3-D Channel and Spatial Attention-Based Multiscale Spatial–Spectral Residual Network for Hyperspectral Image Classification

Zhenyu Lu, *Member, IEEE*, Bin Xu, Le Sun, *Member, IEEE*, Tianming Zhan, and Songze Tang

***Abstract*—With the rapid development of aerospace and various remote sensing platforms, the amount of data related to remote sensing is increasing rapidly. To meet the application requirements of remote sensing big data, an increasing number of scholars are combining deep learning with remote sensing data. In recent years, based on the rapid development of deep learning methods, research in the field of hyperspectral image (HSI) classification has seen continuous breakthroughs. In order to fully extract the characteristics of HSIs and improve the accuracy of image classification, this article proposes a novel 3-D channel and spatial attention-based multiscale spatial–spectral residual network (termed CSMS-SSRN). The CSMS-SSRN framework uses a three-layer parallel residual network structure by using different 3-D convolutional kernels to continuously learn spectral and spatial features from their respective residual blocks. Then, the extracted depth multiscale features are stacked and input into the 3-D attention module to enhance the expressiveness of the image features from the two aspects of channel and spatial domains, thereby improving the accuracy of classification. The CSMS-SSRN framework proposed in this article can achieve better classification performance on different HSI datasets.**

***Index Terms*—Attention, deep learning, hyperspectral image, multiscale spatial–spectral residual network.**

I. Introduction

With the progress of remote sensing technology, the types of remote sensing observation data have become enriched, and the amount of data has increased rapidly [1], [2]. Remote sensing information technology is entering the era of big data, which is characterized by intelligent analysis [3], [4]. An increasing number of scholars have combined deep learning and remote sensing technology to achieve tremendous advancement in target recognition [5], image segmentation [6], and parameter inversion [7]. Hyperspectral remote sensing images are obtained by satellites equipped with various spectral sensors [8]. These satellites can obtain abundant information about the surface of the earth, which provides a new way for people to observe the landform and understand the world. Unlike the ordinary Red-Green-Blue image, the hyperspectral image (HSI) has hundreds of bands, which contain rich spectral information [9]–[11]. The spectral information contained in the HSI helps to distinguish land cover, while the fine spatial resolution provides rich information about spatial structure [12]. Therefore, HSIs have played an important role in geological disaster monitoring, mineral analysis, agricultural testing, and marine environment surveying [13]–[15]. One important research direction in remote sensing science is to classify each pixel in an HSI and divide the geomorphic information [16]–[18].

In recent years, artificial intelligence and other technologies have developed rapidly [19]–[22], and some HSI classification methods based on traditional machine learning have been improved and have achieved good results [23]–[25]. Licciardi *et al.* [26] used principal component analysis (PCA) to combine spectral and spatial information to achieve classification of HSIs. The k-neighbor classification (k-NN) method [27] completes HSI classification by computing the distance between the test data and the training data and judging the similarity between the samples. To prevent the influence of nonlinear features on a k-NN classifier, Chen *et al.* [28] studied a combined manifold learning method and k-NN classifier to improve classification accuracy by maintaining nonlinear features in an HSI. Li *et al.* [29] used rich texture information to classify HSIs and used the local binary mode method to extract the local information in an image. This method works well for analyzing macrotextures, but often cannot handle microtextures. Therefore, Deng *et al.* [30] proposed a model for classification using the microtexture information in HSIs. The model uses a local response mode to retain more structural information and maintain a lower sensitivity to the image conditions. The results show that the proposed framework is effective in terms of both recognition rate and robustness. Camps Valls *et al.* [31] combined the support vector

Manuscript received May 12, 2020; revised July 6, 2020 and July 20, 2020; accepted July 21, 2020. Date of publication July 27, 2020; date of current version August 11, 2020. This work was supported in part by the Natural Science Foundation of China under Grant 61773220, Grant 61971233, Grant 61976117, Grant 61972206, and Grant 61702269, in part by the Key Projects of University Natural Science Fund of Jiangsu Province, China, under Grant 19KJA360001, and in part by the National Science Foundation of Jiangsu Province under Grant BK20191409. *(Corresponding author: Le Sun.)*

Zhenyu Lu is with the School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: luzhenyu@163.com).

Bin Xu is with the School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: binxu@nuist.edu.cn).

Le Sun is with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China, and also with the Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: sunlecncom@nuist.edu.cn).

Tianming Zhan is with the School of Information Engineering, Nanjing Audit University, Nanjing 211815, China (e-mail: ztm@nau.edu.cn).

Songze Tang is with the College of Nanjing Forest Police, Nanjing Forest Police College, Nanjing 210023, China (e-mail: tangsz@nfpc.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2020.3011992

machine (SVM) method to construct a set of composite kernel functions that learn rich spatial and spectral feature information from HSIs to improve classification performance. Chen *et al.* [32] proposed an HSI classification algorithm based on sparseness. This algorithm optimized the sparse constraint problem and improved classification performance by incorporating with context information.

The traditional machine learning method can only extract the shallow features of HSIs. Not all the information contained in HSIs has been fully tapped, and this presents the problems of complicated calculation and low classification performance. With the continuous improvement in computing hardware technology, deep learning [33]–[36] algorithms can extract deep features with strong representation capabilities from images in a layered manner [37]. Therefore, deep learning is widely used to learn the deep features of images and improve their classification accuracy [38]. Many studies related to HSI classification have determined that the extraction of depth features is beneficial to the accuracy of image classification. Chen *et al.* [39] introduced a new framework that combines PCA and logistic regression in a deep learning model. In this framework, spatial information is the main axis of classification, and a stack-type automatic encoder is used to obtain the deep features of the image combined with spectral information. Deep neural network research in spectral space shows that methods based on deep learning have great potential for hyperspectral data classification [40], [41]. Makantasis *et al.* [42] proposed a supervised deep learning method, which reduced the dimension of HSIs by using random PCA, then encoded the spectral and spatial feature information of pixels by using a convolutional neural network (CNN), and finally classified them by a multilayer perceptron (MLP). Zhao *et al.* [43] combined a local discrimination embedding algorithm with a CNN; the former was used to extract spectral information from the image, and the latter was used to extract spatial information continuously. Finally, the obtained spectral features and spatial features were superimposed to obtain a new fusion feature. Finally, the fusion features were trained by the classifier. Due to the problems of limited training samples and unbalanced classes in the HSIs, Chen *et al.* [44] combined virtual sample enhancement technology with a CNN to effectively extract the spectral and spatial information. The use of L2 regularization and dropout can also alleviate overfitting problems that may occur during training. In order to effectively overcome the high-dimensionality problem of hyperspectral data, Alipourfard *et al.* [45] proposed a feature learning method based on subspace to reduce the dimension, and combined with CNN to extract image features. Yue *et al.* [46] combined a deep convolutional neural network (DCNN) and logical regression to classify by generating spectral and spatial feature maps. Zhang *et al.* [47] learned spatial–spectral context-sensitivity using CNN networks in different regions and enhanced the recognition ability of the network by combining various distinguishable appearance factors. Mei *et al.* [48] combined contextual information and spectral information and proposed a five-layer classification neural network for HSI classification. Xu *et al.* [49] combined CNN and a long-term short-term memory model based on band grouping and proposed a spectral-spatial unified network (SSUN) model. Zhong *et al.* [50] designed a supervised residual network consisting of spatial residual blocks and spectral residual blocks to jointly learn spatial and spectral information in HSIs. Because the training time of the spatial–spectral residual network (SSRN) was too long, Wang *et al.* [51] designed a fast dense spectral-spatial convolution network (FDSSC) that was faster than SSRN networks. To learn more representative features from a limited number of training samples, Haut *et al.* [52] combined visual attention with a DCNN and proposed an attention-driven mask mechanism (A-ResNet), which was used to filter the features obtained by the network and improve the results of the model. Since SSRN and FDSSC networks need a lot of training samples to obtain good classification results, it is necessary to learn more representative features from the limited training samples. Multiscale strategy [53]–[55] is one of the effective ways to improve the classification accuracy of HSI. Wu *et al.* [56] proposed a multiscale spectral–spatial joint network to classify HSIs by jointly extracting multiscale spectral–spatial features. Pooja *et al.* [57] combined multiscale strategy with CNN network to achieve effective HSI classification. In order to reduce the interference of adjacent pixels and improve the expressiveness of features, Sun *et al.* [58] proposed a special spatial-attention network (SSAN). Later, a simple spectral–spatial network is combined with the attention mechanism to extract spatial and spectral features of images.

Although the above methods have tremendous advantages over traditional machine learning in extracting hyperspectral spatial–spectral features, obtaining spatial and spectral information using neural networks is still a considerable challenge [59]–[63]. First, because the HSI datasets have been extensively studied, the number of available training samples and test samples is relatively small. Furthermore, the imbalance of differently labeled samples also reduces the accuracy of HSI classification. Therefore, it is worth examining how to use the limited number of samples to obtain more sample features. Second, because an HSI is collected, it inevitably contains a lot of redundant information, which also greatly reduces the classification accuracy.

