Hindawi
Journal of Sensors
Volume 2023, Article ID 9854512, 1 page
https://doi.org/10.1155/2023/9854512

Retraction

Retracted: Advances in Hyperspectral Image Classification with a Bottleneck Attention Mechanism Based on 3D-FCNN Model and Imaging Spectrometer Sensor

Journal of Sensors

Received 12 December 2023; Accepted 12 December 2023; Published 13 December 2023

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] D. Yuan, X. Xie, G. Gao, and J. Xiao, "Advances in Hyperspectral Image Classification with a Bottleneck Attention Mechanism Based on 3D-FCNN Model and Imaging Spectrometer Sensor," *Journal of Sensors*, vol. 2022, Article ID 7587157, 16 pages, 2022.

Hindawi
Journal of Sensors
Volume 2022, Article ID 7587157, 16 pages
https://doi.org/10.1155/2022/7587157

Research Article

Advances in Hyperspectral Image Classification with a Bottleneck Attention Mechanism Based on 3D-FCNN Model and Imaging Spectrometer Sensor

Deren Yuan, Xiaochun Xie, Gao Gao, and Ju Xiao

School of Physics and Electronic Information, Gannan Normal University, Ganzhou 341000, China

Correspondence should be addressed to Xiaochun Xie; xiexiaochun@gnnu.edu.cn

Received 9 June 2022; Revised 22 June 2022; Accepted 29 July 2022; Published 16 August 2022

Academic Editor: C. Venkatesan

Deep learning approaches have significantly enhanced the classification accuracy of hyperspectral images (HSIs). However, the classification process still faces difficulties such as those posed by high data dimensions, large data volumes, and insufficient numbers of labeled samples. To enhance the classification accuracy and reduce the data dimensions and training needed for labeled samples, a 3D fully convolutional neural network (3D-FCNN) model was developed by including a bottleneck attention module. In such a model, the convolutional layer replaces the downsampling layer and the fully connected layer, and 3D full convolution is adopted to replace the commonly used 2D and 1D convolution operations. Thus, the loss of data in the dimensionality reduction process is effectively avoided. The bottleneck attention mechanism is introduced in the FCNN to reduce the redundancy of information and the number of labeled samples. The proposed method was compared to some advanced HSI classification approaches with deep networks, and five common HSI datasets were employed. The experiments showed that our network could achieve considerable classification accuracies by reducing the data dimensionality using a small number of labeled samples, thereby demonstrating its potential merits in the HSI classification process .

1. Introduction

The hyperspectral image (HSI) classification process is vital for the use of hyperspectral remote sensing data. The spectral resolution of HSI data ranges from visible light to short-wave infrared, with wavelengths reaching the order of nanometers. By exploiting the spectral characteristics of HSIs, one can effectively distinguish various objects, which has enabled the application of HSIs in a wide range of disciplines such as agriculture, early warning systems in disaster management, and national defense. Deep learning models for HSI classification are well developed. Many techniques, such as auto encoder [1], deep belief network [2], recurrent neural network [3], and convolutional neural network (CNN) models (e.g., the network described by Gu et al. [4]), are commonly used.

A convolution-related neural framework refers to a typical approach for deep learning [5–8] and HSI classification. It employs three types of models for the processing of a variety of characteristics by the CNN. The first type represents a 1D-CNN that uses only spectral data to extract the characteristics. This method requires a considerable number of training samples. The second type involves a spatial characteristics-based approach termed a 2D-CNN. Spatial characteristics are written by using a sparse representation method [9]; however, Makantasis et al. [10] developed a classification framework that uses particular scenes. The third type refers to the 3D-CNN approach that exploits spectral and spatial characteristics. It uses information on changes in local signals contained in spatial and spectral data without any pre- and postprocessing operations. The 3D convolution technique was initially employed to process videos, and it is currently used extensively in the HSI classification process [11–15]. Other methods are referred to as hybrid CNNs, and many such approaches have been developed for various uses [16, 17]. For instance, various hybrid approaches that adopt 1D-CNN and 2D-CNN were presented by Yang et al. [18] and Zhang et al. [17].

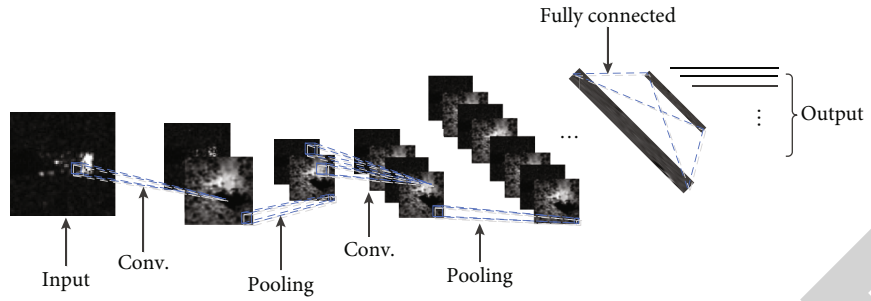


Figure 1: Convolutional neural network (CNN) structure.

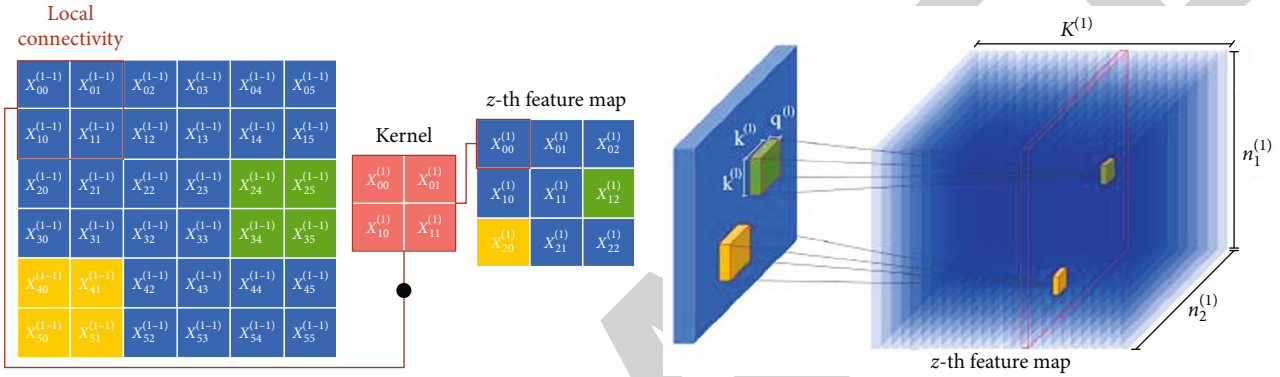


Figure 2: Convolutional layer working mode.

Previous studies on HSI classification based on deep learning have primarily discussed the building of deep networks to enhance accuracy. However, the number of training parameters was proportional to the complexity of the networks. For instance, approximately 360,000 training parameters were used in the classification network proposed by Zhong et al. [19]. Hamida et al. [20] proposed a 3D-1D hybrid CNN method that employs a maximum of 61,949 parameters. In the network proposed by Roy et al. [21], a 3D-2D hybrid CNN used 5,122,176 parameters. The adoption of such a high number of training parameters makes it difficult to train the network and is liable to result in overfitting. Other key issues also require attention, such as high data dimensionality, too few training-labeled samples, and spatial variability of spectral characteristics.

In this study, we present a 3D fully convolutional neural network (3D-FCNN) model with a bottleneck attention mechanism. The downsampling and fully connected layers are substituted by the convolutional layer. A 3D convolution operation is adopted to replace the commonly used 2D and 1D convolution operations, and a bottleneck attention mechanism is introduced to the FCNN to maintain end-to-end classification. A pooling layer is employed for dimension reduction and the final prediction of the classification result.

The major contributions of this study are as follows:

(1) The downsampling layer and the fully connected layer are substituted by convolutional layers, and multiple datasets are adopted to separately alter the model and network depth. The developed network shows improved performance in comparison with several advanced HSI classification approaches with deep networks

(2) Network parameters are significantly reduced without adopting the fully connected layer

(3) A bottleneck attention mechanism is added to determine the latest classification accuracy in a dataset that includes limited training data. Moreover, the time consumed by the developed network is significantly decreased

The rest of the paper is organized as follows: In Section 2, literature related to CNN is presented; in Section 3, the proposed 3D-FCNN structure following the bottleneck attention mechanism is elucidated; in Section 4, the experimental results are presented and analyzed; in Section 5, conclusions are drawn, and the direction of future research is highlighted.

