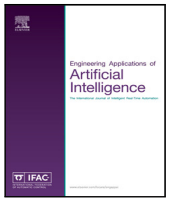
Engineering Applications of Artificial Intelligence 121 (2023) 106017

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



# MSRA-G: Combination of multi-scale residual attention network and generative adversarial networks for hyperspectral image classification

Jinling Zhao [a], Lei Hu [a,b], Linsheng Huang [a], Chuanjian Wang [a,*], Dong Liang [a,*]

[a] National Engineering Research Center for Agro-Ecological Big Data Analysis & Application, Anhui University, Hefei 230601, China
[b] School of Electronics and Information Engineering, Anhui University, Hefei 230601, China

ARTICLE INFO

*Keywords:*
Hyperspectral image classification
Multi-scale feature extraction
Attention mechanism
Residual attention module
Data augmentation

ABSTRACT

Deep learning-based technology has been introduced to increase the classification accuracy of hyperspectral imagery (HSI). Nevertheless, it is still a challenging issue to derive a satisfying classification accuracy from limited training samples. A novel method (MSRA-G) that combines multi-scale residual attention (MSRA) with Generative Adversarial Networks (GANs) was proposed. In view of the low classification accuracy with limited training samples, the is first used to generate more separable synthetic samples. A network is then proposed to extract multi-scale context information for improving HSI classification. The proposed method constructs two multi-scale feature extraction modules to identify high-level spatial–spectral features based on the 3D–2D hybrid network. In addition, the residual connection mode and the attention mechanism are combined to establish the channel and spatial residual attention modules. Different weights are assigned to different features in the channel dimension and spatial dimension, and the features are selectively learned. Furthermore, to verify the performance of MSRA-G, experiments were carried out on three publicly available HSI datasets of Indian Pines, University of Pavia and Salinas Valley. The experimental results show that our proposed MSRA-G is superior to several popular classification models. It can still achieve satisfactory classification accuracies, even in the case of insufficient training samples.

## 1. Introduction

The rapid development of sensors and platforms has greatly facilitated the improvement of spatial, spectral, temporal and radiometric resolutions of satellite imagery. In comparison with multispectral and optical images, abundant spectral–spatial information for a hyperspectral image (HSI) can be fully utilized to perform a finer classification. HSIs have been widely applied to various application scenarios, such as environmental monitoring, precision agriculture, resource management and urban planning (Audebert et al., 2019; Chen et al., 2014; Hennessy et al., 2020; Sothe et al., 2020). In these applications, the HSI classification has become one of the most critical issues. HSIs contain rich spatial and spectral information, but the classification performance is generally unsatisfactory for most classification models due to the limited training data (Zhao et al., 2016). Therefore, it is very important to make full use of spatial–spectral information in HSIs with limited samples. It aims to train a classifier based on some labeled pixel samples, and predict other pixel samples' labels to obtain the spatial distribution and quantity of different objects in the image (Li et al., 2018; Zhang et al., 2022a). In general, data annotation is usually difficult, time-consuming and laborious in the process of HSI classification. Limited training samples and insufficient extraction of spectral–spatial features have brought huge challenges to HSI classification (Deng et al., 2018; Pan et al., 2018; Zhang et al., 2021a, 2022b).

HSIs have hundreds of spectral bands and a lot of redundant data between adjacent spectral bands. This is a challenge for computer hardware requirements and a negative impact on the classification results (Kang et al., 2017; Sun and Du, 2019). In response to this problem, feature extraction of HSIs can be performed before inputting them into classifiers, which can reduce the dimensionality of hyperspectral data, reduce the load on computer hardware, and improve the computing efficiency. Commonly used dimensionality reduction methods include principal component analysis (PCA) (Xia et al., 2013), linear discriminant analysis (LDA) (Li and Qian, 2011), locally linear embedding (LLE) (Zhang et al., 2018), etc. These methods can extract most of the essential information that can represent the original HSIs, while achieving the dimensionality reduction. In order to solve the problem of hyperspectral classification, traditional classifiers are used such as k-nearest neighbor (KNN) (Guo et al., 2018), extreme learning machine (ELM) (Cao et al., 2019), support vector machine (SVM) (Okwuashi and Ndehedehe, 2020), etc., by combining with feature selection methods. The early HSI classification methods usually have poor classification performance based on only spectral information, such as REF-SVM,

* Corresponding authors.
*E-mail addresses:* wcj_si@ahu.edu.cn (C. Wang), dliang@ahu.edu.cn (D. Liang).

https://doi.org/10.1016/j.engappai.2023.106017
Received 6 May 2022; Received in revised form 21 December 2022; Accepted 16 February 2023
Available online 26 February 2023
0952-1976/

random forests (RF) (Xia et al., 2017), multiple logistic regression (MLR) (Khodadadzadeh et al., 2014). Various studies have proved that taking spatial information into consideration can play a positive role in improving classification accuracy. Li et al. (2017a) proposed a HSI classification method called spectral–spatial kernel-based support vector machine (SSF-SVM) which used area median filtering (AMF) to extract spatial features and combined spatial and spectral features into SVM classifier for improving the classification accuracy. When the training sample size is small, the performance of SSF-SVM is poor. Gu et al. (2018) proposed a classification method based on the combination of spatial–spectral features and ensemble extreme learning machines (EML), and the generalization performance of model was improved. Although the spatial information and spectral information are combined, the deep features are not fully explored. The classification accuracy mainly depends on the quality of feature selection and extraction. It is often difficult for traditional feature extraction methods to achieve the desired classification results with limited samples (Zhu et al., 2020).

In recent years, the deep learning (DL) theory has achieved excellent results in the fields of natural language processing and image classification by its powerful automatic learning capabilities (Chen et al., 2021; Cheng et al., 2020; Li et al., 2019; Yang et al., 2018; Zhang et al., 2021b; Zhang and Zhang, 2022). Compared with traditional machine learning methods, DL theory does not require manual design features and can realize end-to-end learning. For example, convolutional neural network (CNN) model can process two-dimensional (2D) and three-dimensional (3D) images, and has unique advantages in feature extraction. CNN usually uses a non-linear activation function to extract the non-linear features of images, which naturally arouses people's attention (Yu et al., 2020). Hu et al. (2015) used the CNN first in HSI classification with only a one-dimensional (1D) convolution kernel and a focus on the spectral features of HSIs. Makantasis et al. (2015) used PCA to remove redundancy on HSIs, and then input the dimensionality-reduced HSI data into 2D-CNN for classification after extracting spatial–spectral features. Chen et al. (2015) proposed a deep belief network (DBN) classification model which combined spatial–spectral features to improve classification accuracy. Studies have shown that, compared to 2D-CNN, 3D convolution kernel is more suitable for HSI classification. Li et al. (2017b) proposed a new framework for spatial–spectral feature extraction based on 3D-CNN, by using the original HSIs as input and effectively extracting the combined features of deep spatial–spectral features. Zhang et al. (2019) used the information of different scales in the network structure and the 3D dense connection structure to aggregate features at different levels, and thus proposed a multi-scale dense network (MSDN) for HSI classification with an improved stability with respect to accuracy. Zhong et al. (2017) proposed a spatial–spectral residual network (SSRN) by referring to ResNet, which extracted spatial and spectral features of HSIs respectively, however, the network design was redundant. Wang et al. (2018) proposed an end-to-end fast dense spatial–spectral convolution (FDSSC) framework. Different convolution kernels have been used to extract multi-scale spatial–spectral features, showing the advantages of extracting effective features from different receptive fields. In addition, Roy et al. (2020) designed a hybrid neural network (HybridSN) that combined 3D and 2D CNNs. Compared with the 3D-CNN alone, the HybridSN reduced the complexity of the model and the potential of hybrid convolutional network was experimentally verified in HSI classification. Feng et al. (2019) designed an 11-layer CNN model Residual-HybridSN (R-HybridSN) from the perspective of network optimization, which could better learn the deep spatial–spectral features with limited training samples.

Although the CNN-based method has achieved good results in the field of HSI classification, during the model training process, the contribution of feature maps output by convolutional layers to classification is different. There is also correlation between feature maps (Haut et al., 2019; Mou and Zhu, 2019). Therefore, to process various feature maps differently and focus on useful features, the attention mechanism is used to refine the feature maps. Hu et al. (2018) constructed squeeze-and-excitation network (SENet) that achieved remarkable results in the Large Scale Visual Recognition Challenge (ILSVRC) 2017 classification competition. Fang et al. (2019) proposed an end-to-end 3D dense network (MSDN-SA) by introducing the spectral attention mechanism, which improved the classification performance of training model. Woo et al. (2018) proposed the convolutional block attention module (CBAM) that could respectively extract more refined information from channel dimension and spatial dimension. Sun et al. (2019) designed a spectral–spatial attention network (SSAN) that achieved good classification results by introducing attention modules to suppress the influence of interfering pixels. A multi-attention fusion network (MAFN) was proposed to merge multiple attention features for classification with excellent potential (Li et al., 2021).

The number of training samples is one of the key factors affecting the performance of CNN models, however, collecting a large number of training samples from high-resolution images is a challenging task (Deng et al., 2018). Therefore, data augmentation is an effective method to solve the problem of serious lack of labeled samples for HSIs. Random flipping, cropping, scaling or adding noise are typical data augmentation techniques, but they usually have fewer positive effects on classification. Recently, generative models have received extensive attention, because they can generate high-quality samples and alleviate the over-fitting problem (Zhu et al., 2018; Huang and Chen, 2020). Goodfellow et al. (2014) proposed the new generative adversarial networks (GANs), which trained the network in an adversarial way and generated new data samples that could estimate the potential distribution of samples. It provides a way to learn depth representation without labeling the training data, and experiments have proved the potential of the framework. In recent years, GANs has been widely used in various application fields such as object detection and image translation (Ma et al., 2018; Courtrai et al., 2020).

To fully extract the spatial–spectral features of HSIs under limited training samples, the original HSIs were first divided into training set, validation set and testing set. Secondly, GANs was used to generate new samples that simulated the real data distribution, and to augment training set to alleviate the over-fitting problem. Thirdly, the PCA was used to reduce the dimensionality of the real and synthetic samples. Finally, a deep hybrid CNN network was proposed, and a multi-scale residual attention (MSRA) module was designed to improve the performance of HSI classification under the condition of limited training data. The main technical contributions of this work are summarized as follows.

(1) A GANs-based data augmentation strategy is proposed to generate more separable synthetic samples.

(2) To obtain the spatial–spectral features in different receptive fields so as to enhance the classification capability, a dual-dimensional multi-scale feature extraction module is constructed based on the 3D–2D hybrid network to extract rich multi-scale spatial–spectral features.

(3) Since different features in a HSI contribute differently to the classification results, the residual connection mode and attention mechanism are combined to establish multi-scale residual attention (MSRA) module to refine feature maps.

## 2. Related work

### 2.1. GANs-based data augmentation

GANs is mainly inspired by the idea of zero-sum game in game theory. When applied to DL, it is the game between generator (G) and discriminator (D) (Tembine, 2019). The GANs proposed in this study sets up four-layer fully connected networks in G and D respectively and uses LeakyReLU as the nonlinear activation function. For each layer, the number of nodes is 512, the batch size is 256 and the learning rate is 5e-5. In addition, the cross-entropy loss function is used to measure the error, and the Root Mean Square Prop (RMSprop) is used to optimize the G and D. As shown in Fig. 1, the alternating

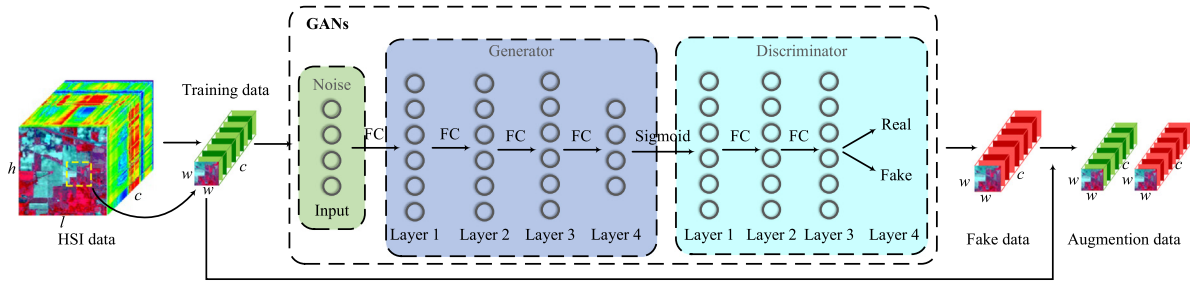



**Fig. 1.** Flowchart of proposed data augmentation based on the GANs.

iterative training is mainly adopted. More specifically, a $r$ ratio of real cube training data set is first randomly selected from HSIs with original size of $h \times l \times c$. The size of selected data is $w \times w \times c$ where $h$, $l$ and $c$ are respectively the height, width and number of bands, and $w$ is the window size. Secondly, the initial noise with the size of [256, $n$] is input into the network and converted into the false sample set of [256, $c$] by the generator, where $n$ is the noise amount obeying the standard normal distribution and its value is set to 100 (Zhan et al., 2018; Zhu et al., 2018; Wang et al., 2021a). Thirdly, the parameters of G remain unchanged, and supervised classification is carried out for D so that D can better judge true and false samples. Then, to fix the parameters of D, and G is trained and the parameters is updated according to the losses derived from D. The above processes are repeated over 10,000 iterations to obtain trained G and D. Finally, as a data augmentation training strategy, false samples are generated from trained G and combined with real training samples.

