Received 15 May 2023, accepted 25 July 2023, date of publication 1 August 2023, date of current version 7 August 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3300422*

RESEARCH ARTICLE

# Multi-Branch Hybrid Network Based on Adaptive Selection of Spatial-Spectral Kernel for Hyperspectral Image Classification

**CAILING WANG**[1], **HE FU**[1], **AND HONGWEI WANG**[2]

[1]College of Computer Science, Xi'an Shiyou University, Xi'an 710065, China
[2]College of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, China

Corresponding author: Cailing Wang (azering@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61401439, Grant 41301480, and Grant 41301382; in part by the Shaanxi Provincial Key Research and Development Program under Grant 2019GY-112; in part by the Higher Education Research Project of the Xi'an Aeronautical Institute under Grant 2018GJ1005; in part by the Shaanxi 13th Five-Year Plan of Education Science under Grant SGH18H435; and in part by the Postgraduate Innovation and Practice Training Program of Xi'an Shiyou University under Grant YCS22213165.

**ABSTRACT** The current limited sample set and mixed spatial-spectral information make effective feature extraction in hyperspectral image (HSI) classification challenging. To better extract spatial-spectral features, enhance the robustness of the learned features against the orientation and scale changes and improve the convergence of the network used for HSI classification, we propose a multi-branch hybrid network (MHNet) based on adaptive selection of spatial-spectral kernels in this paper. Specifically, we use the Gabor convolutional layer as the first layer of this network model. Since the predefined multi-scale and multi-directional Gabor filters in this layer can better characterize the internal spatial-spectral structure of HSI data from different perspectives, the robustness of the model to orientation-scale changes is enhanced. Then the performance of joint spatial-spectral feature extraction is improved by learning adaptive selective 3D convolution kernels. Subsequently, a two-branch network is employed to further fully extract spatial and spectral information for classification accuracy. Experimental results on three public hyperspectral datasets show that the proposed MHNet not only has better classification performance than several existing widely used machine learning and deep learning-based methods, but also it has fast model convergence.

**INDEX TERMS** Multi-branch network, hyperspectral image, selecting spatial-spectral kernels, Gabor, residual network, spatial-spectral features.

## I. INTRODUCTION

Hyperspectral images (HSIs) captured by hyperspectral sensors are typically composed of hundreds of spectral data channels in the same area [1], [2], [3], which contain rich spectral and spatial information. Each pixel in HSIs has hundreds of narrow spectral bands, so HSIs have high spectral resolutions. Due to their high spectral resolutions and abundant spatial-spectral information, HSIs are extensively utilized in many different fields, such as environmental monitoring [4], agricultural remote sensing [5], mineral exploitation [6], ocean remote sensing [7] and geological mapping [8]. In recent years, the primary focus of HSI processing is HSI classification, spectral unmixing and HSI anomaly detection. Among them, HSI classification is the most dynamic research area in the hyperspectral field [3] and has received extensive attention in the field of remote sensing image analysis. So, this paper focuses on HSI classification.

The purpose of HSI classification is to determine the class of objects per pixel, and this technique has been frequently utilized in fields like ground object recognition [9], hyperspectral image change detection [10], and object detection [11], [12], among others. Classification methods for HSIs can be broadly classified into two categories [13], one based

The associate editor coordinating the review of this manuscript and approving it for publication was Abdullah Iliyasu.

on hand-crafted feature extraction methods and the other based on learning feature extraction methods. Traditional HSI classification methods are mostly based on hand-crafted features [14], [15], [16], [17], [18], such as morphological profile (MP) features [14], [15], texture descriptors [16] and spectrum [17], etc. These methods achieve classification by designing different hand-crafted features and feeding them into classifiers, such as random forest (RF) [19] and support vector machine (SVM) [20] based on supervised machine learning. Most studies focus on feature extraction to mine the features of the target more efficiently. For example, Melgani and Bruzzone [20] evaluated the potential of SVM in the hyperdimensional spectral feature space and demonstrated that SVM outperforms traditional pattern recognition methods (i.e. radial basis function network and K-nearest neighbor method) for HSI classification. Zhong and Zhang [21] designed an artificial antibody network (ABNet) to extract HSI spectral features for classification accuracy. However, early hand-crafted feature extraction methods focused more on extracting spectral features and ignored the spatial feature information of neighboring pixel locations. It is difficult to achieve satisfactory classification results using only spectral information. Therefore, numerous joint spatial-spectral classification algorithms have been proposed to fully utilize the spatial-spectral information for improving the classification performance. For example, Marpu et al. [22] proposed automatic construction of extended attribute profiles with standard deviation attributes for remote sensing image classification. For the purpose of extracting spectral spatial characteristics, Kang et al. [23] designed a method based on edge-preserving filtering. Guo et al. [24] developed a multiclass support tensor machine (STM) for identifying information classes of tensor spaces in HSIs. However, the flexibility of feature extraction and the capacity to automatically learn the model's parameters are lacking in these traditional hand-crafted feature extraction methods.

Among the deep learning-based feature extraction methods, convolutional neural networks are extensively used in various fields of visual information processing because of their powerful image processing capabilities, such as image categorization [25], semantic segmentation [26], depth estimation [27], and object identification [28]. Convolutional neural networks also show their powerful performance in the area of processing hyperspectral images. For example, principal component analysis, logistic regression, and a deep learning framework were all combined by Chen et al. [29] to learn the joint spectral-spatial features of HSIs for categorization. Tao et al. [30] developed two feature learning variations, namely multi-scale spatial feature learning and sparse spectral feature learning, and used an unsupervised approach of stacked sparse autoencoders to learn spatial-spectral information. Gabor filter is now widely used in image processing due to its ability to capture discriminative characteristics from various scales and orientations. Most commonly, it is embedded into the CNN framework as prior knowledge in some preprocessing models for enhancing the CNN performance. Kwolek [31] combined the Gabor filter with CNN for face detection. He first transformed the original image with Gabor filters to obtain feature maps with Gabor characteristics and then sent these Gabor feature maps to CNN for classification. Yao et al. [32] proposed Gabor-CNN for target recognition in natural scenes, combining CNN with Gabor filters to enhance the extraction of edge texture information. Gabor filters also have applications in the field of HSI processing. Chen et al. [33] classified HSIs using Gabor filtering and convolutional neural networks to alleviate the overfitting problem. Jia et al. [34] incorporated a 3D Gabor filter into the convolutional kernel to process HSIs, resulting in a significant reduction in model parameters and avoiding overfitting. In order to adequately extract spatial-spectral characteristics to improve classification accuracy, a great number of new models have been generated. Hamida et al. [35] designed a 3D deep learning method to process joint spectral and spatial information. Song et al. [36] developed a deep feature fusion network (DFFN) to categorize HSI. A multiscale three-dimensional convolutional neural network (M3D-DCNN) was used in [37] to jointly learn two-dimensional multiscale spatial features and one-dimensional spectral features in an end-to-end way. In [38], local spatial spectral correlations between nearby single pixel vectors are concurrently exploited to optimize and explore local contextual interactions. To reduce the complexity of network design, Paoletti et al. [39] designed a new CNN architecture Capsnet, which has improved the accuracy of HSI classification. Hong et al. [40] designed miniGCN network to infer out-of-sample data, i.e., without retraining the network and improving the classification performance. Some successful pre-trained CNNs were used in [41] for high resolution remote sensing (HRRS) scene classification. Meanwhile, there have been further attempts to use different branching networks or divergence networks to jointly extract spatial-spectral features. For example, Kang et al. [42] combined residual network and dense convolutional network into a dual-path network (DPN) to classify HSI. Han et al. [43] designed a two-stream network based on different scales and combined with a spatial enhancement strategy to jointly extract spatial-spectral features. Yang et al. [44] established a two-branch network, one for extracting spectral domain features and another for extracting spatial domain features, and connected the learned spectral domain features and spatial domain features for classification.

Currently, some deep learning-based methods, on the one hand, make the model construction complicated due to the increase of network depth, generate a great number of parameters and redundant features during training, and slow down the convergence speed [34], [42], [45]. On the other hand, the correlation between the spectral and spatial domains in HSIs is ignored, and a significant quantity of valuable data is wasted [43], [44]. To solve the above problems, in this paper we propose a novel multi-branch hybrid network based on adaptive selection of spatial-spectral kernels for HSI classification. The model integrates a Gabor layer to accelerate convergence and enhance the robustness to changes in direction and scale. The two-branch network is employed to fully extract spatial and spectral feature information from HSIs to improve classification accuracy. The main contributions of this paper are as follows.

1) A convolutional layer with multi-scale and multi-directional Gabor filters is used as the first layer of this network model to enhance the robustness of the model against changes in orientation and scale, as well as to increase the speed of model convergence. Predefined multi-scale and multi-directional Gabor filters can describe the internal spatial-spectral structure of HSI data from different angles.
2) A strategy for learning adaptive selective 3D spatial-spectral kernels is designed to improve the ability of joint extraction of spatial-spectral features.
3) A two-branch network framework is designed for improving classification accuracy. One introduces an effective feature recalibration (EFR) mechanism. The classification accuracy is improved by automatically adjusting the size of neuronal receptive fields (RF) and enhancing cross-channel dependence among features. The other uses a hybrid combination of 2D and 3D convolution (2D-3DCNN). The 3D CNN learns joint spatial-spectral features from HSI data, and the 2D CNN further learns abstract spatial features to improve classification performance and reduce the model complexity.

The rest of the paper is organized as follows. Section II details the proposed network model framework, Section III gives the experimental results and analysis, and finally, conclusions are presented in Section IV.

## II. PROPOSED METHOD

Fig. 1 shows the general framework diagram of the proposed multi-branch hybrid network (MHNet) based on adaptive selection of spatial-spectral kernels. The overall framework can be divided into three major modules, Gabor CNN module, adaptive selection of spatial-spectral kernel network (SKNet) module, and multi-branch hybrid network (Multi-branchNet) module. Spatial-spectral feature information is extracted jointly by these three modules. Let $X_{raw} \in \mathbb{R}^{H \times W \times B}$ be the original unprocessed 3D HSI with height $H$, width $W$ and number of spectral channels $B$. The overall count of categories contained in the corresponding groundtruth for a block of pixels $x_{i,j} \in X_{raw}\ (i = 1, 2, \ldots, W.\quad j = 1, 2, \ldots, H)$ on a HSI is $C$, and the land cover category is noted as $Y = y_1, y_2, \ldots, y_C$. A subset $P \in \mathbb{R}^{S \times S \times B}$ of neighbor blocks is created from the $X_{raw}$ cube, $S$ represents the width and height of neighbor blocks. The superimposed neighbor blocks $P \in \mathbb{R}^{S \times S \times B}$ are provided as the input of the model. The category of the middle pixel of neighbor blocks is used as the category of neighbor blocks.