In order to effectively deal with the problems mentioned above, this article proposes a novel 3-D channel and spatial attention-based multiscale SSRN (termed CSMS-SSRN). This network was based on a residual network, using convolutional kernels of different sizes to extract spatial and spectral feature information from HSIs, so that the network could learn more features under the condition of a limited number of samples. Second, we introduce an attention mechanism, which enhances the representation ability of specific region features and learns more representative features in the face of a large amount of redundant information.

The three main contributions in this article are as follows.

1) The proposed CSMS-SSRN network uses residual connections and learns the image features' spectral dimension and spatial dimension through a residual block structure. At the same time, we use a parameter correction linear unit

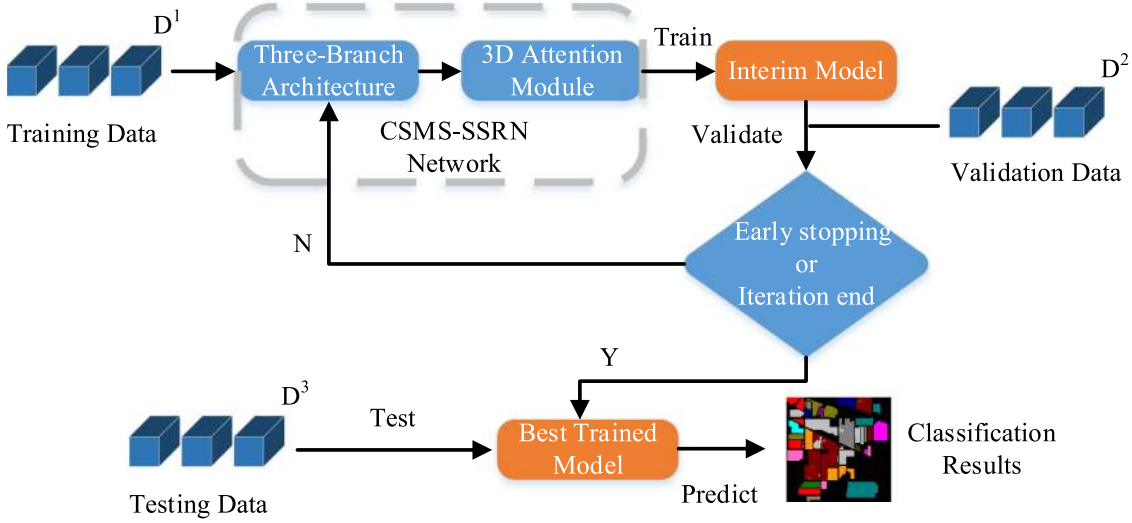


Fig. 1. Schematic diagram of HSI classification of CSMS-SSRN model. D^1, D^2, D^3 are used for training, validation, and testing, respectively.

(ReLU), batch normalization (BN), and a dropout layer to standardize the learning process.

2) Due to the limitations of the HSI single-scale convolutional kernel receptive field, a multiscale residual network model is proposed by using three parallel residual block structures. The use of convolutional kernels of different sizes gives the network the ability to extract features from different fields. The extracted information is more comprehensive and includes both global and local information. Finally, the detailed information is fused into a deep multiscale feature map.
3) To improve the capability of network multiscale feature representation, a simple and effective 3-D attention module is added to the existing model. This method can help the network to learn more representative features. At the same time, the attention mechanism, which is a plug and play module, can be integrated into existing network architectures.

The rest of this article is structured as follows: Section II introduces the proposed CSMS-SSRN framework. Section III introduces the three commonly known HSI datasets used in this study, lists the results of HSI classification, and compares them with other classification methods. Section IV presents our conclusions about the CSMS-SSRN.

II. Proposed Framework

In this section, the CSMS-SSRN framework will be explained in detail, including how the spectral and spatial features are separated from the HSI, and how the deeper spectral and spatial mixed features are extracted from the parallel structure of the three-layer independent residual network, and the use of 3-D attention modules to improve the representation of multiscale features in the network. In this process, in order to explain the CSMS-SSRN network, we use formulas and schematic diagrams to describe the details and steps of the framework and illustrate the advantages of the network.

Before classification, the HSI data and labels need to be divided. Suppose that the HSI dataset X contains N labeled pixels $\{x_1, x_2, \ldots, x_N\} \in R^{1\times1\times b}$, and the corresponding one-hot label vectors are Y with $\{y_1, y_2, \ldots, y_N\} \in R^{1\times1\times T}$, where b represents the number of bands and T represents the number of land-cover categories. For cross validation, the network takes the cube of $w \times w \times L$ in the original HSI data as input. Fig. 1 shows that the input data cube is divided into training data D^1, validation data D^2, and testing data, D^3. At the same time, Y_1, Y_2, and Y_3 are the corresponding label vector data for D^1, D^2, and D^3.

After the hyperparameters for the training are set, the model utilizes training data D^1 and validation data D^2 to train the CSMS-SSRN network. Due to early stopping and the dynamic learning rate, the best CSMS-SSRN model can be obtained. During the training process, the network updates the parameters of the CSMS-SSRN model through the gradient of the cross-entropy objective function in (1). The objective function is used to measure the difference between the predicted label $y' = \{y'_1, y'_2 \ldots, y'_T\}$ and the real label $y = \{y_1, y_2, \ldots, y_T\}$ of the model. After that, using validation data D^2 to monitor the training of the model, the evaluation of the classification performance of the interim model is completed. The CSMS-SSRN achieves the optimal training model using cross validation and testing data D^3 to obtain classification accuracy and classify the testing data

$$C\left(y', y\right) = \sum_{i=1}^{L} y_i \left(\log \sum_{j=1}^{L} e^{y'_j} - \widehat{y}_i \right). \tag{1}$$

In this study, the input data of the CSMS-SSRN is the original 3-D cube. Due to the challenge of multidimensional input data, the 3-D convolutional layer is usually used to extract spectral and spatial features. At the same time, each convolutional layer in the CSMS-SSRN contains a BN layer. This strategy means that the gradient converges faster, making the training process of deep learning models more efficient. Because our framework contains

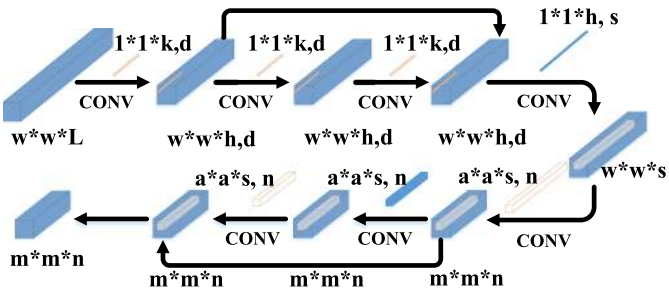


Fig. 2. Residual block structure. The network consists of a spectral residual block and a spatial residual block.

a large number of training parameters, in order to prevent the model from overfitting during the training process and to reduce the calculation cost, this article uses ReLU as the activation function. The formula for ReLU is as follows:

$$f(x) = \max(0, x). \tag{2}$$

A. Residual Block Structure

As shown in Fig. 2, the designed residual block structure is composed of two parts: the spatial feature block and the spectral feature block. These two parts learn deeper spectral and spatial features from the HSI, respectively. Compared with the CNN, the designed residual block reduced the decrease in precision by adding continuous residual blocks and skipping the connection between each other layer. We extracted a data cube with the size of $w \times w \times L$ from the original image pixels as the input data for the model, where $w \times w$ is the size of the input block in the spatial domain, and L represents the number of spectral dimensions. The spectral feature block only uses two 3-D convolutional layers and one spectral residual block. Because the convolution kernel, $1 \times 1 \times k$, did not consider the correlation between the pixels in the spatial field, the advantage is that while the spectral features are extracted, the spatial features are retained perfectly. Therefore, in the first convolutional layer, we applied d kernels of size $1 \times 1 \times k$ to extract the shallow spectral features of the data. Due to the phenomenon of information redundancy between channels in the image spectral dimension, this layer not only achieves dimensionality reduction for the input cube but also extracts the low-order spectral features of the HSI. The spectral residual block includes two convolutional layers and one identity mapping. In each convolutional layer, $1 \times 1 \times k$ convolutional kernels are used to learn deep spectral features. The last convolutional layer of the spectral feature block uses s $1 \times 1 \times h$ convolutional kernels to maintain discriminative spectral features. The information extracted from the spectral feature block is used as the input feature of the next convolutional layer. In order to extract spatial features individually, the dimensions of the convolutional kernel used here are consistent with the dimensions of the input features. The spatial feature block is composed of a 3-D convolutional layer and a spatial residual block. First, n $a \times a \times s$ convolutional kernels are applied to learn the spatial features of the shallow layer, and the spatial size of the feature block is reduced. Finally, the spatial residual block contains two 3-D convolutional layers, which are used to learn deep spatial features with n spatial convolutional kernels of size $a \times a \times s$. It is worth noting that in each spectral and spatial residual block, the size of the 3-D feature block copies the value of the boundary area to the filling area using a padding strategy, so as to keep the output size and input size consistent.