2. Convolutional Neural Network (CNN)

The CNN exploits feature extraction and a weight sharing mechanism to decrease the number of network training parameters required; its structure is illustrated in Figure 1. The working mechanism involves inputting image data and passing it to the convolutional layer for image feature extraction. The downsampling layer reduces the features of the current results. After several cycles of alternating learning of the convolution and downsampling layers, the data are acquired via the rectified linear unit (ReLU) activation

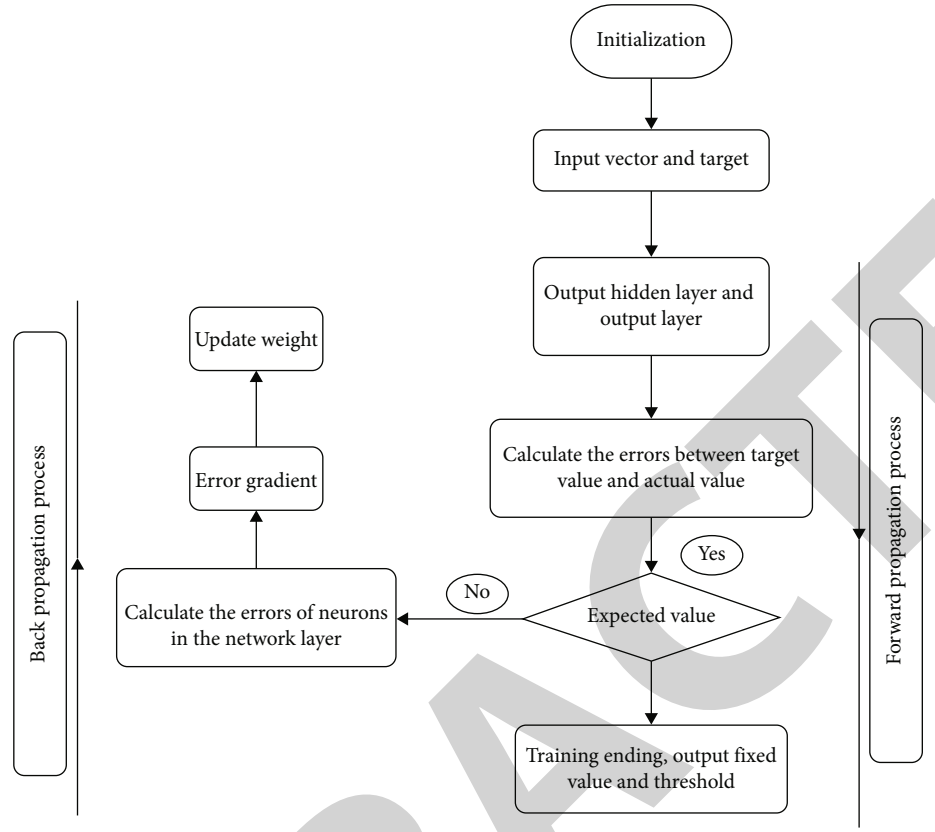


Figure 3: Process of CNN training.

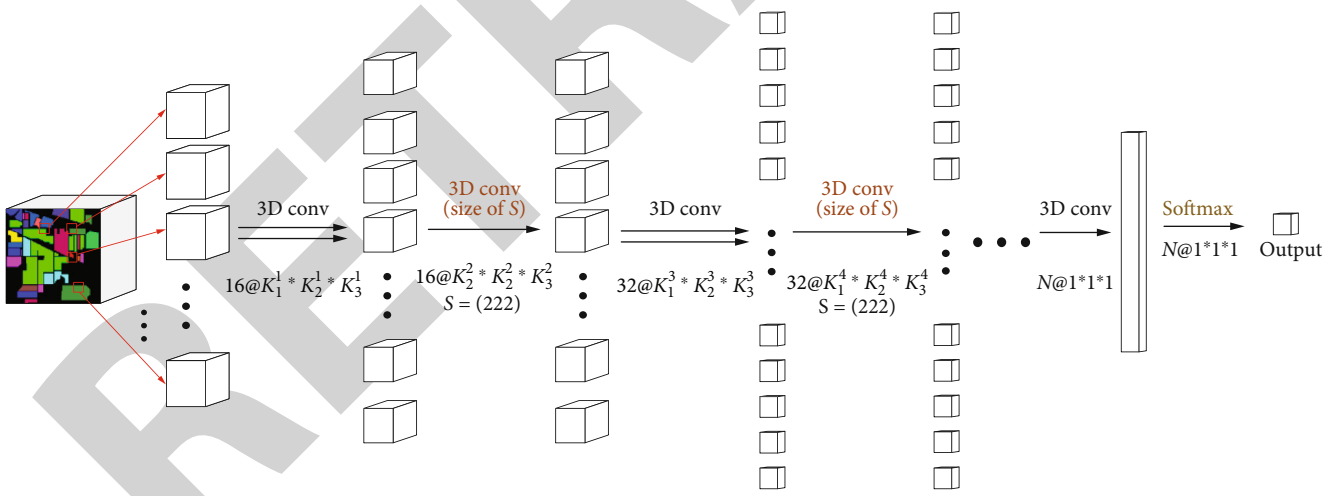


Figure 4: Recognition process based on the 3D-FCNN.

function with high-level abstract characteristics. The acquired abstract characteristics are introduced into a 1D vector, passed to the fully connected layer, subsequently passed to the learning of several fully connected layers, and finally outputted to the classifier to complete the entire classification of the image.

2.1. Convolutional Layer. The convolutional layer is a vital component of the CNN. The generation of multiple feature maps is achieved by multiple learnable filters in respective convolutional layers for convolution processing of input image data. The working mode of the convolutional layer is illustrated in Figure 2. Assuming that X is the input data, its size is $m \times n \times d$, where $m \times n$ denotes the spatial pixel size of X, d is the number of channels, and x_i is the i-th feature of the X feature map. Each layer covers k filters. The parameters w_j and b_j can be employed to represent the weight and offset between the j-th filter and the feature

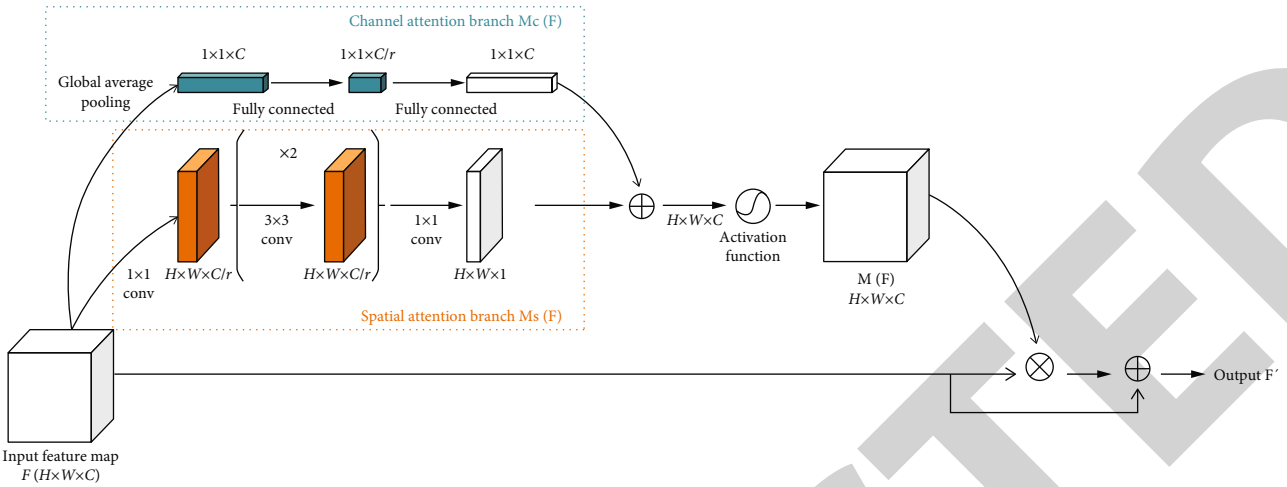


Figure 5: Bottleneck attention module.

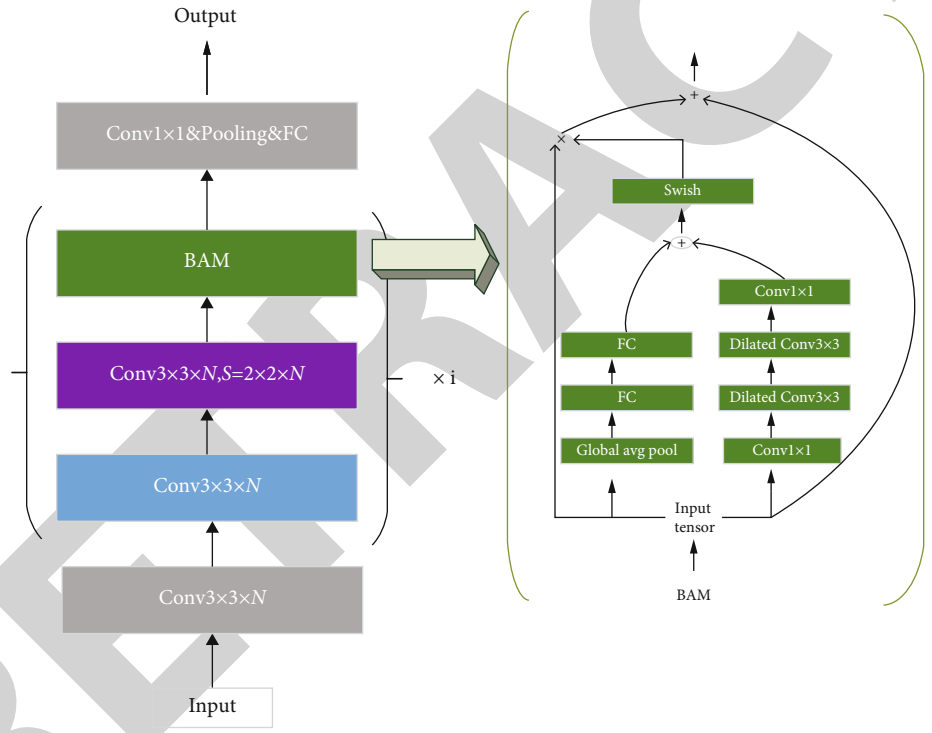


Figure 6: 3D-FCNN model with bottleneck attention module (BAM).

map. Subsequently, the j-th output of the convolutional layer is written as follows:

$$y_j = \sum_{i=1}^{d} f\left(x_i * w_j + b_j\right), \quad j = 1, 2, \cdots, k, \tag{1}$$

where $*$ denotes the convolution operator and $f(.)$ represents the activation function adopted to enhance the network nonlinearity.

