# Patch-Free Bilateral Network for Hyperspectral Image Classification Using Limited Samples

Bing Liu and Xuchu Yu

***Abstract*—Recently, data-driven methods represented by deep learning have been widely used in hyperspectral image (HSI) classification and achieved the promising success. However, using less labeled samples to obtain higher classification accuracy is still a challenging task. In this study, we propose a patch-free bilateral network (PBiNet) for HSI classification. In order to make better use of the features with different scales, PBiNet uses the spatial path and the semantic path to obtain different level features for classification. The spatial path with small stride is used to retain the spatial detail information. The semantic path with fast down sampling rate is used to retain high-level semantic information. Using fast downsampling rates is to expand the scope of receptive field, so that semantic branch can better focus on global information. Then we design a feature fusion module to fuse the features obtained by the two paths. Finally, we use the classification maps produced by different scale features to calculate the loss function to optimize the whole model. Due to the better use of different levels of features, the proposed method could achieve higher classification accuracy with limited labeled samples. More importantly, because the whole HSI is used as the input, the proposed method has higher computational efficiency. To verify the effectiveness of the proposed method, we carried out classification experiments on four popular HSI datasets. Quantitative and qualitative experimental results show that the accuracy of the proposed method exceeds the compared methods.**

***Index Terms*—Bilateral network, deep learning (DL), hyperspectral image (HSI) classification.**

## I. Introduction

HYPERSPECTRAL images (HSIs) composed of hundreds of bands contain plentiful spectral and spatial information, which offer great potentials for land-cover mapping [1]–[5]. However, using these plentiful information to achieve accurate classification of HSIs has always been one of the research hotspots in the field of remote sensing [6]. HSI classification task usually faces two problems: lack of labeled training samples and high data dimension. Early HSI classification methods mainly rely on feature extraction to deal with the above two problems. For example, support vector machine (SVM) combined with extended morphological profiles (EMP) features could effectively improve the classification accuracy [7]. Meanwhile, the researchers also explored using local binary pattern (LBP) [8], Gabor features [9], slow features [10], salient features [11], invariant attribute profiles (IAPs) [12], joint and progressive subspace analysis (JPSA) [13], and other features to improve the accuracy of HSI classification. Although feature extraction methods could improve HSI classification accuracy, there is an irreparable deficiency, that is, they need to manually design feature extraction rules [14]. More importantly, these feature extraction methods need to set different hyperparameters for different HSIs to ensure high classification accuracy.

The deep learning (DL) method, which can automatically learn and extract features from data for classification tasks, could make up for the deficiency of manually designing feature extraction rules [15], [16]. Therefore, DL method is widely used to improve the classification accuracy of HSIs. At present, HSI classification based on DL can be grouped into subpixel, pixel, patch, and scene level. The pixel level method takes the spectral vector or the extracted 1-D feature of the selected sample as the input of the DL model. Typical pixel level DL classifiers include 1-D convolutional neural network (1-D-CNN) [17], recurrent neural network (RNN) [18], deep belief network (DBN) [19], etc.

The pixel level method could not use the neighborhood spatial information for HSI classification. To deal with this deficiency, the patch level DL classifiers take the local cube within a certain neighborhood of the central pixel as the input of the DL model, so that the DL method could better mine the features suitable for classification tasks. To this end, the patch level DL classifiers received extensive attention in HSI classification. For example, 3-D-CNN [20], 2-D-RNN [21], capsule network [22] are used to improve HSI classification accuracy, respectively. Meanwhile, the latest deep network architectures such as cascaded recurrent neural networks [23], attention mechanism [24], [25], graph convolutional networks [26], residual learning [27], and densely connected network [28] are used to improve the results of HSI classification. In order to further improve the classification accuracy, D-CNN designs a new discriminative objective function [29]. Meanwhile, Liu *et al.* [30] combined the superpixel method and DL method to obtain better classification accuracy. Generally, training DL model usually requires a large number of labeled samples to optimize thousands of parameters in the model. However, different from natural images, it is more difficult to obtain artificial label information in HSIs. Therefore, the number of labeled samples that could be used to train the deep models in HSI classification is usually limited [31]. Although the patch level DL classifier has achieved good results, it still faces the problem of lacking enough labeled training samples.

Manuscript received August 19, 2021; revised October 5, 2021; accepted October 12, 2021. Date of publication October 26, 2021; date of current version November 3, 2021. (*Corresponding author: Bing Liu.*)

The authors are with the PLA Strategic Support Force Information Engineering University, Zhengzhou 450001, China (e-mail: liubing220524@126.com; xuchu_yu@sina.com).

Digital Object Identifier 10.1109/JSTARS.2021.3121334

Semisupervised learning methods could solve the problem of limited labeled samples in supervised learning method and insufficient feature learning in unsupervised learning method at the same time, and have become one of the research hotspots in HSI classification. Traditional semisupervised algorithms for HSI classification usually include semisupervised SVM, graph-based algorithms, label propagation, and self-learning. For example, Wang *et al.* [32] proposed a label propagation algorithm combining spectral and spatial information, achieving higher classification accuracy than SVM and semisupervised SVM. Tan *et al.* [33] conducted a comprehensive study on the utilization of spatial information in HSIs semisupervised classification, further improving the classification accuracy with limited training samples. The traditional semisupervised classification algorithm can improve the accuracy of HSI classification to a certain extent, but still can not achieve satisfactory results. As for DL-based method, semisupervised learning methods such as self-training [34] and cotraining [35] have been applied to the training process of deep model. In addition, advanced semisupervised deep models such as generative adversarial network (GAN) [36], [37] have also been widely used in HSI classification. The core idea of transfer learning is to extract effective information from interrelated tasks to assist in solving target tasks. Most of the existing transfer learning-based methods usually adopt the mechanism of "pretraining + fine-tuning." For example, Yang *et al.* [38] proposed an active transfer learning network, in which hierarchical stack autoencoder was first used to extract the sptial-spectral features, and then active transfer learning strategy was used to carry out knowledge transfer and fine-tuning. In addition, domain adaptation technology has also been applied to HSI classification to reduce the distribution difference between training sets and test sets, further improving the transfer learning performance and the classification accuracy of target HSIs [39]. Another way to deal with the lack of training samples is to train the DL model to learn and extract the discriminant features. For example, we can train the stacked autoencoder to extract discriminative features [40], or we can also explore the power of off-the-shelf CNN models [41]. Moreover, the latest machine learning research results such as contrastive learning [42], meta learning [43], active learning [44] are used to improve the classification accuracy of HSI with a small number of labeled samples.

The patch level DL classifier only uses the local data cube in a certain neighborhood around the sample points to determine the class attribution, but the context information of the whole image is equally important for identifying different classes of ground objects. In order to make better use of the global context information in HSI, the image level HSI classification method is proposed. The image level DL classifier takes the whole HSI as the input directly, and then uses several convolution layers to output the classification results of the whole image. In this way, there is no need to crop the data in advance, which not only improves the calculation efficiency, but also improves the classification accuracy. Nevertheless, HSI classification task is very different from semantic segmentation task. The training samples of semantic segmentation task are a group of labeled images. In contrast, the training samples of HSIs is highly sparse and only contains a set of discrete labeled pixels. This leads to the poor effect of directly using semantic segmentation models such as U-Net [45], DeepLab [46], PSPnet [47], and SegNet [48] in HSI classification. FreeNet [49] is the first successful image level HSI DL classifier, which shows the great potential of the image level HSI classification method. Then SSFCN-CRF [50] proposes a novel mask matrix to assist the training of global fully convolutional network for HSI classification. DSSNet [51] is specially designed for HSI classification. FCN-Pyramid [52] introduces attention mechanism in the framework of fully convolution network (FCN) to improve the classification accuracy of HSI. FCSN [53] is designed to simultaneously identify land-cover labels of all pixels in an HSI cube.

Generally, the deeper the layers in the DL model, the more abstract the extracted features are. However, due to the limitation of receptive field, the deep abstract features could not retain enough spatial detail information. This requires us to make a direct balance between preserving high-resolution spatial detail information and abstract semantic information. The above image level DL classifiers could not well choose between high-level semantic information and spatial detail information. Different from the above existing works, we propose to use bilateral networks to preserve high-resolution spatial detail information and high-level semantic information of the HSIs at the same time. The proposed method is simple but efficient, and could greatly improve the classification accuracy of HSIs.

The main contributions of this article can be summarized as follows.

1) From the perspective of comprehensive utilization of global and local information of HSIs, we propose an image level classification method with bilateral network architecture. The proposed bilateral network could preserve spatial details and semantic information at the same time. By fusing the features of the two paths, the classification accuracy of the model could be greatly improved.
2) Considering that the classification map output from different scale features should be consistent, we fuse the loss function of different levels of feature output. Ablation experiments on four HSIs show that the fusion of loss functions with different scales could further improve the classification accuracy.

The rest of this article is organized as follows. Section II describes the proposed classification method. The classification results and corresponding analysis are presented in Section III. Section IV concludes this article.