### A. GABOR FILTER

The kernel functions of Gabor filters [46] have a sinusoidal plane wave based on a specific frequency and orientation, as shown in Fig. 2, which enables them to extract the spatial frequency structure of images [47]. Gabor features are a type of features that can be used to characterize an image's texture information. Additionally, Gabor wavelets are sensitive to the image's edges and can offer superior scale and orientation selection properties, which can extract pertinent features at various scales and orientations in the frequency domain. The Gabor filter offers good adaptability to changes in illumination, tolerates some degree of rotation and distortion of the image, and has certain robustness to illumination and posture.

Gabor wavelets [46] use complex functions as the basis for the Fourier transform in information theoretic applications. It was shown that Gabor wavelets (filters) are comparable to the human visual system in frequency and directional expression [48], [49]. Gabor wavelets (filters) are defined as follows:

$$\Psi_{\omega,\theta}(x, y) = \frac{1}{2\pi\sigma^2} exp\left(-\frac{x'^2 + y'^2}{2\sigma^2}\right) exp\left(j\omega x'\right)$$
$$x' = x\cos\theta + y\sin\theta \quad , \quad y' = -x\sin\theta + y\cos\theta \tag{1}$$

where $(x, y)$ represents the position of the pixel in the spatial domain, $\omega$ represents the central angular frequency of the sinusoidal plane wave, $\theta$ represents the direction of the Gabor filter, and $\sigma$ represents the standard deviation of the Gaussian function along the $x$ and $y$ directions in the spatial domain [48]. According to the experimental parameter settings in [50], in this experiment, we set the value of $\sigma$ to $\sigma \approx \frac{\pi}{\omega}$.

In the Gabor convolutional layer of the proposed model, we chose a filter bank with 5 scales and 8 orientations, as shown in Fig. 2. The frequencies and orientations of the Gabor filters are defined as follows:

$$\omega_m = \frac{\pi}{2} \times \sqrt{2}^{-(m-1)} \quad , \quad \theta_n = \frac{\pi}{8}(n-1)$$
$$(m = 1, 2, \ldots, 5 \quad , \quad n = 1, 2, \ldots, 8) \tag{2}$$

HSI data $P(x, y)$ through the convolution operation with Gabor filter $\Psi_{\omega,\theta}(x, y)$ can be expressed as

$$G_{m,n}(x, y) = P(x, y) * \Psi_{\omega,\theta}(x, y) \tag{3}$$

where $*$ stands for convolution operation.

### B. ADAPTIVE SELECTION OF SPATIAL-SPECTRAL KERNEL MODULE

The receptive field (RF) plays an important role in deep CNNs [51], [52], [53], and its size directly affects the information of the captured target objects [53]. To jointly extract spatial-spectral features and allow neurons to adjust the size of their receptive fields in an adaptive manner, a selection among multiple convolutional kernels with different size receptive fields is required [51]. Specifically, three operational processes are employed, namely, split, fusion and

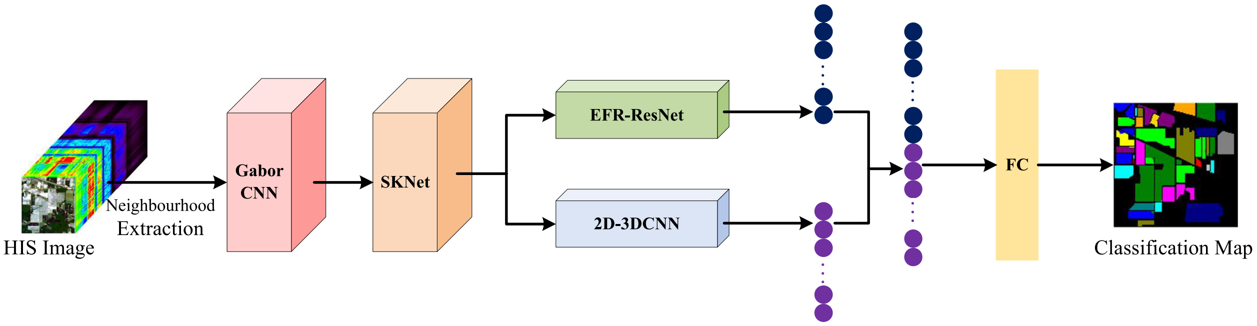


**FIGURE 1.** General framework of the proposed MHNet method.

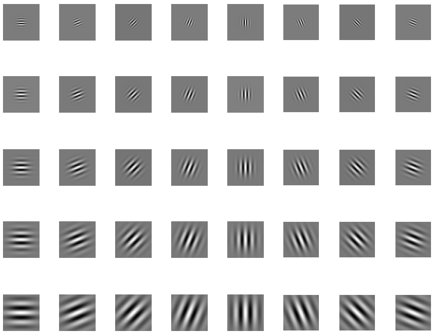


**FIGURE 2.** Gabor filter at 5 scales and 8 orientations.

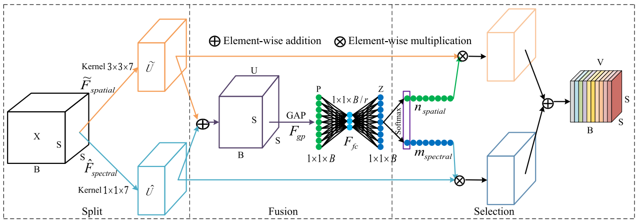


**FIGURE 3.** Adaptive selection of spatial-spectral kernel module.

selection. As shown in Fig. 3, $X \in \mathbb{R}^{S \times S \times B}$ represents the feature map of the input HSI and $V \in \mathbb{R}^{S \times S \times B}$ represents the output feature map that is processed by adaptively selected spatial-spectral kernel convolution. The process can be expressed as

$$V = \mathcal{F}_{SK}(X) \tag{4}$$

### 1) SPLIT

For a given input feature map $X \in R^{S \times S \times B}$, the spatial transformation $X \overset{\tilde{\mathcal{F}}_{spatial}}{\rightarrow} \tilde{U} \in \mathbb{R}^{S \times S \times B}$ and the spectral transformation $X \overset{\hat{\mathcal{F}}_{spectral}}{\rightarrow} \hat{U} \in \mathbb{R}^{S \times S \times B}$ are performed respectively. $\tilde{U}$ and $\hat{U}$ are obtained by 3D convolution, batch normalization (BN) [54] and activation function ReLU [55]. The transformation $\tilde{\mathcal{F}}_{spatial}$ is used for extracting spatial features, and the size of the convolution kernel for its operation is $3 \times 3 \times 7$. The transformation $\hat{\mathcal{F}}_{spectral}$ is used for extracting spectral features, and the size of the convolution kernel for its operation is $1 \times 1 \times 7$. The whole convolution process is as follows:

$$\tilde{U} = \tilde{\mathcal{F}}_{spatial}(X) = X * W_{spatial} + b \tag{5}$$

$$\hat{U} = \hat{\mathcal{F}}_{spectral}(X) = X * W_{spectral} + b \tag{6}$$

where $*$ represents 3D convolution operation. $W$ and $b$ represent the weight parameter and biases, respectively. The size of $W_{spatial}$ is $3 \times 3 \times 7$ and the size of $W_{spectral}$ is $1 \times 1 \times 7$. The subsequent BN [54] and ReLU [55] operations can be expressed as

$$X^{l+1} = ReLU\left(\sum_{j=1}^{x^l} \mathcal{F}_{bn}\left(X_j^l\right) * W^{l+1} + b^{l+1}\right) \tag{7}$$

$$\mathcal{F}_{bn}\left(X_j^l\right) = \frac{X_j^l - \mu\left(X_j^l\right)}{\sqrt{\sigma^2\left(X_j^l\right) + \epsilon}} \cdot \gamma + \beta \tag{8}$$

$X_j^l$ represents the $j$th feature map of $X^l$ in layer $l$, $x^l$ represents the number of feature maps in $X^l$ in layer $l$, and $W^{l+1}$ and $b^{l+1}$ represent the weight parameter and biases of the convolutional filter bank in layer $l+1$, respectively. $\mu(\cdot)$ and $\sigma^2(\cdot)$ are the batch means and variances of the corresponding input data $X_j^l$. $\gamma$ and $\beta$ are the learned parameter vectors. BN is to prevent gradient disappearance and to speed up model training. ReLU introduces nonlinearity into the convolutional features.

### 2) FUSION

As shown in Fig. 3, fusion is a combination and aggregation of information from spatial and spectral paths, which leads to a global and integrated representation of the selection weights of spatial and spectral information. Just as fusion in [51] is to enable neurons to adaptively adjust the size of their receptive fields depending on the learned features, in this paper, the aim of fusion is to allow neurons to jointly extract spatial-spectral features by adaptively adjusting the size of their receptive fields while enhancing their multiscale streaming information. To achieve this, we first use element-wise summation to fuse information from multiple branches, i.e., add $\hat{u}_{i,j,c} \in \hat{U}^{(l+1)}$ and $\tilde{u}_{i,j,c} \in \tilde{U}^{(l+1)}$ element by element to produce the output $U^{(l+1)} \in \mathbb{R}^{S\times S\times B}$, denoted as follows:

$$U^{l+1} = \hat{U}^{(l+1)} \oplus \tilde{U}^{(l+1)} \tag{9}$$

where $\oplus$ represents the sum of elements.

Then global average pooling (GAP) is done for $U^{(l+1)}$:

$$s_b^{l+1} = \frac{1}{S\times S}\sum\nolimits_{i=1}^{S}\sum\nolimits_{j=1}^{S} u_{i,j,b}^{(l+1)} \tag{10}$$

The above equation can be viewed as compressing the $b$th feature map in $U^{(l+1)}$ into one element value along the spatial direction. For a total of $B$ feature maps, the channel description vector is $s_b^{l+1} \in \mathbb{R}^{1\times 1\times B}$ at layer $l+1$.

Learning of compact features and cross-channel dependencies is accomplished via GAP and a fully connected (FC) layer. The computation of the FC layer can be expressed as

$$z^{(l+1)} = \mathcal{F}_{fc}\left(s_b^{l+1}\right) = ReLU\left(W^{(l+1)}\cdot s_b^{l+1}\right) \tag{11}$$

where compact feature $z^{(l+1)} \in \mathbb{R}^{d\times 1}$ is created to achieve accurate and adaptive selection of the spatial-spectral kernel. We use the crucial factor $d = \max\left(\frac{B}{r}, L\right)$ to accomplish model convergence. $r$ is the dimensionality reduction ratio of $z^{(l+1)}$, it is set 2 [56]. $L$ is set to 32 according to [51].