2.2. Downsampling Layer. The downsampling layer is periodically inserted after several convolutional layers in the CNN to reduce redundant information in the image data. Network training parameters and the time consumed by network training are effectively reduced through dimensionality reduction of the feature map. Moreover, if the input pixel shows a slight change in the neighborhood, the downsampling layer exerts its local translation invariance characteristics to ensure the stability of the network and exerts a certain anti-interference effect. Average pooling and max pooling are considered common. To be specific, for the $p \times p$ window size field denoted as S, the average pooling operation is written as follows:

$$z = \frac{1}{F} \sum_{(i,j) \in S} x_{ij}, \tag{2}$$

Table 1: Average accuracy evaluation results for the five datasets derived using different methods.

Class	SVM	1D-NN	1D-CNN	2D-CNN	3D-CNN	3D-FCNN
IP	73.03	83.89	87.68	96.69	98.66	**99.32**
PC	94.70	96.18	96.21	97.23	98.57	**98.82**
UP	90.39	91.48	91.97	96.04	97.34	**99.07**
BS	80.63	81.05	89.81	90.60	90.97	**97.23**
SV	90.36	93.38	95.87	96.66	96.90	**98.59**

Table 2: Overall accuracy evaluation results for the five datasets derived using different methods.

Class	SVM	1D-NN	1D-CNN	2D-CNN	3D-CNN	3D-FCNN
IP	81.27	84.77	86.20	95.27	99.07	**99.25**
PC	98.22	98.74	98.87	98.90	98.93	**99.63**
UP	91.54	92.60	93.44	94.07	95.72	**99.60**
BS	77.83	80.44	88.96	89.72	90.69	**97.02**
SV	87.01	89.09	92.37	93.00	94.40	**96.97**

Table 3: Kappa evaluation results for the five datasets derived using different methods.

Class	SVM	1D-NN	1D-CNN	2D-CNN	3D-CNN	3D-FCNN
IP	78.61	64.39	84.21	94.64	98.93	**99.51**
PC	97.50	98.22	98.40	98.51	98.48	**99.47**
UP	89.07	90.17	91.52	92.25	94.40	**99.47**
BS	75.14	78.80	88.04	88.26	89.91	**96.07**
SV	85.48	87.86	91.49	90.22	93.77	**96.62**

where F denotes the number of elements in S and x_{ij} is the activation value at position (i, j).

2.3. Fully Connected Layer. The CNN output is acquired after the last one or two fully connected layers. Each node is connected to all the nodes in the previous layer, and the characteristics extracted after convolution downsampling are feature fused and subsequently transmitted to the classifier for classification prediction. The classifier is capable of employing logistic regression, SoftMax, support vector machine, or sigmoid [22] to be converted into probability methods. The output of the fully connected layer L is determined by the weighted summation of the input as well as the response of the activation function:

$$y_j^l = f\left(\sum w_{ji}^l * x_i^{l-1} + b_j^l\right), \tag{3}$$

where the j-th output unit $y^l{}_j$ of the layer performs weighting and bias calculations and summation on all the output feature maps of $x^{l-1}{}_i$ of the previous layer, which is obtained by the $f(.)$ classifier; $w^l{}_{ji}$ denotes the weight coefficient of the fully connected network, and $b^l{}_j$ represents the bias term of the l-th fully connected layer.

2.4. Network Training. The training process of the CNN covers two stages, i.e., forward propagation with low-level propagation and high-level propagation and back propagation with high-level propagation and low-level propagation. Figure 3 presents the entire CNN training process.

The input weight parameters are first initialized to avoid gradient propagation problems, reduced training speeds, and consumption of training time. Then, the actual output is obtained after a series of forwarding propagations (e.g., a convolutional layer, downsampling layer, and fully connected layer). The error between the actual output value and the target value is calculated. If the error generated is not consistent with the expected value, the error is retransmitted to the network for training, and the backpropagation sequentially calculates the fully connected, downsampling, and convolutional layers. The weight is updated following the calculated error value, and the mentioned steps are repeated until the error is less than the expected value; then, the training is terminated.

3. 3D-FCNN Structure with a Bottleneck Attention Mechanism

In this section, a new 3D fully convolutional neural network model will be presented to overcome difficulties in the process of hyperspectral images classification. In this model,

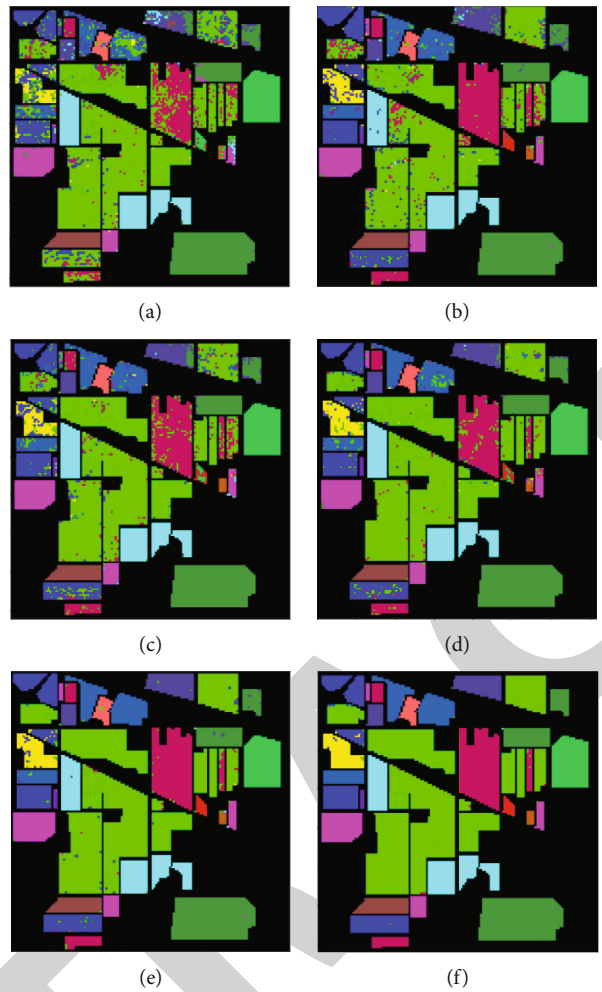


Figure 7: Classification effect diagrams of the IP dataset under the different models: (a) SVM; (b) 1D-NN; (c) 1D-CNN; (d) 2D-CNN; (e) 3D-CNN; and (f) 3D-FCNN.

the downsampling layer and the fully connected layer are replaced with a 3D-CNN, and a bottleneck attention mechanism is embedded. The structure of the elementary block of the developed model is first illustrated, and then the method by which the block extracts and fuses the characteristics is elucidated. Lastly, the bottleneck attention mechanism architecture is detailed.

