.38	± 0.57	± 0.00	± 1.53	± 0.30	± 0.63	± 0.00
10	49	78.76	55.85	93.60	93.05	92.58	91.49	98.03	93.89	96.98
	(923)	± 2.08	± 10.4	± 0.25	± 0.32	± 1.32	± 0.32	± 0.18	± 0.25	± 0.16
11	123	93.59	75.99	91.30	94.68	96.52	94.46	97.89	96.07	97.71
	(2332)	± 1.91	± 1.46	± 0.21	± 0.43	± 0.26	± 0.21	± 0.56	± 0.29	± 0.27
12	30	72.61	50.89	75.99	91.41	97.80	89.94	95.85	96.40	98.65
	(563)	± 0.76	± 1.46	± 0.14	± 0.88	± 0.62	± 0.34	± 0.15	± 0.37	± 0.20
13	11	78.34	82.27	82.27	98.80	100	97.32	100	98.94	100
	(194)	± 0.13	± 1.06	± 1.06	± 1.53	± 0.00	± 0.22	± 0.00	± 0.16	± 0.00
14	64	92.42	93.26	90.89	98.37	98.26	98.66	99.03	98.62	98.35
	(1201)	± 0.39	± 1.30	± 0.13	± 1.58	± 0.33	± 0.37	± 0.13	± 0.38	± 0.05
15	20	62.08	45.75	90.95	94.35	97.65	89.54	97.45	92.81	96.39
	(366)	± 0.13	± 1.35	± 0.13	± 0.37	± 0.24	± 0.43	± 0.15	± 0.29	± 0.31
16	5	28.43	79.89	85.05	96.96	95.08	81.27	88.82	89.46	92.99
	(88)	± 1.36	± 0.81	± 0.74	± 2.5	± 1.09	± 0.53	± 0.38	± 0.59	± 0.07
OA(%)		82.86	77.89	92.28	95.35	96.72	94.90	97.45	96.32	97.83
		± 1.43	± 1.62	± 3.05	± 0.14	± 0.17	± 0.14	± 0.27	± 0.17	± 0.11
AA(%)		70.37	63.94	82.60	95.87	97.54	91.56	96.81	94.47	98.30
		± 2.72	± 2.77	± 4.79	± 0.16	± 0.21	± 0.39	± 0.06	± 0.20	± 0.18
$\kappa \times 100$		80.12	75.56	90.51	94.69	96.26	94.10	97.09	95.8	97.53
		± 1.77	± 0.89	± 0.35	± 0.16	± 0.20	± 0.10	± 0.09	± 0.28	± 0.09

Table 4. The overall accuracy (OA) of classification methods on the IN dataset using various percentages of training samples. The best performances for each test case are in bold.

Sample (Per Class)	Classification Method								
	SVM	HS-GAN	3D-GAN	SS-GAN	AD-GAN	AWF2-GANs			
						F ² - Con.	F ² - Add.	AWF ² - Con.	AWF ² - Add.
110 (1%)	60.38 ± 1.60	59.28 ± 1.47	69.79 ± 0.76	72.81 ± 0.62	75.84 ± 0.55	77.59 ± 0.47	80.17 ± 0.39	80.75 ± 0.35	83.77 ± 0.23
314 (3%)	72.59 ± 1.33	71.87 ± 1.35	83.40 ± 0.57	85.57 ± 0.67	87.67 ± 0.53	92.47 ± 0.64	95.73 ± 0.42	93.85 ± 0.44	95.32 ± 0.17
520 (5%)	81.70 ± 1.39	76.67 ± 1.44	91.71 ± 0.67	94.46 ± 0.46	95.89 ± 0.39	94.49 ± 0.057	96.82 ± 0.28	95.79 ± 0.34	97.53 ± 0.21
726 (7%)	84.77 ± 1.28	78.89 ± 1.16	93.24 ± 0.44	95.77 ± 0.52	96.77 ± 0.48	96.30 ± 0.31	97.83 ± 0.33	97.21 ± 0.30	98.16 ± 0.25
1031 (10%)	88.90 ± 1.01	83.40 ± 1.20	95.79 ± 0.21	97.79 ± 0.13	97.56 ± 0.18	97.33 ± 0.14	98.70 ± 0.12	98.57 ± 0.22	99.07 ± 0.04