B. Three-Branch Architecture

It can be seen from the above introduction that the residual block structure mainly uses the convolutional kernel $1 \times 1 \times k$ and the 3-D convolutional layer to reduce the spectral dimensions and extract the spectral characteristics of the image. Then, the obtained features and information are used as input, and a convolutional kernel of size $a \times a \times s$ is applied to extract the spatial features of the hyperspectral data in the network.

However, from the perspective of selecting the size of the convolutional kernel, too large a convolutional kernel size will lead to too complex image features extraction, while too small a convolutional kernel size will represent very few useful features.

Therefore, the selection of convolutional kernel size has an important impact on the effects of feature extraction. The features extracted by a large-scale convolutional kernel have strong correlation and can bring large receptive field. The small-scale convolutional kernel can bring small receptive field, and the extracted features have greater detail. Therefore, in order to solve the limitation of a single-scale convolutional kernel receptive field, we introduce multiscale filter banks and build a multiscale SSRN model.

In this part, a three-layer independent and parallel residual block structure is adopted, and different convolutional kernels allow the network to obtain different receiving fields. The extracted information is more comprehensive and includes both global information and local detailed information, which is finally merged into a deep multiscale feature.

Taking the Indian pines data sample with an input size of $9 \times 9 \times 200$ as an example, because the original input data contain abundant and redundant spectral information, in the three-branch structure, we used three convolutional kernels of different sizes, i.e., $1 \times 1 \times 5$, $1 \times 1 \times 7$, and $1 \times 1 \times 9$, each of which has 32 convolutional kernels, to continuously extract the relevant information between local channels, and also to achieve dimension reduction. In the last convolutional layer of the learning part, 64 $1 \times 1 \times 96$, 64 $1 \times 1 \times 97$, and 64 $1 \times 1 \times 98$ spectral convolution kernels convoluted 32 9×9 feature tensors, respectively, and generated 9×9 feature volumes as the input of the spatial feature learning part. After passing through the spectral feature block, the $9 \times 9 \times 64$ size of the feature block was obtained and used as the input for the next layer.

For the features of spatial dimension, we used three convolutional kernels of different sizes, i.e., $3 \times 3 \times 64$, $5 \times 5 \times 64$, and $7 \times 7 \times 64$, each of which has 32 convolutional kernels, to continuously extract the features of the HSI spatial dimension through 3-D convolution. In the three-layer independent and parallel residual structure, the use of filling strategy ensures that the size of the output cube is the same as that of the input cube in the process of convolution. Because HSI classification is a

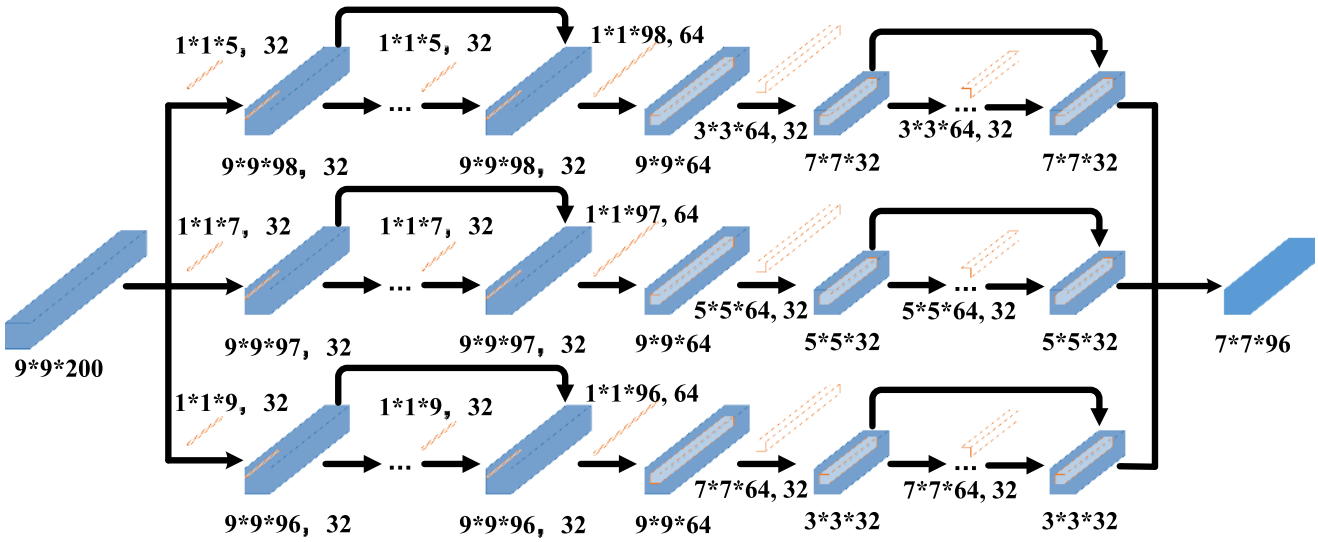


Fig. 3. Three-branch architecture. The network includes three residual block structures. $1 \times 1 \times 5$, $1 \times 1 \times 7$, and $1 \times 1 \times 9$, are three different sizes of convolution kernels are used to extract spectral features. $3 \times 3 \times 64$, $5 \times 5 \times 64$, and $7 \times 7 \times 64$ are three different sizes of convolution kernels are used to extract spatial features.

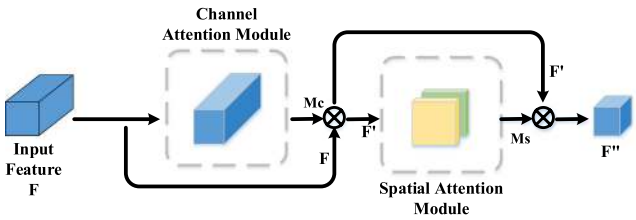


Fig. 4. 3-D attention module. ⊗ denotes elementwise multiplication.

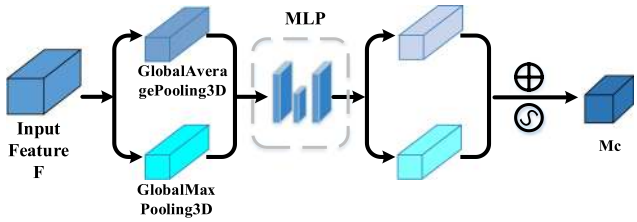


Fig. 5. Channel attention module. MLP is the multilayer perceptron. ⊗ denotes elementwise addition.

multiclass labeling problem, we adopted a softmax function

$$L_s = -\sum_{i=1}^{n} \log \frac{e^{w_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{m} e^{w_j^T x_i + b_j}}. \tag{3}$$

Three feature maps were obtained by the three-branch architecture, and the influence of each feature map on classification was adjusted reasonably through parameter updating. In order to maintain the same spatial size of the three kinds of feature maps, we up-sampled the feature maps of different scales to the same size, then performed a concatenate operation, and finally, the feature cubes of size $7 \times 7 \times 96$ were output. In the convolution process, several different convolutional kernels were used to learn the features of different scales, and then these different features were combined to obtain more abundant features than could be obtained using a single convolutional kernel. The network structure is shown in Fig. 3.

C. 3-D Attention Module

The 3-D attention module is shown in Fig. 4. It consists of a channel attention module and a spatial attention module. It is used in the forward CNN to provide attention feature maps from the channel and spatial dimensions. This module is a general-purpose network module that can be applied to any network structure and has good expressiveness in the classification of images. This module can effectively express the region of interest and help the network pay attention to important features, so the attention mechanism can improve the performance of the network. In addition, by understanding the information to be emphasized or suppressed, the attention mechanism is also conducive to information flow within the network. The main formulas are as follows:

$$F' = M_c(F) \otimes F \tag{4}$$

$$F'' = M_s(F') \otimes F'. \tag{5}$$

Assuming that an intermediate feature map $F \in M^{H \times W \times C}$ is given as input, the 3-D attention module, in turn, derives a 1-D channel attention feature map $M_c \in M^{1 \times 1 \times C}$ and a 2-D space attention feature map $M_s \in M^{H \times W \times 1}$. $\otimes$ represents elementwise multiplication.

As shown in Fig. 5, the channel attention module uses channel relationships between features to generate a channel attention graph. Each channel of the feature represents a special image feature detector. In order to effectively calculate the attention of the channel, we compressed the spatial dimension of the input feature map. In order to obtain spatial information, we used

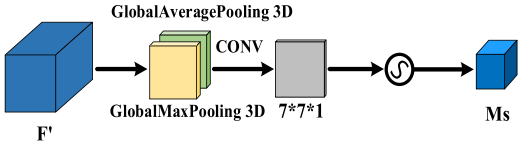


Fig. 6. Spatial attention module.

3-D global average pooling and 3-D global max pooling, which greatly enhance the representation ability of the network. The detailed operation is described below.