3.1. 3D-FCNN Module. Most HSI classification models based on CNNs alternately cover multiple convolutional and downsampling layers, and several fully connected layers. Network parameters can be significantly reduced with convolutional layers instead of fully connected layers. Although the downsampling layer can increase the translation invariance of the characteristics of the CNN, it slightly improves the classification performance of the network. The downsampling of the pooling layer will give the high-level characteristics a larger receptive field while causing some loss of local characteristics. Zhang et al. [23] used a convolutional layer with a step size of 2 to replace the downsampling layer to improve the network classification performance. The 3D-FCNN proposed in the present study is used for pixel-level HSI classification. The main components are 3D convolution and 3D convolution with a step size of S. The model is mainly composed of an input layer, a 3D convolution layer, a 3D convolution layer with a step size of S, and an output layer. Preprocessing operations during training are not required. The image cube is composed of pixels in a small spatial neighborhood (rather than in the entire image) and directly extracted as the input from the entire spectrum. The spectral-spatial characteristics are extracted through the 3D-FCNN model. Lastly, the output of the classification results from the network, that is, the specific HSI classification process based on 3D-FCNN, as shown in Figure 4. The output of the convolutional layer with step size S is represented as follows:

$$v_{(l+1)j}^{xyz} = f\left(\sum_{m}\sum_{h=0}^{H_{(l+1)}-1}\sum_{w=0}^{W_{(l+1)}-1}\sum_{r=0}^{R_{(l+1)}-1} k_{(l+1)jm}^{hwr} v_{lm}^{(xs+h)(ys+w)(zs+r)} + b_{(l+1)j}\right), \tag{4}$$

where l represents the l-th layer, v represents the output feature body, and H, W, and R represent the length, width, and

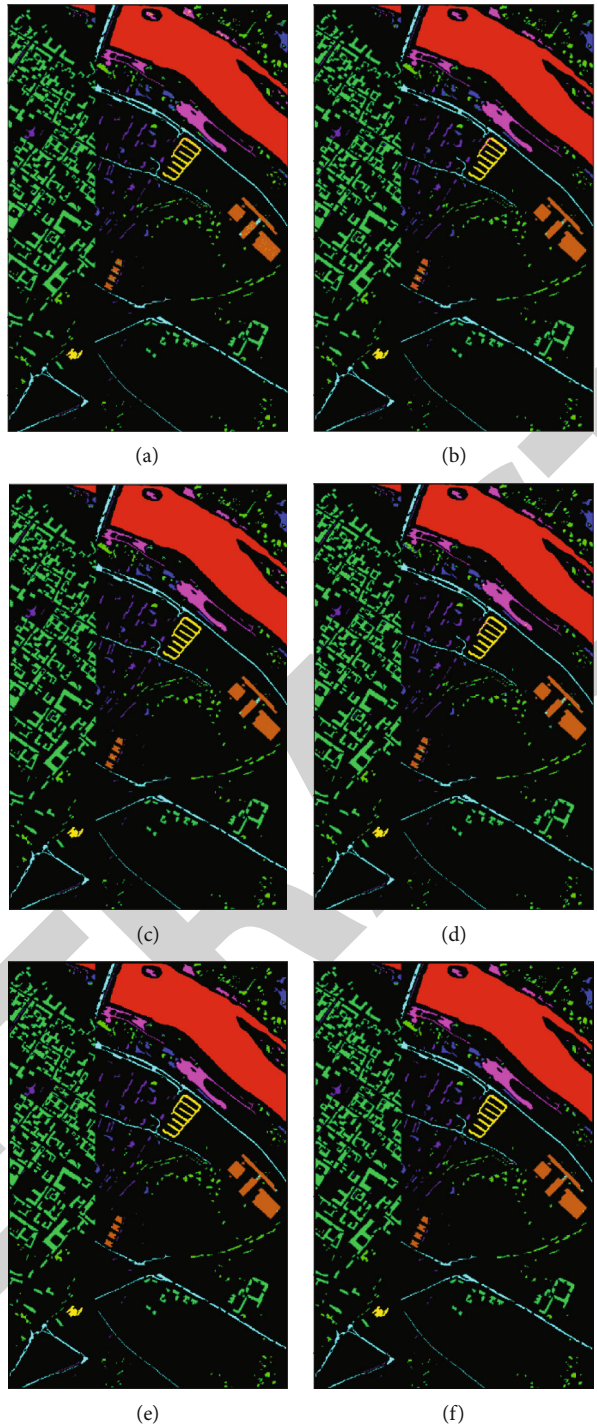


Figure 8: Classification effect diagrams of the PC dataset under the different models: (a) SVM; (b) 1D-NN; (c) 1D-CNN; (d) 2D-CNN; (e) 3D-CNN; and (f) 3D-FCNN.

spectral dimensions of the feature body, respectively. The number of convolution kernels in the current layer is j.

The proposed model primarily consists of three steps:

(1) Extraction of training samples. The $N \times N \times L$ image cube is extracted from the HSI with the input size of $H \times W \times L$, where $N \times N$ denotes the size of the neighborhood space (window size) and L represents the number of spectral bands

(2) Spectral-spatial feature extraction based on 3D-FCNN. The model in the present study substitutes all downsampling layers with convolutional layers with a step size of S

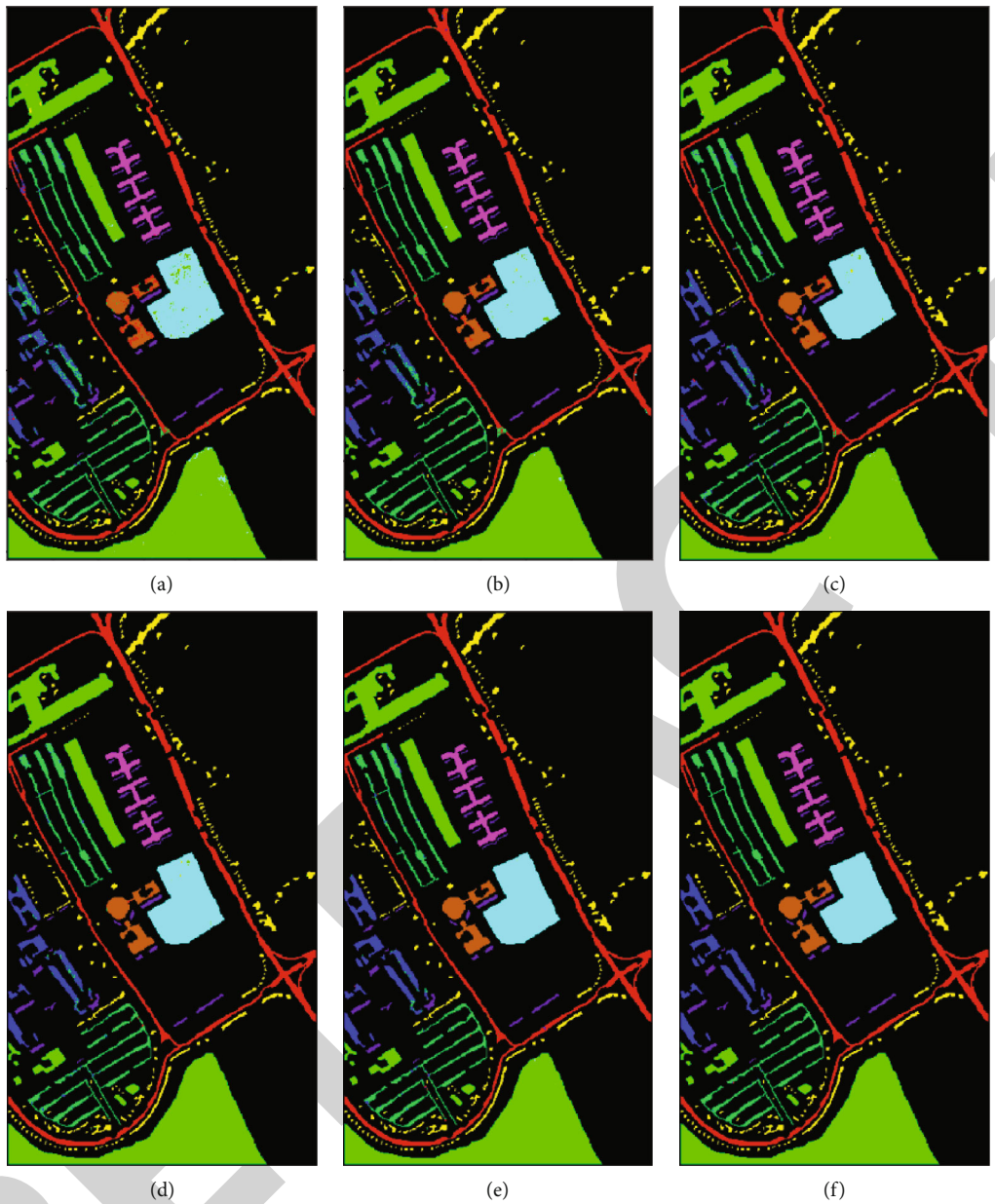


Figure 9: Classification effect diagrams of the UP dataset under the different models: (a) SVM; (b) 1D-NN; (c) 1D-CNN; (d) 2D-CNN; (e) 3D-CNN; and (f) 3D-FCNN.

(3) Classification based on spatial-spectral features. The characteristics of the last layer, i.e., the $1 \times 1 \times 1 \times N$ tensor, are input into the SoftMax classifier to acquire the final classification result

3.2. Bottleneck Attention Mechanism Module. The bottleneck attention module (BAM) [20, 24] is embedded based on the 3D-FCNN classification network. The BAM extracts vital information from the spectral and spatial dimensions of the HSI through the channel and spatial attention branches, respectively, and exploits the characteristics separately without any feature engineering. The end-to-end characteristics are maintained, and the problem of information redundancy is effectively solved.