### 2.2. Extraction of multi-scale features

Here, multi-scale convolution kernel (Shi and Pun, 2018; Tu et al., 2019) is introduced into the hybrid network, and 3D multi-scale feature extraction block and 2D multi-scale feature extraction block are constructed, as shown in Figs. 2(a) and 2(b) respectively. As shown in Fig. 2, both of 3D and 2D multi-scale feature extraction blocks contain three branches and four convolution layers. The batch normalization (BN) operation is performed after each convolution layer, and ReLU is used as the activation function. Here, the multi-scale convolution kernels of $3 \times 3{\times}3, 3{\times}3{\times}5, 3{\times}3 \times 7$ and $3{\times}3, 5{\times}5, 7{\times}7$ are respectively adopted for 3D and 2D blocks to derive the different scale features from the input images. Then, the output feature maps at all scales are connected in series. Finally, fusion features are obtained by using the $1 \times 1 \times 1$ and $1 \times 1$ convolution kernel, respectively, with the sizes of $w \times w \times n_1$ for the $k$ channel and $w_3 \times w_3$ for the $2k$ channel, respectively. Sharing features among various scales can enhance the information flow of the network. Our model uses multi-scale kernels to extract more discriminating features, for alleviating the problem of reduced accuracy caused by limited training samples.

### 2.3. Attention mechanism

Intuitively, different channels and spatial features contribute to HSI classification to different degrees (Mei et al., 2019). Based on this assumption, we connect an attention block behind the current multi-scale network. The purpose of attention block is to enable the network to effectively focus on learning, thereby improving the representational ability (Gao et al., 2021). The attention block takes the feature maps captured from the upper-layer network as input and then uses them to learn the attention maps that act on itself, thereby reinforcing the importance of more recognizable features. Considering the dual dimensions of hybrid networks, the channel attention mechanism and spatial attention mechanism are respectively introduced in 3D and 2D dimensions to strengthen the features that contribute greatly to the classification results, so as to improve the overall performance.

#### 2.3.1. Channel attention module

According to the importance of different channels, the channel attention module redistributes the weight information to strengthen the channel features that can improve network performance, and to suppress the insignificant channels to a certain extent (Hang et al., 2020). The detailed structure of the channel attention module is shown in Fig. 3. The input feature map is $\mathbf{F} \in R^{w \times w \times n_1 \times k}$, where $w \times w$ is the window size, $n_1$ and $k$ respectively represent the number of bands and channels. Firstly, 3D global average pooling and 3D global maximum pooling are performed on the input feature map to obtain two feature descriptors $\mathbf{F}^{ca}_{avg}$ and $\mathbf{F}^{ca}_{max}$ with the size of $1 \times 1 \times k$. Then two feature maps with the size of $1 \times 1 \times k$ are obtained through the share network (SN) which contains two 3D convolution layers and a ReLU activation layer. Finally, the output feature vector is obtained through element merging, the channel attention map $CA_F$ within the range of (0, 1) is obtained through the Sigmoid activation function. The calculation process of channel attention is as shown in Eq. (1).

$$\begin{aligned} CA_F &= \delta\left(SN\left(AvgPool\left(\mathbf{F}\right)\right) + SN\left(MaxPool\left(\mathbf{F}\right)\right)\right) \\ &= \delta\left(W_1\delta'\left(W_0\left(\mathbf{F}^{ca}_{avg}\right)\right) + W_1\delta'\left(W_0\left(\mathbf{F}^{ca}_{max}\right)\right)\right) \end{aligned} \tag{1}$$

where $\delta$ and $\delta'$ respectively represent Sigmoid and ReLU activation functions, and $W_0$ and $W_1$ represent the weights of SN. The generated channel attention map $CA_F$ is multiplied with the original input feature map. Different weights are assigned to each channel to achieve the correction of the importance, and finally the output feature map $\mathbf{F}' \in R^{w \times w \times n_1 \times k}$ is acquired. The calculation expression can be shown in Eq. (2).

$$\mathbf{F}' = CA_F \otimes \mathbf{F} \tag{2}$$

where $CA_F$ represents the channel attention map, $\otimes$ denotes the matrix multiplication operation and $F$ is the 3D input feature map.

#### 2.3.2. Spatial attention module

The spatial attention mechanism focuses on the significance regions in the spatial dimension, which is the main difference from the channel attention mechanism (Dong et al., 2020). The detailed structure of spatial attention module is shown in Fig. 4. For input features $\mathbf{F}^* \in R^{w \times w \times k}$, two types of pooling methods are used to obtain different feature descriptors along the channel direction: $\mathbf{F}^{sa}_{avg}$ and $\mathbf{F}^{sa}_{max}$ with the sizes of $w \times w$. Then, an output feature descriptor $\left[\mathbf{F}^{sa}_{avg}; \mathbf{F}^{sa}_{max}\right]$ is obtained through the joint operation. Finally, through a 3D convolution layer with Sigmoid function, the spatial attention map $SA_F$ is generated. The calculation process of channel attention is as shown in Eq. (3).

$$\begin{aligned} SA_F &= \delta\left(f^{N \times N}\left(\left[AvgPool\left(\mathbf{F}\right); MaxPool\left(\mathbf{F}\right)\right]\right)\right) \\ &= \delta\left(f^{N \times N}\left[\mathbf{F}^{sa}_{avg}; \mathbf{F}^{sa}_{max}\right]\right) \end{aligned} \tag{3}$$

where $\delta$ represents the Sigmoid activation function, $f^{N \times N}$ denotes the 2D convolution operation, and the convolution kernel size is $N \times N$. Finally, the spatial attention map $SA_F$ is multiplied by the original input feature map to obtain the spatial refinement feature map focusing

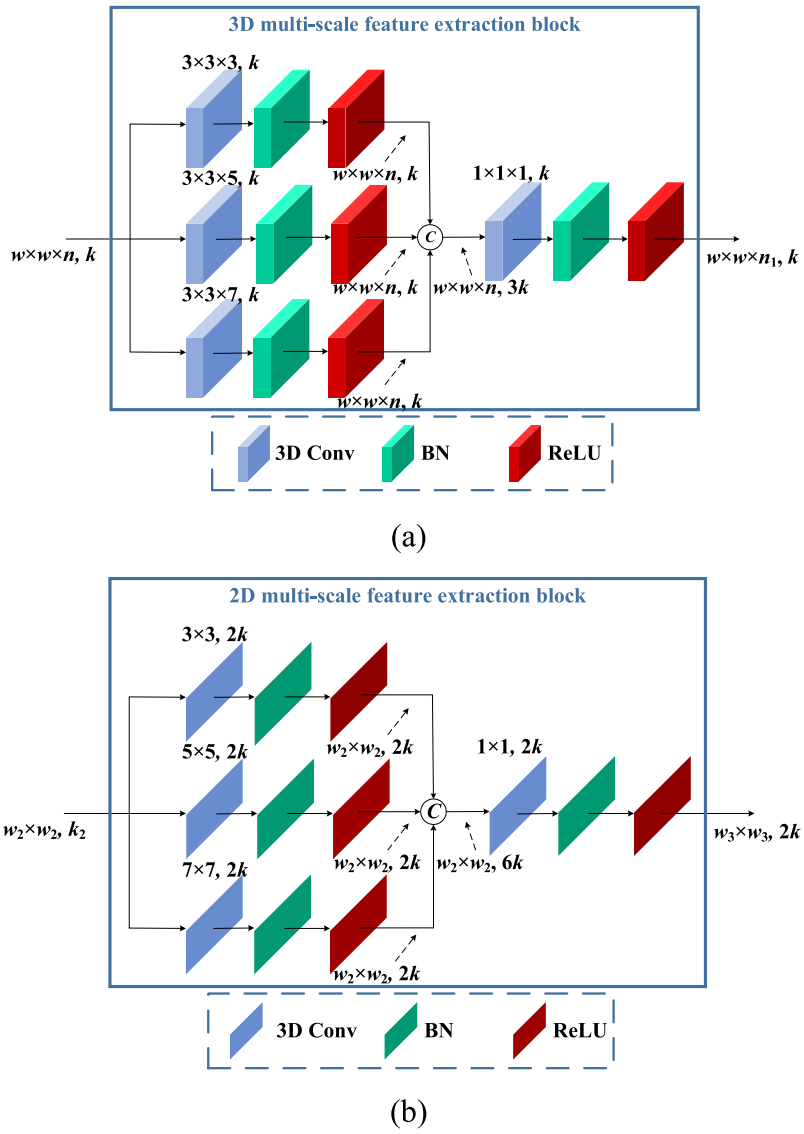



**Fig. 2.** Structure of 3D (a) and 2D (b) multi-scale feature extraction block used in our framework.

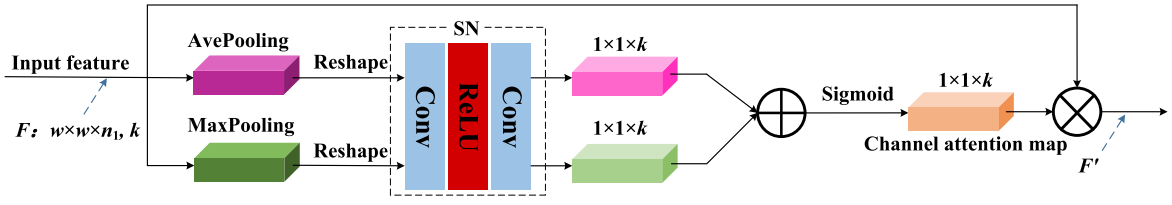



**Fig. 3.** Structure of channel attention module used in our framework.

on the most information-rich regions. The output feature map $\mathbf{F}'' \in R^{w\times w\times k}$ is obtained, and the calculation expression can be shown in Eq. (4).

$$\mathbf{F}'' = SA_F \otimes \mathbf{F}* \tag{4}$$

where $SA_F$ represents the spatial attention map, $\otimes$ represents the matrix multiplication operation, and $\mathbf{F}*$ represents the 2D input feature map.

### 2.4. Multi-scale residual attention module

In DL, the deeper the network structure is, the more likely the phenomenon of gradient disappearance is to occur, resulting in poor network training effect (Paoletti et al., 2018). In order to solve the network degradation problem, residual network is proposed. Due to the existence of skip connection, gradient is more easily and effectively propagated (Wu et al., 2020; Lu et al., 2020). By integrating the multi-scale feature extraction blocks, attention module and residual connection, the MSRA module is established. The concrete connection structure in the hybrid network is shown in Fig. 5. Firstly, the multi-scale module is used to extract the different scale features of the input image, and then the attention module is used to enhance the important features that are more helpful to the classification results. The purpose of the attention module is to allow the network to focus on learning effectively so as to improve the representation ability of the network.

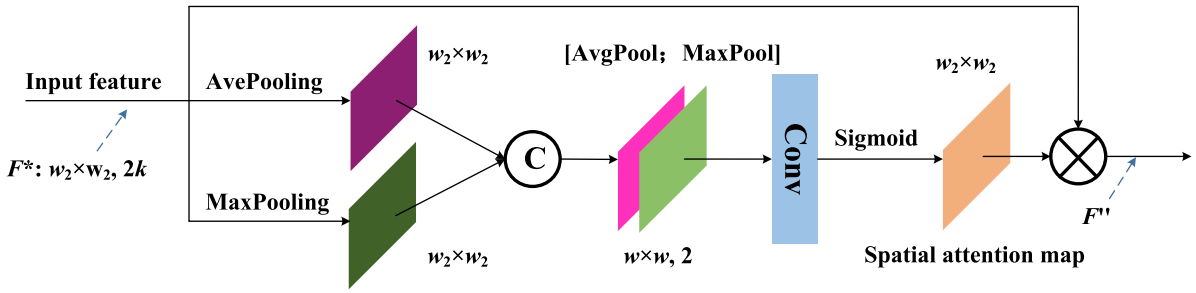



**Fig. 4.** Structure of spatial attention module used in our framework.

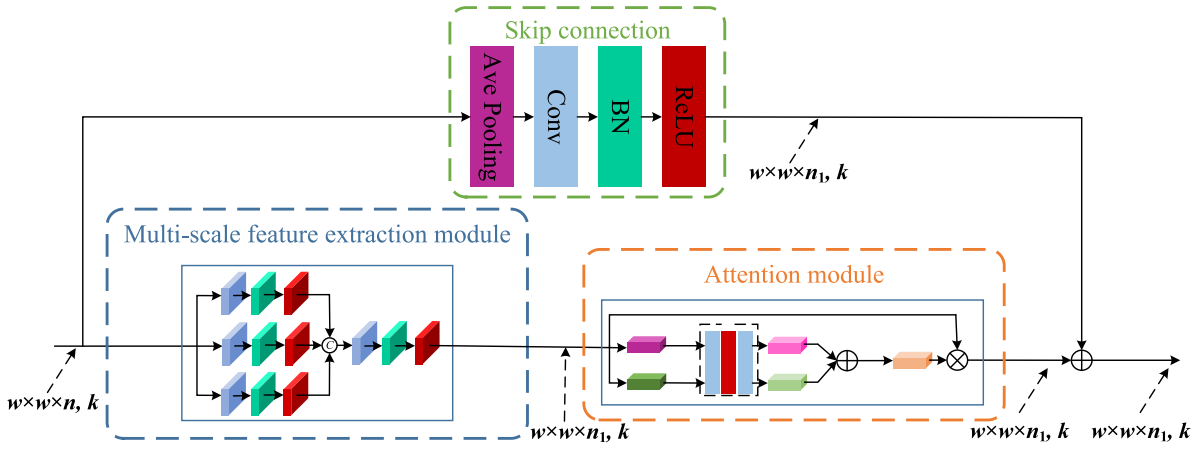



**Fig. 5.** Structure of multi-scale residual attention module used in our framework.

**Table 1**
Parameters of the three hyperspectral datasets.

| Attribute | Datasets | | |
|---|---|---|---|
| | IP | UP | SV |
| Sensor | AVIRIS | ROSIS-3 | AVIRIS |
| Pixels | 145 × 145 | 610 × 340 | 512 × 217 |
| Spatial resolution | 20 m/pixel | 1.3 m/pixel | 3.7 m/pixel |
| Classes | 16 | 9 | 16 |
| Bands | 200 | 103 | 204 |
| Wavelength | 0.4~2.45 μm | 0.43~0.86 μm | 0.4~2.5 μm |

The residual connection is adopted to make the gradient more easily propagated, thus improving the overall performance.