In order to extract spatial information, first, the 3-D global average pooling and 3-D global max pooling were used to compress feature map in the spatial dimension, and two $M^{1\times1\times C}$ channel feature maps are obtained. The advantage of this is that the 3-D global average pooling effectively learns the target object, while the 3-D global max pooling collects another important clue about the unique object features to infer the attention of the channel. After obtaining two $M^{1\times1\times C}$ channel descriptions, two $M^{1\times1\times C}$ channel descriptions were put into a neural network composed of a hidden layer and an MLP. To decrease the parameter overhead, the hidden activation size was set to $M^{C/r\times1\times1}$, where r is the compression ratio. It is worth noting that the features between W_0 and W_1 in the MLP model had to be processed using a sigmoid function as the activation function, and then the results of 3-D global average pooling and 3-D global maximum pooling were added and processed using the sigmoid function. Finally, the result was multiplied by the original input feature F to obtain the new scaled feature M_c. Multiplication here is equivalent to applying different weights to each channel.

Where σ represents the sigmoid function, the algorithm formula is as follows:

$$\begin{aligned} M_c(F) &= \sigma\left(\mathrm{MLP}\left(\mathrm{GAP}\left(F\right)\right)+\mathrm{MLP}\left(\mathrm{GMP}\left(F\right)\right)\right) \\ &= \sigma\left(W_1\left(W_0\left(F^c_{\mathrm{GAP}}\right)\right)+W_1\left(W_0\left(F^c_{\mathrm{GMP}}\right)\right)\right). \end{aligned} \quad (6)$$

As shown in Fig. 6, after the channel attention module, the feature map F' is introduced to the spatial attention module. The algorithm formula is as follows:

$$\begin{aligned} M_s\left(F\right) &= \sigma\left(f^{7\times7\times1}\left(\left[\mathrm{GAP}\left(F\right);\mathrm{GMP}\left(F\right)\right]\right)\right) \\ &= \sigma\left(f^{7\times7\times1}\left(\left[F^s_{\mathrm{GAP}};F^s_{\mathrm{GMP}}\right]\right)\right). \end{aligned} \quad (7)$$

First, the GlobalAveragePooling3D and GlobalMaxPooling3D operations are performed on the input features in the channel dimension. Each channel pooling is equivalent to compressing the channel to one dimension, and finally, two 2-D $M^{H\times W\times1}$ feature maps are obtained. Then, two feature maps $M^{H\times W\times1}$ were spliced together according to the channel dimensions to get a feature map $M^{H\times W\times2}$, and then a hidden layer, including a single 3-D convolutional kernel of size $7\times7\times1$, is used to convolute the feature map $M^{H\times W\times2}$ to ensure that the final feature is consistent with the input feature map in the spatial dimension. The final result F'' is obtained by multiplying the result M_s by F' through the channel attention module. The multiplication here is equivalent to giving different weights to the pixels at different positions in each space.

Algorithm 1: CSMS-SSRN Framework for HSI Classification.

Input: An HSI with ground-truth.
Step 1: Extract patches separately with the available pixels as the center. The size of the patch is $9\times9\times L$, where L represents the number of spectral dimensions.
Step 2: The labeled data are randomly assigned to training data D_1, validation data D_2, and testing data D_3.
Step 3: The value of the convolution kernel in the network is initialized to a Gaussian distribution with a mean of 0 and a standard deviation of 0.1. The bias values of the convolution kernels are initialized to 0.
Step 4: Input the training data D_1, validation data D_2, and the corresponding label data Y_1, Y_2 to the network.
Step 5: Perform iterative optimization with a gradient descent algorithm 200 times to obtain the best model. If the condition for early stopping is satisfied during the training process, the training will be stopped in advance.
Step 6: Put the testing data D_3 into the trained model to predict the classification results and generate the confusion matrix.
Output:
1) Classification maps of the HSI.
2) OA, AA, Kappa

D. General Framework

As shown in Fig. 7, the general framework is composed of a three-branch parallel architecture part, 3-D attention module part, dropout layer, pooling layer, and fully connected layer. Taking a data cube with an input size of $9\times9\times$ L as an example, in the three independent residual block structure part, we used 32 convolutional kernels of size $1\times1\times5$, $1\times1\times7$, and $1\times1\times9$ to extract the spectral information of the original image, and we then used 32 convolutional kernels of different sizes, $3\times3\times64$, $5\times5\times64$, and $7\times7\times64$ to extract the spatial information of the feature cube, and get three $7\times7\times32$ cubes. At the same time, in all the spectral and spatial residual blocks, the size of the 3-D feature block copied the value of the boundary area to the filled area through the padding strategy, thereby keeping the output size and input size consistent. In order to reduce the training time and prevent overfitting during network training, we used an ReLU function for parameter correction, dynamic learning rates, and other technical improvements. In the 3-D attention module, we use GlobalAveragePooling3D and GlobalMaxPooling3D, from the channel and spatial dimensions, by adding weights to different positions, using the attention mechanism to improve the performance of the network and help the network to pay attention to important features and restrain dispensable features. After the $7\times7\times96$ cube passes through the 3-D attention module, the output is also $7\times7\times96$, and after the 3-D average pooling, it is stretched into a 1-D vector with a size of $1\times1\times96$. After passing through the dropout

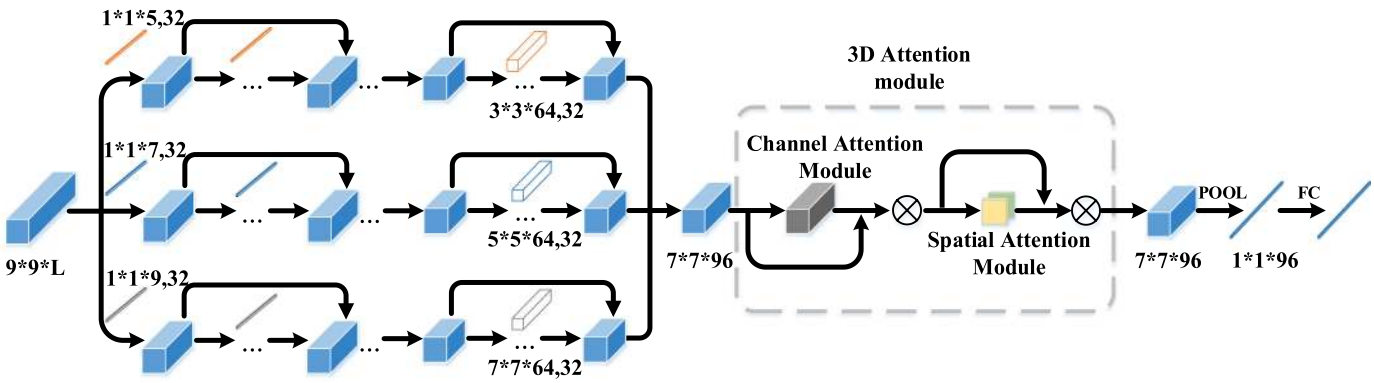


Fig. 7. Three branch architecture. The network includes three residual block structure. $1 \times 1 \times 5$, $1 \times 1 \times 7$, and $1 \times 1 \times 9$ are the three kinds of convolution kernels used to extract spectral features. $3 \times 3 \times 64$, $5 \times 5 \times 64$, and $7 \times 7 \times 64$ are the three kinds of convolution kernels used to extract spatial features.

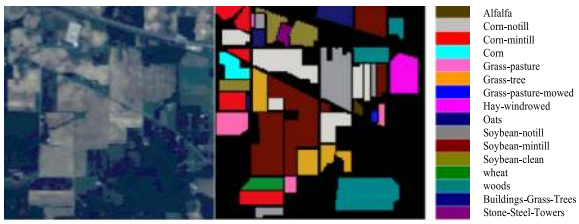


Fig. 8. IN dataset. (a) Pseudo color image (b) Ground truth. (c) Map color.

layer, we use the fully connected layer and softmax classifier to classify HSI.

III. Results and Discussion

In this section, we selected three common HSI datasets, including Indian Pine (IN), Pavia University (UP), and Kennedy Space Center (KSC). To quantifiably assess the classification performance of the proposed CSMS-SSRN in this article, three classification indicators such as total accuracy (OA), average accuracy (AA), and kappa coefficient (κ) are employed. At the same time, we also compared the classification accuracy of the CSMS-SSRN on the three datasets IN, UP, and KSC with other methods to show the advantages of the proposed CSMS-SSRN model for HSI classification. In these three cases, we conducted experiments using randomly selected data samples from the dataset and reported the OA, AA, and Kappa and classification results of each class.

A. Experimental Datasets

The IN dataset was imaged by the Airborne Visible Infrared imaging spectrometer sensor AVIRIS in 1992 on a field of Indian pines in Indiana, USA. The image contains 145×145 pixels and 224 spectral channels in the 0.4–2.45 μm region of the visible and infrared spectra. Because noise removes 24 spectral bands, the remaining 200 spectral bands, ranging from 400 to 2500 nm, were used for experiments. As shown in Table I, the IN dataset covers 16 classes of interest. Fig. 8 shows the pseudo color image and ground truth classification map of the Indian Pine dataset.