In image processing, the core of the attention mechanism refers to mask learning on the image, injecting information from each region into the algorithm, and improving the region conducive to accuracy improvement. Figure 5 illustrates the detailed structure of the BAM. For a given input feature map $F \in R^{C \times H \times W}$, the BAM derives a 3D attention feature map $M(F) \in R^{C \times H \times W}$, and the feature map F' generated after multiplying and adding the original input feature map is obtained as follows:

$$F' = F + F \otimes M(F), \tag{5}$$

where $\otimes$ denotes multiplication by the corresponding elements, and the addition term refers to adding the

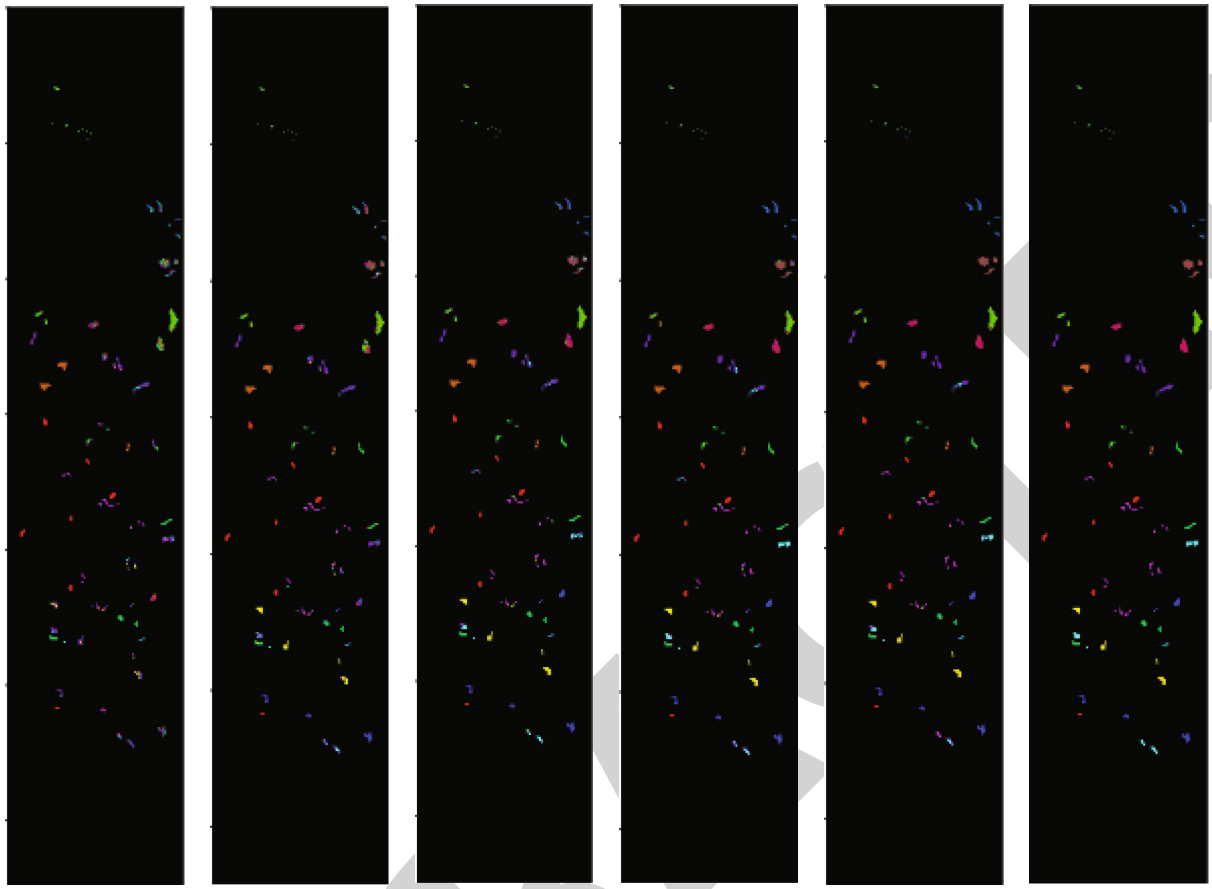


Figure 10: Classification effect diagrams of the BS dataset under the different models: (a) SVM; (b) 1D-NN; (c) 1D-CNN; (d) 2D-CNN; (e) 3D-CNN; and (f) 3D-FCNN.

corresponding elements. A residual structure is introduced to the BAM structure to promote gradient flow. The BAM has two attention mechanism branches, i.e., channel attention $M_c(F) \in R^C$ and spatial attention $M_s(F) \in R^{H\times W}$. The final attention mapping can be illustrated as follows:

$$M(F) = \sigma(M_c(F) + M_s(F)), \tag{6}$$

where σ denotes the sigmoid activation function, and the space size of the two branches is transformed into $R^{C\times H\times W}$ after the addition.

3.2.1. Channel Attention Branch. In the BAM proposed in this study, a channel attention branch is set to enhance or inhibit the characteristics of the band. To aggregate the characteristics in each channel, the global average pooling on the feature map F is employed to generate the channel vector $M_c(F) \in R^{C\times 1\times 1}$. Such a vector masks global information in each channel. To estimate the cross-channel attention from the channel vector F_C, a multilayer perceptron (MLP) with a hidden layer is adopted. To save the parameter overhead, the size of the hidden layer is set to $R^{C/r\times 1\times 1}$, where r denotes the compression ratio. After MLP inclusion, a batch normalization layer is introduced to regulate the scale to match the spatial branch output. Accordingly, the channel attention calculation formula is written as follows:

$$\begin{aligned} M_C(F) &= BN(MLP(AvgPool(F))) \\ &= BN(W_1(W_0 AvgPool(F) + b_0) + b_1), \end{aligned} \tag{7}$$

where $W_0 \in R^{C/r\times C}$, $b_0 \in R^{C/r}$, $W_1 \in R^{C\times C/r}$, and $b_1 \in R^C$.

3.2.2. Spatial Attention Branch. The spatial attention branch generates a spatial attention map $M_S(F) \in R^{H\times W}$, which is adopted to enhance or inhibit characteristics in various spatial positions. The application of context-related data is critical for acquiring spatial locations that require highlighting. Accordingly, a receptive field at a large scale is required to significantly exploit context-related data. Thus, cavity convolution is adopted for expanding the receptive field and enhancing efficiency. The spatial branch employs the "bottleneck structure" developed by ResNet [25], thereby saving on the number of parameters required as well as computation overhead. To be specific, the feature vector $F \in R^{C\times H\times W}$ merges the feature map into a low-dimensional $R^{C/r\times H\times W}$ through 1×1 convolution, which is equated with the integration