## II. Proposed Method

As shown in Fig. 1, the proposed method takes the HSI cube as the input, which is composed of four parts: spatial branch, semantic branch, feature fusion module and segmentation head. The size of the input cube is $H \times W \times C$, where $H$ and $W$ is the height and width of an HSI, respectively, and $C$ is the number of bands. Spatial branch is mainly used to retain high-resolution spatial detail information, and semantic branch is mainly used to obtain high-level semantic information. The feature fusion model is responsible for fusing the features output by spatial

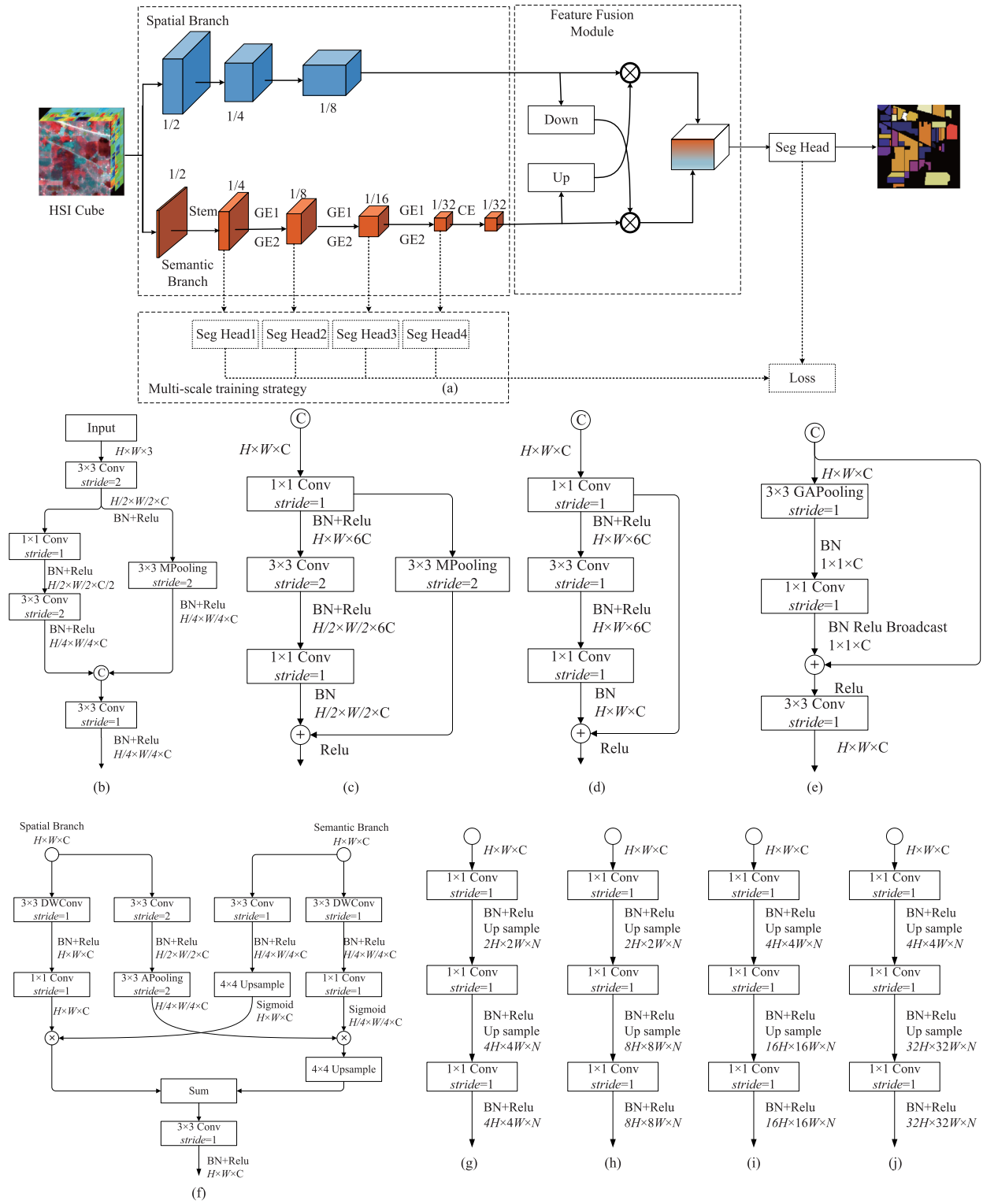


Fig. 1. Pipeline of the proposed method. $\otimes$ represents element-wise product. $\oplus$ represents element-wise plus. DWConv represents depthwise convolution. BN + Relu means to perform batch normalization operation first and then apply the Relu activation function. (a) Multiscale training strategy. (b) Stem block. (c) Context embedding block (CE). (d) Gather-and-expansion block (GE1). (e) Gather-and-expansion block (GE2). (f) Feature fusion module (FFM). (g) Seg head1. (h) Seg head2. (i) Seg head3. (j) Seg head4.

TABLE I
DETAILS OF THE SPATIAL BRANCH

| Stage | operation | k | c | s |
|---|---|---|---|---|
| Stage1 | Conv2D | 3 | 64 | 2 |
| | Conv2D | 3 | 64 | 1 |
| Stage2 | Conv2D | 3 | 64 | 2 |
| | Conv2D | 3 | 64 | 1 |
| Stage3 | Conv2D | 3 | 128 | 2 |
| | Conv2D | 3 | 128 | 1 |

$\mathrm{Conv2-D}$ Represents the Convolutional Layers, $k$ Represents the Convolutional Kernel Size, $c$ Represents the Output Channels, $s$ Represents the Stride of Convolutional Layers

TABLE II
DETAILS OF THE SPATIAL BRANCH

| Stage | operation | k | c | e | s |
|---|---|---|---|---|---|
| Stage1 | Stem | 3 | 16 | - | 4 |
| Stage2 | GE | 3 | 32 | 6 | 2 |
| | GE | 3 | 32 | 6 | 1 |
| Stage3 | GE | 3 | 64 | 6 | 2 |
| | GE | 3 | 64 | 6 | 1 |
| Stage4 | GE | 3 | 128 | 6 | 2 |
| | GE | 3 | 128 | 6 | 1 |
| Stage5 | CE | 3 | 128 | - | 1 |

$k$ Represents the Convolutional Kernel Size, $c$ Represents the Output Channels, $e$ is the Expansion Factor of GE, $s$ Represents the Stride of Convolution. GE is the Gather-and-Expansion Layer. CE is the Context Embedding Block

TABLE III
DETAILS OF FOUR TESTING DATASETS

| | UP | IP | SA | Houston2013 |
|---|---|---|---|---|
| Spatial size | $610 \times 340$ | $145 \times 145$ | $512 \times 217$ | $349 \times 1905$ |
| Spectral range | 430-860 | 400-2500 | 400-2500 | 380-1050 |
| No. of bands | 103 | 200 | 204 | 144 |
| GSD | 1.3 | 20 | 3.7 | 2.5 |
| Sensor type | ROSIS | AVIRIS | AVIRIS | ITRES CASI-1500 |
| Areas | Pavia | Indian | California | Houston |
| No. of classes | 9 | 16 | 16 | 15 |

University of Pavia (UP), Indian Pines (IP), Salinas (SA),Ground Sample Distance (GSD)(m), Spatial Size (pixel), Spectral Range (nm), Reflective Optics System Imaging Spectrometer (ROSIS), Airborne Visible Infrared Imaging Spectrometer (AVIRIS)

branch and semantic branch. The fused features are input into the segmentation head to output the classification results of the whole HSI scene.

### A. Spatial Branch

Spatial branch is mainly used to extract low-level features with high resolution. Therefore, we use convolution layer to achieve this purpose. Spatial branch can be divided into three stages. As shown in Table I, the stride of the first convolution layer is set to 2, the stride of the other convolution layers are set to 1. This setting can ensure that the spatial size of the feature map output in each stage is reduced to one fourth of the original size, so as to retain enough spatial detail information.

TABLE IV
ABLATION EXPERIMENTS

| SpB | SeB | FFM | MLoss | PU | IP | SA | H |
|---|---|---|---|---|---|---|---|
| ✓ | | | | 91.67 | 96.83 | 98.27 | 92.39 |
| ✓ | ✓ | | | 93.21 | 97.71 | 99.11 | 95.67 |
| ✓ | ✓ | ✓ | | 94.16 | 98.13 | 99.23 | 96.21 |
| ✓ | ✓ | ✓ | ✓ | 96.40 | 98.62 | 99.81 | 97.38 |

University of Pavia (UP), Indian Pines (IP), Salinas (SA), Houston2013 (H)

In order to train the model more stably, a batch normalization (BN) layer [54] is connected after each convolution layer, and the rectified linear units (Relu) [55] activation function is also applied.

*1) Convolutional Layer:* The core idea of convolution layer is parameter sharing and local connection. Given $C$ feature maps with size of $H \times W$ and $C_1$ convolution kernels with size of $k \times k$, the convolution layer can be formally expressed as

$$Y_i = f\left(\sum_{j=0}^{C}(w^i X^j) + b^i\right) \tag{1}$$

where $w^i$ is the $i$th convolution kernel, $X^j$ is the $j$th feature map of the input, $b^i$ is the bias term for the $i$th feature map, $f(\cdot)$ is an activation function. Given $C_1$ convolution kernels, the convolution layer will output $C_1$ feature maps. The number of feature maps output in the first two stages is set to 64, and the number of feature maps output in the third stage is set to 128.

*2) Batch Normalization (BN) Layer:* The BN layer is helpful for the training of deep network, thus we apply BN layer and Relu activation function after convolution layer. The BN layer requires data normalization, scaling, and shifting. Because we only have one HSI during training, the BN layer in this article only needs to be scaled and shifted. The scale and shift operations are as follows:

$$\hat{Y}_i = \gamma^i Y_i + \beta^i \tag{2}$$

where $\gamma^i$ and $\beta^i$ are the parameters to be learned, $Y_i$ is the $i$th output feature map of the convolution layer. After the BN layer, we apply the Relu activation function ($f(x) = \max(0, x)$).

### B. Semantic Branch

Semantic branch is mainly used to obtain high-level semantic information. Inspired by Xception [56], MobileNet [57], and BiSeNet [58], [59], we use Stem block, gather-and-expansion (GE) block, and context embedding (CE) block to construct semantic branches. The specific parameter configuration of semantic branch is shown in Table II. In the first stage, Stem block is used to extract abstract features from the HSI cube, and then six GE blocks and one CE are used to aggregate features, while reducing the spatial size of feature map. Because the spatial branch could provide spatial details, semantic branches use fewer channels, so that semantic branches can pay more attention to semantic information.