The UP dataset was acquired in 2001 by the ROSIS reflection optical imaging spectrometer in northern Italy. The images contain 115 bands, with 610×340 pixels, a wavelength ranging from 430 to 860 nm, and a spatial resolution of 1.3 m. As shown in Table II, there are nine types of ground truth. The pseudo color

TABLE I
Number of Training Samples, Test Samples, and Verification Samples of IN

NO.	Class	Train.	Val.	Test
1	Alfalfa	3	3	40
2	Corn-notill	72	72	1284
3	Corn-mintill	42	42	746
4	Corn	12	12	213
5	Grass-pasture	25	25	433
6	Grass-tree	37	37	656
7	Grass-pasture-mowed	2	2	24
8	Hay-windrowed	24	24	430
9	Oats	1	1	18
10	Soybean-notill	49	49	874
11	Soybean-mintill	123	123	2209
12	Soybean-clean	30	30	533
13	wheat	11	11	183
14	woods	64	64	1137
15	Buildings-Grass-Trees	20	20	346
16	Stone-Steel-Towers	5	5	83
	Total	520	520	9209

TABLE II
Number of Training Samples, Test Samples, and Verification Samples of UP

NO.	Class	Train.	Val.	Test
1	Asphalt	199	199	6233
2	Meadows	560	560	17529
3	Gravel	63	63	1973
4	Trees	92	92	2880
5	Metal Sheets	41	41	1263
6	Bare soil	151	151	4745
7	Bitumen	40	40	1232
8	Bricks	111	111	3460
9	Shadows	29	29	889
	Total	1286	1286	40204

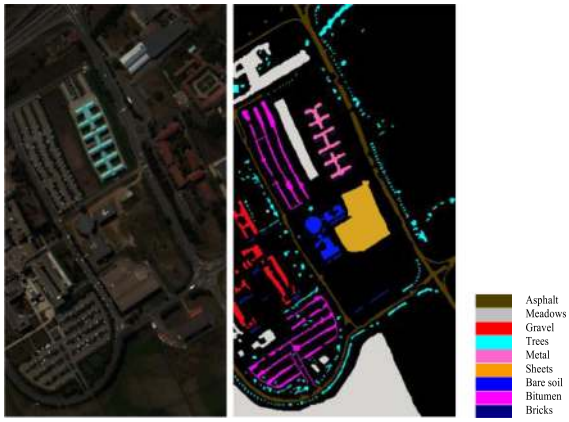


Fig. 9. UP dataset. (a) Pseudo color image (b) Ground truth. (c) Map color.

TABLE III
NUMBER OF TRAINING SAMPLES, TEST SAMPLES, AND VERIFICATION SAMPLES OF KSC

NO.	Class	Train.	Val.	Test
1	Scrub	39	39	683
2	Willow swamp	13	13	217
3	CP hammock	13	13	230
4	Slash pine	13	13	226
5	Oak/Broadleaf	9	9	143
6	Hardwood	12	12	205
7	Swap	6	6	93
8	Graminoid marsh	22	22	387
9	Spartina marsh	26	26	468
10	Cattail marsh	21	21	362
11	Salt marsh	21	21	377
12	Mud flats	26	26	451
13	Water	47	47	833
	Total	268	268	4675

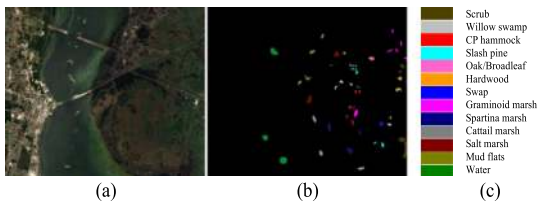


Fig. 10. KSC dataset. (a) Pseudo color image (b) Ground truth. (c) Map color.

image and ground truth classification map of the University of Pavia dataset are shown in Fig. 9.

The KSC dataset was photographed and collected by the AVIRIS sensor in Florida in 1996. It contains 224 bands of 614 $\times$ 512 pixels, and has a spatial resolution of 18 m. As can be seen from Table III, there are 13 highland and wetland class in the KSC dataset, excluding water absorption and low SNR bands, and 176 bands are left in the image. The pseudo color image and ground truth classification map of the KSC dataset are shown in Fig. 10.

In the two datasets of IN and KSC, the number of training data, validation data, and test data are set to 5%, 5%, and 90% of all labeled data, respectively. In UP dataset, the number of

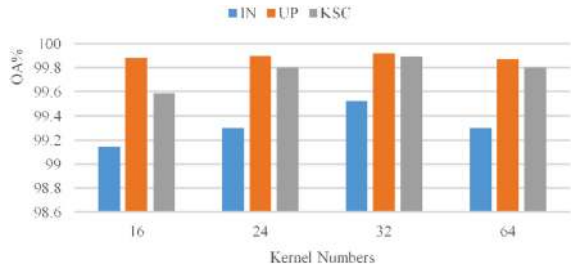


Fig. 11. Overall accuracy histogram of different convolution kernels numbers on three datasets.

training data, verification data, and test data are set to 3%, 3%, 94%, respectively. At the same time, all the input data of these three datasets are transformed into standard normal distribution. Tables I–III list the number of classes used for training, validation, and testing in the three datasets: IN, UP, and KSC.

B. Framework Setting

After designing the framework of the CSMS-SSRN, we carried out extensive experiments. Through an analysis of the training process, we found that three factors affected the training time and classification accuracy of the proposed CSMS-SSRN network, which were the learning rate, the number of convolutional kernels, and the size of the input data. During the training process, the training batch size was set to 16, and the RMSProp optimizer was used to optimize the training of the model. During the training process of 80 epochs, the model with the highest classification performance in the validation group was saved, and the best training results were generated.

First, the learning rate was an important hyperparameter for deep learning, as it determines whether the objective function can converge to a local minimum, and when it converges to the minimum. For the model, a good learning rate promotes the training process and helps the objective function to converge to the local minimum value at an appropriate time. Therefore, choosing an appropriate learning rate is very important for the training of the model. We carried out experiments for each dataset, hoping to find the best learning rate for CSMS-SSRN network training from {0.01, 0.003, 0.001, 0.0003, 0.0001, 0.00003}. According to the results, 0.0003 was the best learning rate for the IN, UP, and KSC datasets.

Second, the number of convolutional filter banks was an important factor for determining the cost of the CSMS-SSRN network. As shown in Fig. 7, the proposed network had the same number of convolutional filter banks in each convolutional layer of the spectral and spatial residual blocks. Therefore, to find general data, we use different kernel numbers in each convolutional layer. As shown in Fig. 11, each convolutional filter bank with 32 cores achieved the highest classification accuracy in the three datasets IN, UP, and KSC.

Third, for the deep learning framework used for HSI classification, the size of input the data cubes is one of the key factors that determine the result of network classification. Therefore, we tested many times for different sizes of input data cubes.

TABLE IV
OVERALL ACCURACY (%) FOR INPUT PATCHES WITH DIFFERENT SPATIAL SIZES FOR THREE DATASETS

Spatial size	IN	UP	KSC
7×7×L	99.13	99.83	98.95
9×9×L	99.62	99.96	99.96
11×11×L	98.71	98.48	99.86
13×13×L	98.39	99.85	99.91

TABLE V
CLASSIFICATION RESULTS OF VARIOUS METHODS ON IN DATASET

	SVM	CNN	SSUN	SSRN	FDSSC	MS-SSRN	CS -SSRN	CSMS -SSRN
OA%	60.67	76.91	89.81	92.95	93.95	93.28	94.02	95.58
AA%	70.61	86.14	90.89	92.14	92.82	94.39	95.36	95.85
Kappa	53.33	73.50	88.49	91.99	93.10	92.33	93.18	95.58
1	100	100	69.70	100	96.30	100	100	93.48
2	63.59	61.72	94.35	96.88	90.73	96.07	96.82	94.40
3	39.47	47.80	93.17	94.56	93.65	90.04	89.82	92.17
4	71.91	53.92	88.89	59.48	90.00	98.46	95.57	97.47
5	64.41	97.49	98.74	96.24	97.14	97.61	98.06	90.79
6	96.42	92.29	86.71	94.91	98.09	97.53	98.65	95.27
7	0	100	88.00	100	77.78	100	100	96.43
8	92.67	90.00	95.87	91.24	96.30	95.52	92.84	100
9	0	100	100	100	100	100	100	100
10	100	89.26	60.33	91.43	94.81	83.93	85.60	89.87
11	50.35	80.67	99.65	97.55	96.42	93.37	94.45	97.13
12	93.88	96.06	96.81	81.17	87.85	90.40	93.65	95.03
13	95.22	99.00	100	95.79	100	100	100	99.76
14	69.87	91.73	95.55	94.84	92.86	96.13	95.75	99.29
15	91.92	80.59	98.00	94.85	91.07	92.81	93.94	98.45
16	100	97.67	88.54	85.29	82.08	78.38	90.53	94.09

In the test, the size of the input data cubes was set to 7 × 7 × L, 9 × 9 × L, 11 × 11 × L, and 13 × 13 × L. Table IV shows that if the size of the input data cube is greater than 9 × 9 × L, the proposed CSMS-SSRN classification performance is significantly reduced. In the three datasets, when the size of the input cube was set to 9 × 9 × L, the model performance was optimal. We kept the size of the input data cubes consistent to fairly compare different classification methods.