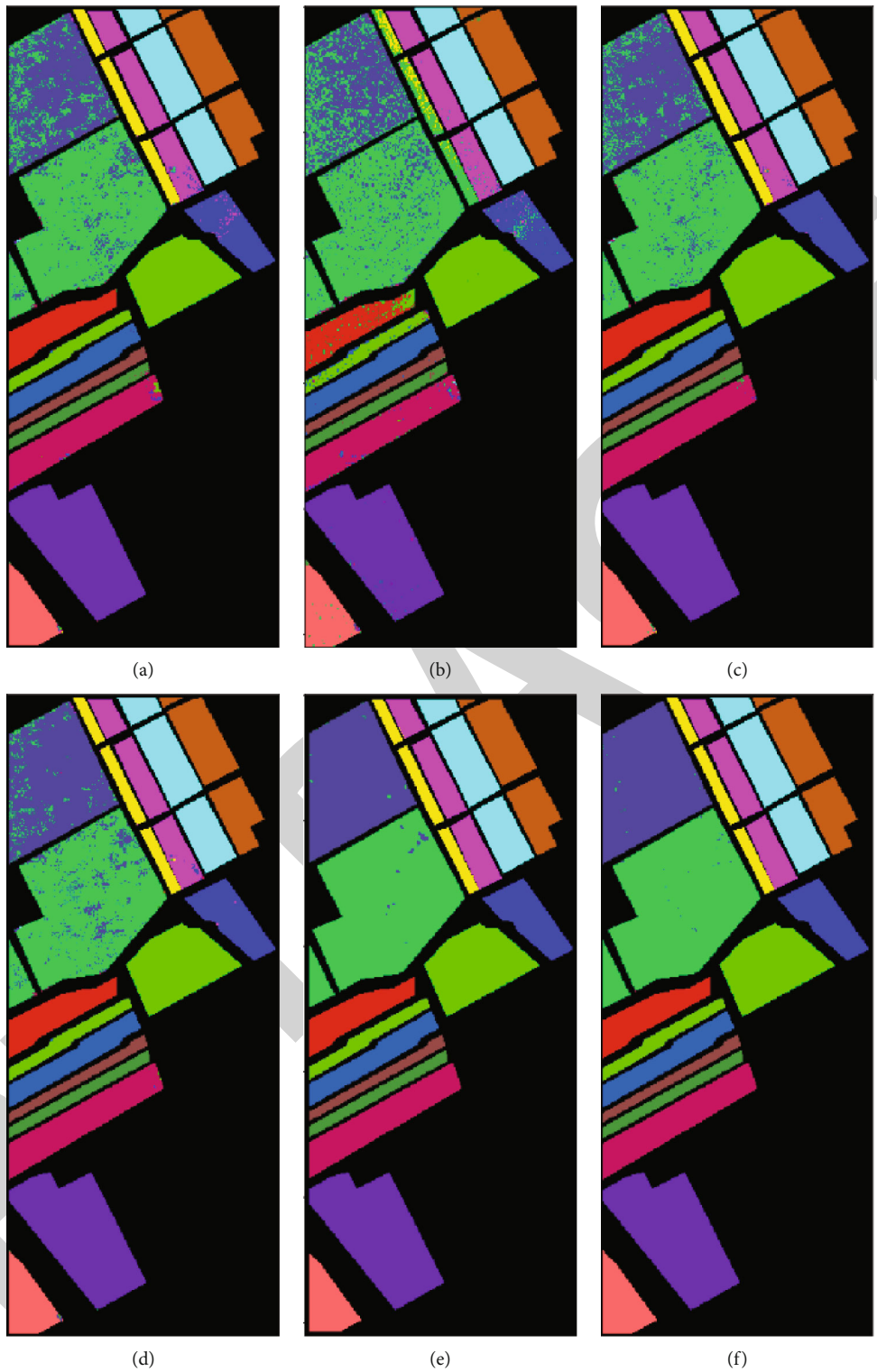


Figure 11: Classification effect diagrams of the SV dataset under the different models: (a) SVM; (b) 1D-NN; (c) 1D-CNN; (d) 2D-CNN; (e) 3D-CNN; and (f) 3D-FCNN.

and compression of the feature map of the channel dimension. Here, a compression rate identical to that of the channel attention branch is adopted. After dimensionality reduction, two 3×3 hole convolutions are used to effectively utilize context information. Lastly, a 1×1 convolution is adopted for reducing the feature to the size of $R^{1 \times H \times W}$ space. For scale adjustment, a batch normalization layer is added to the end of the spatial

Table 4: Performances of different network depths for the 3D-CNN and 3D-FCNN models.

Model	Dataset	3	5	7	9	11
3D-CNN	IP	87.78	**99.07**	77.69	75.76	73.04
	PC	95.62	**98.93**	97.03	96.22	95.00
	UP	93.79	94.01	**95.72**	95.11	94.25
	BS	88.04	**90.69**	88.96	87.13	85.64
	SV	93.08	**94.40**	94.05	93.33	92.57
3D-FCNN	IP	89.60	**99.25**	98.51	96.35	95.63
	PC	99.33	99.63	**99.68**	98.72	97.77
	UP	94.80	98.25	98.49	**98.55**	98.41
	BS	88.44	96.13	**97.02**	95.45	94.28
	SV	93.76	96.38	**96.97**	95.87	95.44

attention branch. Accordingly, spatial attention can be expressed as follows:

$$M_s(F) = BN\left(f\int_3^{1\times1}\left(f\int_2^{3\times3}\left(f\int_0^{1\times1}(F)\right)\right)\right), \quad (8)$$

where f is defined as the convolution operation process, BN is a batch normalization operation, and the superscript of the convolution operation is denoted as the size of the convolution filter. Three 1×1 convolutions are adopted to compress the channel dimension, and two 3×3 dilated convolutions are used to expand the receptive field to aggregate more context-related information.

3.2.3. Merging of the Two Attention Branches. After the channel $M_C(F)$ and the spatial $M_S(F)$ attention branches are obtained, these are merged to generate the final 3D attention feature map $M(F)$. The summation maps of the attention feature maps of each branch to the size of R are obtained and are impacted by the different shapes of the attention feature maps generated by the two branches. In a range of combination methods (e.g., summation, multiplication, or maximum value operations), the corresponding elements act as the operation method. After the summation, the swish function is adopted to activate the final 3D attention feature mapping $M(F)$. The generated 3D attention feature map $M(F)$ is subsequently introduced to the original input feature map F to multiply the corresponding elements in it and generate the redefined feature map F' as expressed in the formula, i.e., to generate the BAM-processed feature map.

3.2.4. Swish Activation Function. The swish activation function refers to a novel type of activation function proposed by Ramachandran et al. [26] for Google Brain; its formula is written as follows:

$$f(x) = x * \text{sigmoid}(x). \quad (9)$$

The common activation function in deep learning is the ReLU activation function characterized by a lower bound, no upper bound, and smoothness. Swish has a lower bound and no upper bound similar to ReLU, whereas the nonmonotonicity of swish is inconsistent with other common activation functions. Moreover, swish exhibits both first-order derivative and second-order derivative smoothness.

3.2.5. 3D-FCNN Model with BAM. The major convolution part of the model network covers a convolutional layer and a convolutional layer with a step length of S. The $N\times N\times L$ image cube of an HSI with the size $H\times W\times L$ is extracted as a sample input of the network. $N\times N$ denotes the size of the neighborhood space (window size), and L represents the spectral band number. The type of the center pixel of the cube acts as the target label. After inputting the data samples, it first passes through a $3\times3\times L$ convolutional layer. The second refers to a small-structure network covering a convolutional layer, a convolutional layer with a step size of S, and an added BAM. The number of times the small network module is superimposed is i. The last attention mechanism feature map generated undergoes a 1×1 convolution, global pooling, and fully connected operation. Then, the SoftMax function is adopted to output the final classification. The model is illustrated in Figure 6.

4. Results and Discussion

To evaluate the accuracy and efficiency of the developed model, experimental processes with respect to five datasets were created for comparison and verification with other approaches. For accurate measurements of each approach, quantitative metrics of Kappa (K), average accuracy (AA), and overall accuracy (OA) were employed. Here, OA denotes the rate of true classification of whole pixels, AA refers to the average accuracy characteristic of all types, and Kappa indicates the consistency characteristic of ground truth with the classification result. The higher these metrics are, the more effective the classification result is.

4.1. Introduction to the Dataset. Five extensively applied HSI datasets, namely, the Indian Pines (IP), Pavia Center (PC), Pavia University (UP), Salinas Valley (SV), and Botswana (BS) datasets, were applied. These datasets are briefly described below:

(i) Indian Pines (IP): generated by the airborne visible infrared imaging spectrometer (AVIRIS) sensor in north-western Indiana, the IP dataset covers 200 spectral bands exhibiting a wavelength scope of 0.4 to 2.5 μm and 16 land cover classes. IP covers 145 $\times$ 145 pixels and exhibits a resolution of 20 m/pixel

(ii) Pavia University (UP) and Pavia Center (PC): collected by the reflective optics imaging spectrometer (ROSIS-3) sensor at the University of Pavia, northern Italy, the UP dataset covers 103 spectral bands exhibiting a wavelength scope of 0.43 to 0.86 μm and 9 land cover classes. UP encompasses 610×340 pixels and exhibits a resolution of 1.3 m/pixel. The PC reaches 1096×715 pixels

(iii) Salinas Valley (SV): collected by the AVIRIS sensor from Salinas Valley, CA, USA, the SV dataset covers

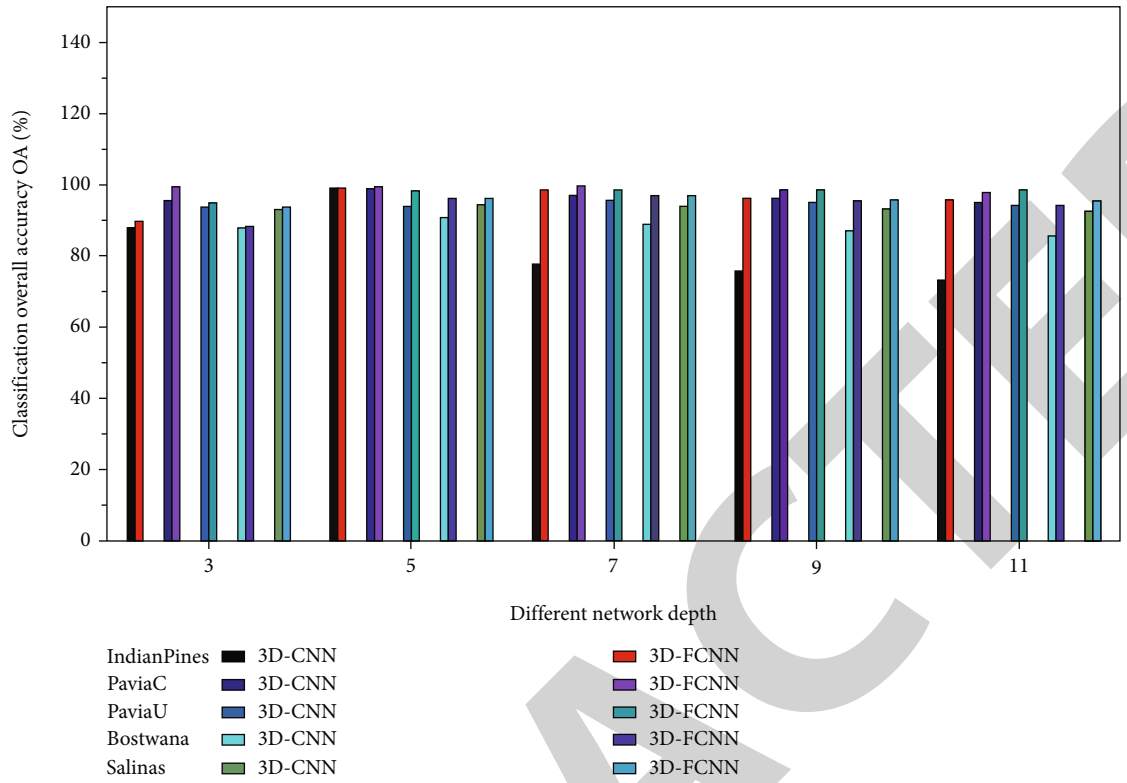


Figure 12: Performances of the 3D-CNN and 3D-FCNN models with each dataset at various depths.