*1) Stem Block:* As shown in Fig. 1(b), the Stem block [60] first uses a convolution layer with a stride of 2 and a convolution

TABLE V
CLASS-SPECIFIC ACCURACY, OA, AA, AND $\kappa$ OF DIFFERENT METHODS FOR THE UNIVERSITY OF PAVIA DATASET

| Class No. | SVM | EMP | DLRGF | 3D-CNN | S-DMM | GCN | U-Net | SSFCN | FContNet | PBiNet |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 93.96 | 97.19 | 98.37 | 99.00 | 99.41 | 98.34 | 99.64 | 91.29 | 99.48 | 94.46 |
| 2 | 96.65 | 98.15 | 98.70 | 96.98 | 98.76 | 98.49 | 98.11 | 99.36 | 97.65 | 99.51 |
| 3 | 71.60 | 84.88 | 85.88 | 86.08 | 90.63 | 83.69 | 80.94 | 96.80 | 82.95 | 97.97 |
| 4 | 76.68 | 91.78 | 96.44 | 94.94 | 83.60 | 96.89 | 99.31 | 72.52 | 98.41 | 76.79 |
| 5 | 98.31 | 90.85 | 99.93 | 100.0 | 100.0 | 99.70 | 99.85 | 95.63 | 100.0 | 96.93 |
| 6 | 73.28 | 85.88 | 91.99 | 82.16 | 84.74 | 93.07 | 69.32 | 99.86 | 91.58 | 100.0 |
| 7 | 63.43 | 90.38 | 98.85 | 93.68 | 96.93 | 68.63 | 87.26 | 99.85 | 96.46 | 99.85 |
| 8 | 86.37 | 91.10 | 84.44 | 86.05 | 97.11 | 89.44 | 92.40 | 94.43 | 97.17 | 96.71 |
| 9 | 99.89 | 100.0 | 82.60 | 100.0 | 98.37 | 99.89 | 99.68 | 80.79 | 100.0 | 92.21 |
| OA (%) | 87.46 | 94.24 | 95.24 | 93.61 | 93.13 | 94.88 | 91.85 | 95.03 | 96.47 | 96.40 |
| AA (%) | 84.46 | 92.25 | 93.02 | 93.21 | 94.39 | 92.02 | 91.84 | 92.28 | 95.96 | 94.93 |
| $\kappa$ | 83.76 | 92.42 | 93.71 | 91.58 | 91.05 | 93.25 | 89.53 | 93.43 | 95.33 | 95.27 |

TABLE VI
CLASS-SPECIFIC ACCURACY, OA, AA, AND $\kappa$ OF DIFFERENT METHODS FOR THE INDIANA PINES DATASET

| Class No. | SVM | EMP | DLRGF | 3D-CNN | S-DMM | GCN | U-Net | SSFCN | FContNet | PBiNet |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 83.33 | 85.19 | 93.88 | 95.83 | 100.0 | 86.54 | 95.83 | 100.0 | 100.0 | 85.19 |
| 2 | 68.57 | 79.09 | 98.63 | 68.57 | 90.80 | 81.58 | 92.82 | 95.99 | 97.35 | 98.92 |
| 3 | 57.31 | 76.22 | 98.26 | 67.73 | 80.53 | 83.38 | 93.60 | 100.0 | 97.78 | 100.0 |
| 4 | 41.74 | 66.76 | 99.58 | 73.89 | 84.64 | 65.91 | 96.34 | 98.75 | 97.93 | 99.16 |
| 5 | 88.22 | 92.94 | 96.40 | 86.88 | 84.53 | 95.99 | 95.78 | 98.15 | 99.59 | 98.37 |
| 6 | 96.11 | 99.86 | 98.50 | 93.05 | 80.65 | 96.41 | 99.59 | 96.78 | 99.86 | 94.80 |
| 7 | 80.00 | 84.85 | 100.0 | 66.67 | 63.64 | 93.10 | 90.32 | 84.85 | 100.0 | 100.0 |
| 8 | 100.0 | 99.79 | 96.76 | 100.0 | 79.53 | 99.37 | 100.0 | 100.0 | 100.0 | 100.0 |
| 9 | 71.43 | 60.61 | 61.29 | 71.43 | 85.71 | 60.61 | 100.0 | 90.91 | 95.24 | 80.00 |
| 10 | 64.34 | 75.37 | 93.45 | 87.03 | 80.04 | 75.78 | 84.30 | 99.28 | 94.74 | 99.28 |
| 11 | 83.42 | 92.64 | 98.72 | 90.17 | 95.17 | 84.46 | 95.97 | 99.45 | 98.01 | 99.50 |
| 12 | 68.82 | 85.34 | 97.93 | 63.24 | 85.38 | 70.84 | 94.63 | 99.66 | 97.97 | 99.49 |
| 13 | 94.39 | 99.02 | 87.34 | 98.09 | 70.45 | 91.11 | 99.03 | 94.47 | 99.51 | 94.91 |
| 14 | 96.59 | 98.94 | 97.29 | 98.12 | 90.77 | 96.21 | 100.0 | 99.29 | 99.61 | 98.43 |
| 15 | 67.68 | 85.01 | 97.96 | 80.57 | 84.27 | 87.22 | 96.74 | 96.02 | 98.72 | 98.47 |
| 16 | 84.55 | 97.89 | 65.44 | 94.90 | 76.85 | 88.57 | 95.88 | 93.94 | 86.92 | 95.83 |
| OA (%) | 76.48 | 87.23 | 96.87 | 83.30 | 86.41 | 85.32 | 94.99 | 98.38 | 98.04 | 98.62 |
| AA (%) | 77.91 | 86.22 | 92.59 | 83.51 | 83.31 | 84.82 | 95.68 | 96.72 | 97.71 | 96.40 |
| $\kappa$ | 73.59 | 85.57 | 96.44 | 81.14 | 84.65 | 83.34 | 94.31 | 98.16 | 97.77 | 98.43 |

TABLE VII
CLASS-SPECIFIC ACCURACY, OA, AA AND $\kappa$ OF DIFFERENT METHODS FOR THE SALINAS DATASET

| Class No. | SVM | EMP | DLRGF | 3D-CNN | S-DMM | GCN | U-Net | SSFCN | FContNet | PBiNet |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 99.24 | 99.55 | 100.0 | 93.79 | 100.0 | 99.21 | 100.0 | 96.35 | 99.85 | 100.0 |
| 2 | 98.64 | 99.86 | 100.0 | 99.75 | 99.92 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| 3 | 90.43 | 97.08 | 100.0 | 89.74 | 97.40 | 97.19 | 98.32 | 100.0 | 99.80 | 100.0 |
| 4 | 97.34 | 98.93 | 98.86 | 98.58 | 99.54 | 98.93 | 99.93 | 93.22 | 98.31 | 99.71 |
| 5 | 96.25 | 99.73 | 99.00 | 98.43 | 99.81 | 94.94 | 99.78 | 99.59 | 100.0 | 99.48 |
| 6 | 99.95 | 99.55 | 100.0 | 99.82 | 100.0 | 99.72 | 99.75 | 100.0 | 100.0 | 100.0 |
| 7 | 98.89 | 98.97 | 100.0 | 96.24 | 100.0 | 99.58 | 99.92 | 100.0 | 100.0 | 99.94 |
| 8 | 81.86 | 90.39 | 93.54 | 87.04 | 83.53 | 78.88 | 94.84 | 99.41 | 97.35 | 99.88 |
| 9 | 99.09 | 99.55 | 99.98 | 99.42 | 100.0 | 98.71 | 99.95 | 99.98 | 99.98 | 100.0 |
| 10 | 82.92 | 93.17 | 98.69 | 93.50 | 87.28 | 96.12 | 98.32 | 100.0 | 99.57 | 100.0 |
| 11 | 76.06 | 95.98 | 99.72 | 85.62 | 94.79 | 94.48 | 100.0 | 100.0 | 98.05 | 99.81 |
| 12 | 95.07 | 98.87 | 99.74 | 99.69 | 99.19 | 99.22 | 98.97 | 99.95 | 99.59 | 100.0 |
| 13 | 94.83 | 98.81 | 100.0 | 98.18 | 99.51 | 99.46 | 90.51 | 100.0 | 100.0 | 99.89 |
| 14 | 88.93 | 89.54 | 96.64 | 95.38 | 98.47 | 98.80 | 99.91 | 100.0 | 97.18 | 100.0 |
| 15 | 56.44 | 79.81 | 90.41 | 81.26 | 62.91 | 79.83 | 84.85 | 98.39 | 95.97 | 99.07 |
| 16 | 98.50 | 97.66 | 97.99 | 91.22 | 99.71 | 99.83 | 98.64 | 100.0 | 100.0 | 100.0 |
| OA (%) | 86.12 | 94.03 | 97.07 | 92.62 | 87.03 | 91.91 | 96.26 | 99.31 | 98.71 | 99.81 |
| AA (%) | 90.09 | 96.09 | 98.41 | 94.23 | 95.13 | 95.93 | 97.71 | 99.18 | 99.10 | 99.86 |
| $\kappa$ | 84.65 | 93.37 | 96.73 | 91.79 | 85.65 | 90.97 | 95.84 | 99.23 | 98.57 | 99.79 |

kernel size of $3 \times 3$ to extract features, and then uses convolution and pooling to further reduce the spatial dimension of the feature map. $1 \times 1$ convolution is mainly used to increase the nonlinearity of the module. The parallel structure and asymmetric convolution kernel structure in Stem block can reduce the amount of computation while ensuring that the loss of information is small enough. Therefore, the Stem block is first used to extract features in the semantic branch.