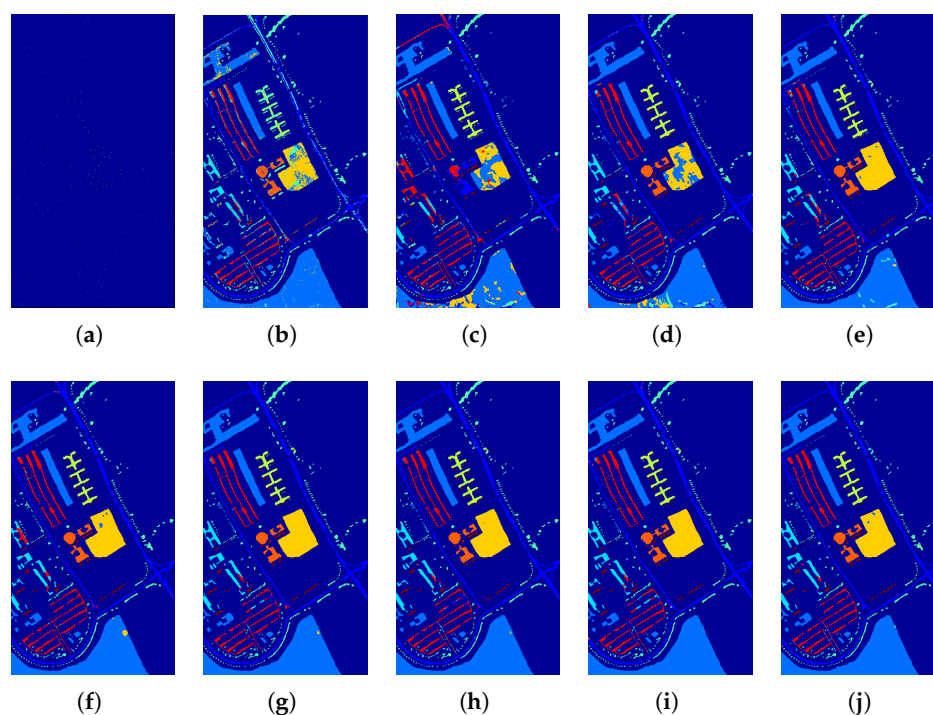


Figure 9. Classification maps for the UP dataset with 350 labeled training samples: (a) training samples (b) SVM (EMAPs) (c) HS-GAN (d) 3D-GAN (e) SS-GAN (f) AD-GAN (g) F²-Concat. (h) F²-Add. (i) AWF²-Concat. and (j) AWF²-Add.

Table 5 leads us to similar conclusions as the results of our quantitative experiments on the IN dataset. AWF²-GANs with various feature fusion options all have a higher OA compared with the other methods. In addition, AWF²-Add has the highest individual classification accuracy for most classes. For instance, it achieved accurate classification for the “Asphalt,” “Painted metal sheets,” and “Bitumen” classes. Moreover, 3D-GAN, SS-GAN, AD-GAN, and AWF²-GANs provided more homogeneous classification maps than the SVM and HS-GAN. Conversely, the HS-GAN classification map contained a large amount of noisy pixels compared to other methods. This demonstrates that the HSI classifier can be significantly improved by taking advantage of spectral–spatial characteristics.

In contrast to the 3D-GAN, SS-GAN, and AD-GAN, the classification maps of AWF²-GANs with various feature fusion options had the most clear boundaries and most uniform regions. This illustrates that spectral–spatial characteristics extracted from spectral–spatial combinations led to greater generalization efficiency than spectral–spatial filter banks.

Table 5. The overall accuracy (OA), average accuracy (AA), kappa coefficient (κ), and individual class accuracies for the UP dataset with 350 labeled and unlabeled samples for training. The best results are highlighted in bold typeface.