204 spectral bands exhibiting a wavelength scope of 0.4 to 2.5 μm and 16 land cover classes. SV encompasses 512×217 pixels and exhibits a resolution of 3.7 m/pixel

(iv) Botswana (BS): captured by the NASA EO-1 satellite over the Okavango Delta, Botswana, the BS dataset covers 145 spectral bands exhibiting a wavelength scope of 0.4 to 2.5 μm and 14 land cover classes. BS encompasses 1476×256 pixels and exhibits a resolution of 30 m/pixel

Deep learning algorithms are data driven and rely on large numbers of labeled training samples. As more labeled data are fed into the training, the accuracy improves. However, more data for training implies increased time consumption and higher computation complexity. The five datasets used by the 3D-FCNN are the same as those used by the other networks discussed, and we set the parameters based on experience. For the IP dataset, 50% of the samples were selected for training, and 5% were randomly selected for verification. Since the samples were sufficient for UP, PC, BS, and SV, only 10% of the samples were used for training, and the remaining 90% were used as test data. Of the 10% of samples used for training, 50% (5% of the total) were randomly selected. Accordingly, different models and different network depths were compared under identical data conditions. Notably, in the absence of training samples, the model based on the BAM was capable of maintaining excellent performance. Thus, in the experiment, the sizes of the training and verification samples were set to the minimum level. The IP and SV datasets were employed for the experimental processes. Owing to the uneven distribution of the number of types in the IP dataset, the ratio of training-set : test-set was maintained at 1 : 1. As the number of labeled samples in the SV dataset is identical among different types, the ratio of training-set : test-set was maintained at 1 : 9.

4.2. Experimental Settings. To assess the effectiveness of the model, deep learning-based classifiers (SVM, 1D-NN, 1D-CNN, 2D-CNN, and 3D-CNN) were utilized to compare with our proposed framework. Under identical conditions, comparisons of the generalization ability and nonlinear expression ability at different network depths were conducted. The BAM added with the parameter $r = 5$ was employed in the CNN model. Two other methods, SE-Net [27] (squeeze-and-excitation (SE)) and frequency band weighted module [28] (band attention module, (BandAM)), were also employed. The classification results were compared. To ensure the validity of the experiment, the same depth was maintained for all involved models, and 10 experiments were carried out to eliminate randomness.

The patch size of each classifier was set as specified in the corresponding original paper. To compare the classification performances, all experiments were performed on the same platform with 32 GB of memory and an NVIDIA GeForce RTX 2080 Ti GPU. All classifiers based on deep learning were implemented by adopting PyTorch, TensorFlow, and Keras libraries.

Table 5: Classification effects of different modules on the IP dataset.

Class	3D-FCNN	SE+3D-FCNN	BandAM+3D-FCNN	BAM+3D-FCNN
1	53.33	100	52.27	**100**
2	82.74	98.10	**99.19**	95.49
3	59.61	98.04	88.09	**98.66**
4	64.68	**100**	80.89	97.65
5	67.78	27.93	94.12	**97.47**
6	99.03	**99.11**	98.70	98.93
7	0	96.15	74.07	**100**
8	94.29	100	100	**100**
9	0	88.89	73.68	**94.44**
10	94.24	94.27	79.74	**97.60**
11	90.09	99.25	97.13	**99.91**
12	67.12	95.79	82.77	**98.88**
13	99.01	100	91.79	**100**
14	97.60	99.05	**99.50**	99.03
15	89.79	97.45	92.64	**99.42**
16	65.22	100	**100**	98.81
OA (%)	82.29	93.01	93.66	**98.54**
AA (%)	71.00	93.36	88.13	**98.51**
Kappa	79.64	91.98	92.75	**98.33**

Table 6: Classification effects of different modules on the SV dataset.

Class	3D-FCNN	SE+3D-FCNN	BandAM+3D-FCNN	BAM+3D-FCNN
1	100	98.99	100	**100**
2	100	100	100	**100**
3	100	99.90	100	**100**
4	100	100	**99.76**	98.49
5	94.19	95.44	99.75	**99.96**
6	100	98.55	100	**100**
7	100	100	**100**	99.76
8	99.93	97.65	**100**	99.08
9	100	100	100	**100**
10	99.97	99.32	100	**100**
11	100	99.62	100	**100**
12	100	96.59	**100**	99.78
13	100	98.90	100	**100**
14	99.90	99.72	99.79	**100**
15	79.80	93.39	91.48	**99.96**
16	99.94	99.94	100	**100**
OA (%)	96.88	98.05	98.83	**99.73**
AA (%)	98.27	98.59	99.39	**99.81**
Kappa	96.52	97.83	98.70	**99.70**

4.3. Experimental Results. For SVM, 1D-NN, 1D-CNN, 2D-CNN, and 3D-CNN, the same architecture and parameter settings as in the present study were used. For those settings that are not explicitly given in the present study, we used commonly used values in the HSI classification (for example, the merge span is 2). Detailed analysis results are presented in Tables 1–3. The classification effect diagrams of various datasets under different models are presented in Figure 7 for IP, Figure 8 for PC, Figure 9 for UP, Figure 10 for BS, and Figure 11 for SV.

Our 3D-FCNN network replaces the downsampling layer and the fully connected layer with a CNN, which reduces the network training parameters, consumes less training time under identical conditions, and has a higher convergence speed, thus showing better overall performance. Furthermore, the model developed in the present study has the best classification performance with a classification accuracy of 99.63% and minimum classification error based on the three evaluation criteria. Adopting CNNs to replace the downsampling layer and the fully connected layer is suggested as a potentially feasible approach for training the deep network.

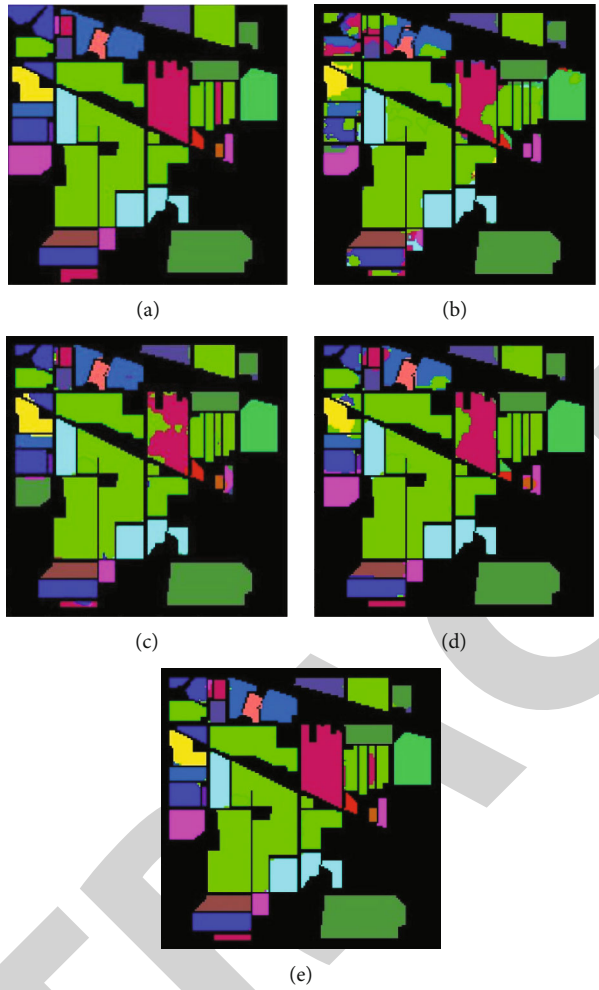
The number of network model layers (depth) is another critical parameter that should be considered. In the case of a fixed input data cube size, different network layers are employed for multiple datasets to further demonstrate the impact of the depth parameter on the classification results. The experimental processes were performed on the datasets and compared with the 3D-CNN model under identical conditions. The number of layers was 3, 5, 7, and 9. Table 4 shows the comparative results. Figure 12 presents the performances of the two models on the respective datasets at various depths.

The results show that, regardless of depth, the model developed in this study outperforms the 3D-CNN model. The 3D-FCNN model developed in the present study has better performance generalization and nonlinear expression abilities under identical conditions.