C. Classification Results

This section compares SVM [31], CNN [42], SSUN [49], SSRN [50], FDSSC [51], and our proposed CSMS-SSRN classification method on three datasets. In order to verify the effectiveness of the multiscale strategy in CSMS-SSRN framework, we also test the network with three-Branch Architecture (MS-SSRN). To verify the effectiveness of attention mechanism, we also add attention mechanism to SSRN (CS-SSRN) and compare it with SSRN to determine the superiority of the attention mechanism in classification performance. For a fair experimental comparison, we set the input block volume of all methods to 9 × 9 × L, and adjusted all the comparison algorithms to the best setting. In the IN and KSC, the number of training data, validation data, and test data were set to 5%, 5%, and 90% of all labeled data, respectively. In the UP dataset, the number of training data, verification data, and test data were set to 3%, 3%, and 94%, respectively.

TABLE VI
CLASSIFICATION RESULTS OF VARIOUS METHODS ON UP DATASET

	SVM	CNN	SSUN	SSRN	FDSSC	MS-SSRN	CS-SSRN	CSMS -SSRN
OA%	78.65	85.57	92.62	97.08	97.58	97.99	98.63	99.29
AA%	68.99	67.86	93.97	97.34	97.21	97.72	98.09	99.30
Kappa	70.80	80.74	90.11	96.12	96.79	97.34	98.18	99.05
1	78.09	79.32	95.06	93.11	96.75	98.88	96.53	99.72
2	86.25	94.24	95.29	99.00	98.97	99.88	99.91	99.58
3	50.00	0.00	92.26	97.21	94.26	99.88	89.66	99.73
4	95.91	97.06	99.93	100	99.18	98.86	99.69	99.55
5	100	99.76	98.44	99.85	100	100	100	100
6	53.14	74.50	99.27	96.08	96.19	99.29	99.69	99.92
7	0.00	0.00	99.17	99.14	95.74	98.47	99.19	100
8	66.92	65.87	66.51	91.80	93.76	84.56	98.11	95.18
9	90.56	100	99.79	99.89	100	99.67	100	100

TABLE VII
CLASSIFICATION RESULTS OF VARIOUS METHODS ON KSC DATASET

	SVM	CNN	SSUN	SSRN	FDSSC	MS-SSRN	CS-SSRN	CSMS -SSRN
OA%	65.61	88.32	94.45	95.26	95.96	95.84	96.43	96.96
AA%	68.57	82.38	90.93	92.26	94.03	93.54	94.47	95.66
Kappa	61.87	87.00	93.82	94.72	95.50	95.37	96.03	96.62
1	100	97.71	99.71	98.30	100	99.29	99.56	94.57
2	100	95.92	91.88	91.20	100	98.60	99.06	100
3	7.87	88.71	90.30	100	97.72	88.10	99.09	96.02
4	27.58	41.74	66.78	78.69	63.88	71.56	89.33	88.24
5	100	62.99	85.42	71.58	64.62	92.20	53.44	85.71
6	92.86	61.31	100	95.32	99.37	79.76	99.45	100
7	19.88	58.58	57.76	76.86	100	95.88	91.30	86.11
8	57.69	88.71	98.75	96.14	97.18	97.31	97.68	97.79
9	67.37	92.94	97.76	95.24	100	98.36	100	95.75
10	37.44	89.45	95.06	96.71	99.73	95.60	99.19	100
11	100	100	100	100	100	100	100	100
12	80.69	92.86	98.68	99.33	100	99.34	100	99.57
13	100	100	100	100	100	100	100	99.88

Tables V–VII report the classification accuracy of OA, AA, and kappa coefficients, and all categories of the three datasets for HSI classification. In these three datasets, the classification accuracy of the CSMS-SSRN framework proposed in this study was the highest among all methods, and the classification result for SVM was worse than that of any other method based on deep learning. For example, in the IN dataset, the OA% of the CSMS-SSRN was 95.58%, which was approximately 34.91% higher than that of the SVM. Among these three groups of datasets, the classification results obtained by the FDSSC network were better than those of the SSUN and SSRN, and lower than the CSMS-SSRN. For example, in the IN dataset, the OA% of the CSMS-SSRN was 1.63% higher than that of the FDSSC. The accuracy of the CSMS-SSRN for some classes in the dataset reached 100%. For example, the CSMS-SSRN achieved 100% accuracy in the three categories of the UP dataset: *metal sheets*, *bitumen*, and *shadows*. Similarly, this happened in two other datasets. On the other hand, Experimental data showed that the classification results of MS-SSRN were higher than that of single-scale SSRN network in the three datasets, so the multiscale strategy can improve the classification effect of the network. And CS-SSRN also had good classification performance. The OA% of CS-SSRN was 1.63%, 1.05%, and 0.47% higher than those of FDSSC, respectively. This is due to the existence of attention module, which improves the accuracy

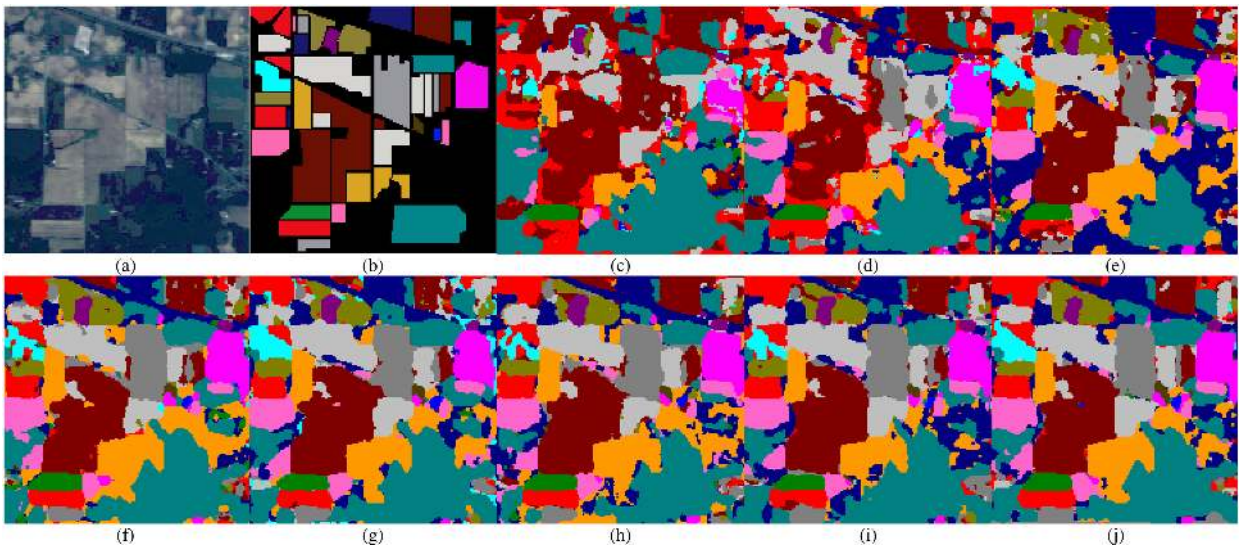


Fig. 12. Classification results of the best models for the IN dataset. (a) Pseudo color image. (b) Ground-truth labels. (c) SVM. (d) CNN. (e) SSUN. (f) SSRN. (g) FDSSC. (h) MS-SSRN. (i) CS-SSRN. (J) CSMS-SSRN.

of network classification. In addition, in the IN dataset, *oats* are typical examples of an unbalanced sample, and the classification accuracy of the CSMS-SSRN was still as high as 100%. Therefore, compared with other networks, CSMS-SSRN network had good robustness in the absence of sufficient samples.

Figs. 12–14 shows the classification result pictures of several comparison models and proposed models from three datasets, and the pseudo color image of the original HSI from three datasets. From the figure, we find that the three classification results of the SVM are not satisfactory and contain much noise. The CNN also has noise in some classes, for example, in the bare soil class in UP and in the slash pine class in KSC. In addition, the CS-SSRN performed very well on these three datasets. In addition to the SVM and CNN, the other methods achieved good results. However, the CSMS-SSRN not only continuously learns spectral and spatial features through convolutional kernels of different scales in a three-branch architecture, but also improves the accuracy of a small number of classes through the attention mechanism. Thus, compared with other methods, the obtained classification map is more accurate and smoother.

In order to test the dependence of the training results of the CSMS-SSRN on the samples used for training, we randomly selected four different percentage samples from 5% to 20% as the training data for the three datasets IN, UP, and KSC. Fig. 15 shows the OA% line chart for various methods at different sampling rates. In the three datasets, the CSMS-SSRN shows better performance than the other comparison methods, especially in the IN dataset, which has obvious advantages. This is because the CSMS-SSRN has more recognition features and stronger representation ability than the other comparative networks. In the case of fewer training samples, the CSMS-SSRN still produced better classification results than the other networks for all three HSI datasets. However, when the sampling rate was higher than 15%, the improvement in the other methods was not obvious, except the SVM and CNN. Because the classification overall accuracy OA was greater than 99%, the improvement was limited.