## 3. Hyperspectral datasets and accuracy assessment metrics

Three publicly available and widely used HSI datasets are used: Indian Pines (IP), University of Pavia (UP) and Salinas Valley (SV). The parameters of the three datasets are shown in Table 1, and their false color images and ground-truth classes are shown in Figs. 6–8, respectively.

Considering that the sample sizes and categories of the three datasets are different and unbalanced, different proportions of training samples for each dataset are used to verify the performance of the proposed model. For the IP dataset, the 5% labeled samples are randomly selected from each class as the training and validation sets, and the remaining 90% samples as test set (Table 2). For the UP dataset, the 1% labeled samples are randomly selected from each class as the training and validation sets, and the remaining 98% as the test set (Table 3). For the SV dataset, the 0.5% of labeled samples are randomly selected from each class as the training and validation sets, and the remaining 99% as the test set (Table 4).

The overall accuracy (OA), average accuracy (AA), and Kappa coefficient (Kappa) based on the confusion matrix are used to evaluate

**Table 2**
Numbers of Training, validation and test sets in IP hyperspectral dataset.

| Label | Class name | Total sample | Training | Validation | Test |
|---|---|---|---|---|---|
| 1 | Alfalfa | 46 | 2 | 2 | 42 |
| 2 | Corn-notill | 1428 | 71 | 71 | 1286 |
| 3 | Corn-mintill | 830 | 42 | 42 | 746 |
| 4 | Corn | 237 | 12 | 12 | 213 |
| 5 | Grass-pasture | 483 | 24 | 24 | 435 |
| 6 | Grass-trees | 730 | 37 | 37 | 656 |
| 7 | Grass-pasture-mowed | 28 | 1 | 1 | 26 |
| 8 | Hay-windrowed | 478 | 24 | 24 | 430 |
| 9 | Oats | 20 | 1 | 1 | 18 |
| 10 | Soybean-notill | 972 | 49 | 49 | 874 |
| 11 | Soybean-mintill | 2455 | 123 | 123 | 2209 |
| 12 | Soybean-clean | 593 | 30 | 30 | 533 |
| 13 | Wheat | 205 | 10 | 10 | 185 |
| 14 | Woods | 1265 | 63 | 63 | 1139 |
| 15 | Buildings-Grass-Trees-Drives | 386 | 19 | 19 | 348 |
| 16 | Stone-Steel-Towers | 93 | 5 | 5 | 83 |

**Table 3**
Numbers of Training, validation and test sets in UP hyperspectral dataset.

| Label | Class name | Total sample | Training | Validation | Test |
|---|---|---|---|---|---|
| 1 | Asphalt | 6631 | 66 | 66 | 6499 |
| 2 | Meadows | 18649 | 186 | 186 | 18277 |
| 3 | Gravel | 2099 | 21 | 21 | 2057 |
| 4 | Trees | 3064 | 31 | 31 | 3002 |
| 5 | Painted metal sheets | 1345 | 13 | 13 | 1319 |
| 6 | Bare Soil | 5029 | 50 | 50 | 4929 |
| 7 | Bitumen | 1330 | 13 | 13 | 1304 |
| 8 | Self-Blocking Bricks | 3682 | 37 | 37 | 3608 |
| 9 | Shadows | 947 | 9 | 9 | 929 |

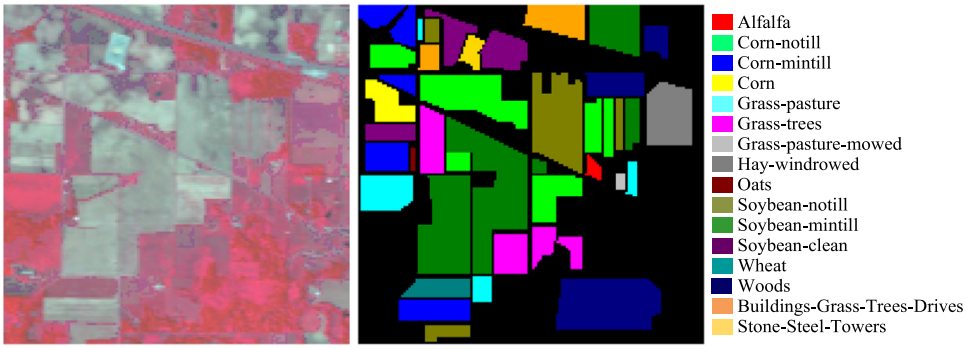



**Fig. 6.** False-color RGB image (a) and ground-truth classes (b) for the IP scene.

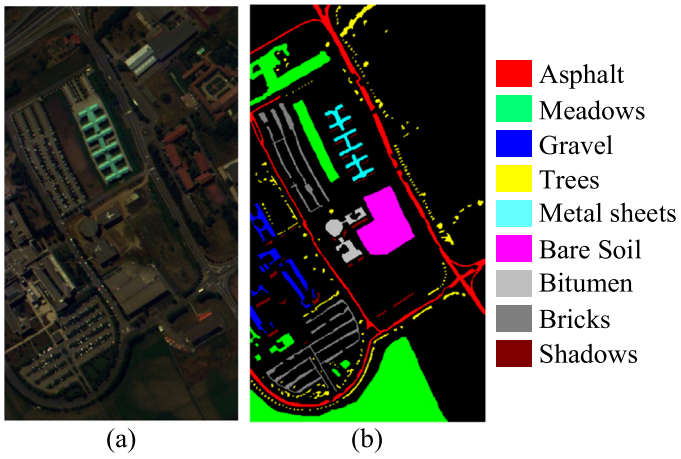



**Fig. 7.** False-color RGB image (a) and ground-truth classes (b) for the UP scene.

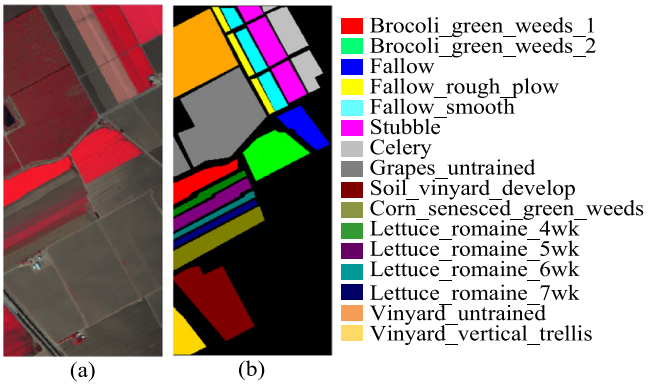



**Fig. 8.** False-color RGB image (a) and ground-truth classes (b) for the SV scene.

**Table 4**
Numbers of training, validation and test sets in SV hyperspectral data set.

| Label | Class name | Total sample | Training | Validation | Test |
|---|---|---|---|---|---|
| 1 | Alfalfa | 2009 | 10 | 10 | 1989 |
| 2 | Corn-notill | 3726 | 19 | 19 | 3689 |
| 3 | Corn-mintill | 1976 | 10 | 10 | 1956 |
| 4 | Corn | 1394 | 7 | 7 | 1380 |
| 5 | Grass-pasture | 2678 | 13 | 13 | 2651 |
| 6 | Grass-trees | 3959 | 20 | 20 | 3919 |
| 7 | Grass-pasture-mowed | 3579 | 18 | 18 | 3543 |
| 8 | Hay-windrowed | 11271 | 56 | 56 | 11158 |
| 9 | Oats | 6203 | 31 | 31 | 6141 |
| 10 | Soybean-notill | 3278 | 16 | 16 | 3245 |
| 11 | Soybean-mintill | 1068 | 5 | 5 | 1057 |
| 12 | Soybean-clean | 1927 | 10 | 10 | 1908 |
| 13 | Wheat | 916 | 5 | 5 | 907 |
| 14 | Woods | 1070 | 5 | 5 | 1059 |
| 15 | Buildings-Grass-Trees-Drives | 7268 | 36 | 36 | 7195 |
| 16 | Stone-Steel-Towers | 1807 | 9 | 9 | 1789 |

the classification performance (Hay, 1988). OA is the ratio between the number of samples predicted correctly and the total samples on the test set. AA is the ratio of the sum of classification accuracy for each class to the number of classes. Kappa is typically used for consistency testing and can also be used to measure the classification accuracy. The higher the values of three evaluation indicators, the better the classification performance.

## 4. Methodology

As shown in Fig. 9, original HSI data are first randomly divided into 3D cube training set, validation set and testing set. Secondly, the training set are introduced into GANs network to learn the potential distribution and generate synthetic samples. Thirdly, the PCA is carried out on the augmented training set to reduce the redundancy, and then is transmitted to MSRA network for training. Meanwhile, the validation set are used to constantly adjust the network hyperparameters to obtain the best training model. Finally, the test set are used to evaluate the performance of the trained model.

As shown in Fig. 10. Taking the IP dataset as an example, the sample size is set to 21 × 21 × 14 in the MSRA network. MSRA_3D is mainly composed of 3D multi-scale feature extraction module and

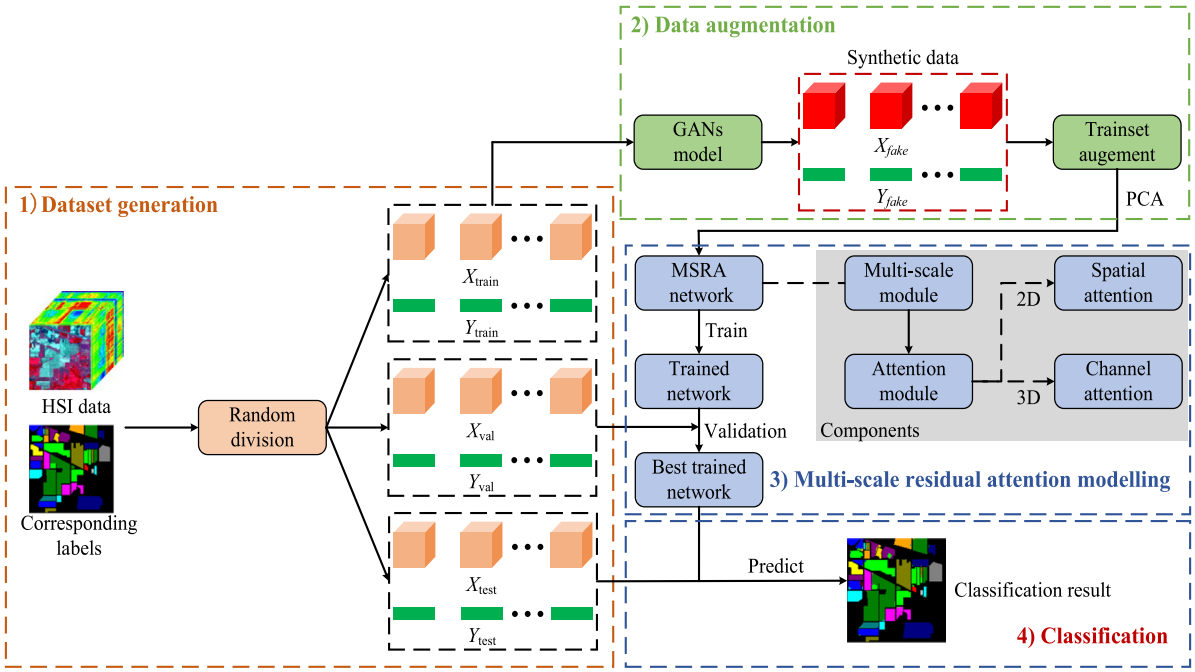



**Fig. 9.** Overall procedure of the proposed MSRA-G framework including dataset generation, data augmentation, MSRA modeling and classification.