Class	Train (Test)	SVM	HS-GAN	3D-GAN	SS-GAN	AD-GAN	AWF ² -GANs			
							F ² -Con.	F ² -Add.	AWF ² -Con.	AWF ² -Add.
1	54	84.53	75.46	96.73	98.97	98.65	96.75	96.80	97.70	100
	(6577)	± 2.38	± 1.53	± 0.49	± 0.14	± 0.15	± 0.29	± 0.01	± 0.19	± 0.00
2	150	95.48	93.57	97.82	99.33	98.96	98.85	98.94	98.89	98.57
	(18499)	± 1.67	± 0.86	± 0.39	± 0.03	± 0.35	± 0.30	± 0.07	± 0.13	± 0.38
3	17	67.50	82.24	95.00	86.40	98.69	93.82	97.68	90.89	99.37
	(2082)	± 0.83	± 4.23	± 0.64	± 0.15	± 0.03	± 0.19	± 0.18	± 0.25	± 0.61
4	25	95.33	85.64	98.87	99.47	99.91	98.00	96.68	97.93	96.48
	(3039)	± 0.10	± 1.13	± 0.59	± 0.10	± 0.05	± 0.14	± 0.19	± 0.51	± 0.38
5	14	51.86	78.98	1.00	99.54	99.92	98.81	99.68	99.73	100
	(1331)	± 0.14	± 5.26	± 0.00	± 0.08	± 0.02	± 0.24	± 0.37	± 0.30	± 0.00
6	41	75.47	70.07	97.09	97.51	97.79	98.49	99.02	99.05	99.80
	(4988)	± 0.05	± 0.72	± 0.46	± 0.19	± 0.22	± 0.26	± 0.13	± 0.14	± 0.30
7	11	80.90	77.67	95.95	95.96	95.63	100	100	97.10	100
	(1319)	± 0.61	± 2.99	± 0.64	± 0.45	± 0.06	± 0.00	± 0.00	± 0.38	± 0.00
8	30	84.19	80.98	89.98	91.73	85.09	94.84	95.98	96.00	97.39
	(3652)	± 0.34	± 0.27	± 0.96	± 0.36	± 0.43	± 0.18	± 0.14	± 0.22	± 0.25
9	8	50.85	90.96	97.78	99.85	100	100	100	99.85	98.62
	(939)	± 0.11	± 0.38	± 0.41	± 0.02	± 0.00	± 0.00	± 0.00	± 0.12	± 0.47
OA(%)		86.26	84.98	97.07	97.25	97.39	98.05	98.24	98.01	98.68
		± 1.19	± 2.33	± 0.19	± 0.15	± 0.09	± 0.24	± 0.16	± 0.19	± 0.22
AA(%)		76.23	81.11	96.99	96.53	97.18	97.68	98.53	97.43	98.81
		± 2.75	± 1.21	± 0.36	± 0.13	± 0.07	± 0.32	± 0.11	± 0.20	± 0.20
$\kappa \times 100$		81.76	80.56	96.49	96.37	96.53	97.31	97.66	97.47	98.12
		± 1.54	± 1.77	± 0.46	± 0.19	± 0.12	± 0.26	± 0.27	± 0.18	± 0.43

To further verify the robustness and practicability of the proposed AWF²-GANs, we employed different training samples from the Pavia University dataset. The labeled training set was generated by randomly selecting 0.1%, 0.2%, 0.4%, 0.8%, and 1% samples per class, and the unlabeled training set was sampled equally to the labeled one. Table 6 presents the OA(%) matrix for each test case. The proposed approach achieved higher classification accuracies with a limited percentage of labeled samples than other methods. For instance, AWF²-Add yielded a 97.31% overall accuracy when only using 0.4% of labeled training samples per class. In contrast to other spectral–spatial classifiers, adaptive weighting fusion features can capture discriminative spectral purity and spatial neighboring contexts, and greatly improve classification accuracy.

Table 6. The overall accuracy (OA) of classification methods on the UP dataset using various percentages of training samples. The best performances for each test case are in bold.

Sample	Classification Method								
	Per Class	SVM	HS-GAN	3D-GAN	SS-GAN	AD-GAN	AWF ² -GANs		
							F ² -Con.	F ² -Add.	AWF ² -Con.
									AWF ² -Add.
48 (0.1%)		67.75 ± 1.54	66.96 ± 1.23	80.12 ± 0.98	81.73 ± 1.01	83.38 ± 0.95	85.75 ± 1.12	87.00 ± 0.97	89.76 ± 0.89
91 (0.2%)		78.33 ± 1.34	75.57 ± 1.19	89.57 ± 1.03	89.98 ± 0.97	91.09 ± 0.89	91.54 ± 0.98	91.97 ± 0.88	93.85 ± 1.16
176 (0.4%)		82.69 ± 1.11	78.87 ± 1.36	93.13 ± 0.88	94.57 ± 0.82	95.38 ± 1.06	95.35 ± 0.61	96.20 ± 0.45	96.54 ± 0.54
347 (0.8%)		86.26 ± 0.97	84.98 ± 1.29	97.07 ± 0.75	97.25 ± 0.79	97.39 ± 0.61	96.90 ± 0.77	97.31 ± 0.48	98.01 ± 0.32
432 (1%)		88.74 ± 1.26	86.99 ± 0.97	97.26 ± 0.65	97.47 ± 0.16	97.64 ± 0.22	97.97 ± 0.13	98.09 ± 0.21	98.26 ± 0.16
									90.74 ± 0.92 94.80 ± 0.77 97.31 ± 0.37 98.68 ± 0.29 98.73 ± 0.09