*2) Gather-and-Expansion Block:* A large number of studies in semantic segmentation tasks show that the gather-and-expansion structure helps to improve the accuracy of the

TABLE VIII
CLASS-SPECIFIC ACCURACY, OA, AA, AND $\kappa$ OF DIFFERENT METHODS FOR THE HOUSTON DATASET

| Class No. | SVM | EMP | DLRGF | 3D-CNN | S-DMM | GCN | U-Net | SSFCN | FContNet | PBiNet |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 85.56 | 95.62 | 99.27 | 99.35 | 99.43 | 99.11 | 90.33 | 99.28 | 95.71 | 99.27 |
| 2 | 98.67 | 98.61 | 99.68 | 98.73 | 98.50 | 97.96 | 99.24 | 99.29 | 97.08 | 97.63 |
| 3 | 100.0 | 99.86 | 96.51 | 99.86 | 100.0 | 98.86 | 99.86 | 100.0 | 100.0 | 100.0 |
| 4 | 98.98 | 99.27 | 98.26 | 97.63 | 98.47 | 98.87 | 100.0 | 99.20 | 93.27 | 95.50 |
| 5 | 88.74 | 96.53 | 99.20 | 98.95 | 99.92 | 99.36 | 95.31 | 96.19 | 95.54 | 97.84 |
| 6 | 99.39 | 98.77 | 99.08 | 98.19 | 99.69 | 91.35 | 100.0 | 97.89 | 86.21 | 99.39 |
| 7 | 82.27 | 92.58 | 94.52 | 94.55 | 95.77 | 97.23 | 97.65 | 97.60 | 94.30 | 94.21 |
| 8 | 86.05 | 95.47 | 93.58 | 98.10 | 97.96 | 95.90 | 97.89 | 98.88 | 96.51 | 95.70 |
| 9 | 94.38 | 82.86 | 95.18 | 91.14 | 92.00 | 93.19 | 92.22 | 97.24 | 94.44 | 95.45 |
| 10 | 73.90 | 96.01 | 92.75 | 92.53 | 95.64 | 83.78 | 85.52 | 92.34 | 99.24 | 98.70 |
| 11 | 76.36 | 94.18 | 93.24 | 95.49 | 94.71 | 92.49 | 98.68 | 95.74 | 98.70 | 100.0 |
| 12 | 68.46 | 85.75 | 94.85 | 97.89 | 95.79 | 93.98 | 91.74 | 93.97 | 97.09 | 98.12 |
| 13 | 91.57 | 78.08 | 90.64 | 94.51 | 96.65 | 85.41 | 98.61 | 97.71 | 94.56 | 91.19 |
| 14 | 94.47 | 95.91 | 99.53 | 97.94 | 99.53 | 98.85 | 99.53 | 91.26 | 100.0 | 100.0 |
| 15 | 99.85 | 99.54 | 99.70 | 94.15 | 98.36 | 100.0 | 96.07 | 99.25 | 97.63 | 99.40 |
| OA (%) | 86.55 | 93.79 | 96.23 | 96.47 | 97.15 | 95.15 | 95.25 | 97.04 | 96.23 | 97.38 |
| AA (%) | 89.24 | 93.93 | 96.40 | 96.60 | 97.49 | 95.09 | 96.18 | 97.06 | 96.02 | 97.49 |
| $\kappa$ | 85.46 | 93.28 | 95.92 | 96.19 | 96.92 | 94.76 | 94.87 | 96.80 | 95.93 | 97.17 |

TABLE IX
EXECUTION TIMES OF TRAINING AND TESTING PROCEDURES

| University of Pavia Data set | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | SVM | EMP | DLRGF | 3D-CNN | S-DMM | GCN | U-Net | SSFCN | FContNet | PBiNet |
| Trianing (s) | 8.7 | 20.9 | 76.38 | 130.6 | 140.3 | 119.5 | 12.91 | 8.6 | 38.2 | 8.5 |
| Feature extraction (s) | 1.5 | 2.4 | 3.3 | 2.7 | 2.9 | 2.2 | <0.1 | <0.1 | 0.2 | <0.1 |
| Indiana Pines data set | | | | | | | | | | |
| | SVM | EMP | DLRGF | 3D-CNN | S-DMM | GCN | U-Net | SSFCN | FContNet | PBiNet |
| Trianing (s) | 12.0 | 16.7 | 59.71 | 179.26 | 182.9 | 173.7 | 8.9 | 6.8 | 12.1 | 3.1 |
| Feature extraction (s) | 1.7 | 1.9 | 2.9 | 1.43 | 1.44 | 1.34 | <0.1 | <0.1 | 0.1 | <0.1 |
| Salinas data set | | | | | | | | | | |
| | SVM | EMP | DLRGF | 3D-CNN | S-DMM | GCN | U-Net | SSFCN | FContNet | PBiNet |
| Trianing (s) | 16.2 | 21.7 | 60.17 | 387.5 | 394.1 | 355.5 | 9.3 | 7.9 | 18.9 | 7.1 |
| Feature extraction (s) | 3.7 | 4.5 | 5.3 | 9.2 | 10.3 | 8.0 | <0.1 | <0.1 | 0.2 | <0.1 |
| Houston2013 data set | | | | | | | | | | |
| | SVM | EMP | DLRGF | 3D-CNN | S-DMM | GCN | U-Net | SSFCN | FContNet | PBiNet |
| Trianing (s) | 16.2 | 39.6 | 47.68 | 94.7 | 98.29 | 168.2 | 86.3 | 49.4 | 79.23 | 43.6 |
| Feature extraction (s) | 3.7 | 3.9 | 3.2 | 1.4 | 1.5 | 1.3 | 0.3 | 0.2 | 0.3 | 0.2 |

model [60]. In order to better extract semantic features, as shown in Fig. 1(c) and (d), we designed two GE structures with $1 \times 1$ convolution kernel and $3 \times 3$ convolution. We used $3 \times 3$ Max-pooling (MPooling) operation on the residual connection path shown in Fig. 3(c) to ensure that the feature dimensions are the same. This structure is to reduce the spatial dimension of the feature map while aggregating features. The structure of Fig. 3(d) is relatively simple, it is mainly used to enhance the nonlinearity of the model. As shown in Table II, the use of GE block is still divided into three stages. In each stage, the structure shown in Fig. 3(c) is first used to reduce the spatial dimension of the feature map, and then the GE structure shown in Fig. 3(d) is used to enhance the nonlinearity of the feature.

*3) Context Embedding Block:* In order to obtain high-level semantic information, we designed a context embedding (CE) block as shown in Fig. 1(e). Specifically, we first use the global average pooling (MAPooling) operation to aggregate features, then use two $1 \times 1$ convolutions to enhance the nonlinearity of features, and finally expand the features to the input feature dimension and connect with the input. In the last stage of semantic branch, we apply this CE block.

### C. Feature Fusion Module

The spatial branch mainly focuses on the low-level spatial details of HSI, and the semantic branch mainly focuses on the high-level semantic information of HSI. In fact, there is a semantic gap between the two features, which leads to the poor effect of adding and fusing the features output by the two branches directly. Therefore, we design a feature fusion model (FFM) as shown in Fig. 1(f). The FFM first transforms and upsamples semantic features, and then multiplies them element by element with spatial features; then, the spatial features are downsampled by convolution operation with a stride of 2, multiplied by semantic features element by element, and then the features of this fusion path are upsampled to maintain the consistency of the feature dimensions of the two fusion paths. Finally, the fused features of the two paths are added.

### D. Training Strategy and Segmentation Head

In order to train the model more stably, we adopt a multi-scale training strategy. It is notable that the multiscale refers to that features have different receptive fields. The larger the receptive field is, the more global information is concerned, on the

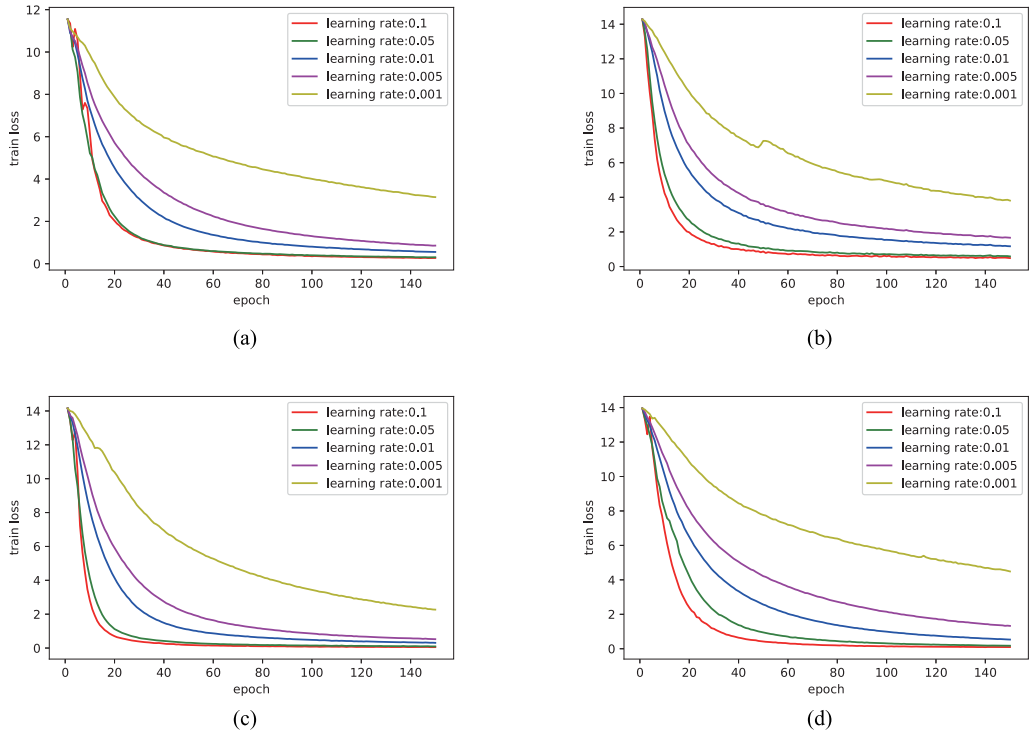


Fig. 2. Context embedding block.