TABLE VIII
COMPARISON OF TRAINING AND TESTING TIME BETWEEN THE THREE CONTRAST MODELS AND PROPOSED MODEL ON THREE DATASETS

		IN	UP	KSC
CNN	Train. (m)	2.35	5.21	1.22
	Test. (s)	8.98	29.36	3.52
SSRN	Train. (m)	9.57	19.23	5.15
	Test. (s)	10.18	36.40	4.46
FDSSC	Train. (m)	4.31	9.62	3.21
	Test. (s)	13.65	44.25	5.32
CSMS-SSRN	Train. (m)	9.78	19.48	5.27
	Test. (s)	19.06	78.89	9.07

The training and testing times using the CNN, SSRN, CS-SSRN, and CSMS-SSRN are shown in Table VIII. All experiments used a TITAN X(Pascal) GPU and 12 GB RAM. The training time of SSRN is about five times that of CNN, which means that the computational cost of SSRN is higher. This is because SSRN needs 200 epochs to obtain good accuracy. However, the training time of FDSSC is much shorter than that of SSRN, because FDSSC only needs 80 epochs to obtain higher accuracy. Since the network structure of CSMS-SSRN is more complex, the training time of CSMS-SSRN is roughly the same as that of SSRN. However, CSMS-SSRN network can achieve higher accuracy. Fortunately, the use of high-performance graphics cards greatly reduced training time. The accuracy and loss curves of the CSMS-SSRN training and verification sets from the IN, UP, and KSC datasets are shown in Fig. 16. For the three datasets, the CSMS-SSRN model converges quickly at the beginning of the training process, and there is no large fluctuation in the loss curve thereafter. The curve converged in 75 epochs. Therefore, we used 80 epochs.

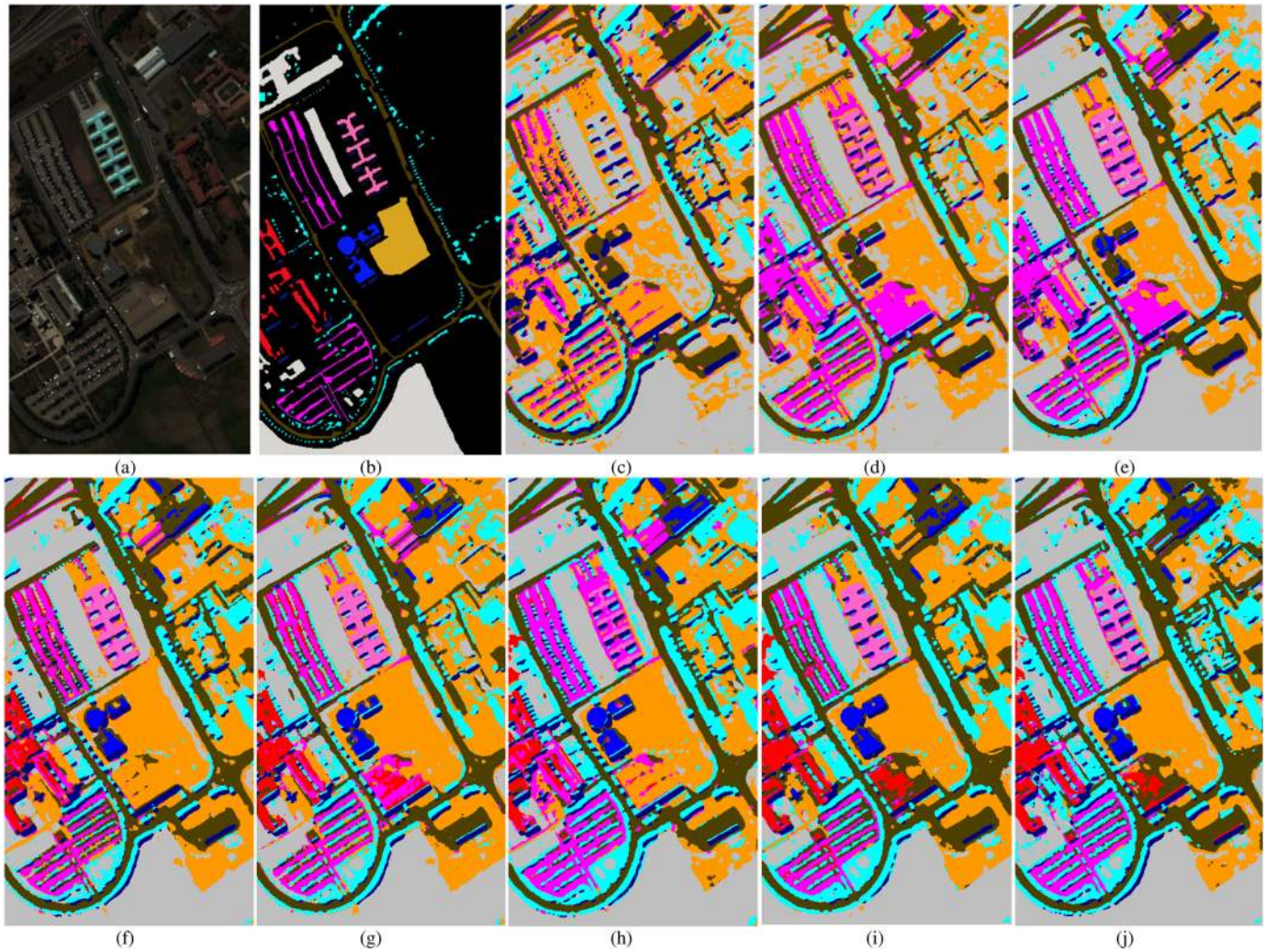


Fig. 13. Classification results of the best models for the UP dataset. (a) Pseudo color image. (b) Ground-truth labels. (c) SVM. (d) CNN. (e) SSUN. (f) SSRN. (g) FDSSC. (h) MS-SSRN. (i) CS-SSRN. (J) CSMS-SSRN.

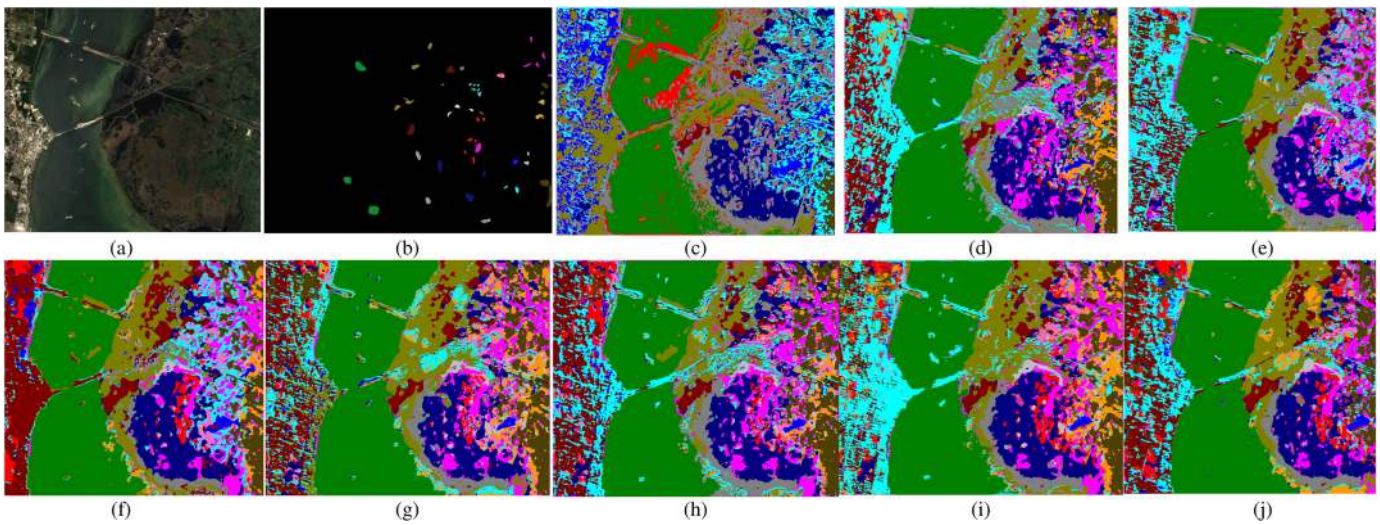


Fig. 14. Classification results of the best models for the KSC dataset. (a) Pseudo color image. (b) Ground-truth labels. (c) SVM. (d) CNN. (e) SSUN. (f) SSRN. (g) FDSSC. (h) MS-SSRN. (i) CS-SSRN. (J) CSMS-SSRN.

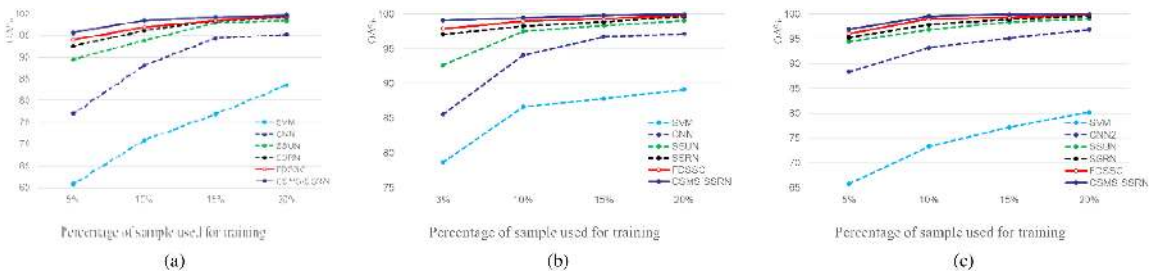


Fig. 15. (a) Line chart of overall accuracy of different models at different sampling rates for IN. (b) Line chart of overall accuracy of different models at different sampling rates for UP. (c) Line chart of overall accuracy of different models at different sampling rates for KSC.