5. Discussion

5.1. Investigation of Running Time

The training and testing times are investigated to assess the efficiency of the various classification methods. In particular, the SVM with EMAPs, as the classical algorithm for combining spectral–spatial features, was tuned with 10-fold cross validation to identify the optimal parameters. For a fair comparison, we set the same maximum batch size to 16 in each GAN-based method. SS-GAN and AD-GAN, as patch-based classification methods, were trained on 9×9 patches centered at the neighboring cuboids of chosen training pixels. For AWF²-GANs, each feature extractor was constructed with 128 output channels; the other parameters are the same as mentioned above. For testing, 5% of samples per class were randomly selected from the IN dataset, and 1% of samples per class were randomly selected from the UP dataset. Once the number of samples for one class was less than 50, 5 samples of that class were selected for training.

Table 7 lists the training and testing times of the studied algorithms. The training times of AWF²-GANs were 5 to 7 times shorter than those of SS-GAN and AD-GAN. GAN-based methods required more training time because adversarial learning needs to converge. Despite this fact, F²-GANs and AWF²-GANs took approximately 1 s per batch for the IN images. All remaining labeled samples were employed in the testing phase. The results revealed the SS-GANs lasted 2 to 3 times longer than the AWF²-GANs. This could be because the pixel-wise classification GANs with a set of spectral–spatial filter banks will require more time when processing image patches. Although the adjacent patches contain a lot of redundant information, and a large amount of computer calculations, the branch extractors treat the spectral and spatial characteristics of HSIs separately. In contrast, spectral–spatial combination based on the branch fusion strategy reduces the computational complexity, and the processing of the testing phase is faster than that of spectral–spatial filter banks.

Table 7. Training and testing times of various classification methods on the Indian Pines and Pavia University datasets.

Methods	Training Time (s)		Testing Time (s)	
	IndianPines	Pavia University	IndianPines	Pavia University
SVM	1.21 ± 1.05	1.05 ± 0.95	1.11 ± 0.19	1.2 ± 0.23
HS-GAN	465.35 ± 34.54	586.77 ± 25.35	0.37 ± 0.05	0.55 ± 0.11
3D-GAN	1047.62 ± 70.29	798.22 ± 34.46	0.97 ± 0.15	2.35 ± 0.23
SS-GAN	770.89 ± 75.55	535.79 ± 59.66	8.36 ± 0.19	16.44 ± 0.07
AD-GAN	574.24 ± 66.75	409.26 ± 50.99	5.79 ± 0.22	6.76 ± 0.12
F²-Con.	187.02 ± 10.16	165.56 ± 8.52	5.23 ± 0.10	4.83 ± 0.31
F²-Add.	112.53 ± 11.59	99.68 ± 7.93	2.16 ± 0.14	4.91 ± 0.07
AWF²-Con.	185.68 ± 32.67	138.59 ± 21.89	5.23 ± 0.09	10.58 ± 0.05
AWF²-Add.	140.99 ± 25.78	118.365 ± 17.87	3.31 ± 0.12	7.22 ± 0.08

5.2. Kernel Setting and Units Selection for Feature Extraction

The effectiveness of the kernel setting and selection of the number of neurons were evaluated when using AWF²-Add. as the network backbone. For the kernel setting, the kernel size and the number of kernels are important factors impacting the extraction of local spatial features. To explore the discriminative spatially separable feature along the channel dimension, the number of intermediate spatially separable convolutional kernels is between 16 to 128 (where we set three intermediate layers in the spatially separable feature extractor). For the sensitive neighboring area, we considered kernel sizes of 3×3 , 5×5 , and 7×7 . Furthermore, spectral purity analysis allows us to verify the spectral feature utilization. To this end, seven combinations of the number of neurons used for the dense spectrum feature extractor were selected: (128,128,128), (256,128,128), (256,256,256), (512,256,256), (512,512,512), (1024,512,256), and (1024,512,512). Figure 10 illustrates the overall accuracy with 150 training epochs and 500 samples on the IN dataset, and with 350 on the UP dataset.

In the kernel setting phase, the overall accuracy peaks when there are 64 and 128 spatial convolutional kernels for the IN and UP datasets, respectively. In addition, overall accuracy does not improve upon increasing the number of kernels. Figure 10b shows that the kernel size of 3×3 results in the best classification performance on both datasets. For the spectral purity analysis, it can be seen that best numbers of neurons in the three intermediate layers is (1024, 512, 512). Since the neighboring HSI cuboid takes a size of 9×9 , and the proposed extractors are dense connection structures, it is sufficient to capture the discriminative feature mapping of HSIs. Therefore, the generalization ability of AWF²-GANs can be effectively expressed regardless of using smaller convolutional kernel size or less neural units during training. Furthermore, small kernel sizes for training could mitigate overfitting.