**Table 5**
Network details of the MSRA_3D.

| Layer name | Kernel shape | Strides | Output shape |
|---|---|---|---|
| Input layer | – | – | Out_1: (21 × 21 × 14, 1) |
| Conv3D-BN-ReLU | Conv3D(Out_1): (1 × 1 × 1) | (1, 1, 1) | Out_2: (21 × 21 × 14, 32) |
| 3D multi-scale Block | Conv3D(Out_2): (3 × 3 × 3) | (1, 1, 1) | Out_3_1: (21 × 21 × 14, 32) |
| | Conv3D(Out_2): (3 × 3 × 5) | (1, 1, 1) | Out_3_2: (21 × 21 × 14, 32) |
| | Conv3D(Out_2): (3 × 3 × 7) | (1, 1, 1) | Out_3_3: (21 × 21 × 14, 32) |
| | Concatenate(Out_3_1-3) | - | Out_3_4: (21 × 21 × 14, 96) |
| | Conv3D(Out_3_4): (1 × 1 × 1) | (1, 1, 2) | Out_3_5: (21 × 21 × 7, 32) |
| Channel_attention block | GAP(Out_3_5) | – | Out_4_1: (1 × 1 × 32) |
| | GMP(Out_3_5) | – | Out_4_2: (1 × 1 × 32) |
| | Conv3D(Out_4_1): (1 × 1 × 1) | (1, 1, 1) | Out_4_3: (1 × 1 × 32) |
| | Conv3D(Out_4_2): (1 × 1 × 1) | (1, 1, 1) | Out_4_4: (1 × 1 × 32) |
| | Add(Out_4_3, Out_4_4) | – | Out_4_5: (1 × 1 × 32) |
| | Multiply(Out_3_5, Out_4_5) | – | Out_4_6: (21 × 21 × 7, 32) |
| Residual connection | AP(Out_2) | (1, 1, 2) | Out_5_1: (21 × 21 × 7, 32) |
| | Conv3D(Out_5_1): (1 × 1 × 1) | (1, 1, 1) | Out_5_2: (21 × 21 × 7, 32) |
| | Add(Out_4_6, Out_5_2) | – | Out_5_3: (21 × 21 × 7, 32) |
| Conv3D-BN-ReLU | Conv3D(Out_5_3): (3 × 3 × 3) | (2, 2, 1) | Out_6: (10 × 10 × 5, 64) |

channel attention module (Table 5). Firstly, the 3D convolution with the kernel size of 1 × 1 × 1 is used to increase the channel number to 32. Then, the 3D multi-scale module is used to extract spatial–spectral features of different scales, and the size of the output feature map is (21 × 21 × 7, 32). In order to refine the features, feature map is input into the channel attention module. New weights are given to different channels according to their importance, and thus refined feature map is gotten. Meanwhile, residual structure is adopted to prevent the gradient from disappearing.

MSRA_2D is mainly composed of 2D multi-scale feature extraction module and spatial attention module (Table 6). For the HSI cube sample size of 10 × 10 × 5 with 64 channels, they are reshaped into 10 × 10 with 320 channels and input into the 2D network. Then, three different 2D convolution kernels are used to extract spatial features, and the key information is highlighted using the spatial attention. The size of the output feature map is (5 × 5, 64).

Finally, the network passes through the fully convolutional layer while using the dropout layer to prevent over-fitting, and the classification results are obtained through Softmax function. The MSRA networks were similarly structured for UP and SV datasets.

## 5. Experiments and results

### 5.1. Parameter setting

To validate the proposed method, a computer hardware was configured including Intel(R) Core(TM) i5-7300HQ CPU (2.50 GHz), GTX1050Ti GPU and 8 GB RAM. The primary software included Microsoft Windows 10 64-bit Operating System, Python compiler and DL framework Keras. The weight parameters of the network were updated by gradient back propagation, and the adaptive moment estimation (Adam) optimization algorithm was used to train the MSRA network. This section focuses on several factors that affect the classification effect

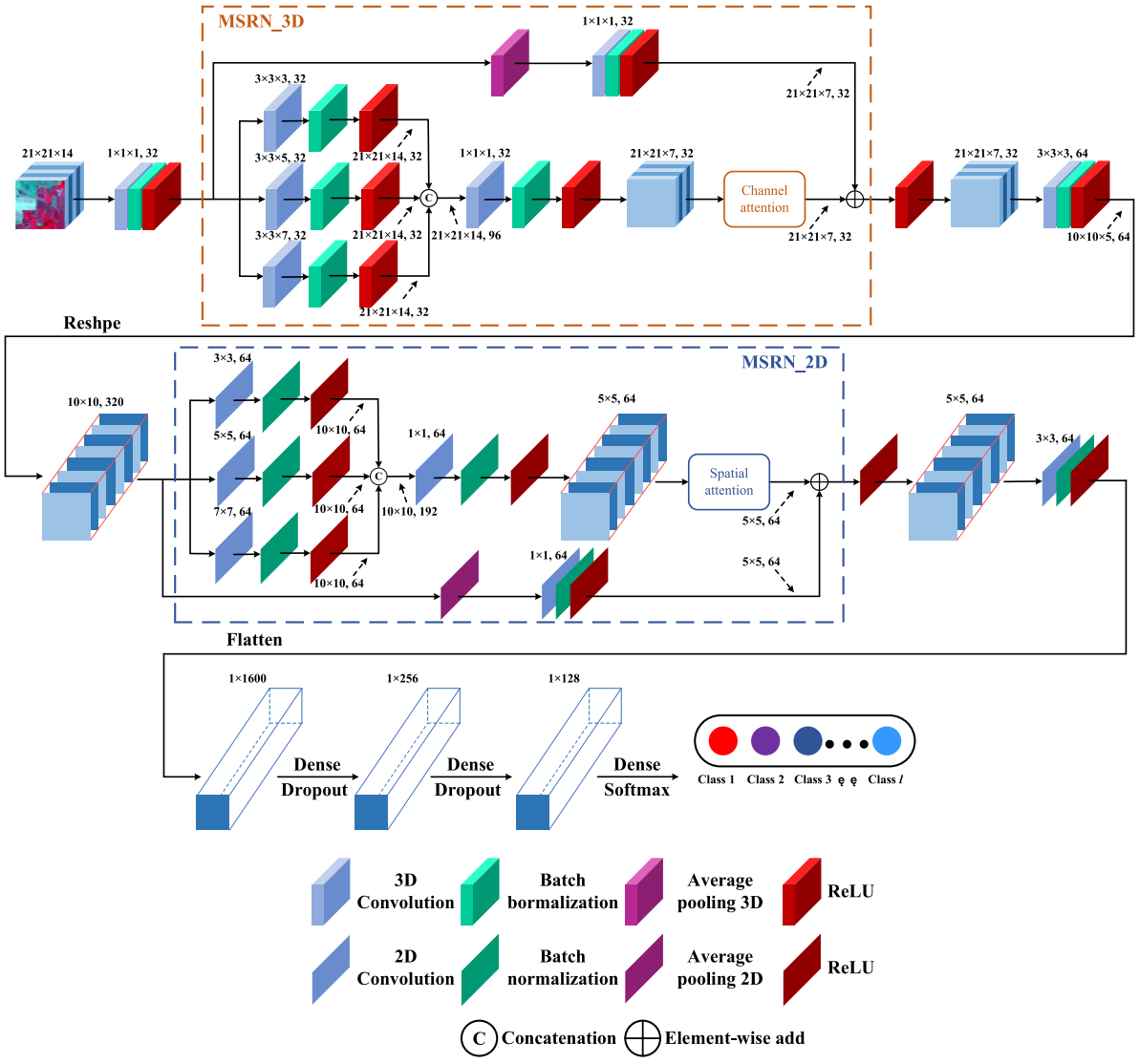



**Fig. 10.** Schematic diagram of multi-scale residual attention (MSRA) network.

**Table 6**
Network details of the MSRA_2D.

| Layer name | Kernel shape | Strides | Output shape |
|---|---|---|---|
| Input layer | – | – | Out2_1: (10 × 10 × 5, 64) |
| Reshape | – | – | Out2_2: (10 × 10, 320) |
| 2D multi-scale block | Conv2D(Out2_2): (3 × 3) | (1, 1, 1) | Out2_3_1: (10 × 10, 64) |
| | Conv2D(Out2_2): (5 × 5) | (1, 1, 1) | Out2_3_2: (10 × 10, 64) |
| | Conv2D(Out2_2): (7 × 7) | (1, 1, 1) | Out2_3_3: (10 × 10, 64) |
| | Concatenate(Out2_3_1-3) | – | Out2_3_4: (10 × 10, 192) |
| Spatial_attention block | Conv2D(Out2_3_4): (1 × 1) | (2, 2) | Out2_3_5: (5 × 5, 64) |
| | GAP(Out2_3_5) | – | Out2_4_1: (5 × 5, 1) |
| | GMP(Out2_3_5) | – | Out2_4_2: (5 × 5, 1) |
| | Concatenate(Out2_4_1, Out2_4_2) | – | Out2_4_3: (5 × 5, 2) |
| | Conv2D(Out2_4_1): (3 × 3) | (1, 1, 1) | Out2_4_4: (5 × 5, 1) |
| | Multiply(Out2_3_5, Out2_4_5) | – | Out2_4_5: (5 × 5, 64) |
| Residual connection | AP(Out2_2) | (2, 2) | Out2_5_1: (5 × 5, 320) |
| | Conv2D(Out2_5_1): (1 × 1 × 1) | (1, 1) | Out2_5_2: (5 × 5, 64) |
| | Add(Out2_4_5, Out2_5_2) | – | Out2_5_3: (5 × 5, 64) |
| Conv2D-BN-ReLU | Conv2D(Out2_5_3): (3 × 3) | (1, 1) | Out2_6: (5 × 5, 64) |

of MSRA, including principal components number ($d$), window size of input sample ($w$), learning rate ($lr$) and dropout proportions. The batch size was set to 64 with the iteration of 200 times, and the average accuracy of 10 experiments was used as the final result.

#### 5.1.1. Effect of d on classification performance

Before inputting the training samples into MSRA network, the samples were processed with the PCA. Here, the $d$ values were optimally set to 10, 12, 14, 16, 18 and 20 on the three datasets after comparing the

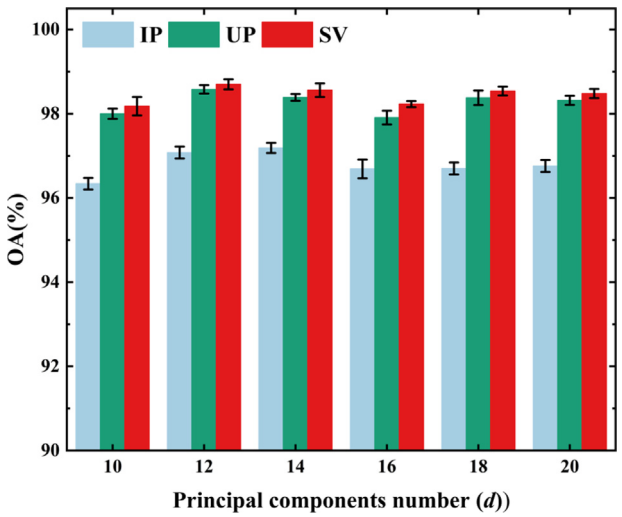



**Fig. 11.** Comparison of OAs of different $d$ values on three datasets.

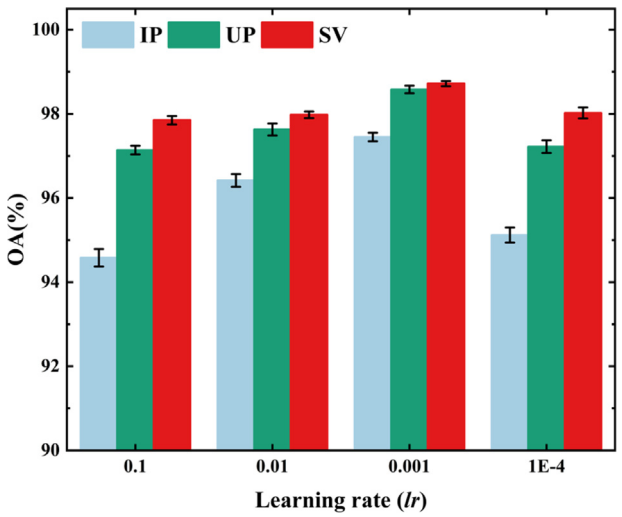



**Fig. 13.** Comparison of OAs of different $lr$ values on three datasets.

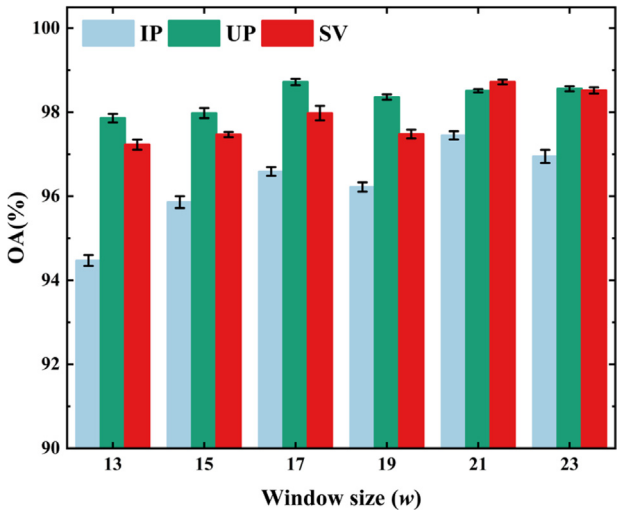



**Fig. 12.** Comparison of OAs of different $w$ values on three datasets.

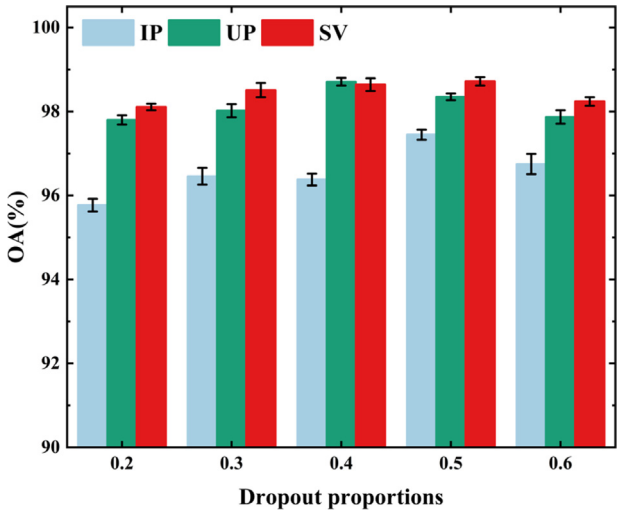



**Fig. 14.** Comparison of OAs of different dropout proportions on three datasets.

contribution and cumulative contribution. As can be seen from Fig. 11, different $d$ values result in various OAs, basically showing a trend of increasing first and decreasing later. For IP dataset, when $d$ is 14, the OA reaches the highest value, and then it is basically stable. They achieve the best classification for UP and SA datasets when $d$ is 12. OA slightly fluctuates up and down, as the $d$ continues to increase. Consequently, to balance classification accuracy with computational costs, $d$ was set to 14, 12 and 12 for IP, UP and SV datasets, respectively.