### 3) SELECTION

The spectral and spatial information is adaptively selected on $z^{(l+1)}$ by cross-channel soft attention. Specifically, the softmax function is applied on $z^{(l+1)}$ to generate the spectral attention vector $m_{spectral}$ and the spatial attention vector $n_{spatial}$, respectively. The expressions are as follows:

$$m_{spectral}^{l+1} = \frac{e^{M^{(l+1)}z^{(l+1)}}}{e^{M^{(l+1)}z^{(l+1)}} + e^{N^{(l+1)}z^{(l+1)}}} \tag{12}$$

$$n_{spatial}^{l+1} = \frac{e^{N^{(l+1)}z^{(l+1)}}}{e^{M^{(l+1)}z^{(l+1)}} + e^{N^{(l+1)}z^{(l+1)}}} \tag{13}$$

where $M^{(l+1)} \in \mathbb{R}^{B\times d}$, $N^{(l+1)} \in \mathbb{R}^{B\times d}$. $m_{spectral}^{l+1}$ is the spectral soft attention vector corresponding to $\hat{U}^{(l+1)}$ and $n_{spatial}^{l+1}$ is the spatial soft attention vector corresponding to $\tilde{U}^{(l+1)}$.

The final feature map $V$ is computed by the attention weighting of the spectral and spatial directions.

$$V = \left(m_{spectral}^{l+1} \otimes \hat{U}^{(l+1)}\right) \oplus \left(n_{spatial}^{l+1} \otimes \tilde{U}^{(l+1)}\right) \tag{14}$$

$$m_{spectral}^{l+1} + n_{spatial}^{l+1} = 1 \tag{15}$$

where $V = [v_1, v_2, \ldots, v_B]$, $v_i \in \mathbb{R}^{S\times S}$, $i = 1, \ldots, B$. $\otimes$ represents the multiplication of elements.

## C. MULTI-BRANCH NETWORK

In this section we use a two-branch network to achieve joint extraction of high-level features in spatial and spectral domains for classification accuracy, as shown in Fig. 1. One is a residual network with an efficient feature recalibration (EFR-ResNet) mechanism. This branch improves classification accuracy by automatically modulating the size of neuronal receptive fields (RF) and enhancing cross-channel dependence between features. The other is a hybrid combination of 2D and 3D convolution (2D-3DCNN), where the 3D CNN learns joint spatial-spectral feature information from HSI data and the 2D CNN goes further to learn abstract spatial feature information, which improves classification performance and reduces model complexity. Afterwards, the information streams of the two branches are stitched together in series and classified through a FC layer.

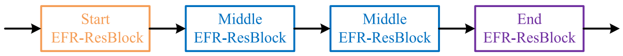


**FIGURE 4. Residual network framework with efficient feature recalibration mechanism.**

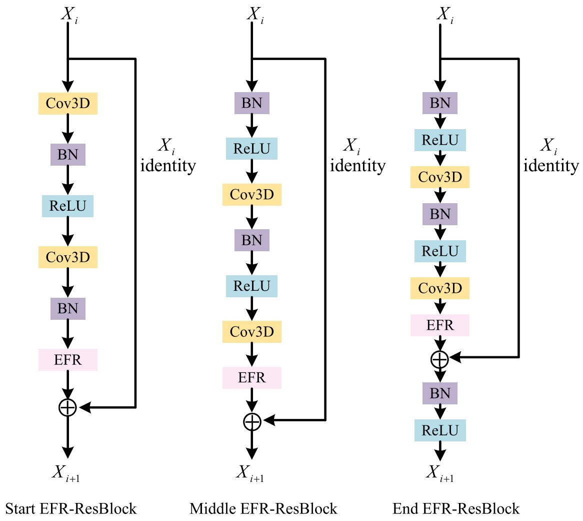


**FIGURE 5. EFR-ResBlock module.**

### 1) RESIDUAL NETWORKS WITH EFFICIENT FEATURE RECALIBRATION (EFR) MECHANISM

Fig. 4 illustrates the residual network framework, which has a total of four residual blocks in this branch network. To improve classification performance and avoid increasing the complexity of the model, we introduced an efficient spectral channel attention module [57], namely the efficient feature recalibration mechanism (EFR) [58], for all four residual blocks. The final composition is the EFR-ResBlock module. The detailed structure is shown in Fig. 5, from which the specific composition of the Start EFR-ResBlock module, Middle EFR-ResBlock module and End EFR-ResBlock module can be seen.

Each EFR-ResBlock module contains three operations with different arrangements of Conv3D, BN and ReLU, as shown in Fig. 5. Each of these EFR-ResBlock modules has two Conv3D operations. The adaptively selected spatial-spectral kernel feature maps generated by entering Equation 14 in each EFR-ResBlock module are subjected to two successive 3D convolution operations. Filter banks of kernel size $k_1 \times k_2 \times k_3$ are used at the $(l+1)$ and $(l+2)$

**TABLE 1. Convolutional kernel size and step size in each EFR-ResBlock.**

| Module | Kernels | Kernel shapes | Stride shapes |
|---|---|---|---|
| | | $k_1^i \times k_2^i \times k_3^i$ | $s_1^i \times s_2^i \times s_3^i$ |
| Start EFR-ResBlock | 24 | (1, 1, 7) | (0, 0, 3) |
| Middle EFR-ResBlock | | (1, 1, 7) | (0, 0, 3) |
| Middle EFR-ResBlock | | (3, 3, 1) | (1, 1, 0) |
| End EFR-ResBlock | | (3, 3, 1) | (1, 1, 0) |

layers, and the number of filter banks for each EFR-ResBlock module is 24. In Fig. 5, let the operation before EFR be the feedforward residual function $\mathcal{F}_{res}\left(X_j^l;\theta_1,\theta_2\right)$. $I\left(X_j^l\right)$ represents the identity mapping of $X_j^l$. $\mathcal{F}_{res}\left(X_j^l;\theta_1,\theta_2\right)$ enters $\mathcal{F}_{EFR}(\cdot)$ defined by the Equation 19, and then add $I\left(X_j^l\right) = X_j^l$. The whole operation process is as follows:

$$2X_j^{l+2} = I\left(X_j^l\right) + \mathcal{F}_{EFR}\left(\mathcal{F}_{res}\left(X_j^l;\theta_1,\theta_2\right)\right) \tag{16}$$

$$\mathcal{F}_{res}\left(X_j^l;\theta_1,\theta_2\right) = ReLU\left(X_j^{l+1}\right) * W_j^{l+2} + b_j^{l+2} \tag{17}$$

$$X_j^{l+1} = ReLU\left(X_j^l\right) * W_j^{l+1} + b_j^{l+1} \tag{18}$$

where $X_j^{l+1}$ and $X_j^{l+2}$ represent the output feature maps after 3D convolution at layers $l+1$ and $l+2$, respectively. $\theta_1 = \left\{W_j^{l+1}, W_j^{l+2}\right\}$, $\theta_2 = \left\{b_j^{l+1}, b_j^{l+2}\right\}$, $W_j = \left\{W_j^{l+i} \mid 1 \leq i \leq 2\right\}$ and $b_j = \left\{b_j^{l+1} \mid 1 \leq i \leq 2\right\}$ represent the weight parameter and bias of the $j$th residual block (ResBlock) at the $(l+1)$th convolution layer and the $(l+2)$th convolution layer, respectively.

In Fig. 4 there are four EFR-ResBlock modules and the number of groups of convolution filters in the 3D convolution layers in each EFR-ResBlock is 24. Convolution kernel size and step size in each EFR-ResBlock are shown in Table 1.

As shown in Table 1, the first two EFR-ResBlocks are employed to extract spectral features and the last two EFR-ResBlocks are employed to extract spatial features, i.e., these four EFR-ResBlocks are used to jointly learn spatial-spectral features for improving the capability of image classification. After the EFR-ResBlock module, GAP is utilized to compress the output feature map into a feature vector of size 1 × 1 × 24 along the spatial scale.

The efficient feature recalibration mechanism (EFR) in each residual block employs a local cross-channel interaction approach without dimensionality reduction, which reduces model complexity while maintaining high performance, as shown in Fig. 6.

The process can be expressed as

$$\begin{aligned} \hat{X} &= \mathcal{F}_{EFR}(X) \\ X &\in R^{S\times S\times B}, \hat{X} \in R^{S\times S\times B} \end{aligned} \tag{19}$$

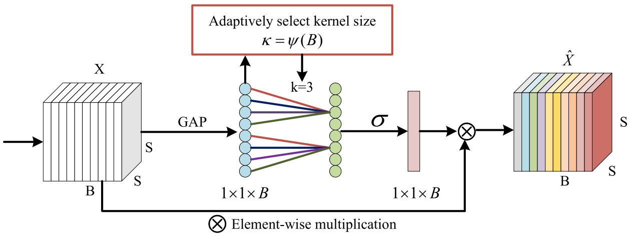


**FIGURE 6. EFR module.**

Specifically, a spectral channel descriptor vector is obtained by doing a spatial squeezing operation in the spatial dimension of the input feature map using GAP. The process is represented as follows:

$$y = GAP(X) = \frac{1}{S\times S}\sum\nolimits_{i=1}^{S}\sum\nolimits_{j=1}^{S} x_{i,j,c} \tag{20}$$

where $y \in R^{1\times1\times B}$, $x_{i,j,c}$ represents the value with coordinates $(i,j)$ on the feature map of the $c$th channel.

The two FC layers in the module are designed to capture nonlinear cross-channel interactions. Compared to the channel attention module in SENet [56], which uses dimensionality reduction to control the complexity of the model and thus causes inefficiencies in capturing dependencies across all channels, the efficient feature recalibration mechanism (EFR) used in this paper captures cross-channel interactions efficiently without dimensionality reduction. As shown in Fig. 6, EFR captures cross-channel interactions to learn efficient feature attention by using each channel and its $k$ local neighbors. It generates the channel weight values by performing a fast one-dimensional convolution with kernel size $k$. $k$ is obtained by adaptive selection of the channel dimension $B$. As described in [57], the weight value of each optimum in the channel descriptor vector can be obtained by linearly interacting each channel with its $k$ nearest neighbors. The same learning parameters are shared among these channels, that is

$$\omega_i = \sigma\left(\sum\nolimits_{j=1}^{k} \beta^j \cdot y_i^j\right), y_i^j \in \Omega_i^k \tag{21}$$

where $\Omega_i^k$ represents the set of $k$ adjoining channels $y_i$ and $\beta^j$ denotes the shareable weights related to each $y_i^j$. $\omega \in \mathbb{R}^B$ is a feature recalibration vector.

The transformation from $X^l$ to $\hat{X}^{l+1}$ with a smaller computational overhead is done here to emphasize the multi-convolutional channel features for higher classification accuracy. The transformation process from $X^l$ to $\hat{X}^{l+1}$ can be expressed as follows:

$$\hat{X}^{l+1} = \mathcal{F}_{scale}\left(x_c \cdot \omega_c\right) \quad \forall c \in [1, 2, \ldots, B] \tag{22}$$

where $\hat{X}^{l+1} \in \mathbb{R}^{S\times S\times B}$, $x_c \in X^l$, $X^l = [x_1, x_2, \ldots, x_B]$.