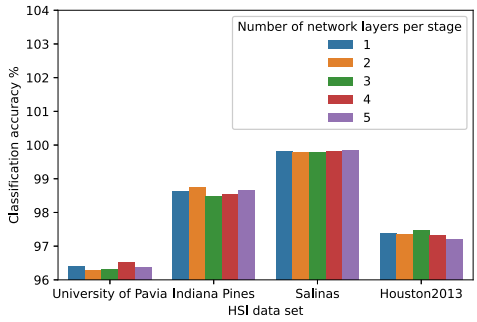


Fig. 3. Classification accuracy of models with different network depths on four HSIs.

contrary, more local information is concerned. Spatial branch have smaller receptive fields, so it pay more attention to local detail information, while semantic branch has a wider receptive field, mainly focusing on global information. Specifically, we output the classification results of HSI through a segmentation head model. Intuitively, the classification results output by these different resolution features should be consistent, so we use these results and the ground truth to calculate the loss function, and finally sum these loss functions as the final loss function. The segmentation head specifically uses three $1 \times 1$ convolution layers combined with the up sampling operation to complete the operation of outputting classification results from feature maps with different resolutions. As shown in Fig. 1(g)–(j), each segmentation head uses three $1 \times 1$ convolutions, and the up sampling rate is set according to the different resolution of the input feature map. The training of the model adopts the SGD optimizer integrated in PyTorch. During training, an HSI is repeatedly input into the network to calculate the corresponding loss function and update the network parameters. It is worth noting that if all pixels participate in the calculation of the loss function, the network will output the background, so only the labeled pixels participate in the calculation of the loss function. After training, the model only needs to calculate the classification results of the fused feature for testing.

## III. Experimental Results and Analysis

In this section, classification experiments are carried out on three widely used HSIs to verify the effectiveness of the proposed method. The hardware environment of the experiment is a personal notebook computer equipped with a Rtx2070 m graphics card, the video memory of the graphics card is 8 G, the memory of the computer is 16 G, and the CPU is Intel(R) Core(TM) i7-9750H. And the proposed method is implemented by the PyTorch library.

### A. Datasets

The four HSIs used for classification are the University of Pavia, Indian Pines, Salinas, and Houston2013 dataset. In order to quantitatively evaluate the classification algorithm, the researchers carefully labeled these HSIs. The detail information of the three HSIs is shown in Table III.

The University of Pavia dataset is acquired by the reflective optics spectrographic imaging system (ROSIS) sensor during a flight campaign over Pavia, Northern Italy. The number of

spectral bands is 103. The image size is $610 \times 340$ pixels. The geometric resolution is 1.3 meters.

The Indian Pines and Salinas datasets are acquired by the airborne visible infrared imaging spectrometer (AVIRIS) sensor. The number of spectral bands is 200 for Indian Pines and 204 for Salinas. The Indian Pines dataset is a $145 \times 145$ pixels image, and the Salinas dataset is $512 \times 271$ pixels. The geometric resolution of the Indian Pines and Salinas datasets are 20 and 3.7 m, respectively.

The Houston2013 dataset is obtained by ITRES CASI-1500 sensor and provided by 2013 IEEE GRSS data fusion competition. The data size is $349 \times 1905$ pixels, and it includes 144 bands ranging from 380 to 1050 nm. The dataset was acquired over the University of Houston campus and the neighboring urban area. The geometric resolution is 2.5 m.

In order to evaluate the classification performance of the proposed method, for each HSI, we randomly select 100 labeled samples from each class as the training samples and the remaining labeled samples as test samples. When the number of samples of a certain class is less than 100, we randomly select half of the samples as training samples.

### B. Parameters Setting and Ablation Experiment

The stochastic gradient descent (SGD) optimizer is used to optimize the patch-free bilateral network (PBiNet). Since there is only one HSI for training, the batch size is 1. We tested different learning rates, and their results are shown in Fig. 2. According to the experimental results, we found that for the SGD optimizer, a large learning rate will make the network better fit the training data. Therefore, we set the learning rate 0.1. The number of epochs is set to 150. Since only one HSI is available in each dataset, the batch size is 1. The weight decay of the SGD optimizer is set to 0.99, the momentum of the SGD optimizer is set to 0.9.

In order to prove the effectiveness of each module in the proposed method, we performed ablation experiments on four HSIs. In Table IV, SpB denotes spatial branch, SeB denotes semantic branch, FFM indicates whether to use the feature fusion module, and Mloss indicates whether to apply the middle layer output to calculate the loss function. We use spatial branch alone as a benchmark and observe the experimental results in Table IV. We find that adding semantic branch to spatial branch could effectively improve the classification accuracy, because the bilateral branch architecture could make more comprehensive use of the local detail information and global semantic information. The further introduction of FFM in the bilateral branch network could slightly improve the classification accuracy, because FFM could better integrate the features learned by the spatial branch and semantic branch. When the Mloss is further introduced, higher classification accuracy could be obtained, because the classification maps output by different scale features should be consistent, which is equivalent to adding a regularization term to the loss function. Overall, SpB+SeB, FFM, and Mloss are helpful to improve the classification accuracy of HSI. However, the improvement of SpB+SeB and Mloss is more obvious, FFM could only slightly improve the classification accuracy.

Fig. 3 shows the classification accuracy of models with different network depths on four HSIs. The numbers in the legend represent the number of repetitions of stage. According to the experimental results in Fig. 3, increasing the network depth will not improve the classification accuracy, but will greatly increase the model complexity. Therefore, we only repeat each stage in the spatial branch and semantic branch once.

### C. Comparison Results With the State-of-the-Art Methods

In this section, we use class-specific accuracy, overall accuracy (OA), average accuracy (AA), and $\kappa$ to quantitatively evaluate the proposed method (PBiNet) and compare it with the pixel level classifier and the patch level classifier. OA refers to the percentage of the number of correctly classified samples in the total number of samples, which is used to evaluate the overall performance of the classification results. Its calculation method is shown in the following:

$$OA = \frac{\sum_{i=1}^{k} C_i}{N} \tag{3}$$

where $N$ is the total number of samples, $k$ denotes the number of classes, and $C_i$ is the number of correctly classified samples in class $i$. AA is the average value of classification accuracy per class, which can effectively measure the effectiveness of classification algorithm in the case of uneven sample distribution. Its calculation method is shown as follows:

$$AA = \frac{\sum_{i=1}^{k} acc_i}{k} \tag{4}$$

where $acc_i$ denotes the classification accuracy of class i. Kappa coefficient ($\kappa$) is calculated based on the confusion matrix, which can overcome the problem that overall accuracy depends on the number of classes and the number of samples, and make a fairer evaluation. Its calculation method is shown as follows:

$$\kappa = \frac{OA - p_e}{1 - p_e}$$
$$p_e = \frac{\sum_{i}^{k} sum(row_i) \cdot sum(column_i)}{N^2} \tag{5}$$

where sum($row_i$) and sum($column_i$) denote the sum of elements on the $i$th row and the $i$th column in the confusion matrix, respectively. As for the pixel level classifier, we take the classification result of SVM as the benchmark, further extract the extended morphological profiles (EMP) [7] and discriminative low-rank Gabor filtering (DLRGF) [61] features, and then use SVM for classification. As for EMP, three principal components are extracted from the original hyperspectral image to build a morphological profile using PCA. Four opening and four closing based on circular structural elements with $R = 3, 5, 7, 9$ are computed for each PC. Therefore, the dimension of the EMPs is 27 for all datasets. SVM and EMP use the SVM classifier with radial basis function (RBF) and the optimal hyperplane parameters $C$ (parameter that controls the amount of penalty during the SVM optimization) and $\gamma$ (spread of the RBF kernel) have been traced in the range of $C = 2^{-2}, 2^{-1}, \ldots, 2^{5}$ and

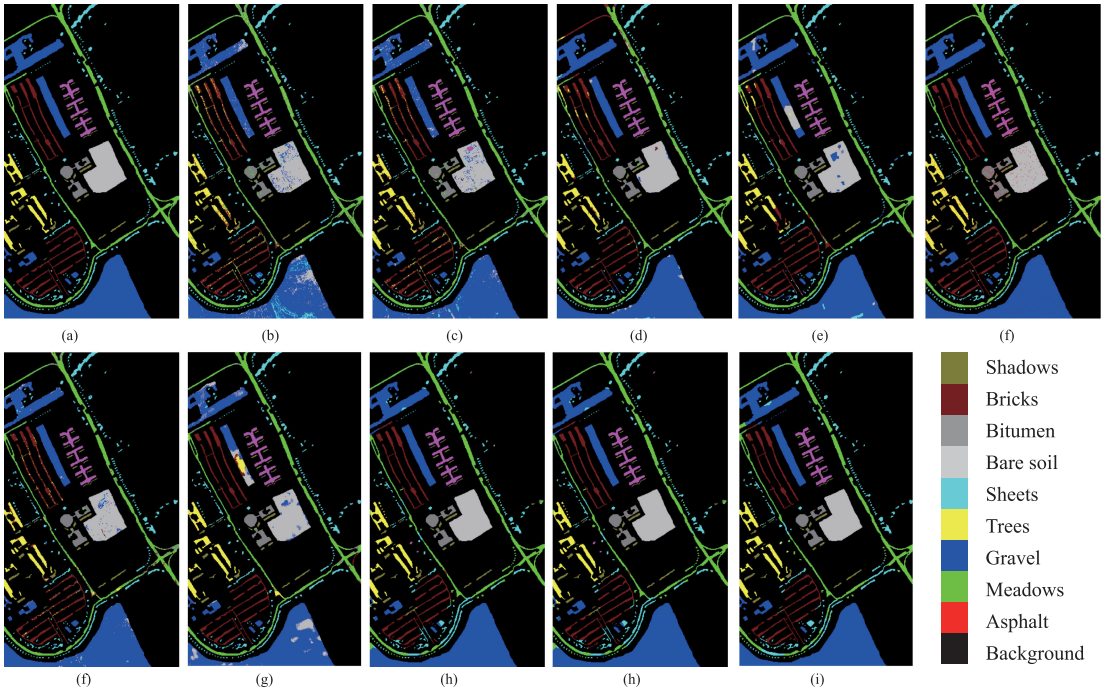


Fig. 4. Classification maps resulting from different methods for the University of Pavia dataset. (a) Ground-truth map. (b) SVM. (c) EMP. (d) DLRGF. (e) 3-D-CNN. (f) S-DMM. (g) U-Net. (h) SSFCN. (i) PBiNet.