### 5.1.2. Effect of w on classification performance

If the input $w$ is too small, it will easily lead to insufficient receptive field, while it is too large, noise will be caused to slow down the training speed. Therefore, for different HSI datasets, appropriate spatial sizes are important. Six $w$ values were set to $13\times13$, $15\times15$, $17\times17$, $19\times19$, $21\times21$ and $23\times23$ for the input sample. Fig. 12 shows that, as $w$ grows, OA begins to increase rapidly and increases by about 1% for each of the three datasets. When $w$ reaches 17, UP dataset has the highest OA, and then the accuracy begins to decrease. For IP and SV datasets, the OA value is highest when $w$ reaches 21.

### 5.1.3. Effect of lr on classification performance

The $lr$ plays an important role in DL-based HSI classification methods, and affects the convergence state of models. If $lr$ is too small, low optimization efficiency maybe caused, and the algorithm is difficult to converge. If $lr$ is too large, the model parameter adjustment changes quickly and the optimal value may be passed over. Therefore, for different data sets, the appropriate $lr$ will have a positive impact on classification accuracy. Four $lr$ values of 0.1, 0.01, 0.001 and 0.0001 were set up to explore the classification performance of MSRA. As shown in Fig. 13, it can be observed that, with the decrease of $lr$, the OAs of three datasets generally show a tendency to increase first and then decrease. When $lr$ is 0.001, the OAs of three datasets reach the maximum value, meanwhile, the classification accuracy is more stable.

### 5.1.4. Effect of dropout proportions on classification performance

The over-fitting is one of the common problems in neural network training, which affects the generalization performance of classification models. To prevent over-fitting, the dropout layer is added after the first two fully connected layers, which is a regularization method by randomly making the weights of some hidden layer nodes inoperative. Three dropout proportions were set to 0.4, 0.5 and 0.6 to compare the classification effect. As shown in Fig. 14, when dropout proportion is 0.4, UP and SV datasets have higher OA values. For IP dataset, the best OA was achieved with the dropout proportion of 0.5.

**Table 7**
Comparison of accuracies between MSRA-G and other methods in IP dataset.

| Class | Color | REF-SVM | 3D-CNN | MSDN | HybridSN | SSRN | R-HybridSN | MAFN | MSRA-G |
|---|---|---|---|---|---|---|---|---|---|
| 1 | | 11.63 | 26.22 | 64.84 | 63.72 | 83.19 | 67.01 | **96.53** | **92.50** |
| 2 | | 77.27 | 93.92 | 86.17 | 93.36 | **96.01** | 95.72 | 95.22 | 95.70 |
| 3 | | 62.31 | 90.21 | 92.71 | 95.96 | 96.26 | **97.12** | 93.15 | 94.25 |
| 4 | | 30.49 | 96.12 | 82.38 | 83.32 | 95.84 | 92.91 | 90.74 | **100.00** |
| 5 | | 85.24 | 94.82 | 95.11 | 94.29 | **98.91** | 96.42 | 97.50 | 98.12 |
| 6 | | 92.13 | 96.97 | 98.84 | 97.48 | 97.02 | 98.81 | **98.97** | 98.75 |
| 7 | | 00.00 | 95.29 | **97.04** | 82.65 | 88.58 | 94.75 | 83.69 | 95.83 |
| 8 | | 98.89 | 85.03 | 99.64 | 98.03 | 98.54 | 99.34 | 99.36 | **100.00** |
| 9 | | 00.00 | 96.59 | 59.38 | 87.14 | **98.65** | 72.99 | 97.95 | 76.67 |
| 10 | | 59.63 | 88.43 | 91.27 | 94.55 | 92.87 | **95.55** | 94.49 | 93.22 |
| 11 | | 84.75 | 88.11 | 96.53 | 98.22 | 94.60 | 98.04 | 98.28 | **99.21** |
| 12 | | 58.35 | 96.22 | 89.24 | 84.74 | 84.65 | 92.18 | 93.84 | **96.17** |
| 13 | | 96.37 | 94.94 | 98.52 | 92.96 | **100.00** | 98.29 | 96.89 | **100.00** |
| 14 | | 92.51 | 90.30 | 99.35 | 98.25 | 99.21 | 98.98 | 99.29 | **99.46** |
| 15 | | 55.65 | 75.19 | 89.83 | 83.43 | 93.71 | 93.27 | 95.09 | **96.76** |
| 16 | | 83.91 | 83.85 | 99.38 | 85.58 | **99.65** | 96.94 | 92.53 | 96.34 |
| OA(%) | | 77.30 ±0.62 | 89.71 ±1.03 | 93.44 ±0.36 | 94.35 ±1.01 | 95.26 ±0.39 | 96.43 ±0.45 | 96.48 ±0.78 | **97.35 ±0.05** |
| AA(%) | | 61.82 ±1.73 | 87.01 ±1.21 | 90.02 ±0.95 | 89.61 ±1.63 | 94.86 ±0.54 | 93.02 ±1.38 | 95.22 ±1.16 | **95.81 ±0.31** |
| Kappa×100 | | 73.80 ±0.72 | 88.87 ±0.42 | 92.59 ±0.18 | 93.45 ±0.25 | 94.64 ±0.47 | 95.04 ±0.13 | 95.99 ±0.88 | **97.00 ±0.06** |

### 5.2. Classification performance

In order to further verify the performance of MSRA-G, IP, UP and SV datasets were used in our experiment, and the classification performance was compared with six HSI classification methods of REF-SVM (Okwuashi and Ndehedehe, 2020), 3D-CNN (Li et al., 2017a), MSDN (Zhang et al., 2019), HybridSN (Roy et al., 2020), SSRN (Zhong et al., 2017), R-HybridSN (Feng et al., 2019) and MAFN (Li et al., 2021). The parameter setting of six comparative methods were consistent with the corresponding references. In addition, the categorical cross-entropy loss function and Adam optimizer were used while setting the $lr$ to 0.001. To observe the performance of MSRA-G, the network was trained for 200 epochs in total. During the training process, the model with the highest accuracy for the validation set was saved. The classification results were derived from 10 experiments by recording the standard deviation.

#### 5.2.1. Classification maps of ip dataset

On the IP dataset, the 5%, 5% and 90% samples were randomly selected from each class as the training set, validation set and test set, respectively. Due to the imbalance distribution of IP samples, the sample size of each class is too small, which makes classification very difficult. Therefore, the GANs was used to generate synthetic samples for MSRA-G, which achieves the goal of augmenting the datasets. PCA was then used to reduce the dimensions of HSIs to 14. The dimension-reduced HSIs were input into MSRA for classification, with the $w$ of 21 × 21 and the dropout proportions of 0.5 (Table 7 and Fig. 15).

It can be found from Table 7, MSRA-G has the highest OA, AA and Kappa of 97.35%, 95.81%, and 97.00%, respectively. Compared with other methods, the OA, AA Kappa of MSRA-G increases approximately by 0.87%–20.05%, 0.59%–34%, 1.01%–23.2%, respectively. The quite unbalanced numbers of class samples in IP dataset, low classification accuracies are caused. For example, the sample sizes of Alfalfa (1) and Grass-pasture-mowed (7) are very small, so the accuracy of traditional REF-SVM is not satisfactory. As a contrast, the DL-based 3D-CNN methods improve classification accuracy to some extent, showing their advantages of processing small sample data. MSRA-G improves the accuracies of these two classes to over 92%. MSDN using a dense structure to extract information on different scales, compared with 3D-CNN, improves the OA, AA and Kappa by 3.73%, 3.01% and 3.72%, respectively. In comparison with HybridSN and SSRN, the HybridSN-based improved R-HybridSN improves the OA by 2.08% and 1.17% and Kappa by 1.59% and 0.4%, respectively, by introducing the residual module to deepen the network depth, but the AA is slightly inadequate. MSRA-G achieves the highest accuracy in 8 classes, which uses GANs to augment the datasets to alleviate the problem of data imbalance and the MSRN module to fully extract the spatial–spectral features. Compared with MAFN, the OA, AA and Kappa of MSRA-G are improved by 0.87%, 0.59% and 1.01%, respectively, and the classification results are more stable. The classification map of MSRA-G has fewer misclassified pixels and the overall performance is the best (Fig. 15h).

#### 5.2.2. Classification maps of up dataset

On the UP dataset, the 1%, 1% and 98% samples were randomly selected from each class as the training, validation and test sets, respectively. The $d$ was set to 12 and MSRA was used as the classifier, where $w$ is set to 17 × 17 and the dropout is 0.4. As shown in Table 8, it can be found that MSRA-G can obtain superior performance compared with other methods, with the OA, AA and Kappa of 98.72%, 97.89%, and 98.30%, respectively, which yields significant improvement in the OA, AA, Kappa of 0.63%–12.23%, 0.69%–15.81% and 0.83%–16.39%, respectively. MSRA-G has the highest classification accuracies in 5 classes with at least 95% accuracy. It also performs best in Bitumen (7) class with only 13 training samples, while the other methods do not perform well in this class. For the Gravel (3) and Self-Blocking Bricks(8), the accuracies of other methods are less than 94% or lower accuracies, however, but MSRA-G can achieve more than 95%. Compared with traditional REF-SVM, the OA of 3D-CNN increases by 4.6% by deeply mining the spatial–spectral features of training samples. MSDN further increases the OA to 94.15% by extracting multi-scale information. With

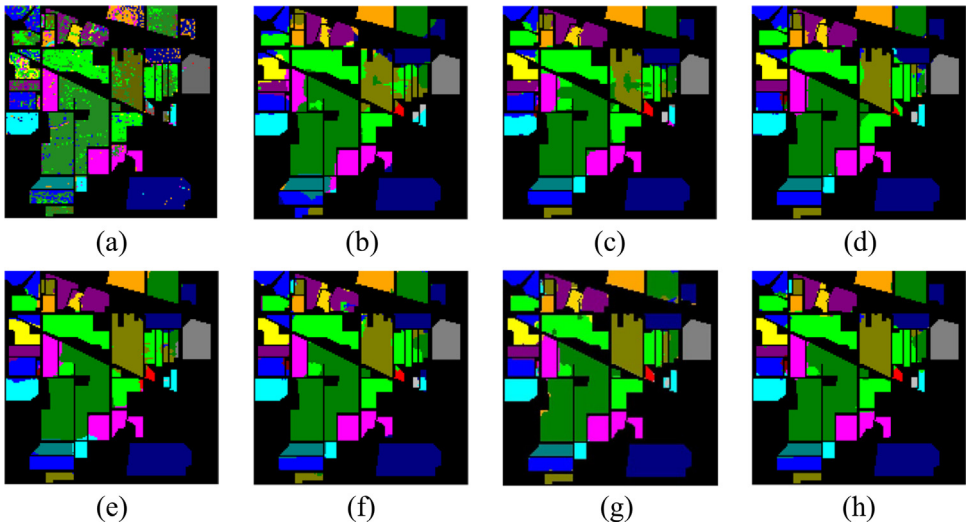



**Fig. 15.** Classification maps derived from the IP dataset. (a) REF-SVM. (b) 3D-CNN. (c) MSDN. (d) HybridSN. (e) SSRN. (f) R-HybridSN. (g) MAFN. (h) MSRA-G.