### 2) NETWORK WITH A HYBRID COMBINATION OF 2D AND 3D CONVOLUTION

The other network in the multi-branch network is a hybrid combination of 2D and 3D convolution. Firstly, the 3D CNN kernel is employed to extract both spectral and spatial features from the HSI data. Then, the 2D CNN kernel is adopted

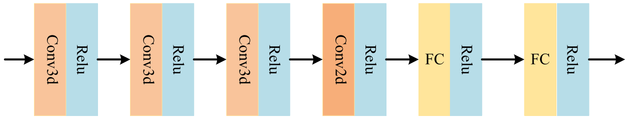


**FIGURE 7.** Flow chart of the framework of branch hybrid network.

**TABLE 2.** Convolutional kernel shape and size of each convolutional layer.

| Convolution kernel | Kernels | Kernel shapes | Stride shapes |
|---|---|---|---|
| | | $k_1^i \times k_2^i \times k_3^i$ | $s_1^i \times s_2^i \times s_3^i$ |
| Conv3d | 24 | (3, 3, 7) | (1, 1, 1) |
| | | (3, 3, 5) | (1, 1, 1) |
| | 32 | (3, 3, 3) | (1, 1, 1) |
| Conv2d | 64 | (3, 3) | (1, 1) |

to extract spatial features even further in the output feature map after convolution with the 3D CNN kernel. Thus, the spatial-spectral information are fully extracted. The framework flowchart of this branch network is shown in Fig. 7.

As shown in Fig. 7, there are three 3D convolutional modules, one 2D convolutional module and two fully connected layer modules in the branch network. The parameters of the convolutional kernels are shown in Table 2.

In the 3D convolution layer, the output feature map is accomplished by convolving the 3D convolution kernel with the 3D image data. The 3D convolution kernel is convolved with the 3D HSI of the adjacent spectral channels to facilitate the extraction of spectral feature information.The activation value at the 3D spatial coordinates $(x, y, z)$ on the $j$th feature map of the $i$th layer can be represented as $v_{i,j}^{x,y,z}$ i.e.

$$v_{i,j}^{x,y,z} = \phi\left(b_{i,j} + \sum_{\tau=1}^{d_{l-1}} \sum_{\lambda=-\eta}^{\eta} \sum_{\rho=-\gamma}^{\gamma} \sum_{\sigma=-\delta}^{\delta} \omega_{i,j,\tau}^{\sigma,\rho,\lambda} \times v_{i-1,\tau}^{x+\sigma,y+\rho,z+\lambda}\right) \tag{23}$$

In Equation 23, $\phi$ denotes the activation function, $b_{i,j}$ is the bias of the $j$th feature map in the $i$th layer, $d_{l-1}$ represents the number of feature maps in the $(l-1)$th layer, and also represents the depth of the corresponding kernel $\omega_{i,j}$ of the $j$th feature map in the $i$th layer. $2\eta + 1$ is the depth of kernel along spectral dimension, $2\gamma + 1$ and $2\delta + 1$ are the width and height of the 3D convolution kernel, respectively. $\omega_{i,j}$ also indicates the weight of the $j$th feature map of the $i$th layer.

In the 2D convolution layer, the convolution kernel is used channel-by-channel to extract spatial feature information of the input feature map. The convolutional features are introduced nonlinearly through the activation function. The activation value at the 2D planar coordinates $(x, y)$ on the $j$th feature map of the $i$th layer can be expressed as $v_{i,j}^{x,y}$, i.e.

$$v_{i,j}^{x,y} = \phi\left(b_{i,j} + \sum_{\tau=1}^{d_{l-1}} \sum_{\rho=-\gamma}^{\gamma} \sum_{\sigma=-\delta}^{\delta} \omega_{i,j,\tau}^{\sigma,\rho} \times v_{i-1,\tau}^{x+\sigma,y+\rho}\right) \tag{24}$$

In Equation 24, $2\gamma + 1$ and $2\delta + 1$ are the width and height of the 2D convolution kernel, respectively. Other parameters are the same as Equation 23.

The spectral information in the input raw HSI data after three 3D convolutions is saved in the output feature map, and then the spatial information of different spectral bands is differentiated by 2D convolution. The whole convolution process uses back propagation to initialize all the weights and update the parameters.

Finally, the information streams learned by the multi-branch networks are fused in series and classified through a FC layer, where a cross-entropy loss function is used, i.e.

$$Loss = -\frac{1}{M} \sum_{m=1}^{M} \sum_{c=1}^{C} y_c^m \log \hat{y}_c^m \tag{25}$$

where $C$ represents the category of land cover class, $M$ represents the sample batch size, $y$ denotes the true value label, and $\hat{y}$ represents the predicted value.

## III. EXPERIMENTAL RESULTS

To demonstrate the effectiveness of the proposed MHNet method, we conducted experiments on three HSI datasets, namely IP, UP and KSC datasets.

### A. DATA SETS

1) Indian Pines (IP) dataset: The IP dataset consists of hyperspectral bands from a single landscape in Indiana, USA, with a pixel size of $145 \times 145$. The data collection contains 220 spectral reflection bands for each pixel that correspond to various regions of the electromagnetic spectrum in the 400-2500 nm range. This is shown in Fig. 8.
2) Pavia University (UP) dataset: The UP dataset is collected by the Reflection Optical System Imaging Spectrometer sensor over the city of Pavia, Italy. The image includes 115 spectral bands and a resolution of $610 \times 340$ pixels. There are 42,776 labeled samples covering 9 categories, namely, asphalt, grass, gravel, trees, metal plates, bare soil, asphalt, bricks, and shadows. As in Fig. 9.
3) Kennedy Space Center (KSC) dataset: The KSC dataset was collected on March 23, 1996 with NASA AVIRIS instrument overhead at Kennedy Space Center (KSC), Florida. 224 bands of 10 nm width were collected, and after some processing, it ended up being 176 bands for analysis. In total, there are 13 land cover classes, as shown in Fig. 10.

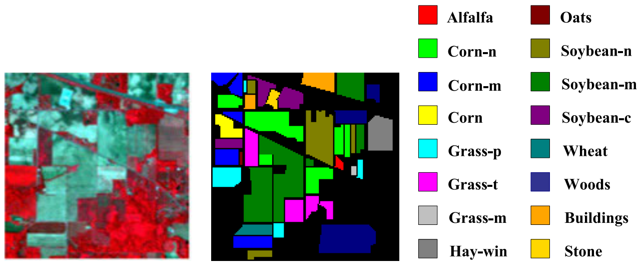
The specific information of the three datasets is shown in Table 3. The experiments are preprocessed by normalizing these three data, and the neighborhood extraction of the three datasets are $9 \times 9 \times 200$, $9 \times 9 \times 103$ and $9 \times 9 \times 176$, respectively. Each of these three datasets was randomly divided into two mutually exclusive training and test sets, with 10% of the training set and 90% of the test set.

### B. PARAMETER SETTING

The Gabor convolutional layer uses a filter bank with 5 scales and 8 directions. It is parameterized and initialized in the manner of [59]. Fig. 3 illustrates the shape and size of the

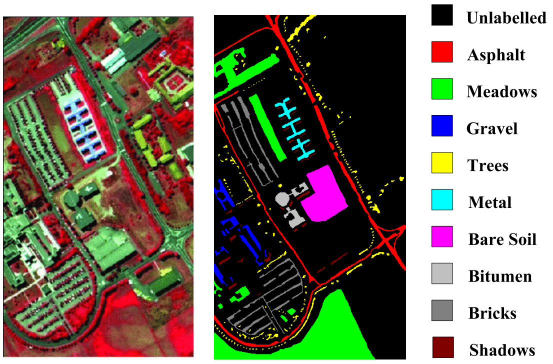
**TABLE 3.** Specific information of the three data sets.

| Description | Dataset | | |
|---|---|---|---|
| | IP | UP | KSC |
| Spatial Dimension | 145 × 145 | 610 × 340 | 512 × 614 |
| Spectral band | 200 | 103 | 176 |
| Ground cover category | 16 | 9 | 13 |
| Total number of samples | 10249 | 42776 | 5202 |



(a) (b) (c)

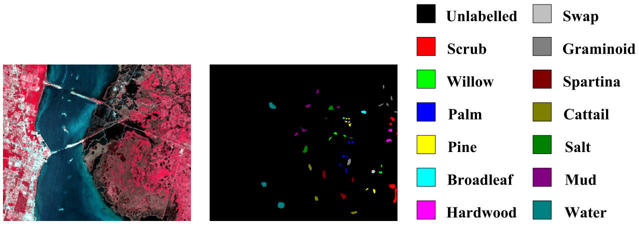
**FIGURE 8.** (a) False-color map (b) ground-truth map (c) labels of the Indian Pines dataset.



(a) (b) (c)

**FIGURE 9.** (a) False-color map (b) ground-truth map (c) labels of the Pavia University dataset.

convolution kernel in the adaptive selection of spatial-spectral kernel module. The parameters of the convolutional kernels in the two branch networks are set as in Table 2 and Table 3. In this paper we use the Adam [60] optimizer to update the weights of the 3D spatial-spectral filter set. For the sake of comparison, we fix the spatial window of all three data sets as 9 × 9 × $B$. Based on the combination of time, computational cost and accuracy, the whole experiment was repeated six times and the average accuracy was calculated to eliminate errors for obtaining the final classification results. That is, the values in the classification result table are the average of the six classification results. BN [54] is applied to prevent overfitting problems. The test set is evaluated using the configuration with the highest accuracy in each epoch, and the model uses a single-cycle strategy [61] to obtain the optimal learning rate. Once the optimal learning rate was obtained, the learning rate was adjusted for all epochs of each data set by using a cosine annealing scheduler.



(a) (b) (c)

**FIGURE 10.** (a) False-color map (b) ground-truth map (c) labels of the Kennedy Space Center dataset.

### C. COMPARISON OF BRANCH NETWORK INTEGRATION METHODS

The combination of branch networks has a remarkable influence on the parameter configuration, complexity and classification performance of the whole network, so it is extremely important to determine the optimal combination. To investigate the effect of the branch network binding methods on the classification performance, in this paper, we compare three branch network binding methods as shown in Fig. 11. These three binding methods are named as method I, method II and method III.

Method I indicates that the joint spatial-spectral features learned by the two branch networks are stitched together in series and then classified through the dense and fully connected layers. Method II is to average the spatial-spectral features learned by the two-branch network and apply a fully connected layer for classification. Method III is to concatenate the features learned by the two networks and input them to the FC layer for classification. To determine the optimal binding method, we conducted experiments on the IP, UP and KSC datasets, respectively, and Table 4 shows the corresponding values of these three binding methods in terms of OA, AA and $\kappa$.

From Table 4, we can observe that method III outperforms both method I and method II on OA, AA and $\kappa$. Therefore, in this paper we choose method III as the optimal combination of the two branch networks.