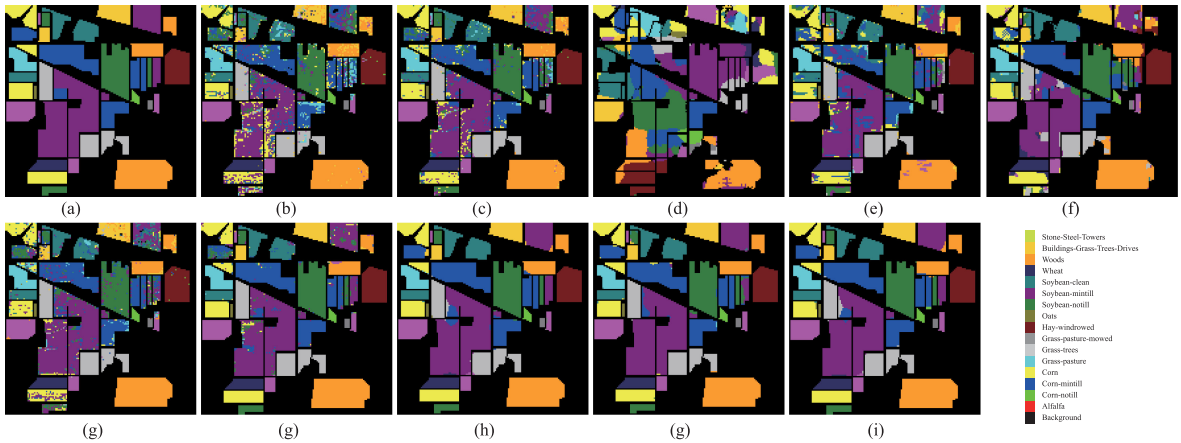


Fig. 5. Classification maps resulting from different methods for the Indiana Pines dataset. (a) Ground-truth map. (b) SVM. (c) EMP. (d) DLRGF. (e) 3-D-CNN. (f) S-DMM. (g) U-Net. (h) SSFCN. (i) PBiNet.

$\gamma = 2^{-2}, 2^{-1}, \ldots, 2^{5}$ using fourfold cross validation [62]. The parameter setting of DLRGF is consistent with that of [61], and we use the author's open source code to implement DLRGF. As for the patch level classifier we selected the most popular 3-D-CNN [63], similarity-based deep metric model (S-DMM) [64], and graph convolution network (GCN) [26]. In addition, we also implement U-Net, SSFCN [50], and FContNet [52] for classification experiments, in which SSFCN is a model specially designed for hyperspectral. The training samples used by the proposed method and the compared methods are the same.

Tables V–VII show the classification accuracy of different methods on four HSIs. In terms of classification accuracy, both the manually designed feature extraction methods (EMP, DLRGF) and the patch level classification methods based on DL (3-D-CNN, S-DMM) are better than SVM using only spectral features. More importantly, the image level classification methods (U-Net, SSFCN) could effectively improve the classification accuracy, and the improvement range is large. Compared with U-Net and SSFCN, the proposed method (PBiNet) could achieve higher classification accuracy. Figs. 4– 7 show the classification maps obtained by different methods. From these classification maps, we could find that the distribution of misclassified samples of the pixel level and patch level classifiers is relatively discrete, while the classification results of the image level classification

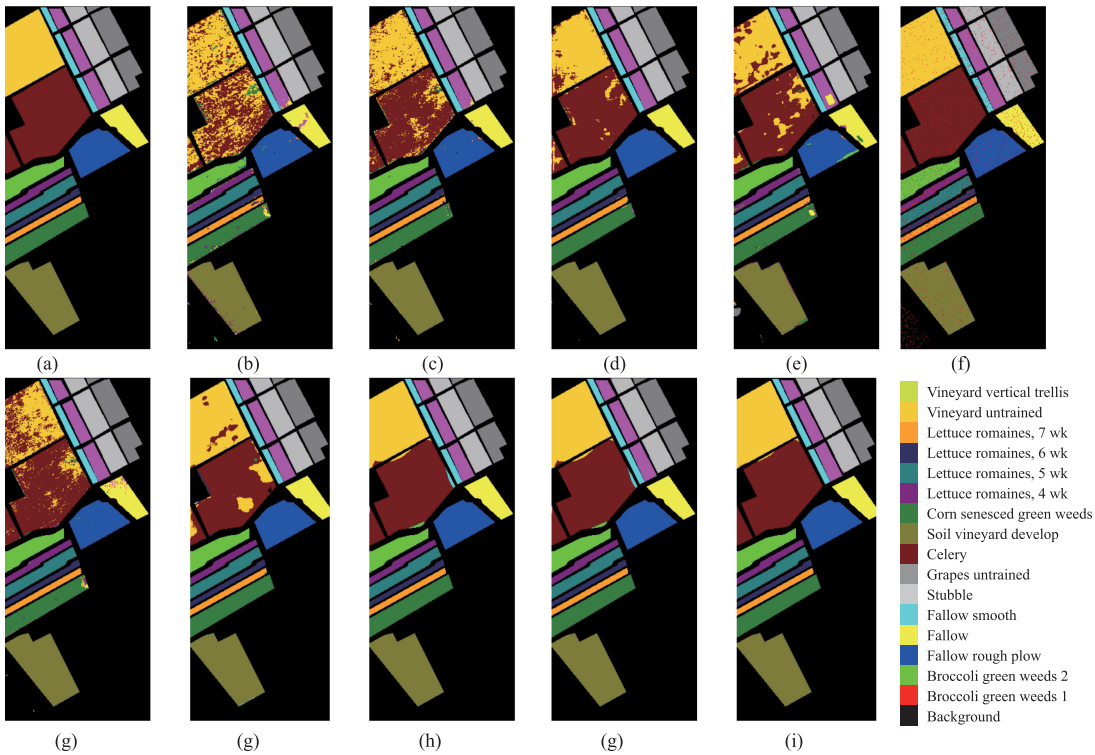


Fig. 6. Classification maps resulting from different methods for the Salinas dataset. (a) Ground-truth map. (b) SVM. (c) EMP. (d) DLRGF. (e) 3-D-CNN. (f) S-DMM. (g) U-Net. (h) SSFCN. (i) PBiNet.

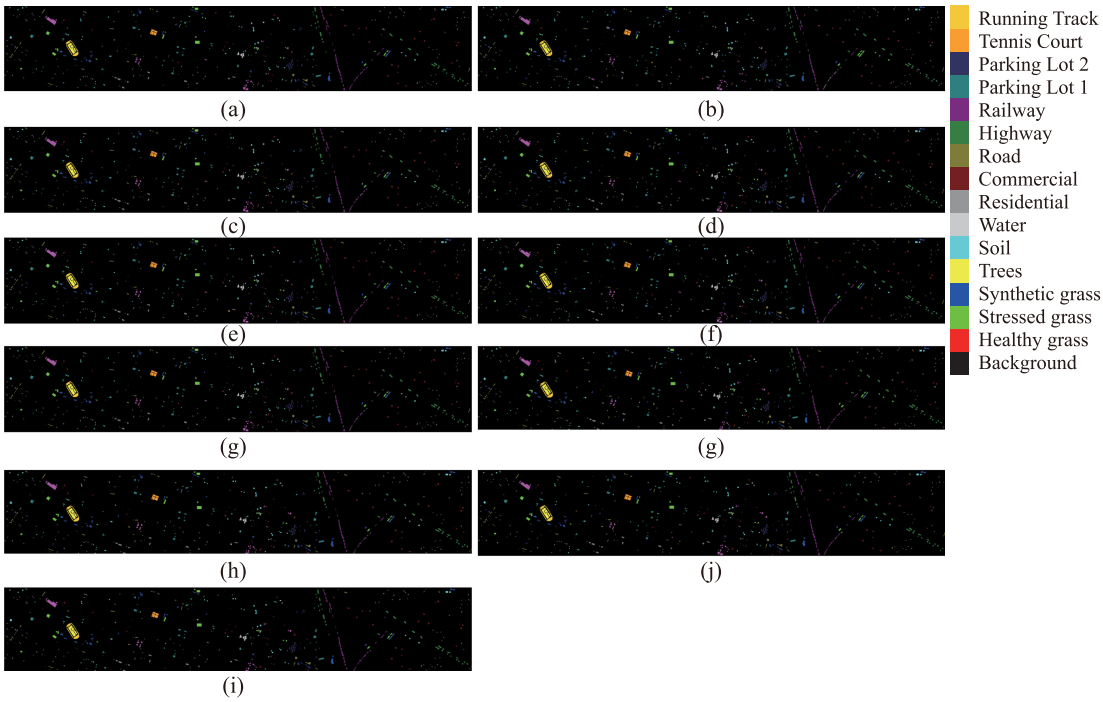


Fig. 7. Classification maps resulting from different methods for the Houston dataset. (a) Ground-truth map. (b) SVM. (c) EMP. (d) DLRGF. (e) 3-D-CNN. (f) S-DMM. (g) U-Net. (h) SSFCN. (i) PBiNet.