**Table 8**
Comparison of accuracies between MSRA-G and other methods in UP dataset.

| Class | Color | REF-SVM | 3D-CNN | MSDN | HybridSN | SSRN | R-HybridSN | MAFN | MSRA-G |
|---|---|---|---|---|---|---|---|---|---|
| 1 | | 86.67 | 90.73 | 97.94 | 91.05 | 99.53 | 97.40 | 98.21 | **99.71** |
| 2 | | 95.02 | 97.51 | 99.66 | 99.29 | 98.43 | 99.76 | 99.46 | **99.78** |
| 3 | | 51.29 | 75.78 | 80.57 | 91.41 | 80.15 | 91.58 | 88.54 | **95.38** |
| 4 | | 91.41 | 96.95 | 92.11 | 90.19 | **100.00** | 93.57 | 99.43 | 97.97 |
| 5 | | 97.50 | 98.68 | 99.86 | 96.88 | **100.00** | 99.38 | 99.52 | 99.62 |
| 6 | | 68.22 | 81.65 | 88.97 | 94.48 | 95.29 | 98.90 | **99.78** | 96.71 |
| 7 | | 65.92 | 71.34 | 80.41 | 95.96 | 91.60 | 96.27 | 97.92 | **98.24** |
| 8 | | 84.17 | 81.11 | 82.48 | 89.90 | 87.67 | 93.72 | 91.92 | **97.26** |
| 9 | | 98.49 | 97.67 | 97.22 | 80.82 | **99.52** | 89.19 | 100.00 | 96.33 |
| OA(%) | | 86.49 | 91.09 | 94.15 | 95.08 | 96.29 | 97.24 | 98.09 | **98.72** |
| | | ±0.46 | ±0.52 | ±0.55 | ±0.43 | ±0.19 | ±0.46 | ±0.24 | **±0.07** |
| AA(%) | | 82.08 | 87.93 | 91.02 | 92.22 | 94.71 | 95.53 | 97.20 | **97.89** |
| | | ±0.57 | ±1.10 | ±0.37 | ±0.67 | ±0.23 | ±0.36 | ±0.34 | **±0.09** |
| Kappa×100 | | 81.91 | 87.82 | 92.17 | 93.75 | 94.69 | 96.29 | 97.47 | **98.30** |
| | | ±1.02 | ±0.72 | ±0.80 | ±0.31 | ±0.12 | ±0.07 | ±0.45 | **±0.02** |

the 3D–2D hybrid network characteristics of HybridSN and the residual structure of SSRN, R-HybridSN effectively improves the classification performance. The OA, AA and Kappa are respectively improved by 6.15%, 7.6% and 8.47%, compared with 3D-CNN using single 3D convolution kernel, showing that 3D–2D-CNN model is more suitable for the HSI classification under limited samples to a certain extent. MSRA-G performs better in OA, AA and Kappa than MAFN. Meanwhile, it can be seen from Fig. 16, there are fewer noise pixels and greater spatial continuity for a certain class.

### 5.2.3. Classification maps of sv dataset

On the SV dataset, because the sample size is relatively large, we randomly selected 0.5%, 0.5% and 99% samples from each class as training, validation and test sets, respectively. The $d$ was set to 12, $w$ to $21 \times 21$ and dropout proportion to 0.4. It can be observed that MSRA-G has the best performance, with the OA, AA and Kappa of 98.72%, 98.94% and 98.58%, respectively (Table 9). In addition, it yields significant improvement in OA, AA, Kappa of 1.48%–11.37%, 1.84%–11.08% and 1.42%–12.72%, respectively. Among all the classification methods, REF-SVM is the worst, with the OA, AA and Kappa of less than 88%, and its classification map also shows a large number of misclassified pixels. The 3D-CNN, MSDN and SSRN alleviate the phenomenon to some extent, but they underperform in the classes of Hay-windrowed (8) and Buildings-Grass-Trees-Drives (15). In addition, HybridSN is more competitive in classification compared with R-HybridSN, the OA, AA and Kappa reach 97.24%, 97.10% and 97.16%, respectively. As a contrast, MSRA-G performs well in all the classes and achieves the highest accuracies in nine classes. Compared with HybridSN, the OA, AA and Kappa yield significant improvement of 1.48%, 1.84% and 1.42%, respectively. Compared with MAFN, the OA, AA and Kappa yield significant improvement of 2.3%, 1.85% and 2.11%, respectively. Meanwhile, the classification map is also smoother and performs best overall (Fig. 17).

## 6. Discussion

### 6.1. Experimental investigation on training percentages

In order to further verify the classification performance of MSRA-G with limited samples, four different proportions of training samples were set up on the three datasets. For IP dataset, the proportions were respectively 1%, 3%, 5% and 10%, while they were respectively 0.5%, 1%, 5% and 10% for UP and SA datasets. Unsurprisingly, as the training

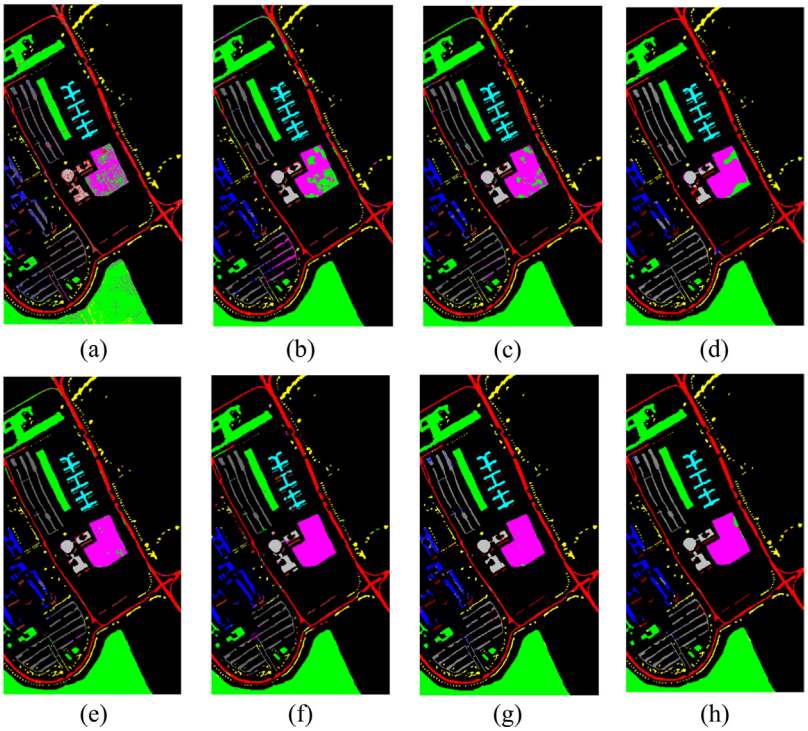



**Fig. 16.** Classification maps derived from the UP dataset. (a) REF-SVM. (b) 3D-CNN. (c) MSDN. (d) HybridSN. (e) SSRN. (f) R-HybridSN. (g) MAFN. (h) MSRA-G.

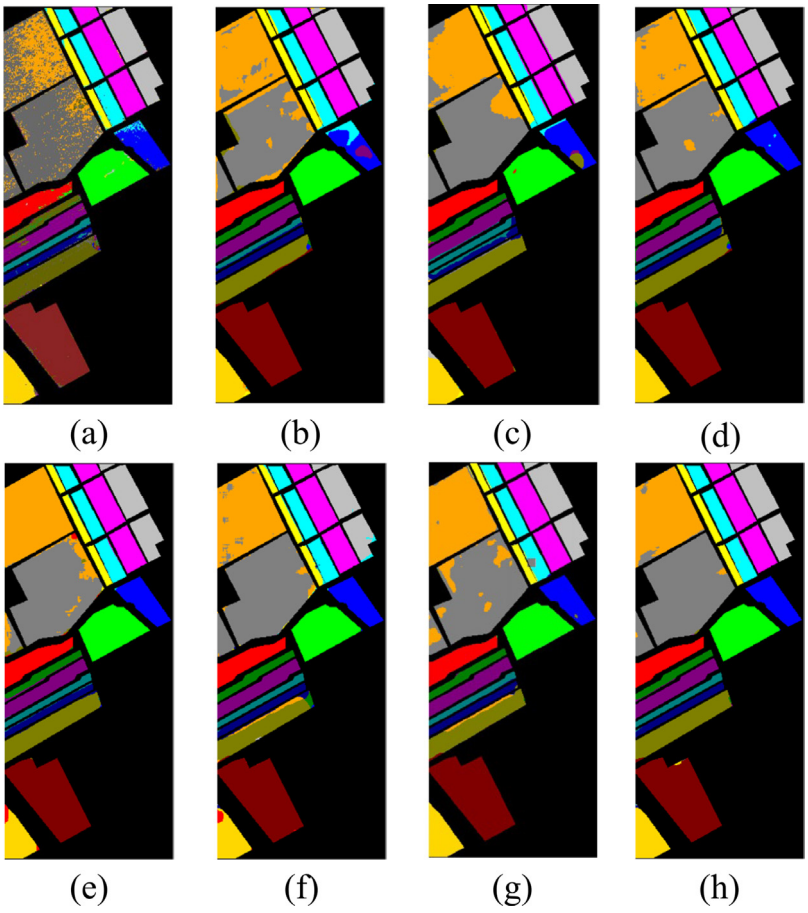



**Fig. 17.** Classification maps derived from the SV dataset. (a) REF-SVM. (b) 3D-CNN. (c) MSDN. (d) HybridSN. (e) SSRN. (f) R-HybridSN. (g) MAFN. (h) MSRA-G.

**Table 9**
Comparison of accuracies between MSRA-G and other methods in SA dataset.

| Class | Color | REF-SVM | 3D-CNN | MSDN | HybridSN | SSRN | R-HybridSN | MAFN | MSRA-G |
|---|---|---|---|---|---|---|---|---|---|
| 1 | | 95.95 | 98.56 | 97.42 | 98.49 | 99.12 | **100.00** | **100.00** | 99.95 |
| 2 | | 98.54 | 90.10 | 96.46 | 99.35 | **100.00** | 99.16 | 99.52 | 99.95 |
| 3 | | 88.61 | 95.59 | 99.36 | 98.32 | 90.07 | 98.13 | 99.84 | **100.00** |
| 4 | | 94.74 | 96.36 | 96.60 | 96.90 | 95.22 | 96.39 | 94.85 | **99.42** |
| 5 | | 98.61 | 94.18 | 94.06 | 97.77 | **99.28** | 98.35 | 98.75 | 98.53 |
| 6 | | 99.42 | 94.53 | 99.75 | 98.43 | 99.36 | 98.40 | 99.48 | **100.00** |
| 7 | | 99.41 | 97.83 | 98.09 | 98.45 | 98.67 | 97.38 | 100.00 | **100.00** |
| 8 | | 88.03 | 87.56 | 90.54 | 93.30 | 88.44 | 95.86 | 96.54 | **99.06** |
| 9 | | 96.79 | 98.46 | 98.25 | **100.00** | 98.86 | 98.49 | 99.42 | 99.58 |
| 10 | | 91.14 | 97.35 | 93.33 | 96.88 | 98.19 | 96.59 | 98.05 | **98.37** |
| 11 | | 20.04 | 82.97 | 95.73 | 94.62 | 93.66 | **97.14** | **99.34** | 95.18 |
| 12 | | 95.41 | 96.91 | 97.76 | 95.98 | 99.86 | 98.38 | 99.57 | **100.00** |
| 13 | | 97.37 | 97.09 | 93.97 | 95.96 | **100.00** | 91.02 | 99.13 | **100.00** |
| 14 | | 90.80 | 95.98 | 96.12 | 98.03 | 98.24 | 94.55 | 95.49 | **99.06** |
| 15 | | 53.43 | 84.01 | 88.46 | 92.60 | 90.31 | **95.16** | 87.21 | 94.80 |
| 16 | | 97.55 | 96.06 | 96.39 | 98.50 | **100.00** | 97.97 | 99.95 | 99.11 |
| OA(%) | | 87.35 | 92.93 | 94.67 | 97.24 | 96.09 | 97.18 | 96.42 | **98.72** |
| | | ±0.35 | ±0.38 | ±0.16 | ±0.12 | ±0.22 | ±0.33 | ±0.49 | **±0.10** |
| AA(%) | | 87.86 | 93.98 | 95.77 | 97.10 | 96.83 | 97.06 | 97.09 | **98.94** |
| | | ±0.41 | ±0.54 | ±0.07 | ±0.10 | ±0.32 | ±0.21 | ±0.56 | **±0.08** |
| Kappa×100 | | 85.86 | 92.05 | 94.36 | 97.16 | 95.49 | 97.12 | 96.47 | **98.58** |
| | | ±0.82 | ±0.36 | ±0.18 | ±0.13 | ±0.25 | ±0.04 | ±0.43 | **±0.03** |

sample size increases, so do the OAs of all classification methods. I can be found that MSRA-G performs better than others in any case. As can be seen from Fig. 18(a), REF-SVM performs worst in the seven methods on the IP dataset, with the OA of below 85% in any proportions. When the training percentage is 1%, SSRN is inferior to HybridSN. MAFN is superior to other comparison methods, but MSRA-G is always better than it in OA, especially under very few training samples. As shown in Fig. 18(b), even with 0.5% of UP training samples, the OA of REF-SVM can reach more than 82%, HybridSN increases it to more than 92%, while MSRA-G can reach 96%. The SSRN, R-HybridSN, MAFN and MSRA-G continue to perform well in all the cases, showing the stability of the three methods. Obviously, MSRA-G performs better than SSRN, R-HybridSN and MAFN in all the proportions. All the methods except REF-SVM perform well with the OA of exceeding 92%, because of the relative abundance of SV samples (Fig. 18(c)). With the training sample size increases, the accuracy gap between all the methods becomes smaller. Interestingly, for SV dataset, HybridSN shows strong competitiveness, and the classification accuracies are higher than SSRN, R-HybridSN and MAFN. Moreover, MSRA-G is superior to HybridSN in all the percentages and has better classification performance.