### D. ANALYSIS OF THE EFFECTIVENESS OF THE NUMBER OF OUTPUT FILTERS IN THREE-DIMENSIONAL CONVOLUTION

The number of filters in the 3D convolution process is an important parameter of the proposed method, which is important to increase the model discriminative ability and feature extraction ability. However, the increase of the number of filters in the 3D convolution process generates a large amount of computational time spent. To explore the effect of the number of filters in the convolutional layer on the classification performance, we conducted several experiments with different numbers of output filters on three datasets. In the experiments, the number of output filters was set to 12, 18, 24 and 30, respectively. Fig. 12 shows the OA, AA and $\kappa$ values obtained by the proposed model on the IP, UP and KSC data sets using different numbers of output filters. It can be observed from Fig. 12 that the overall performance of the

**TABLE 4.** Comparison of the three binding methods in Fig. 11.

| Dataset | IP | | | UP | | | KSC | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | Method I | Method II | Method III | Method I | Method II | Method III | Method I | Method II | Method III |
| OA | 97.605 | 98.032 | **98.673** | 98.045 | 99.391 | **99.518** | 98.772 | 99.131 | **99.442** |
| AA | 93.002 | 96.389 | **97.098** | 96.964 | 99.262 | **99.403** | 98.272 | 98.529 | **99.129** |
| $\kappa$ | 0.97274 | 0.97756 | **0.98488** | 0.9710 | 0.99193 | **0.99361** | 0.98632 | 0.99032 | **0.99378** |

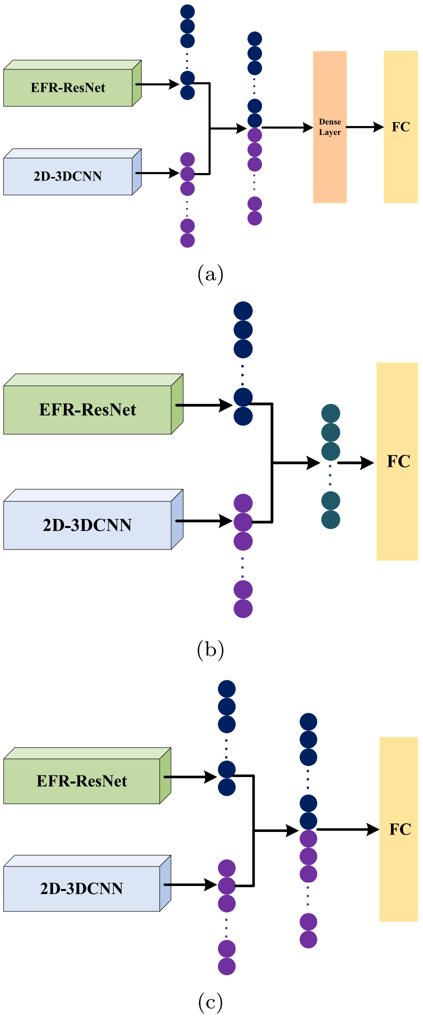


**FIGURE 11.** (a) Branch network combination method I (b) Branch network combination method II (c) Branch network combination method III.

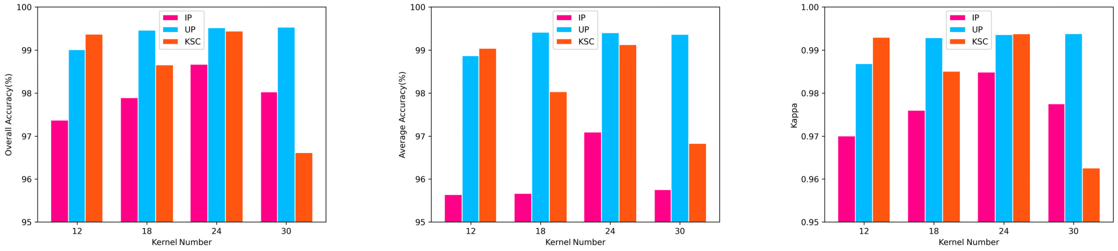
network model reaches the optimum when the number of filters reaches 24, so we choose 24 as the number of filters in the 3D convolution process.

### E. COMPARISON AND ANALYSIS OF CLASSIFICATION RESULTS

To demonstrate the effectiveness of our proposed MHNet method, we selected three extensively used machine learning-based methods and seven advanced deep learning-based methods for comparison. Namely, the machine learning-based methods are RF, SVM and XGBOOST; the deep learning-based methods are 2D-CNN [62], 3D-DL [35], ResNet [25], ContextualCNN [63], M3D-DCNN [37], HSI-CNN [64] and DePyResNet [65]. Overall accuracy (OA), average accuracy (AA) and Kappa coefficient ($\kappa$) were adopted to evaluate the results of the classification experiments. More specifically, OA is the number of correctly classified samples divided by the number of total samples involved in the test, while AA is the classification accuracy of each land cover category divided by the total number of categories. In addition, $\kappa$ is a method used to assess consistency in statistics, especially in multi-classification tasks with unbalanced samples [66]. The range of values of this coefficient is [-1,1]. All experiments were done six times, and the average of OA, AA and $\kappa$ for the six experiments are shown in this paper. Tables 5 to 7 show the classification results on the IP, UP and KSC datasets, respectively.

From Table 5, it can be concluded that the proposed MHNet method clearly outperforms the machine learning-based and deep learning-based methods in terms of OA, AA and $\kappa$ on the IP dataset. Specifically, compared with three popular machine learning-based methods. On OA, AA and $\kappa$, the proposed method MHNet reached 98.673%, 97.098% and 0.98488, respectively. However, the highest values in the machine learning-based methods only reached 90.428%, 85.187% and 0.89061. This shows that our proposed MHNet method outperforms these popular machine learning-based methods by a wide margin. On the other hand, a comparison with advanced deep learning based methods is made. The highest values on OA, AA and $\kappa$ were 96.355%, 94.712% and 0.95849 for the deep learning-based methods, which were implemented by ContextualCNN, DePyResNet and ContextualCNN, respectively. Obviously, it is also our proposed method that prevails.

Also in Table 5, it can be seen that 3D-DL, ResNet, M3D-DCNN and HSI-CNN show excellent classification performance in the first class with 100% classification accuracy. HSI-CNN achieves 100% classification accuracy in the seventh class. The methods that achieve 100% classification accuracy in the eighth, ninth and thirteenth classes are RF, DePyResNet and 3D-DL, respectively. 2D-CNN and 3D-DL achieve 100% classification accuracy in the sixteenth class classification. The proposed MHNet is optimal in the first, second, third, fifth, seventh, tenth, twelfth, fourteenth and fifteenth classes in comparison with other methods in the table. Among them, the classification accuracy reaches 100% in the first, seventh and twelfth categories. It can also be observed in Table 5 that the machine learning-based XGBOOST outperforms some deep learning-based methods. To better express the superior performance of our proposed



(a) OA (b) AA (c) $\kappa$

**FIGURE 12.** 10% of the training samples are randomly selected from IP, UP and KSC, respectively, to study the effect of the number of filters used in the 3D convolution process on OA, AA and $\kappa$.

**TABLE 5.** Comparison of OA,AA and $\kappa$ values of IP dataset.

| class | RF | SVM | XGBOOST | 2D-CNN | 3D-DL | ResNet | ContextualCNN | M3D-DCNN | HSI-CNN | DePyResNet | MHNet |
|---|---|---|---|---|---|---|---|---|---|---|---|
| C1 | 47.826 | 39.130 | 56.522 | 91.892 | **100.000** | **100.000** | 80.987 | **100.000** | **100.000** | 98.039 | **100.000** |
| C2 | 85.154 | 69.608 | 85.784 | 87.402 | 88.287 | 94.499 | 97.433 | 84.168 | 91.330 | 95.205 | **98.401** |
| C3 | 76.747 | 46.627 | 79.518 | 82.530 | 89.278 | 84.503 | 90.556 | 81.118 | 89.084 | 82.409 | **98.036** |
| C4 | 67.933 | 55.696 | 77.215 | 77.778 | 94.778 | **98.878** | 90.420 | 88.020 | 95.670 | 97.255 | 97.409 |
| C5 | 95.652 | 71.429 | 95.652 | 94.574 | 98.422 | 98.407 | 95.364 | 96.335 | 97.944 | 98.818 | **99.490** |
| C6 | 97.945 | 73.973 | 97.945 | **99.486** | 98.983 | 98.553 | 97.945 | 95.018 | 97.135 | 93.641 | 98.997 |
| C7 | 78.571 | 53.571 | 78.571 | 86.957 | 33.333 | 97.222 | 64.011 | 66.667 | **100.000** | 93.651 | **100.000** |
| C8 | **100.000** | 70.502 | 99.372 | 98.953 | 92.566 | 95.615 | 96.466 | 89.755 | 96.470 | 93.612 | 98.977 |
| C9 | 50.000 | 30.000 | 50.000 | 87.500 | 33.333 | 88.889 | 56.170 | 66.667 | 33.333 | 100.000 | 70.002 |
| C10 | 88.580 | 65.844 | 89.815 | 88.546 | 91.116 | 89.397 | 96.108 | 82.400 | 93.259 | 90.029 | **98.716** |
| C11 | 92.872 | 38.778 | 92.505 | 90.071 | 88.254 | 94.957 | **98.974** | 84.635 | 94.551 | 95.445 | 98.067 |
| C12 | 82.293 | 63.238 | 84.148 | 75.106 | 87.570 | 89.871 | 97.333 | 86.623 | 88.285 | 93.760 | **100.000** |
| C13 | 97.073 | 65.854 | 97.561 | 98.781 | **100.000** | 94.569 | 95.078 | 98.716 | 99.177 | 98.404 | 98.773 |
| C14 | 98.182 | 99.051 | 98.103 | 97.925 | 92.500 | 97.791 | 97.669 | 90.974 | 96.423 | 94.598 | **99.902** |
| C15 | 76.684 | 55.440 | 81.347 | 80.906 | 97.958 | 95.644 | 92.905 | 85.369 | 95.378 | 93.539 | **98.137** |
| C16 | 95.699 | 58.065 | 98.925 | **100.000** | **100.000** | 94.464 | 96.870 | 93.413 | 99.020 | 96.984 | 98.667 |
| OA | 89.550 | 62.416 | 90.428 | 89.974 | 91.205 | 93.762 | 96.355 | 86.550 | 93.908 | 93.255 | **98.673** |
| AA | 83.201 | 59.800 | 85.187 | 89.900 | 86.649 | 94.579 | 90.268 | 86.867 | 91.691 | 94.712 | **97.098** |
| $\kappa$ | 0.88037 | 0.57427 | 0.89061 | 0.88555 | 0.89909 | 0.92884 | 0.95849 | 0.84579 | 0.93047 | 0.92305 | **0.98488** |