5.3. Depths of the Feature Extractors

The generalization ability and stability of HSI classification networks are also subject to the capacities of spectral and spatial feature extractors, i.e., their subcomponent depths. In the feature extraction phase, the dense spectrum and the spatially separable feature extractors play important roles, with various subcomponents FC-ReLU and Sep-Conv blocks used to obtain valid features. For AWF²-GANs, the depths of both extractors were validated from 3 to 5 subcomponents on both datasets, as shown in Figure 11. To maintain the stability of the GAN framework, the depth of the generator was fixed to 4 FC-ReLU blocks for training.

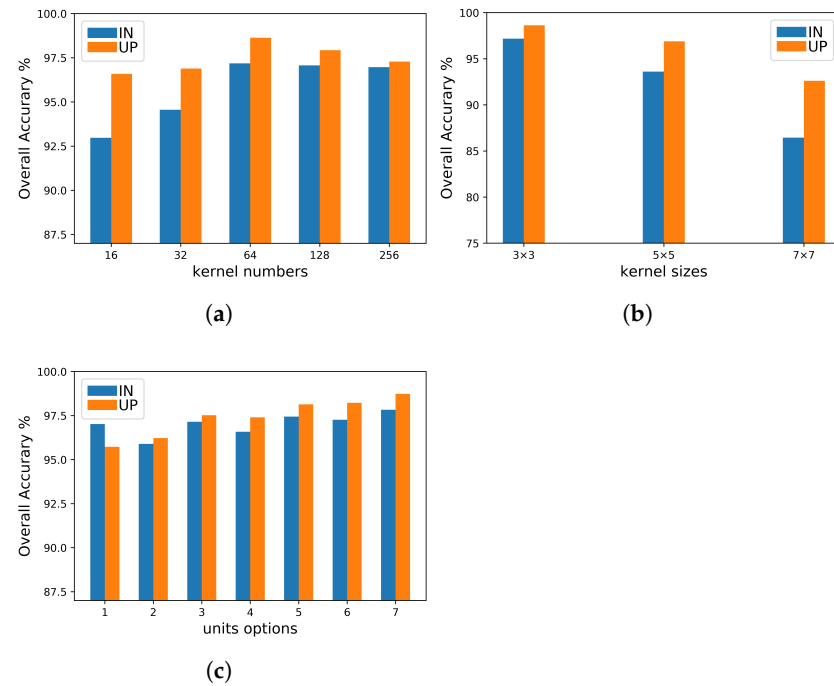


Figure 10. Overall accuracies of AWF²-Add.-GAN with various kernel settings and number of neurons in their spectral and spatial feature extractors, sampled on two datasets for training. (a) Effect of the number of kernels, (b) kernel sizes, and (c) number of neurons for spectral purity analysis.

As illustrated in Figure 11, setting the depths as “4 & 4” led to the best results on both datasets. Furthermore, the overall accuracies obtained with shallow depths differed little. Therefore, the branch feature fusion strategy yields high robustness and practicability compared with the obvious spectral–spatial representation reviewed in [37].

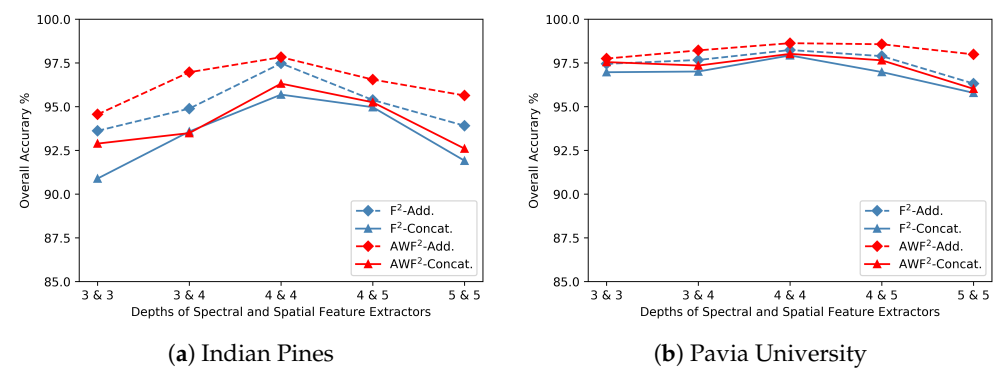


Figure 11. Overall accuracies for different depths of the two feature extractors (3 & 3, 3 & 4, 4 & 4, 4 & 5, and 5 & 5, respectively). (a) On the Indian Pines dataset, and (b) on the Pavia University dataset.