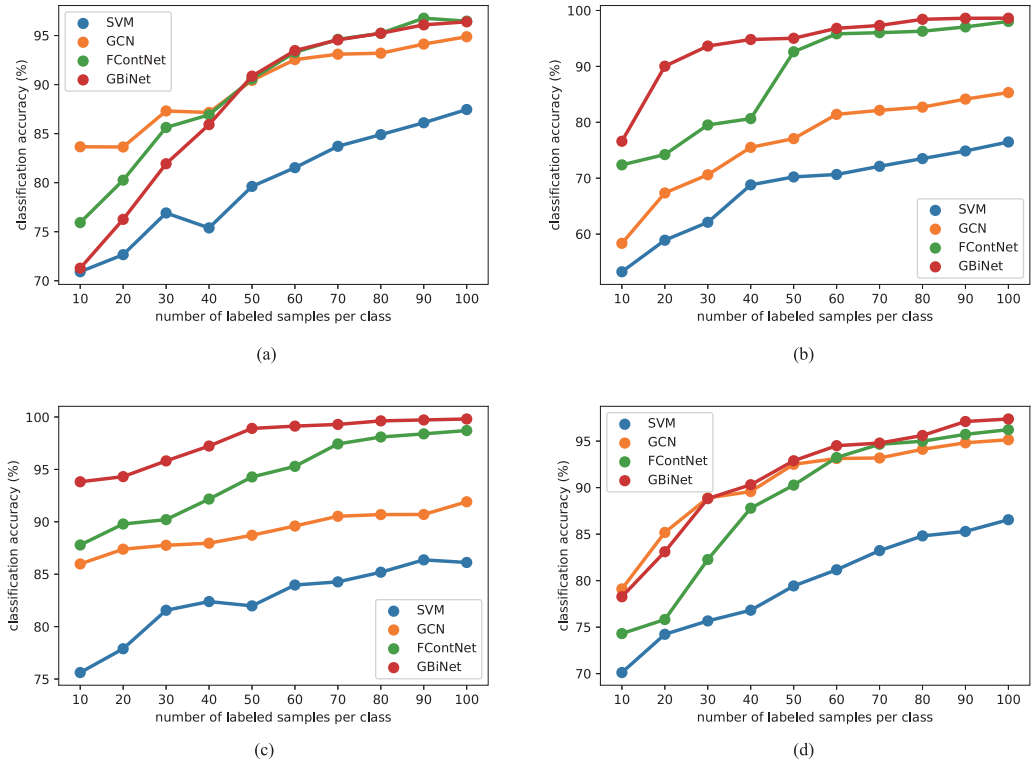


Fig. 8. Classification accuracy for the four HSI datasets with different sample number. (a) University of Pavia dataset. (b) Indiana Pines. (c) Salinas. (d) Houston2013.

methods are more in line with the human visual system, but the misclassified samples are often clustered into blocks. The proposed method (PBiNet) has the least misclassified samples, which is consistent with the quantitative evaluation results in Tables V–VII. The above comparative experiments fully illustrate the effectiveness of the proposed method.

In order to prove the effectiveness of the proposed method for limited samples, we further reduce the number of training samples. We select three representative methods from all compared methods for analysis. The experimental results on four HSIs are given in Fig. 8. The abscissa in Fig. 8 represents the number of randomly selected labeled samples for each class, and the ordinate is the overall classification accuracy. By observing the results in Fig. 8, we can see that the accuracy obtained by the proposed method is much higher than that of SVM, GCN, and FContNet on the Indiana Pines and Salinas dataset, which shows the effectiveness of the proposed method for limited samples. As for the University of Pavia and Houston2013 dataset, the proposed method needs a certain number of samples to ensure the advantages.

### D. Efficiency Analysis

DL methods usually require a lot of iterations to optimize thousands of parameters in the model, while the classical shallow classification methods (such as SVM) require relatively less computation. The training and testing time of the compared methods and the proposed method are given in Table IX. Note that the DL models (3-D-CNN, S-DMM, GCN, U-Net, SSFCN, FContNet) run on a RTX2070 graphics card, and the shallow model (SVM, EMP, DLRGF) runs on the i7-9750 h CPU. The training time of SVM, EMP and DLRGF includes the time of feature extraction and selection of optimal parameters. It could be seen from Table VIII that the training time of the DL method (e.g., 3-D-CNN, S-DMM, GCN) with local image patch as input is long, while the training time of the image level DL method is close to that of shallow models such as SVM. The high computational efficiency of the proposed method is mainly due to taking the whole HSI as the input without cutting the image into overlapping patches, which avoids a large number of repeated calculations. In terms of the testing time, the advantage of the proposed method is very obvious. The testing time on three HSIs is less than 0.1 s, which is much less than that of SVM, EMP, 3-D-CNN.

### E. Feature Visualization and Discussion

In order to better understand the motivation of the proposed method, we take the Indian Pines dataset as an example to visualize the output features of the semantic branch and the spatial branch. As shown in Fig. 9, the first row is four feature channels randomly selected from the spatial branch. The brighter the color in the figure, the greater the feature value. The second row is four feature channels randomly selected from the semantic branch. From the results of feature visualization, the spatial branch pay more attention to local details, while the semantic branch pays more attention to the global abstract information in the whole HSI scene. Thanks to this dual branch

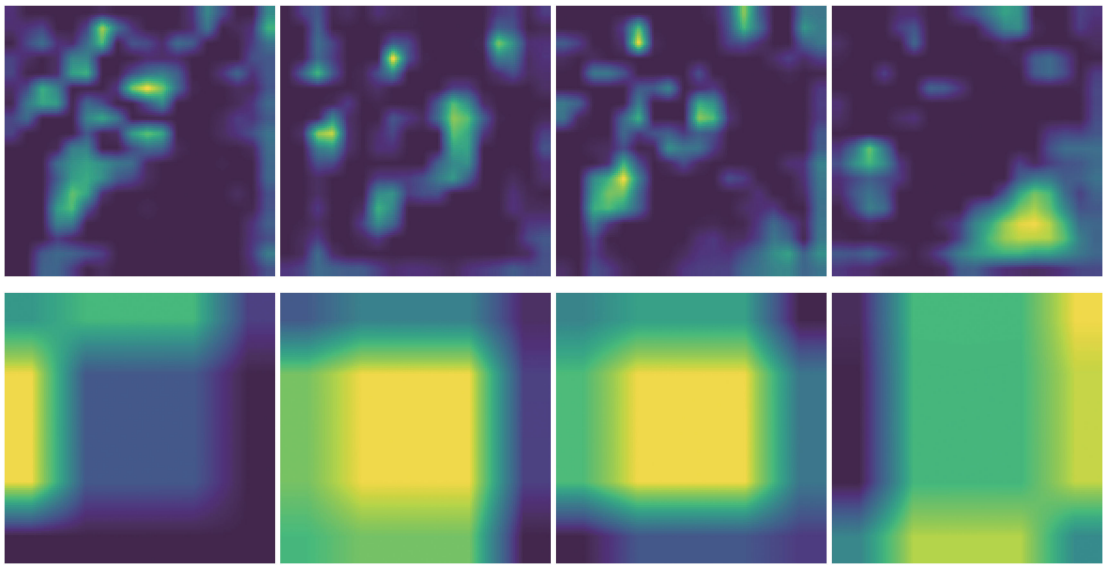

Fig. 9. Feature visualization of the Indian Pines dataset.

network architecture which could comprehensively utilize local and global information, the proposed method can use a shallow network architecture to obtain competitive classification results.

## IV. Conclusion

This study presents an image level HSI classification method. The bilateral network architecture make the model effectively use global semantic information and local spatial information, and the feature fusion module can integrate global semantic information and local spatial information well. To better train the model, we proposes a booster training strategy. The classification experiments on three HSIs show that the proposed method could achieve higher classification accuracy than the comparison method. In addition, the proposed method takes the HSI of the whole scene as the input without any preprocessing, so it has higher classification efficiency.

## Acknowledgment

The author would like to thank Prof. J. Li and Prof. S. Jia for providing the open source codes of DLRGF and S-DMM, respectively.

## References

[1] M. Paoletti, J. Haut, J. Plaza, and A. Plaza, "Deep learning classifiers for hyperspectral imaging: A review," *ISPRS J. Photogrammetry Remote Sens.*, vol. 158, pp. 279–317, 2019.

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

[3] S.-E. Qian, "Hyperspectral satellites, evolution, and development history," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 7032–7056, Jun. 2021, doi: 10.1109/JSTARS.2021.3090256.

[4] B. Rasti *et al.*, "Feature extraction for hyperspectral imagery: The evolution from shallow to deep: Overview and toolbox," *IEEE Geosci. Remote Sens. Mag.*, vol. 8, no. 4, pp. 60–88, Dec. 2020.

[5] L. He, J. Li, C. Liu, and S. Li, "Recent advances on spectral-spatial hyperspectral image classification: An overview and new guidelines," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1579–1597, Mar. 2018.

[6] D. Hong *et al.*, "Interpretable hyperspectral artificial intelligence: When nonconvex modeling meets hyperspectral remote sensing," *IEEE Geosci. Remote Sens. Mag.*, vol. 9, no. 2, pp. 52–87, Jun. 2021.

[7] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 11, pp. 3804–3814, Nov. 2008.

[8] W. Huang, Y. Huang, Z. Wu, J. Yin, and Q. Chen, "A multi-kernel mode using a local binary pattern and random patch convolution for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 4607–4620, Apr. 2021, doi: 10.1109/JSTARS.2021.3076198.

[9] L. He, C. Liu, J. Li, Y. Li, S. Li, and Z. Yu, "Hyperspectral image spectral-spatial-range Gabor filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 7, pp. 4818–4836, Jul. 2020.