Figure 12 shows the results of different network depths. Overall, the network is better with increasing depth. Furthermore, increasing depth facilitates extraction and classification using more advanced functions. However, the results of our model are not proportional to the depth of the network, as the architecture of the developed model balances performance and cost by selecting the optimal network layer.

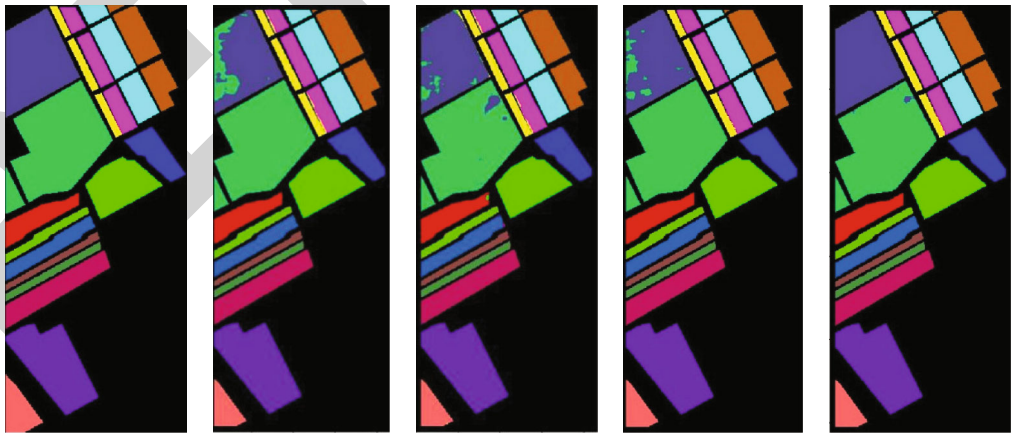
An optimized FCNN acts as the basic network. The network does not perform any operations and directly performs classification. The other three methods use different band weighted inputs, including the BandAM module, SE module, and the BAM proposed in the present study. Tables 5 and 6 present the specific analysis and comparison. The classification effect diagrams of various datasets under different modules (Figure 13 for IP and Figure 14 for SV) are illustrated.

In this study, we explored a novel and effective 3D-FCNN for HSI classification. On this basis, we embedded a module for the extraction of spectral and spatial features. Compared to the latest network, the most significant



(a) (b) (c) (d) (e)

Figure 13: Classification effect diagrams for IP dataset of different modules: (a) ground truth; (b) 3D-FCNN; (c) SE+3D-FCNN; (d) BandAM+3D-FCNN; and (e) BAM+3D-FCNN.



(a) (b) (c) (d) (e)

Figure 14: Classification effect diagrams of the SV dataset of different modules: (a) Ground truth; (b) 3D-FCNN; (c) SE+3D-FCNN; (d) BandAM+3D-FCNN; and (e) BAM+3D-FCNN.Tables 5 and 6 indicate that the proposed BAM considers spatial and spectral information, and it significantly improves classification performance. The 2–3% improvement in each standard demonstrates that the proposed BAM is effective. For HSI classification, the proposed BAM can be considered a plug-and-play supplementary module for most mainstream CNNs.

advantage of the proposed network is that it requires only a small number of network parameters to achieve considerable classification accuracy, in which an end-to-end classification mechanism is maintained. The proposed network uses various training strategies to help it converge better and faster without causing a computational burden.

5. Conclusions

The results of our study suggest the following:

(1) Deep networks that adopt spectral and spatial characteristics achieve significantly higher classification accuracy than deep networks that adopt only spectral characteristics. The results prove that the BAM is beneficial to HSI classification

(2) Deep learning performs well in several remote sensing fields. However, the trend to make the network more complex and deeper adds several parameters to the training process. With the inclusion of more parameters, the model can exhibit better classification capabilities. The results of the present study showed that this attempt has successfully reduced the network parameters and the loss of data information. That is, the developed method successfully replaces the downsampling layer and the fully connected layer with a convolutional layer. Furthermore, the experimental results show that the proposed network exhibits a high generalization ability and classification performance irrespective of its depth

Suggested improvements to the present study in the future are as follows:

(1) Application of the developed framework to HSIs in specific areas, such as forest resources observation and agricultural production management, other than the open-source datasets considered here

(2) The methods applied in the present study are all supervised. Semisupervised or unsupervised methods can be adopted using the considered limited data and achieve relatively higher performance with less labeled data

(3) The reduction in the training time poses an attractive challenge and needs to be addressed

Data Availability

All code will be made available on request to the correspondent author's email with appropriate justification.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (grant number: 61501210) and the Department of Education of Jiangxi Province (grant number: GJJ211410).

References

[1] G. E. Hinton and R. S. Zemel, "Autoencoders, minimum description length and Helmholtz free energy," in *Advances in neural information processing systems, Proceedings of the Neural Information Processing Systems*, pp. 3–10, Denver, Colorado, November 1993.

[2] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[3] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, "DRAW: A recurrent neural network for image generation," in *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1462–1471, Lille, France, July 2015.

[4] J. Gu, Z. Wang, J. Kuen et al., "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

[5] S. Dieleman and B. Schrauwen, "End-to-end learning for music audio," in *2014 Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6964–6968, Florence, Italy, 2014.

[6] A. R. Mohamed, T. N. Sainath, G. Dahl, B. Ramabhadran, G. E. Hinton, and M. A. Picheny, "Deep belief networks using discriminative features for phone recognition," in *2011 Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5060–5063, Prague, Czech Republic, 2011.

[7] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *2012 Proceedings of the 21st International Conference on Pattern Recognition (ICPR 2012)*, pp. 3304–3308, Tsukuba Science City, Japan, November 2012.

[8] D. Yu, L. Deng, and S. Wang, "Learning in the deep-structured conditional random fields," in *2009 Proceedings of the NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, pp. 1–8, Vancouver, Canada, December 2009.

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

[10] K. Makantasis, E. Protopapadakis, A. Doulamis, N. Doulamis, and C. Loupos, "Deep convolutional neural networks for efficient vision based tunnel inspection," in *2015 Proceedings of the IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pp. 335–342, Cluj-Napoca, Romania, September 2015.

[11] C. Chen, J. J. Zhang, C. H. Zheng, Q. Yan, and L. N. Xun, *Classification of hyperspectral data using a multi-channel convolutional neural network*, Intelligent computing methodologies, D. S. Huang, M. M. Gromiha, K. Han, and A. Hussain, Eds., Springer International Publishing, Berlin, 2018.

[12] M. He, B. Li, and H. Chen, "Multi-scale 3D deep convolutional neural network for hyperspectral image classification," in *2017 Proceedings of the IEEE International Conference on Image*

Processing (ICIP), pp. 3904–3908, Beijing, China, September 2017.

[13] Y. Li, H. Zhang, and Q. Shen, "Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sensing*, vol. 9, no. 1, p. 67, 2017.

[14] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 145, pp. 120–147, 2018.

[15] M. Seydgar, A. A. Naeini, M. Zhang, W. Li, and M. Satari, "3-D convolution-recurrent networks for spectral-spatial classification of hyperspectral images," *Remote Sensing*, vol. 11, no. 7, p. 883, 2019.

[16] J. Yang, Y. Q. Zhao, and J. C. W. Chan, "Learning and transferring deep joint spectral–spatial features for hyperspectral classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 8, pp. 4729–4742, 2017.

[17] H. Zhang, Y. Li, Y. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network," *Remote Sensing Letters*, vol. 8, no. 5, pp. 438–447, 2017.

[18] J. Yang, Y. Zhao, J. C. W. Chan, and C. Yi, "Hyperspectral image classification using two-channel deep convolutional neural network," in *2016 Proceedings of the IEEE international geoscience and remote sensing symposium. (IGARSS)*, pp. 5079–5082, Beijing, China, 2016.

[19] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: a 3-D deep learning framework," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 847–858, 2018.

[20] A. Ben Hamida, A. Benoit, P. Lambert, and C. Ben Amar, "3-D deep learning approach for remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 8, pp. 4420–4434, 2018.

[21] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, "HybridSN: exploring 3-D–2-D CNN feature hierarchy for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 2, pp. 277–281, 2020.

[22] J. Mount, "The equivalence of logistic regression and maximum entropy models," 2011, https://github.com/WinVector/Examples/blob/main/dfiles/LogisticRegressionMaxEnt.pdf.

[23] M. Zhang, W. Li, and Q. Du, "Diverse region-based CNN for hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2623–2634, 2018.

[24] R. Li, S. Zheng, C. Duan, Y. Yang, and X. Wang, "Classification of hyperspectral image based on double-branch dual-attention mechanism network," *Remote Sensing*, vol. 12, no. 3, p. 582, 2020.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, United States, June 2016.

[26] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017, https://arxiv.org/abs/1710.05941.

[27] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *2018 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, Salt Lake City, United States, June 2018.

[28] H. Dong, L. Zhang, and B. Zou, "Band attention convolutional networks for hyperspectral image classification," 2019, https://arxiv.org/abs/1906.04379.