**TABLE 6.** Comparison of OA,AA and $\kappa$ values on the UP data set.

| class | RF | SVM | XGBOOST | 2D-CNN | 3D-DL | ResNet | ContextualCNN | M3D-DCNN | HSI-CNN | DePyResNet | MHNet |
|---|---|---|---|---|---|---|---|---|---|---|---|
| C1 | 97.768 | 70.834 | 97.828 | 99.112 | 88.613 | 97.985 | **99.549** | 99.067 | 94.451 | 92.147 | 99.465 |
| C2 | 99.550 | **100.000** | 99.276 | 99.952 | 95.633 | 99.310 | 99.786 | 99.819 | 98.972 | 98.802 | 99.911 |
| C3 | 84.374 | 69.986 | 89.233 | 94.938 | 90.424 | 95.616 | 94.526 | 98.811 | 94.595 | 95.686 | **99.570** |
| C4 | 96.279 | 69.713 | 97.781 | 97.674 | 99.280 | **99.750** | 99.478 | 99.284 | 99.208 | 89.397 | 99.635 |
| C5 | 99.628 | 71.524 | 99.480 | **100.000** | 95.472 | 99.969 | 99.814 | 99.752 | 99.938 | 99.688 | 99.969 |
| C6 | 86.558 | 69.755 | 95.427 | 98.931 | 98.681 | 98.944 | 99.224 | **99.653** | 98.504 | 99.543 | 99.387 |
| C7 | 86.391 | 70.451 | 91.880 | 95.113 | 65.738 | 99.434 | 98.532 | **99.748** | 96.829 | 98.523 | 99.625 |
| C8 | 95.247 | 70.369 | 95.166 | **97.624** | 74.483 | 88.721 | 97.026 | 96.540 | 86.714 | 90.148 | 97.501 |
| C9 | 99.894 | 70.222 | **100.000** | 99.736 | 99.911 | 99.519 | 99.024 | 99.825 | 99.460 | 78.210 | 99.562 |
| OA | 95.998 | 72.623 | 97.438 | 98.932 | 90.727 | 97.919 | 99.065 | 99.318 | 96.887 | 94.730 | **99.518** |
| AA | 93.965 | 73.650 | 96.230 | 98.120 | 89.804 | 97.694 | 98.551 | 99.170 | 96.519 | 93.572 | **99.403** |
| $\kappa$ | 0.94653 | 0.47891 | 0.96598 | 0.98587 | 0.87507 | 0.97238 | 0.98761 | 0.99096 | 0.95863 | 0.93082 | **0.99361** |

method, we visualize the classification effect graph of some methods as shown in Fig 13. The classification effect of our proposed MHNet method is significantly superior to other methods as seen from the classification effect graph in Fig 13.

From Table 6, it can be concluded that the proposed MHNet method outperforms all the other methods in the table on the UP dataset. In comparison with the machine learning-based method, the proposed MHNet achieves 99.518%, 99.403% and 0.99361 on OA, AA and $\kappa$. The highest values in the machine learning based method are indeed 97.438%, 96.230% and 0.96598. In addition, a comparison is made with the deep learning-based methods in the table. The

**TABLE 7.** Comparison of OA,AA and $\kappa$ values on the KSC data set.

| class | RF | SVM | XGBOOST | 2D-CNN | 3D-DL | ResNet | ContextualCNN | M3D-DCNN | HSI-CNN | DePyResNet | MHNet |
|---|---|---|---|---|---|---|---|---|---|---|---|
| C1 | 98.160 | 88.831 | 97.372 | 96.716 | 98.236 | 92.338 | 98.877 | 96.085 | 95.789 | 95.530 | **99.835** |
| C2 | 94.650 | 91.358 | 95.473 | 85.128 | 96.414 | 91.579 | 93.984 | 96.306 | 88.130 | 93.066 | **97.917** |
| C3 | 96.094 | 89.063 | 96.875 | 83.415 | 93.167 | 63.616 | 74.462 | 78.963 | 79.568 | 84.879 | **97.373** |
| C4 | 86.508 | 86.508 | 88.095 | 79.105 | 91.484 | 54.549 | 85.369 | 84.693 | 71.736 | 77.647 | **99.844** |
| C5 | 89.441 | 88.199 | 91.926 | 81.395 | 89.770 | 56.642 | 69.357 | 92.148 | 72.698 | 91.035 | **95.938** |
| C6 | 82.096 | 84.716 | 84.716 | 76.630 | 98.569 | 82.574 | 91.380 | 88.555 | 84.451 | 93.788 | **99.460** |
| C7 | 94.286 | 92.381 | 94.286 | 88.095 | 96.410 | 69.852 | 83.077 | 96.784 | 94.915 | **100.000** | 99.603 |
| C8 | 96.752 | 94.432 | 97.448 | 97.674 | 98.893 | 94.412 | 96.961 | 97.992 | 98.962 | 98.851 | **100.000** |
| C9 | 98.654 | 95.385 | 98.846 | **100.000** | 99.294 | 90.760 | 98.694 | 98.057 | 94.521 | 97.327 | 99.692 |
| C10 | 97.525 | **100.000** | 98.020 | 98.452 | 99.896 | 90.719 | 99.896 | 99.896 | **100.000** | 99.896 | **100.000** |
| C11 | 99.284 | 87.351 | 99.284 | **100.000** | **100.000** | 99.293 | **100.000** | **100.000** | **100.000** | 99.312 | 99.425 |
| C12 | 98.211 | 85.487 | 98.012 | 85.360 | 97.589 | 95.502 | 98.500 | 97.528 | 95.219 | 98.153 | **99.585** |
| C13 | 99.784 | 98.382 | 99.892 | 98.922 | **100.000** | 92.073 | **100.000** | **100.000** | 99.565 | **100.000** | **100.000** |
| OA | 96.546 | 91.960 | 96.891 | 93.237 | 97.943 | 86.271 | 94.913 | 95.934 | 93.359 | 94.826 | **99.442** |
| AA | 94.726 | 90.930 | 95.403 | 90.069 | 96.902 | 82.631 | 91.581 | 94.385 | 90.427 | 94.576 | **99.129** |
| $\kappa$ | 0.96152 | 0.91061 | 0.96891 | 0.92468 | 0.97708 | 0.84650 | 0.94338 | 0.95467 | 0.92592 | 0.94232 | **0.99378** |

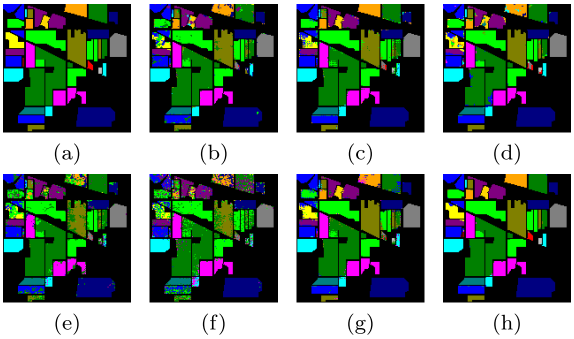


**FIGURE 13.** Classification effect plots on IP dataset (a) ground truth plots (b)-(h) with 3D-DL, ResNet, ContextualCNN, M3D-DCNN, HSI-CNN, DePyResNet and MHNet generated respectively.

highest values in deep learning based methods are 99.318%, 99.170% and 0.99096 for OA, AA and $\kappa$, respectively. They are all achieved by M3D-DCNN. The proposed MHNet is 0.2%, 0.233% and 0.00265 higher than M3D-DCNN on OA, AA and $\kappa$, respectively. All of which fully demonstrate the great advantage of the proposed MHNet in hyperspectral image classification.

It can also be concluded from Table 6 that SVM reached 100% classification accuracy in the second class of the UP dataset, 2D-CNN achieved significant classification performance of 100% in the fifth class of the UP dataset, and XGBOOST achieved 100% classification accuracy in the ninth class. The classification performance of the machine learning-based XGBOOST method on the UP dataset is better than the deep learning-based 3D-DL, HSI-CNN and DePyResNet. The proposed MHNet approach has the best classification performance in the third class contrasted with the other methods in Table 6. In order to more clearly express the superior performance of our proposed approach, we visualize the classification graph as shown in Fig. 14. As can be seen from the visualization in Fig. 14, our MHNet clearly outperforms other deep learning methods.

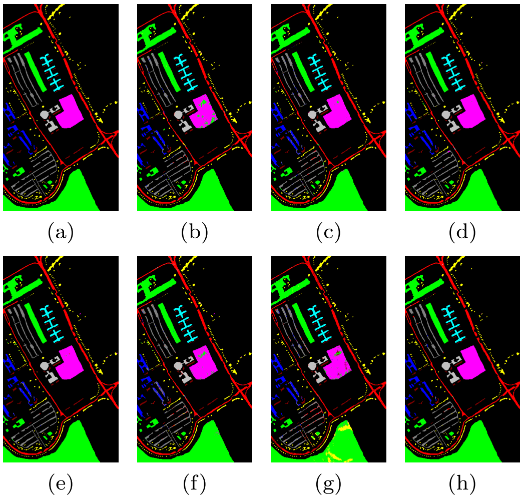


**FIGURE 14.** Classification effect plots on UP dataset (a) ground truth plots (b)-(h) with 3D-DL, ResNet, ContextualCNN, M3D-DCNN, HSI-CNN, DePyResNet and MHNet generated respectively.

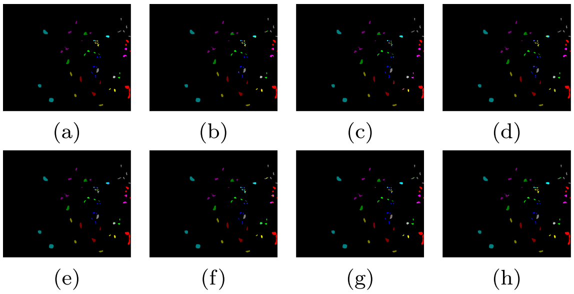
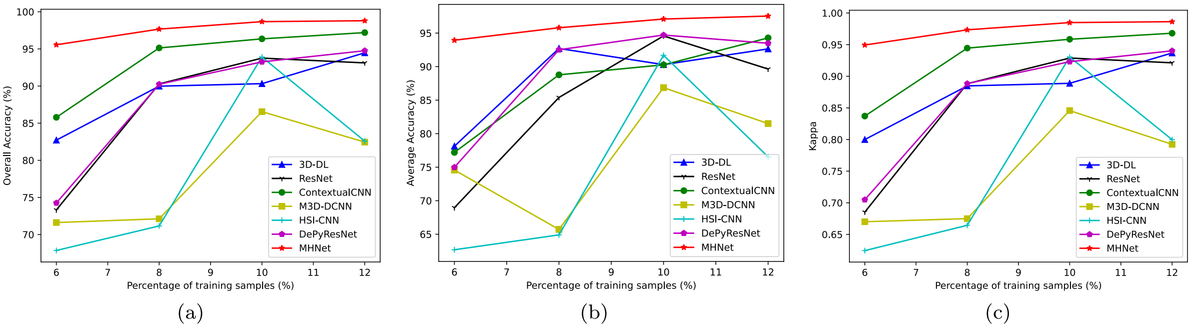


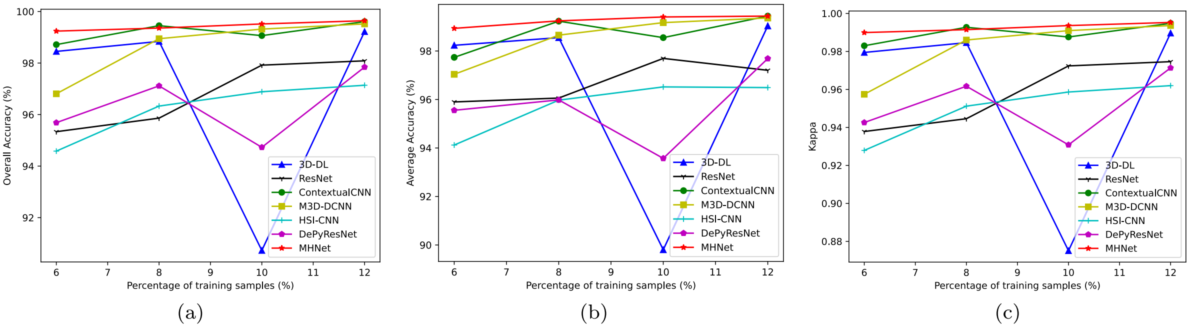
**FIGURE 15.** Classification effect plots on KSC dataset (a) ground truth plots (b)-(h) with 3D-DL, ResNet, ContextualCNN, M3D-DCNN, HSI-CNN, DePyResNet and MHNet generated respectively.