[10] B. Liu, A. Yu, X. Tan, and R. Wang, "Slow feature extraction for hyperspectral image classification," *Remote Sens. Lett.*, vol. 12, no. 5, pp. 429–438, 2021.

[11] X. Yu, R. Wang, B. Liu, and A. Yu, "Salient feature extraction for hyperspectral image classification," *Remote Sens. Lett.*, vol. 10, no. 6, pp. 553–562, Jun. 2019.

[12] D. Hong, X. Wu, P. Ghamisi, J. Chanussot, N. Yokoya, and X. X. Zhu, "Invariant attribute profiles: A spatial-frequency joint feature extractor for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 3791–3808, Jun. 2020.

[13] D. Hong, N. Yokoya, J. Chanussot, J. Xu, and X. X. Zhu, "Joint and progressive subspace analysis (JPSA) with spatial-spectral manifold alignment for semisupervised hyperspectral dimensionality reduction," *IEEE Trans. Cybern.*, vol. 51, no. 7, pp. 3602–3615, Jul. 2021.

[14] B. Liu, X. Yu, P. Zhang, A. Yu, Q. Fu, and X. Wei, "Supervised deep feature extraction for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 1909–1921, Apr. 2018.

[15] B. Liu, X. Yu, A. Yu, P. Zhang, G. Wan, and R. Wang, "Deep few-shot learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2290–2304, Apr. 2019.

[16] L. Zhang, M. Lan, J. Zhang, and D. Tao, "Stagewise unsupervised domain adaptation with adversarial self-training for road segmentation of remote-sensing images," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2021.3104032.

[17] W. Hu, Y. Huang, I. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, no. *Pt.3*, 2015, Art. no. 258619.

[18] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.

[19] Y. Chen, X. Zhao, and X. Jia, "Spectral-spatial classification of hyperspectral data based on deep belief network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2381–2392, Jun. 2015.

[20] H. Zhang, Y. Li, Y. Jiang, P. Wang, Q. Shen, and C. Shen, "Hyperspectral classification based on lightweight 3-D-CNN with transfer learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5813–5828, Aug. 2019.

[21] B. Liu, X. Yu, A. Yu, P. Zhang, and G. Wan, "Spectral-spatial classification of hyperspectral imagery based on recurrent neural networks," *Remote Sens. Lett.*, vol. 9, no. 12, pp. 1118–1127, 2018.

[22] X. Ding *et al.*, "An adaptive capsule network for hyperspectral remote sensing classification," *Remote Sens.*, vol. 13, no. 13, 2021, Art. no. 2445.

[23] R. Hang, Q. Liu, D. Hong, and P. Ghamisi, "Cascaded recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5384–5394, Aug. 2019.

[24] C. Yu, R. Han, M. Song, C. Liu, and C.-I. Chang, "Feedback attention-based dense CNN for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2021.3058549.

[25] R. Hang, Z. Li, Q. Liu, P. Ghamisi, and S. S. Bhattacharyya, "Hyperspectral image classification with attention-aided CNNS," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 2281–2293, Mar. 2021.

[26] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5966–5978, Jul. 2021.

[27] Y. Zheng, J. Li, Y. Li, J. Guo, X. Wu, and J. Chanussot, "Hyperspectral pansharpening using deep prior and dual attention residual network," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 11, pp. 8059–8076, Nov. 2020.

[28] J. Xie, N. He, L. Fang, and P. Ghamisi, "Multiscale densely-connected fusion networks for hyperspectral images classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 1, pp. 246–259, Jan. 2021.

[29] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2811–2821, May 2018.

[30] Q. Liu, L. Xiao, J. Yang, and Z. Wei, "CNN enhanced graph convolutional network with pixel and superpixel level feature fusion for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 10, pp. 8657–8671, Oct. 2021.

[31] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019.

[32] L. Wang, S. Hao, Q. Wang, and Y. Wang, "Semi-supervised classification for hyperspectral imagery based on spatial-spectral label propagation," *ISPRS J. Photogrammetry Remote Sens.*, vol. 97, pp. 123–137, 2014.

[33] K. Tan, E. Li, Q. Du, and P. Du, "An efficient semi-supervised classification approach for hyperspectral imagery," *ISPRS J. Photogrammetry Remote Sens.*, vol. 97, pp. 36–45, 2014.

[34] J. Wang, N. Jiang, G. Zhang, B. Hu, and Y. Li, "Automatic framework for semi-supervised hyperspectral image classification using self-training with data editing," in *Proc. 7th Workshop Hyperspectral Image Signal Process.: Evol. Remote Sens.*, 2015, pp. 1–4.

[35] S. Samiappan and R. J. Moorhead, "Semi-supervised co-training and active learning framework for hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2015, pp. 401–404.

[36] X. Wang, K. Tan, Q. Du, Y. Chen, and P. Du, "Caps-tripleGAN: GAN-assisted capsnet for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 7232–7245, Sep. 2019.

[37] R. Hang, F. Zhou, Q. Liu, and P. Ghamisi, "Classification of hyperspectral images via multitask generative adversarial networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 2, pp. 1424–1436, Feb. 2021.

[38] J. Yang, Y.-Q. Zhao, and J. C.-W. Chan, "Learning and transferring deep joint spectral-spatial features for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4729–4742, Aug. 2017.

[39] X. Ma, X. Mou, J. Wang, X. Liu, H. Wang, and B. Yin, "Cross-data set hyperspectral image classification based on deep domain adaptation," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 10164–10174, Dec. 2019.

[40] P. Zhou, J. Han, G. Cheng, and B. Zhang, "Learning compact and discriminative stacked autoencoder for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4823–4833, Jul. 2019.

[41] G. Cheng, Z. Li, J. Han, X. Yao, and L. Guo, "Exploring hierarchical convolutional features for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6712–6722, Nov. 2018.

[42] B. Liu, A. Yu, X. Yu, R. Wang, K. Gao, and W. Guo, "Deep multiview learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*,vol. 59, no. 9, pp. 7758–7772, Sep. 2021.

[43] K. Gao, W. Guo, X. Yu, B. Liu, A. Yu, and X. Wei, "Deep induction network for small samples classification of hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 3462–3477, Jun. 2020, doi: 10.1109/JSTARS.2020.3002787.

[44] B. Liu *et al.*, "Active deep densely connected convolutional network for hyperspectral image classification," *Int. J. Remote Sens.*, vol. 42, no. 15, pp. 5915–5934, 2021.

[45] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., Cham, Switzerland: Springer, 2015, pp. 234–241.

[46] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[47] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6230–6239.

[48] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[49] Z. Zheng, Y. Zhong, A. Ma, and L. Zhang, "FPGA: Fast patch-free global learning framework for fully end-to-end hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 8, pp. 5612–5626, Aug. 2020.

[50] Y. Xu, B. Du, and L. Zhang, "Beyond the patchwise classification: Spectral-spatial fully convolutional networks for hyperspectral image classification," *IEEE Trans. Big Data*, vol. 6, no. 3, pp. 492–506, Sep. 2020.

[51] B. Pan, X. Xu, Z. Shi, N. Zhang, H. Luo, and X. Lan, "DSSNet: A simple dilated semantic segmentation network for hyperspectral imagery classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 11, pp. 1968–1972, Nov. 2020.

[52] D. Wang, B. Du, and L. Zhang, "Fully contextual network for hyperspectral scene parsing," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2021.3050491.

[53] H. Sun, X. Zheng, and X. Lu, "A supervised segmentation network for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 30, pp. 2810–2825, Feb. 2021, doi: 10.1109/TIP.2021.3055613.

[54] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*.

[55] L. Xu, C.-S. Choy, and Y.-W. Li, "Deep sparse rectifier neural networks for speech denoising," in *Proc. IEEE Int. Workshop Acoust. Signal Enhancement*, 2016, pp. 1–5.

[56] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1800–1807, doi: 10.1109/CVPR.2017.195.

[57] A. G. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017.

[58] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *Proc. European Conf. Comput. Vis.*, 2018, pp. 334–349.

[59] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "Bisenet V2: bilateral network with guided aggregation for real-time semantic segmentation," *Int. J. Comput. Vis.*, pp. 3051–3068, 2021, doi: 10.1007/s11263-021-01515-2.

[60] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," *CoRR*, vol. abs/1602.07261, 2016.

[61] L. He, J. Li, A. Plaza, and Y. Li, "Discriminative low-rank Gabor filtering for spectral-spatial hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 3, pp. 1381–1395, Mar. 2017.

[62] P. Ghamisi, J. Plaza, Y. Chen, J. Li, and A. Plaza, "Advanced supervised spectral classifiers for hyperspectral images: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 1, pp. 8–32, Mar. 2017.

[63] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

[64] B. Deng, S. Jia, and D. Shi, "Deep metric learning-based feature embedding for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 2, pp. 1422–1435, Feb. 2020.

**Bing Liu** received the B.S. degree in measurement and control engineering, the M.S. degree in pattern recognition and intelligent system, and the Ph.D. degree in surveying and mapping science and technology from Information Engineering University, Zhengzhou, China, in 2013, 2016, and 2019, respectively.

He is currently working with the Information Engineering University as an Associate Professor. His research interests include machine learning, pattern recognition, and signal processing in earth observation.

Dr. Liu is an active Reviewer for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, IEEE *Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, IEEE TRANSACTIONS ON CYBERNETICS, *International Journal of Remote Sensing*, *Remote Sensing Letter*, and *Journal of Applied Remote Sensing*.

**Xuchu Yu** received the Ph.D. degree in photogrammetry and remote sensing from the Institute of Surveying and Mapping, Zhengzhou, China, in 1997.

He is currently working with the Information Engineering University, Zhengzhou, China, as a Professor and Doctoral Supervisor. His research interests include photogrammetry, remote sensing, and pattern recognition.