## 6.2. Ablation study

### 6.2.1. Effectiveness of augmentation

We aim to study the quality of synthetic spectra by GANs and its effect on classification results. Firstly, three GANs were trained using random training samples on the three datasets, and then a certain class was selected from each dataset to make an intuitive comparison by plotting the mean spectra and standard deviation between real and synthetic samples. As shown in Fig. 19, the fourth-class Corn, the fourth-class Trees and the thirteenth-class Wheat were selected from the IP, UP and SV datasets, respectively. Their spectral reflectivities were plotted using the real and synthetic samples. It can be found that the spectral shapes are precisely learned by GANs. In addition, the samples near cluster center tend to have higher classification accuracies (Wang et al., 2021b). Compared with real samples, synthetic samples are closer to the cluster center, that is, the new augmented samples have better cluster degree than real samples.

To further verify the effectiveness of augmentation samples, the classification models were trained under different augmentation factors $N$ in the training samples, and the OA on the real test samples was used as an evaluation indicator. Fig. 20 is an OA histogram of different augmentation factors $N$ on the three datasets, where $N_0$ represents no augmentation and $N_1$–$N_5$ represent the augmented multiples than $N_0$. For example, $N_1$ indicates that the number of augmented training samples is more than double $N_0$. An expanded training sample can effectively increase the OA. The OAs of IP, UP and SA datasets reach the highest when the augmented multiples are $N_3$, $N_2$, and $N_2$, respectively. Obviously, the augmented training samples multiples are not endless. For UP dataset, when OA reaches the highest, it tends to decrease instead of increase as the $N$ continues to increase. The reason may be that too many similar samples will affect the classification performance.

### 6.2.2. Effectiveness of attention mechanism

To validate the effectiveness of channel attention mechanism and spatial attention mechanism, three additional experiments were performed on the three datasets, i.e., without attention mechanism (denoted as Model1), only with the spatial attention mechanism (denoted as Model2) and only using the channel attention mechanism (denoted as Model3). It can be observed that the models that use attention mechanism have higher OAs, which proves the effectiveness of attention mechanism (Fig. 21). Furthermore, the Model3 performs better than Model2, which demonstrates the channel-attention mechanism is more effective than spatial attention mechanism. MSRA-G has the highest OAs as a whole.

## 6.3. Experimental investigation on running time

Table 10 shows the comparison of training and test time for seven methods on the IP, UP and SV datasets, respectively. It is obvious that

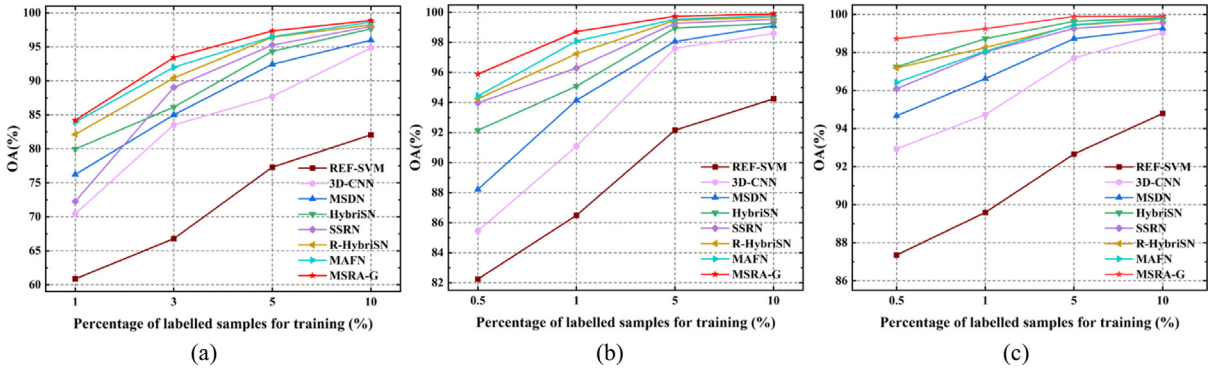



**Fig. 18.** The OAs of REF-SVM, 3D-CNN, MSDN, HybridSN, SSRN, R-HybridSN, MAFN and MSRA-G with varying proportions of training samples on the (a) IP, (b) UP and (c) SV.

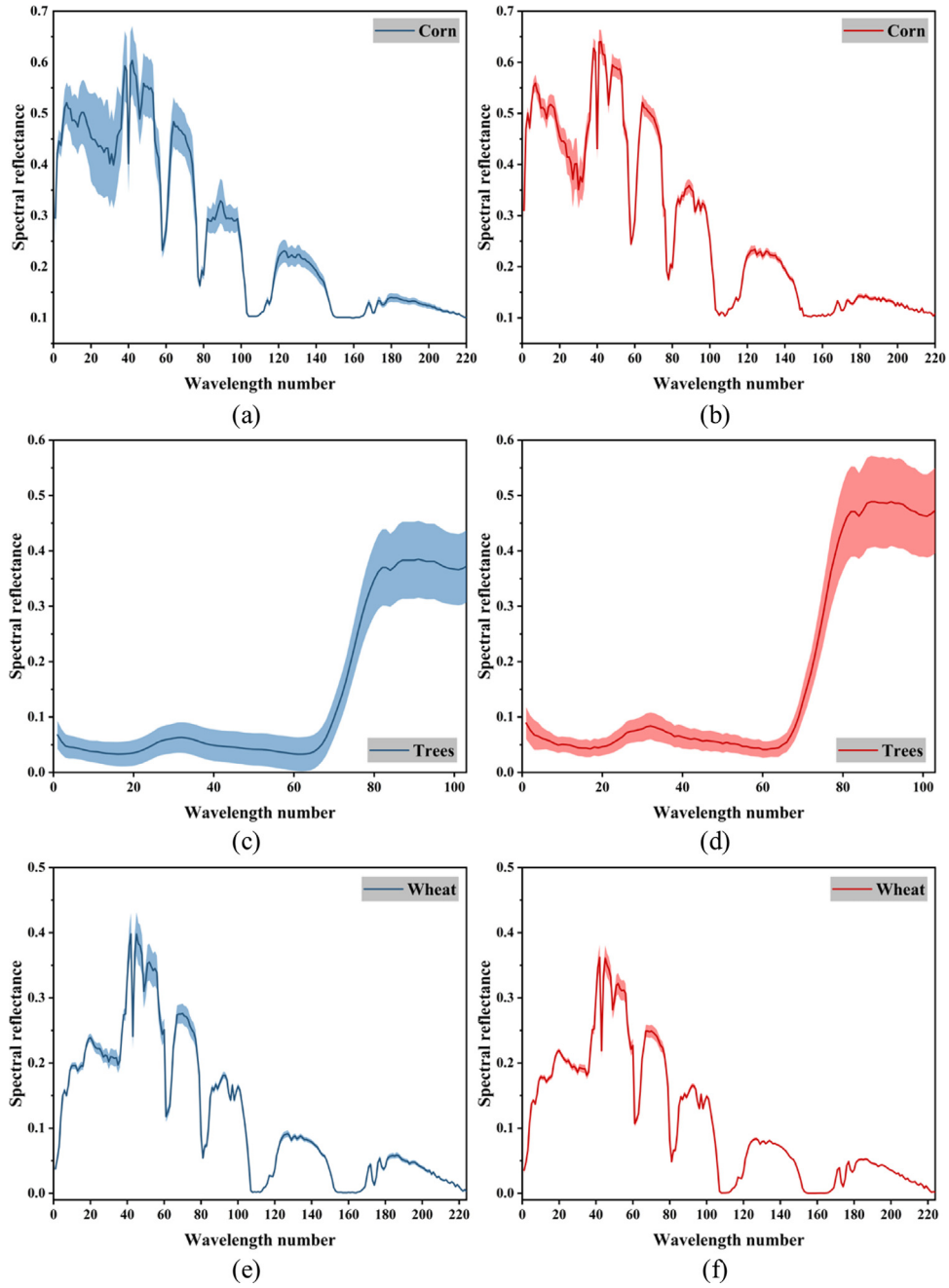



**Fig. 19.** The true spectral (blue) reflectance and synthesized spectral (red) reflectance on the three datasets. (a) and (b) show the 4th class of IP dataset. (c) and (d) show the 4th class of UP dataset. (e) and (f) show the 13th class of SV dataset.

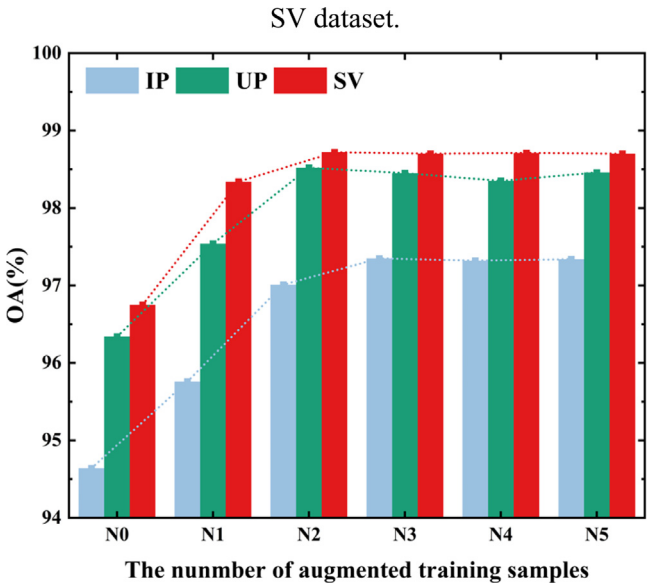



**Fig. 20.** Comparison of OAs for different augmented factors $N$ on the three datasets.

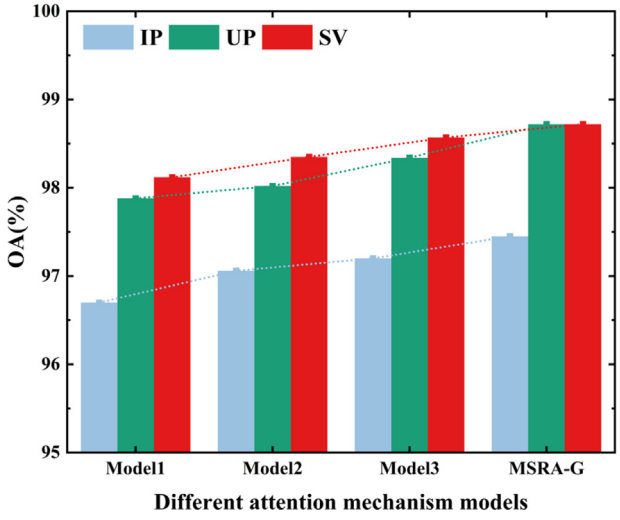



**Fig. 21.** Comparison of OAs with and without attention mechanism models on the three datasets.

RBF-SVM takes less time than 3D cubes-based classification methods because of its simple structure and fewer parameters. 3D-CNN requires more time to train the network, due to more training parameters. The training time of MSDN and SSRN is less than 3D-CNN, but the classification accuracy is much higher than 3D-CNN, showing the superiority of MSDN and SSRN. Compared with HybridSN, R-HybridSN with better classification performance takes longer time to train, because of the use of residual modules. MSRA-G takes slightly more time in the training and less time in the test than R-HybridSN. The reason is that it uses multi-scale attention module and GANs to alleviate the problem of sample imbalance, so more time are needed to train more parameters. MSRA-G takes about the same training time as R-HybridSN, but achieves higher classification accuracy than R-HybridSN.

## 7. Conclusion

In view of insufficient HSI feature extraction under limited samples, the classification accuracy is greatly affected. We propose a classification method combining MSRA and GANs (MSRA-G). Our work is mainly divided into two aspects. Firstly, the training samples are augmented by using the GANs. Secondly, a spatial–spectral feature extraction framework is designed. For the designed data augmentation method, it is essential to learn the original spectral properties from HSIs through the game between G and the D, thereby creating new samples reasonably. Experiments prove that augmenting the training samples can effectively improve HSI classification accuracy to a certain extent. In addition, a classification framework based on the hybrid networks is proposed. The multi-scale feature extraction module is used to fully extract the spatial–spectral features of HSIs at different scales. Different weights are given to channel dimension and spatial dimension through introducing attention mechanism, so the important features of HSIs are selectively learned. Meanwhile, in order to alleviate the phenomenon of gradient disappearance, the residual connection mode is used. In addition, the BN and dropout layer are also introduced to prevent overfitting. In comparison with several popular methods, the classification performance of proposed MSRA-G is very competitive. It also achieves satisfying classification accuracy, even under the condition of limited samples.

**Table 10**
Training and test time of seven methods on the three datasets.

| Method | Time | IP | UP | SV |
|---|---|---|---|---|
| RBF-SVM | Training (s) | 3.85 | 7.53 | 2.19 |
| | Test (s) | 1.34 | 5.07 | 3.44 |
| 3D-CNN | Training (m) | 11.35 | 11.26 | 10.70 |
| | Test (s) | 25.48 | 33.34 | 36.50 |
| MSDN | Training (m) | 10.28 | 12.94 | 10.02 |
| | Test (s) | 21.48 | 31.30 | 20.58 |
| HybridSN | Training (m) | 6.03 | 8.15 | 5.35 |
| | Test (s) | 5.8 | 10.66 | 8.24 |
| SSRN | Training (m) | 9.58 | 15.48 | 8.6 |
| | Test (s) | 15.48 | 30.10 | 25.21 |
| R-HybridSN | Training (m) | 8.69 | 9.15 | 7.52 |
| | Test (s) | 22.15 | 46.27 | 33.14 |
| MAFN | Training (m) | 12.16 | 17.80 | 15.33 |
| | Test (s) | 21.12 | 37.47 | 30.58 |
| MSRA-G | Training (m) | 9.93 | 10.42 | 7.65 |
| | Test (s) | 20.75 | 28.12 | 18.29 |

## CRediT authorship contribution statement

**Jinling Zhao:** Formal analysis, Writing – original draft, Funding acquisition. **Lei Hu:** Investigation, Formal analysis, Validation. **Linsheng Huang:** Formal analysis, Validation. **Chuanjian Wang:** Methodology, Writing – review & editing. **Dong Liang:** Conceptualization, Writing – review & editing.