From Table 7, compared with the machine learning based method, our method has values of 99.442%, 99.129% and 0.99378 for OA, AA and $\kappa$. The highest values in the machine learning-based method are 96.891%, 95.403% and



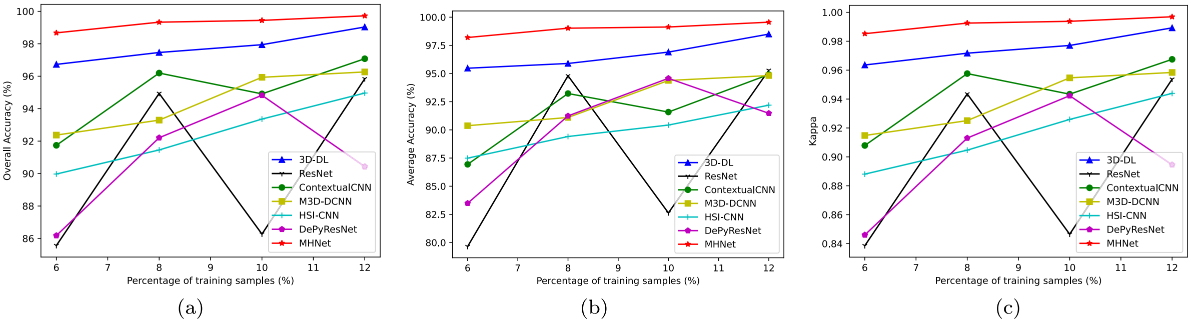
(a) (b) (c)

**FIGURE 16.** **Classification accuracy versus different percentage of training samples on the Indian Pines dataset. (a) OA (%).(b) AA (%). (c) $\kappa$.**



(a) (b) (c)

**FIGURE 17.** **Classification accuracy versus different percentage of training samples on the Pavia University dataset. (a) OA (%).(b) AA (%). (c) $\kappa$.**



(a) (b) (c)

**FIGURE 18.** **Classification accuracy versus different percentage of training samples on the Kennedy Space Center dataset. (a) OA (%). (b) AA (%). (c) $\kappa$.**

0.96891 for these three aspects. Specifically, the proposed MHNet method is 2.551%, 3.726% and 0.02487 higher than the highest values of machine learning based methods, respectively. Among the deep learning based methods, the highest values are 97.943%, 96.902% and 0.97708 for OA, AA and $\kappa$. They are all achieved by 3D-DL. The proposed MHNet is 1.499%, 2.227% and 0.02487 higher than 3D-DL in these three aspects, respectively. In conclusion, our MHNet clearly outperforms all other methods in the table. In particular, the classification performance is significant on the KSC dataset.

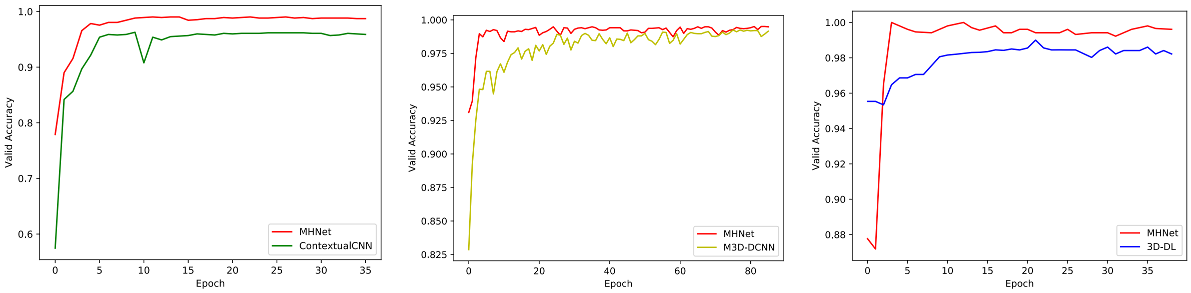
In addition, it can be concluded from Table 7 that DePyResNet in the seventh class and 2D-CNN in the ninth class has superior classification performance with an accuracy of 100%. SVM and HSI-CNN have 100% classification accuracy in the tenth class. The methods that achieve 100% classification accuracy in the eleventh class are 2D-CNN, 3D-DL, ContextualCNN, M3D-DCNN and HSI-CNN. The methods that achieve 100% classification accuracy in the thirteenth class are 3D-DL, ContextualCNN, M3D-DCNN and DePyResNet. The classification accuracy of our proposed MHNet in the first, second, third, fourth, fifth, sixth, eighth, tenth, twelfth and thirteenth classes is optimal compared to the other methods in Table 7. Among them, the classification accuracies in the eighth, tenth and thirteenth classes all reach 100%. To more clearly demonstrate the superior classification performance of the proposed MHNet, we show the classification effect graphs of some of the methods as Fig. 15. As can be seen in Fig. 15, the proposed MHNet clearly outperforms other deep learning methods.

### F. EXPERIMENTS WITH SMALL TRAINING SAMPLES

To verify the robustness and validity of the proposed MHNet method on a small sample training set, we conducted

**TABLE 8. Performance analysis and comparison of different modules in the network.**

| Data Set | Accuracy | SKNet | SKNet+GaborNet | SKNet+Multi-branchNet | GaborNet+SKNet+Multi-branchNet |
|---|---|---|---|---|---|
| Indian Pines | OA(%) | 95.544 | 92.179 | 98.328 | **98.673** |
| | AA(%) | 94.543 | 93.408 | 96.491 | **97.098** |
| | $\kappa$ | 0.94923 | 0.91072 | 0.98095 | **0.98488** |
| Pavia University | OA(%) | 98.672 | 97.257 | 99.404 | **99.518** |
| | AA(%) | 98.482 | 96.199 | 99.253 | **99.403** |
| | $\kappa$ | 0.98236 | 0.96362 | 0.99210 | **0.99361** |
| Kennedy Space Center | OA(%) | 98.501 | 97.959 | 99.259 | **99.442** |
| | AA(%) | 97.421 | 97.057 | 98.934 | **99.129** |
| | $\kappa$ | 0.98331 | 0.97726 | 0.99174 | **0.99378** |



(a) (b) (c)

**FIGURE 19. Validation accuracy of the model during training. (a) IN. (b) UP. (c) KSC.**

comparative experiments on all data sets. Specifically, the training samples are randomly selected as 6%, 8%, 10% and 12% for each dataset. Fig. 16 to Fig. 18 show the performance of each method on different proportions of training datasets on IP, UP and KSC datasets. As seen from the figures, the performance of our proposed MHNet is optimal for various different proportions of training samples compared to other methods. In particular, this advantage becomes significant when the training sample set is reduced. For example, on 6% training samples, the differences in OA values between the proposed method and the second-best method are +9.771%, +0.524% and +1.943% for the IP, UP and KSC datasets, respectively. In conclusion, our proposed MHNet has a huge advantage in the small sample training set.

### G. COMPARISON OF CONVERGENCE

Next, we further verify the convergence of MHNet. Since the classification performance of MHNet has been shown to outperform other models on OA, AA and $\kappa$, in terms of convergence speed, we only compare the proposed MHNet with the second-best classification performance on IP, UP and KSC, respectively. On IP, it can be seen from Table 5 that although ResNet outperforms ContextualCNN on AA, ContextualCNN outperforms ResNet on OA and $\kappa$. Therefore, the second-best model is ContextualCNN. On UP, from Table 6, the second-best model is M3D-DCNN. On KSC, from Table 7, the second best model is 3D-DL. The experimental results are shown in Fig. 19. The convergence of the proposed MHNet is faster than the second-best model on IP, UP and KSC. In conclusion, MHNet has a fast convergence speed and high model efficiency.

### H. ABLATION STUDY

To validate the effectiveness of each module in the proposed MHNet approach, we conducted ablation experiments on the IP, UP and KSC datasets. For a fair comparison, the configuration of each module was kept constant (as described in Section II).

1) SKNet: As the base network module of the proposed network.
2) SKNet+GaborNet: The model generated after removing the Multi-branchNet module from the proposed network framework.
3) SKNet+Multi-branchNet: the model obtained by removing the GaborNet module from the proposed network framework.

The experimental results are shown in Table 8, from which it can be observed that SKNet+Multi-branchNet outperforms SKNet on both IP, UP and KSC datasets in terms of OA, AA and $\kappa$. This indicates that the Multi-branchNet module can effectively and adequately extract the joint spatial-spectral feature information in HSI and improve the classification accuracy. GaborNet+SKNet+Multi-branchNet again outperforms SKNet+Multi-branchNet. It illustrates that the GaborNet layer eliminates most of the variability in images caused by illumination conditions and contrast changes, and enhances robustness against illumination and pose changes. It also helps the network to efficiently extract useful image information in edges and textures, which greatly improves the classification accuracy. In conclusion, the proposed network reaches excellent performance, and all modules in the network are effective.

## IV. CONCLUSION

In this paper, we propose a new end-to-end depth model MHNet for HSI classification. First, the Gabor filter is introduced into this model, which improves the generalization of the model in terms of rotation and scale variation. Then comes the adaptive selective spatial-spectral kernel-based module, in which we introduce an attention mechanism to improve the performance of joint extraction of spatial-spectral features by learning adaptive selective 3D convolution kernels. This is followed by a two-branch network to learn deep joint spatial-spectral features. The spatial-spectral features learned by both networks are concatenated and passed through a fully connected layer in order to capture higher-level spatial-spectral joint features. In our experiment, we contrast the proposed MHNet with three machine learning-based and seven deep learning-based methods and use three HSI datasets to evaluate the performance of the methods. Experiments with fixed and different numbers of training samples were also performed. The experiments demonstrate that the proposed MHNet outperforms the other methods on OA, AA and $\kappa$ and converges faster than the model with the second-best classification performance on all datasets, which proves the effectiveness and superiority of the MHNet method. It is also demonstrated that MHNet is more advantageous on small sample datasets. Finally, an ablation study for different modules was conducted to further validate the rationality of our proposed method.