5.4. Influence of Unlabeled Real HSI Cuboids for AWF²-GANs

To evaluate the influence of unlabeled real HSI cuboids, we tested the proposed AWF²-GANs with four feature fusion options using different numbers of unlabeled HSI samples. We used both IN and UP datasets for this evaluation. We randomly selected 0, 300, 1000, and 5000 unlabeled samples, and 300 labeled samples for training. To verify the influence by adding unlabeled samples, we also take semi-supervised GANs like HS-GAN and SS-GAN into account. Table 8 shows that adding too many real unlabeled HSI cuboids for training has less effect than that of little unlabeled sizes, and even jeopardizes HSI

classification performance. This is caused by the sample distribution differences between labeled and unlabeled HSI pixels.

Table 8. The overall accuracies (OA%) of comparison methods using different numbers of unlabeled and 300 labeled HSI samples; experiments conducted on the Indian Pines and Pavia University datasets.

Dataset	Models	0	300	1000	5000
Indian Pines	HS-GAN	65.89	65.38	60.16	59.86
	SS-GAN	89.69	91.28	88.22	79.85
	F ² -Con.	94.79	93.07	89.73	67.49
	F ² -Add.	95.95	96.60	94.83	96.47
	AWF ² -Con.	93.86	94.83	95.32	79.82
	AWF ² -Add.	96.06	96.47	95.16	78.57
Pavia University	HS-GAN	83.42	83.69	80.87	78.89
	SS-GAN	96.64	96.75	94.82	92.80
	F ² -Con.	98.02	97.99	95.66	84.35
	F ² -Add.	98.03	98.27	94.46	82.17
	AWF ² -Con.	97.91	98.01	93.39	90.59
	AWF ² -Add.	98.16	98.37	96.69	91.26

Furthermore, it can be seen that using 300 unlabeled samples equal to the labeled samples resulted in improved accuracy compared to other methods, which is consistent with the conclusion reported in [35]. Therefore, in contrast to the negative effects with unlabeled samples reported in [37], the consistent HSI classification performance of AWF²-GANs use unlabeled samples equal to their labeled samples demonstrated that adding unlabeled samples mitigate the small samples effects in other deep learning models. It is worth mentioning that, F²-Con. (with basic concatenation feature fusion option) always yielded the highest accuracy with no unlabeled samples, which is caused by its channel concatenation retains abundant spectral radiation and detail characteristics, and improve its feature effectiveness. When the equivocal neighboring distribution has been introduced, the detail characteristics would be guided in a wrong direction, and result in a accuracy—decreasing.

6. Conclusions

Hyperspectral image classification faces challenges due to redundant spectral information, weak spectral–spatial representation, and limited labeled samples. It has been demonstrated that generative adversarial networks have a strong ability to expand sample sets and generalize models for classification and feature representation. In this paper, we proposed patch-based and semi-supervised GAN-based classification framework with various feature fusion strategies for HSI classification and to overcome the above-mentioned issues. Experimental results revealed that the feature fusion spectral–spatial combinations are more effective than the fixed spectral–spatial extractions, which took three times longer in testing. The feature fusion model contained four fusion options to adapt complementary and interconnected information for classification. The AWF²-GANs were designed to integrate the global spectral signatures and separable spatial contexts via various fusion options, and provided generated data for small sample issues. Furthermore, the considered joint loss function with the center loss captured the intra-class sensitivity from local neighboring areas, and gave an efficient spatial regularization result. Quantitative experiments on two hyperspectral image datasets demonstrated that the proposed AWF²-GANs can achieve promising classification accuracy and robust performance.

Author Contributions: Methodology, H.L. and W.B.; Project administration, W.B.; Validation, X.S.; Writing—original draft, H.L.; Writing—review & editing, H.L. and X.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Natural Science Foundation of Ningxia Province of China (Project No. 2020AAC02028).