## Declaration of competing interest

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

## Data availability

Data will be made available on request.

## Acknowledgments

This work was supported by National Natural Science Foundation of China (31971789), Natural Science Foundation of Anhui Province, China (2008085MF184) and Science and Technology Major Project of Anhui Province (202003a06020016). We also thank the anonymous reviewers for their feedback and helpful suggestions.

## References


Audebert, N., Le Saux, B., Lefèvre, S., 2019. Deep learning for classification of hyperspectral data: A comparative review. IEEE Geosci. Remote Sens. Mag. 7 (2), 159–173.

Cao, F., Yang, Z., Ren, J., Chen, G., Shen, Y., 2019. Local block multilayer sparse extreme learning machine for effective feature extraction and classification of hyperspectral images. IEEE Trans. Geosci. Remote Sens. 57 (8), 5580–5594.

Chen, H., Li, W., Shi, Z., 2021. Adversarial instance augmentation for building change detection in remote sensing images. IEEE Trans. Geosci. Remote Sens. 60, 5603216.

Chen, Y., Lin, Z., Zhao, X., Wang, G., Gu, Y., 2014. Deep learning-based classification of hyperspectral data. IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 7 (6), 2094–2107.

Chen, Y., Zhao, X., Jia, X., 2015. Spectral-spatial classification of hyperspectral data based on deep belief network. IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 8 (6), 2381–2392.

Cheng, G., Xie, X., Han, J., Guo, L., Xia, G., 2020. Remote sensing image scene classification meets deep learning: Challenges, methods, benchmarks, and opportunities. IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens. 13, 3735–3756.

Courtrai, L., Pham, M.-T., Lefèvre, S., 2020. Small object detection in remote sensing images based on super-resolution with auxiliary generative adversarial networks. Remote Sens. 12 (19), 3152.

Deng, F., Pu, S., Chen, X., Shi, Y., Yuan, T., Pu, S., 2018. Hyperspectral image classification with capsule network using limited training samples. Sensors 18 (9), 3153.

Dong, Z., Cai, Y., Cai, Z., Liu, X., Yang, Z., Zhuge, M., 2020. Cooperative spectral–spatial attention dense network for hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. 18 (5), 866–870.

Fang, B., Li, Y., Zhang, H., Chan, J.C.W., 2019. Hyperspectral images classification based on dense convolutional networks with spectral-wise attention mechanism. Remote Sens. 11 (2), 159.

Feng, F., Wang, S., Wang, C., Zhang, J., 2019. Learning deep hierarchical spatial–spectral features for hyperspectral image classification based on residual 3D-2D CNN. Sensors 19 (23), 5276.

Gao, H., Miao, Y., Cao, X., Li, C., 2021. Densely connected multiscale attention network for hyperspectral image classification. IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 14, 2563–2576.

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al., 2014. Generative adversarial nets. In: Proceedings of the 27th International Conference on Neural Information Processing Systems. NIPS, pp. 2672–2680.

Gu, Y., Xu, Y., Guo, B.F., 2018. Hyperspectral image classification by combination of spatial–spectral features and ensemble extreme learning machines. Acta Geod. Cartogr. Sin. 47 (9), 1238.

Guo, Y., Cao, H., Han, S., Sun, Y., Bai, Y., 2018. Spectral–spatial hyperspectral image classification with K-nearest neighbor and guided filter. IEEE Access 6, 18582–18591.

Hang, R., Li, Z., Liu, Q., Ghamisi, P., Bhattacharyya, S.S., 2020. Hyperspectral image classification with attention-aided CNNs. IEEE Trans. Geosci. Remote Sens. 59 (3), 2281–2293.

Haut, J.M., Paoletti, M.E., Plaza, J., Plaza, A., Li, J., 2019. Visual attention-driven hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 57 (10), 8065–8080.

Hay, A.M., 1988. The derivation of global estimates from a confusion matrix. Int. J. Remote Sens. 9 (8), 1395–1398.

Hennessy, A., K., Clarke., Lewis, M., 2020. Hyperspectral classification of plants: A review of waveband selection generalizability. Remote Sens. 12 (1), 113.

Hu, W., Huang, Y., Wei, L., Zhang, F., Li, H., 2015. Deep convolutional neural networks for hyperspectral image classification. J. Sensors 2015, 258619.

Hu, J., Shen, L., Sun, G., 2018. Squeeze-and-excitation networks. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, IEEE, pp. 7132–7141.

Huang, L., Chen, Y., 2020. Dual-path siamese CNN for hyperspectral image classification with limited training samples. IEEE Geosci. Remote Sens. Lett. 18 (3), 518–522.

Kang, X., Xiang, X., Li, S., Benediktsson, J.A., 2017. PCA-based edge-preserving features for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 55 (12), 7140–7151.

Khodadadzadeh, M., Li, J., Plaza, A., Bioucas-Dias, J.M., 2014. A subspace-based multinomial logistic regression for hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. 11 (12), 2105–2109.

Li, L., Ge, H., Gao, J., 2017a. A spectral-spatial kernel-based method for hyperspectral imagery classification. Adv. Space Res. 59 (4), 954–967.

Li, J., Qian, Y., 2011. Dimension reduction of hyperspectral images with sparse linear discriminant analysis. In: 2011 IEEE International Geoscience and Remote Sensing Symposium. IGARSS, IEEE, pp. 2927–2930.

Li, S., Song, W., Fang, L., Chen, Y., Ghamisi, P., Benediktsson, J.A., 2019. Deep learning for hyperspectral image classification: An overview. IEEE Trans. Geosci. Remote Sens. 57 (9), 6690–6709.

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

Li, J., Zhao, X., Li, Y., Du, Q., Xi, B., Hu, J., 2018. Classification of hyperspectral imagery using a new fully convolutional neural network. IEEE Geosci. Remote Sens. Lett. 15 (2), 292–296.

Li, Z., Zhao, X., Xu, Y., Li, W., Zhai, L., Fang, Z., et al., 2021. Hyperspectral image classification with multiattention fusion network. IEEE Geosci. Remote Sens. Lett. 19, 5503305.

Lu, Z., Xu, B., Sun, L., Zhan, T., Tang, S., 2020. 3-D channel and spatial attention based multiscale spatial–spectral residual network for hyperspectral image classification. IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 13, 4311–4324.

Ma, S., Fu, J., Chen, C.W., Mei, T., 2018. DA-GAN: Instance-level image translation by deep attention generative adversarial networks. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, IEEE, pp. 5657–5666.

Makantasis, K., Karantzalos, K., Doulamis, A., Doulamis, N., 2015. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In: 2015 IEEE International Geoscience and Remote Sensing Symposium. IGARSS, IEEE, pp. 4959–4962.

Mei, X., Pan, E., Ma, Y., Dai, X., Huang, J., Fan, F., et al., 2019. Spectral-spatial attention networks for hyperspectral image classification. Remote Sens. 11 (8), 963.

Mou, L., Zhu, X.X., 2019. Learning to pay attention on spectral domain: A spectral attention module-based convolutional network for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 58 (1), 110–122.

Okwuashi, O., Ndehedehe, C.E., 2020. Deep support vector machine for hyperspectral image classification. Pattern Recogn. 103, 107298.

Pan, B., Shi, Z., Xu, X., 2018. MugNet: Deep learning for hyperspectral image classification using limited samples. ISPRS J. Photogramm. Remote Sens. 145, 108–119.

Paoletti, M.E., Haut, J.M., Fernandez-Beltran, R., Plaza, J., Plaza, A.J., Pla, F., 2018. Deep pyramidal residual networks for spectral–spatial hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 57 (2), 740–754.

Roy, S.K., Krishna, G., Dubey, S.R., Chaudhuri, B.B., 2020. HybridSN: Exploring 3D-2D CNN feature hierarchy for hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. 17 (2), 277–281.

Shi, C., Pun, C.-M., 2018. Multi-scale hierarchical recurrent neural networks for hyperspectral image classification. Neurocomputing 294, 82–93.

Sothe, C., De Almeida, C.M., Schimalski, M.B., La Rosa, L.E.C., Castro, J.D.B., Feitosa, R.Q., et al., 2020. Comparative performance of convolutional neural network, weighted and conventional support vector machine and random forest for classifying tree species using hyperspectral and photogrammetric data. GISci. Remote Sens. 57 (3), 369–394.

Sun, W., Du, Q., 2019. Hyperspectral band selection: A review. IEEE Geosci. Remote Sens. Mag. 7 (2), 118–139.

Sun, H., Zheng, X., Lu, X., Wu, S., 2019. Spectral–Spatial attention network for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 58 (5), 3232–3245.

Tembine, H., 2019. Deep learning meets game theory: Bregman-based algorithms for interactive deep generative adversarial networks. IEEE Trans. Cybern. 50 (3), 1132–1145.

Tu, B., Li, N., Fang, L., He, D., Ghamisi, P., 2019. Hyperspectral image classification with multi-scale feature extraction. Remote Sens. 11 (5), 534.

Wang, W., Dou, S., Jiang, Z., Sun, L., 2018. A fast dense spectral–Spatial convolution network framework for hyperspectral images classification. Remote Sens. 10 (7), 1068.

Wang, J., Gao, F., Dong, J., Du, Q., 2021a. Adaptive DropBlock-Enhanced generative adversarial networks for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 59 (6), 5040–5053.

Wang, W., Liu, X., Mou, X., 2021b. Data augmentation and spectral structure features for limited samples hyperspectral classification. Remote Sens. 13 (4), 547.

Woo, S., Park, J., Lee, J.Y., Kweon, I.S., 2018. CBAM: Convolutional block attention module. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 3–19.

Wu, P., Cui, Z., Gan, Z., Liu, F., 2020. Residual group channel and space attention network for hyperspectral image classification. Remote Sens. 12 (12), 2035.

Xia, J., Chanussot, J., Du, P., He, X., 2013. (Semi-) supervised probabilistic principal component analysis for hyperspectral remote sensing image classification. IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 7 (6), 2224–2236.

Xia, J., Falco, N., Benediktsson, J.A., Du, P., Chanussot, J., 2017. Hyperspectral image classification with rotation random forest via KPCA. IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 10 (4), 1601–1609.

Yang, X., Ye, Y., Li, X., Lau, R.Y., Zhang, X., Huang, X., 2018. Hyperspectral image classification with deep learning models. IEEE Trans. Geosci. Remote Sens. 56 (9), 5408–5423.

Yu, C., Han, R., Song, M., Liu, C., Chang, C.I., 2020. A simplified 2D-3D CNN architecture for hyperspectral image classification based on spatial–spectral fusion. IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 13, 2485–2501.

Zhan, Y., Hu, D., Wang, Y., Yu, X., 2018. Semisupervised hyperspectral image classification based on generative adversarial networks. IEEE Geosci. Remote Sens. Lett. 15 (2), 212–216.

Zhang, C., Li, G., Du, S., 2019. Multi-scale dense networks for hyperspectral remote sensing image classification. IEEE Trans. Geosci. Remote Sens. 57 (11), 9201–9222.

Zhang, Y., Li, W., Sun, W., Tao, R., Du, Q., 2022a. Single-source domain expansion network for cross-scene hyperspectral image classification. arXiv preprint arXiv: 2209.01634.

Zhang, Y., Li, W., Tao, R., Peng, J., Du, Q., Cai, Z., 2021a. Cross-scene hyperspectral image classification with discriminative cooperative alignment. IEEE Trans. Geosci. Remote Sens. 59 (11), 9646–9660.

Zhang, Y., Li, W., Zhang, M., Qu, Y., Tao, R., Qi, H., 2021b. Topological structure and semantic information transfer network for cross-scene hyperspectral image classification. IEEE Trans. Neur. Net. Learn. Syst http://dx.doi.org/10.1109/TNNLS.2021.3109872.

Zhang, Y., Li, W., Zhang, M., Wang, S., Tao, R., Du, Q., 2022b. Graph in formation aggregation cross-domain few-shot learning for hyperspectral image classification. IEEE Trans. Neur. Net. Learn. Syst http://dx.doi.org/10.1109/TNNLS.2022.3185795.

Zhang, Q., Wei, X., Xiang, D., Sun, M., 2018. Supervised PolSAR image classification with multiple features and locally linear embedding. Sensors 18 (9), 3054.

Zhang, L., Zhang, L., 2022. Artificial intelligence for remote sensing data analysis: A review of challenges and opportunities. IEEE Geosci. Remote Sens. Mag. 10 (2), 270–294.

Zhao, C., Gao, X., Wang, Y., Li, J., 2016. Efficient multiple-feature learning-based hyperspectral image classification with limited training samples. IEEE Trans. Geosci. Remote Sens. 54 (7), 4052–4062.

Zhong, Z., Li, J., Luo, Z., Chapman, M., 2017. Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework. IEEE Trans. Geosci. Remote Sens. 56 (2), 847–858.

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

Zhu, M., Jiao, L., Liu, F., Yang, S., Wang, J., 2020. Residual spectral–spatial attention network for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 59 (1), 449–462.