## REFERENCES

[1] D. Landgrebe, "Hyperspectral image data analysis," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 17–28, Jan. 2002.

[2] 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.

[3] P. Ghamisi, N. Yokoya, J. Li, W. Liao, S. Liu, J. Plaza, B. Rasti, and A. Plaza, "Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 37–78, Dec. 2017.

[4] A. Hollinger, M. Bergeron, M. Maskiewicz, S. Qian, H. Othman, K. Staenz, R. Neville, and D. Goodenough, "Recent developments in the hyperspectral environment and resource observer (HERO) mission," in *Proc. IEEE Int. Symp. Geosci. Remote Sens.*, Jul. 2006, pp. 1620–1623.

[5] M. Teke, H. S. Deveci, O. Haliloglu, S. Z. Gürbüz, and U. Sakarya, "A short survey of hyperspectral remote sensing applications in agriculture," in *Proc. 6th Int. Conf. Recent Adv. Space Technol. (RAST)*, Jun. 2013, pp. 171–176.

[6] N. Yokoya, J. Chan, and K. Segl, "Potential of resolution-enhanced hyperspectral data for mineral mapping using simulated EnMAP and Sentinel-2 images," *Remote Sens.*, vol. 8, no. 3, p. 172, Feb. 2016.

[7] E. Zafra, A.-M. Sánchez, E. Torrecilla, and J. Piera, "Low-cost computationally hyperspectral simulator for highly dynamic marine environments," in *Proc. 6th Workshop Hyperspectral Image Signal Process., Evol. Remote Sens. (WHISPERS)*, 2014, pp. 1–4.

[8] R. D. Hewson, T. J. Cudahy, M. Caccetta, A. Rodger, M. Jones, and C. Ong, "Advances in hyperspectral processing for province- and continental- wide mineral mapping," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2009.

[9] D. Li, X. Hu, S. Wang, C. Zhang, R. Zhou, and H. Zhou, "Hyperspectral images ground object recognition based on split attention," in *Proc. IEEE 2nd Int. Conf. Big Data, Artif. Intell. Internet Things Eng. (ICBAIE)*, Mar. 2021, pp. 324–330.

[10] B. Taskesen, A. Koz, A. Alatan, and O. Weatherbee, "Change detection for hyperspectral images using extended mutual information and oversegmentation," in *Proc. 9th Workshop Hyperspectral Image Signal Process., Evol. Remote Sens. (WHISPERS)*, Sep. 2018, pp. 1–5.

[11] D. Manolakis and G. Shaw, "Detection algorithms for hyperspectral imaging applications," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 29–43, Jan. 2002.

[12] Y. E. Esin, O. Öztürk, S. Öztürk, and Ö. Özdil, "Deep learning based enhancement in hyperspectral object detection," in *Proc. 28th Signal Process. Commun. Appl. Conf. (SIU)*, Oct. 2020, pp. 1–4.

[13] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. A. Benediktsson, "Advances in hyperspectral image classification: Earth monitoring with statistical learning methods," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 45–54, Jan. 2014.

[14] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 480–491, Mar. 2005.

[15] 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.

[16] Y. Qian, M. Ye, and J. Zhou, "Hyperspectral image classification based on structured sparse logistic regression and three-dimensional wavelet texture features," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 4, pp. 2276–2291, Apr. 2013.

[17] L. Zhang, L. Zhang, D. Tao, and X. Huang, "On combining multiple features for hyperspectral remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 879–893, Mar. 2012.

[18] Y. Shao, N. Sang, C. Gao, and L. Ma, "Spatial and class structure regularized sparse representation graph for semi-supervised hyperspectral image classification," *Pattern Recognit.*, vol. 81, pp. 81–94, Sep. 2018.

[19] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 492–501, Mar. 2005.

[20] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[21] Y. Zhong and L. Zhang, "An adaptive artificial immune network for supervised classification of multi-/Hyperspectral remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 894–909, Mar. 2012.

[22] P. R. Marpu, M. Pedergnana, M. D. Mura, J. A. Benediktsson, and L. Bruzzone, "Automatic generation of standard deviation attribute profiles for spectral–spatial classification of remote sensing data," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 2, pp. 293–297, Mar. 2013.

[23] X. Kang, S. Li, and J. A. Benediktsson, "Spectral–spatial hyperspectral image classification with edge-preserving filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2666–2677, May 2014.

[24] X. Guo, X. Huang, L. Zhang, L. Zhang, A. Plaza, and J. A. Benediktsson, "Support tensor machines for classification of hyperspectral remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3248–3264, Jun. 2016.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[26] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[27] V. K. Repala and S. R. Dubey, "Dual CNN models for unsupervised monocular depth estimation," in *Pattern Recognition and Machine Intelligence* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2019, pp. 209–217.

[28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. NIPS*, vol. 28, 2015, pp. 1–9.

[29] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.

[30] C. Tao, H. Pan, Y. Li, and Z. Zou, "Unsupervised spectral–spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2438–2442, Dec. 2015.

[31] B. Kwolek, "Face detection using convolutional neural networks and Gabor filters," in *Artificial Neural Networks: Biological Inspirations (ICANN)* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2005, pp. 551–556.

[32] H. Yao, L. Chuyi, H. Dan, and Y. Weiyu, "Gabor feature based convolutional neural network for object recognition in natural scene," in *Proc. 3rd Int. Conf. Inf. Sci. Control Eng. (ICISCE)*, Jul. 2016, pp. 386–390.

[33] Y. Chen, L. Zhu, P. Ghamisi, X. Jia, G. Li, and L. Tang, "Hyperspectral images classification with Gabor filtering and convolutional neural network," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2355–2359, Dec. 2017.

[34] S. Jia, J. Liao, M. Xu, Y. Li, J. Zhu, W. Sun, X. Jia, and Q. Li, "3-D Gabor convolutional neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5509216.

[35] A. Ben Hamida, A. Benoit, P. Lambert, and C. Ben Amar, "3-D deep learning approach for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4420–4434, Aug. 2018.

[36] W. Song, S. Li, L. Fang, and T. Lu, "Hyperspectral image classification with deep feature fusion network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3173–3184, Jun. 2018.

[37] M. He, B. Li, and H. Chen, "Multi-scale 3D deep convolutional neural network for hyperspectral image classification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3904–3908.

[38] H. Lee and H. Kwon, "Going deeper with contextual CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4843–4855, Oct. 2017.

[39] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. Plaza, J. Li, and F. Pla, "Capsule networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2145–2160, Apr. 2019.

[40] 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.

[41] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sens.*, vol. 7, no. 11, pp. 14680–14707, Nov. 2015.

[42] X. Kang, B. Zhuo, and P. Duan, "Dual-path network-based hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 3, pp. 447–451, Mar. 2019.

[43] M. Han, R. Cong, X. Li, H. Fu, and J. Lei, "Joint spatial–spectral hyperspectral image classification based on convolutional neural network," *Pattern Recognit. Lett.*, vol. 130, pp. 38–45, Feb. 2020.

[44] J. Yang, Y.-Q. Zhao, and J. C. 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.

[45] S. Luan, C. Chen, B. Zhang, J. Han, and J. Liu, "Gabor convolutional networks," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4357–4366, Sep. 2018.

[46] D. Gabor, "Theory of communication. Part 1: The analysis of information," *J. Inst. Elect. Eng. III, Radio Commun. Eng.*, vol. 93, no. 26, pp. 429–441, Nov. 1946.

[47] A. K. Jain, N. K. Ratha, and S. Lakshmanan, "Object detection using Gabor filters," *Pattern Recognit.*, vol. 30, no. 2, pp. 295–309, Feb. 1997.

[48] T. Sing Lee, "Image representation using 2D Gabor wavelets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 10, pp. 959–971, Oct. 1996.

[49] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition," *IEEE Trans. Image Process.*, vol. 11, no. 4, pp. 467–476, Apr. 2002.

[50] H.-B. Deng, L.-W. Jin, L.-X. Zhen, and J.-C. Huang, "A new facial expression recognition method based on local Gabor filter bank and PCA plus LDA," *Int. J. Inf. Technol.*, vol. 11, no. 11, pp. 86–96, 2005.

[51] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 510–519.

[52] T. Alipour-Fard, M. E. Paoletti, J. M. Haut, H. Arefi, J. Plaza, and A. Plaza, "Multibranch selective kernel networks for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 6, pp. 1089–1093, Jun. 2021.

[53] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Proc. NIPS*, vol. 29, 2016, pp. 1–9.

[54] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 448–456.

[55] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.

[56] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020.

[57] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11531–11539.

[58] S. K. Roy, S. Manna, T. Song, and L. Bruzzone, "Attention-based adaptive spectral–spatial kernel ResNet for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7831–7843, Sep. 2021.

[59] C. Liu and H. Wechsler, "Independent component analysis of Gabor features for face recognition," *IEEE Trans. Neural Netw.*, vol. 14, no. 4, pp. 919–928, Jul. 2003.

[60] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[61] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1—Learning rate, batch size, momentum, and weight decay," 2018, *arXiv:1803.09820*.

[62] S. Liu, R. S. Chu, X. Wang, and W. Luk, "Optimizing CNN-based hyperspectral image classification on FPGAs," in *Applied Reconfigurable Computing* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2019, pp. 17–31.

[63] H. Lee and H. Kwon, "Contextual deep CNN based hyperspectral classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 3322–3325.

[64] Y. Luo, J. Zou, C. Yao, X. Zhao, T. Li, and G. Bai, "HSI-CNN: A novel convolution neural network for hyperspectral image," in *Proc. Int. Conf. Audio, Lang. Image Process. (ICALIP)*, 2018, pp. 464–469.

[65] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. J. Plaza, and F. Pla, "Deep pyramidal residual networks for spectral–spatial hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 740–754, Feb. 2019.

[66] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, Apr. 1960.

**CAILING WANG** received the B.S. degree from Tianjin University, Tianjin, China, in 2006, and the Ph.D. degree in signals and information processing from the Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, Shaanxi, China, in 2011. She is currently an Associate Professor with the College of Computer Science, Xi'an Shiyou University. Her major research interests include remote sensing image processing and artificial intelligence.

**HE FU** is currently pursuing the master's degree in computer science and technology with Xi'an Shiyou University, Xi'an, China. His research interests include remote sensing image processes and hyperspectral image classification.

**HONGWEI WANG** received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 2006, and the Ph.D. degree in optical engineering from the Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, Shaanxi, China, in 2011. He is currently an Associate Professor with the School of Artificial Intelligence, Optics and Electronics, Northwestern Polytechnical University. His major research interests include image processing and data analysis.