Acknowledgments: We thank the Image and Intelligence Information Processing Innovation Team of the National Ethnic Affairs Commission of China for their support.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Berger, K.; Atzberger, C.; Danner, M.; D’Urso, G.; Mauser, W.; Vuolo, F.; Hank, T. Evaluation of the PROSAIL model capabilities for future hyperspectral model environments: A review study. *Remote Sens.* **2018**, *10*, 85. [\[CrossRef\]](#)
- Gerhards, M.; Schlerf, M.; Mallick, K.; Udelhoven, T. Challenges and future perspectives of multi-/Hyperspectral thermal infrared remote sensing for crop water-stress detection: A review. *Remote Sens.* **2019**, *11*, 1240. [\[CrossRef\]](#)
- Vali, A.; Comai, S.; Matteucci, M. Deep learning for land use and land cover classification based on hyperspectral and multispectral earth observation data: A review. *Remote Sens.* **2020**, *12*, 2495. [\[CrossRef\]](#)
- Chang, C.I.; Song, M.; Zhang, J.; Wu, C.C. Editorial for Special Issue “Hyperspectral Imaging and Applications”. *Remote Sens.* **2019**, *11*, 17. [\[CrossRef\]](#)
- Pullanagari, R.R.; Kereszturi, G.; Yule, I.J.; Ghamisi, P. Assessing the performance of multiple spectral–spatial features of a hyperspectral image for classification of urban land cover classes using support vector machines and artificial neural network. *J. Appl. Remote Sens.* **2017**, *11*, 026009. [\[CrossRef\]](#)
- Zhang, N.; Yang, G.; Pan, Y.; Yang, X.; Chen, L.; Zhao, C. A Review of Advanced Technologies and Development for Hyperspectral-Based Plant Disease Detection in the Past Three Decades. *Remote Sens.* **2020**, *12*, 3188. [\[CrossRef\]](#)
- Kycko, M.; Zagajewski, B.; Lavender, S.; Dabija, A. In situ hyperspectral remote sensing for monitoring of alpine trampled and recultivated species. *Remote Sens.* **2019**, *11*, 1296. [\[CrossRef\]](#)
- Ding, C.; Li, Y.; Xia, Y.; Wei, W.; Zhang, L.; Zhang, Y. Convolutional neural networks based hyperspectral image classification method with adaptive kernels. *Remote Sens.* **2017**, *9*, 618. [\[CrossRef\]](#)
- Luo, F.; Du, B.; Zhang, L.; Zhang, L.; Tao, D. Feature learning using spatial-spectral hypergraph discriminant analysis for hyperspectral image. *IEEE Trans. Cybern.* **2018**, *49*, 2406–2419. [\[CrossRef\]](#)
- Audebert, N.; Le Saux, B.; Lefèvre, S. Deep learning for classification of hyperspectral data: A comparative review. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 159–173. [\[CrossRef\]](#)
- Meng, Z.; Li, L.; Jiao, L.; Feng, Z.; Tang, X.; Liang, M. Fully Dense Multiscale Fusion Network for Hyperspectral Image Classification. *Remote Sens.* **2019**, *11*, 2718. [\[CrossRef\]](#)
- Shi, G.; Huang, H.; Liu, J.; Li, Z.; Wang, L. Spatial-Spectral Multiple Manifold Discriminant Analysis for Dimensionality Reduction of Hyperspectral Imagery. *Remote Sens.* **2019**, *11*, 2414. [\[CrossRef\]](#)
- Hu, W.; Huang, Y.; Wei, L.; Zhang, F.; Li, H. Deep convolutional neural networks for hyperspectral image classification. *J. Sens.* **2015**, *2015*, 333–344. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- Mei, S.; Ji, J.; Hou, J.; Li, X.; Du, Q. Learning sensor-specific spatial-spectral features of hyperspectral images via convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4520–4533. [\[CrossRef\]](#)
- Li, W.; Wu, G.; Zhang, F.; Du, Q. Hyperspectral image classification using deep pixel-pair features. *IEEE Trans. Geosci. Remote Sens.* **2016**, *55*, 844–853. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- Li, Y.; Zhang, H.; Shen, Q. Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sens.* **2017**, *9*, 67. [\[CrossRef\]](#)
- Zhang, M.; Li, W.; Du, Q. Diverse region-based CNN for hyperspectral image classification. *IEEE Trans. Image Process.* **2018**, *27*, 2623–2634. [\[CrossRef\]](#) [\[PubMed\]](#)
- Liang, M.; Jiao, L.; Meng, Z. A superpixel-based relational auto-encoder for feature extraction of hyperspectral images. *Remote Sens.* **2019**, *11*, 2454. [\[CrossRef\]](#)
- Liu, H.; Li, J.; He, L.; Wang, Y. Superpixel-guided layer-wise embedding CNN for remote sensing image classification. *Remote Sens.* **2019**, *11*, 174. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- Wang, W.; Dou, S.; Jiang, Z.; Sun, L. A fast dense spectral–spatial convolution network framework for hyperspectral images classification. *Remote Sens.* **2018**, *10*, 1068. [\[CrossRef\]](#)
- Zhu, K.; Chen, Y.; Ghamisi, P.; Jia, X.; Benediktsson, J.A. Deep convolutional capsule network for hyperspectral image spectral and spectral-spatial classification. *Remote Sens.* **2019**, *11*, 223. [\[CrossRef\]](#)

26. Cui, X.; Zheng, K.; Gao, L.; Zhang, B.; Yang, D.; Ren, J. Multiscale spatial-spectral convolutional network with image-based framework for hyperspectral imagery classification. *Remote Sens.* **2019**, *11*, 2220. [\[CrossRef\]](#)
27. Zhang, Y.; Huynh, C.P.; Ngan, K.N. Feature fusion with predictive weighting for spectral image classification and segmentation. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6792–6807. [\[CrossRef\]](#)
28. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the CVPR 2017—2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017; pp. 2117–2125.
29. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [\[CrossRef\]](#)
30. Fang, S.; Quan, D.; Wang, S.; Zhang, L.; Zhou, L. A Two-Branch Network with Semi-Supervised Learning for Hyperspectral Classification. In Proceedings of the IGARSS 2018—2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 3860–3863.
31. Hu, Y.; An, R.; Wang, B.; Xing, F.; Ju, F. Shape Adaptive Neighborhood Information-Based Semi-Supervised Learning for Hyperspectral Image Classification. *Remote Sens.* **2020**, *12*, 2976. [\[CrossRef\]](#)
32. Wan, S.; Gong, C.; Zhong, P.; Du, B.; Zhang, L.; Yang, J. Multiscale dynamic graph convolutional network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 3162–3177. [\[CrossRef\]](#)
33. Zhao, W.; Chen, X.; Chen, J.; Qu, Y. Sample Generation with Self-Attention Generative Adversarial Adaptation Network (SaGAAN) for Hyperspectral Image Classification. *Remote Sens.* **2020**, *12*, 843. [\[CrossRef\]](#)
34. He, Z.; Liu, H.; Wang, Y.; Hu, J. Generative adversarial networks-based semi-supervised learning for hyperspectral image classification. *Remote Sens.* **2017**, *9*, 1042. [\[CrossRef\]](#)
35. 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. [\[CrossRef\]](#)
36. 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. [\[CrossRef\]](#)
37. Zhong, Z.; Li, J.; Clausi, D.A.; Wong, A. Generative adversarial networks and conditional random fields for hyperspectral image classification. *IEEE Trans. Cybern.* **2020**, *50*, 3318–3329. [\[CrossRef\]](#)
38. Gao, H.; Yao, D.; Wang, M.; Li, C.; Liu, H.; Hua, Z.; Wang, J. A Hyperspectral Image Classification Method Based on Multi-Discriminator Generative Adversarial Networks. *Sensors* **2019**, *19*, 3269. [\[CrossRef\]](#)
39. 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. [\[CrossRef\]](#)
40. Wang, J.; Gao, F.; Dong, J.; Du, Q. Adaptive DropBlock-Enhanced Generative Adversarial Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, 1–14. [\[CrossRef\]](#)
41. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. In Proceedings of the International Conference on Learning Representations ICLR, Toulon, France, 20 January 2016; pp. 1–16.
42. 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. [\[CrossRef\]](#)
43. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2672–2680.
44. Odena, A.; Olah, C.; Shlens, J. Conditional Image Synthesis With Auxiliary Classifier GANs. In *International Conference on Machine Learning*; PMLR: Sydney, Australia, 2017; pp. 2642–2651.
45. Wen, Y.; Zhang, K.; Li, Z.; Qiao, Y. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*; Springer: Amsterdam, The Netherlands, 2016; pp. 499–515.
46. Cai, Y.; Dong, Z.; Cai, Z.; Liu, X.; Wang, G. Discriminative Spectral-Spatial Attention-Aware Residual Network for Hyperspectral Image Classification. In Proceedings of the 2019 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS), Amsterdam, The Netherlands, 24–26 September 2019; pp. 1–5.
47. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.
48. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; pp. 1026–1034.
49. Marpu, P.R.; Pedernana, M.; Dalla Mura, M.; Benediktsson, J.A.; Bruzzone, L. Automatic generation of standard deviation attribute profiles for spectral-spatial classification of remote sensing data. *IEEE Geosci. Remote Sens. Lett.* **2012**, *10*, 293–297. [\[CrossRef\]](#)