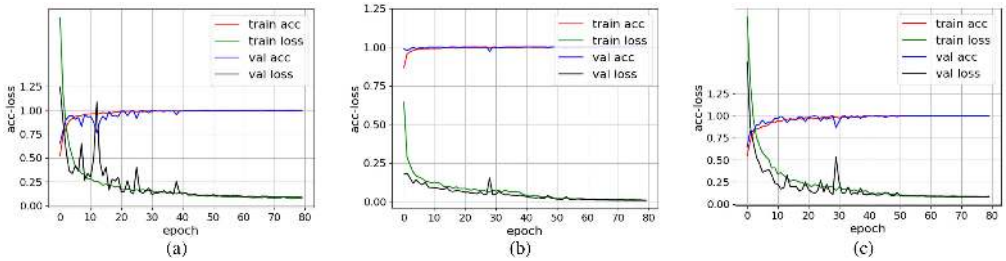


Fig. 16. Accuracy and loss function curves of the training and validation sets for (a) IN, (b) UP, and (c) KSC.

TABLE IX
A-RESNET AND CSMA-SSRN ON UP AND IN DATASETS BASED ON THE CLASSIFICATION RESULTS OF 100 RANDOM SAMPLES OF EACH CLASS

		IN (15%)	UP (10%)
A-ResNet	OA%	98.75	99.86
	AA%	97.05	99.76
	Kappa	98.58	99.82
CSMS-SSRN	OA%	99.17	99.91
	AA%	99.12	99.86
	Kappa	99.05	99.88
		IN (10%)	UP (3%)
SSAN	OA%	95.49	98.02
	AA%	94.17	96.90
	Kappa	94.85	97.37
CSMS-SSRN	OA%	98.41	99.29
	AA%	96.54	99.30
	Kappa	98.19	99.05

We compared the CSMS-SSRN with recent hyperspectral image classification methods based on the fusion of the attention mechanism and deep learning. In [52], based on a combination of visual attention and deep neural networks, Haut *et al.* proposed a mask mechanism that was driven by attention and automatically filtered network features (A-ResNet). Because there was no code in the original text, we kept the parameters and sampling strategy of CSMS-SSRN consistent with A-ResNet. During the training process, 15% of IN dataset samples were used for training, while in UP dataset, 10% of labeled samples were used for network training. As shown in Table IX, the OA, AA, Kappa values of CSMS-SSRN were better than those of A-ResNet on the IN and UP datasets. In the IN dataset, the OA result of the CSMS-SSRN was 0.42% higher than that of A-ResNet. Moreover, we also compared the proposed CSMS-SSRN with the SSAN network proposed in [58]. The OA% value of CSMS-SSRN was 2.92% higher than that of SSAN when 10% of IN dataset was used as training samples. Similarly, the OA% value of CSMS-SSRN was 1.27% higher than that of SSAN when 3% of UP dataset was used as training samples. This is because the CSMS-SSRN network structure is more unique and diverse, so the extracted features are more representative.

IV. CONCLUSION

This article presented a novel CSMS-SSRN for hyperspectral classification. The designed CSMS-SSRN consisted of a three-branch architecture, 3-D attention module, pooling layer, dropout layer, and fully connected layer. The three-branch architecture part included three levels of independent and parallel residual block structures. By using several different convolutional kernels to learn the features of HSIs, the network obtained different receptive fields and more comprehensive multiscale features. Second, this study used an attention mechanism to enhance the expressiveness of the image features from the two aspects of channel and spatial domains.

Through a comparison of the experimental data, it was proven that the CSMS-SSRN network model proposed in this article was superior to other machine learning methods, such as CNN, SSUN, FDSSC, SSRN, etc., in terms of the accuracy and robustness of HSI classification. There are three reasons why the CSMS-SSRN was superior to other deep learning networks. First, the network processed the spatial and spectral features of the HSIs, respectively, through the recurrent block structure. Second, in order to make full use of all the kinds of information in HSIs with limited samples and effectively solve the problem

of the limited receptive field of a single-scale convolutional kernel, we introduced multiscale filter banks and established a three-branch architecture. More discriminant features were extracted by differently scaled filter banks. Finally, the 3-D attention module was used to improve the deep multiscale feature representation ability, which improved the classification accuracy of the CSMA-SSRN network.

In the experiment, the insufficient training samples had the problem of unbalanced classes, which showed that the CSMS-SSRN network could achieve similar or even better accuracy than existing methods with difficult samples. At the same time, the CSMS-SSRN included deep multiscale features and the general attention mechanism module, which could be easily applied to other remote sensing classification problems.

REFERENCES

[1] L. Sun, W. Ge, Y. Chen, J. Zhang, and B. Jeon, "Hyperspectral unmixing employing l 1− l 2 sparsity and total variation regularization," *Int. J. Remote Sens.*, vol. 39, no. 19, pp. 6037–6060, Jun. 2018.

[2] L. Sun, T. Zhan, Z. Wu, L. Xiao, and B. Jeon, "Hyperspectral mixed denoising via spectral difference-induced total variation and low-rank approximation," *Remote Sens.*, vol. 10, no. 12, Dec. 2018, Art. no. 1956.

[3] Y. Ma *et al.*, "Remote sensing big data computing: Challenges and opportunities," *Future Gener. Comput. Syst.*, vol. 51, pp. 47–60, Oct. 2015.

[4] X. Huang, L. Wang, J. Yan, Z. Deng, S. Wang, and Y. Ma, "Towards building a distributed data management architecture to integrate multi-sources remote sensing big data," in *Proc. HPCC Smart City DSS*, Jan. 2018, pp. 83–90.

[5] T. Shuai, K. Sun, B. Shi, and J. Chen, "A ship target automatic recognition method for sub-meter remote sensing images," in *Proc. 4th Int. Workshop Earth Obs. Remote Sens. Appl.*, 2016, pp. 153–156.

[6] L. Liwei, M. Jianwen, C. Xue, W. Qi, and X. Xiaoyan, "High spatial resolution remote sensing image segmentation using temporal independent pulse coupled neural network," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.,* 2007, pp. 1915–1917.

[7] W. Fu, H. Pei, X.-W. Gao, C. Bai, H. Tian, and Q.-Y. Zhu, "Model inversion of BBPV based on DWD of MISR RS image," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 8, pp. 1390–1393, Aug. 2014.

[8] H. Gao, D. Jupp, Y. Qin, X. Gu, and T. Yu, "Cross-calibration of the HSI sensor reflective solar bands using hyperion data," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 4127–4137, Jul. 2015.

[9] S. Zhang, J. Li, Z. Wu, and A. Plaza, "Spatial discontinuity-weighted sparse unmixing of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 5767–5779, Oct. 2018.

[10] P. Ghamisi *et al.*, "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.

[11] 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.

[12] 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.

[13] M. Long and Y. Zeng, "Detecting iris liveness with batch normalized convolutional neural network," *Comput. Mater. Continua*, vol. 58, pp. 493–504, Jan. 2019.

[14] D. Zhang, G. Yang, F. Li, J. Wang, and A. K. Sangaiah, "Detecting seam carved images using uniform local binary patterns," *Multimedia Tools Appl.*, vol. 79, pp. 8415–8430, Jul. 2018.

[15] L. Sun, F. Wu, T. Zhan, W. Liu, J. Wang, and B. Jeon, "Weighted nonlocal low-rank tensor decomposition method for sparse unmixing of hyperspectral images," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 13, no. 1, pp. 1174–1188, Mar. 2020.

[16] T. V. Bandos, L. Bruzzone, and G. Camps-Valls, "Classification of hyperspectral images with regularized linear discriminant analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 3, pp. 862–873, Mar. 2009.

[17] J. Zhang, X. Jin, J. Sun, J. Wang, and A. K. Sangaiah, "Spatial and semantic convolutional features for robust visual object tracking," *Multimedia Tools Appl.*, vol. 79, pp. 15095–15115, 2020.

[18] D. Zeng, Y. Dai, F. Li, R. S. Sherratt, and J. Wang, "Adversarial learning for distant supervised relation extraction," *Comput. Mater. Continua*, vol. 55, no. 1, pp. 121–136, Jan. 2018.

[19] W. Wei, J. Yongbin, L. Yanhong, L. Ji, W. Xin, and Z. Tong, "An advanced deep residual dense network (DRDN) approach for image super-resolution," *Int. J. Comput. Intell. Syst.*, vol. 12, no. 2, pp. 1592–1601, 2019.

[20] D. Zeng, Y. Dai, F. Li, J. Wang, and A. K. Sangaiah, "Aspect based sentiment analysis by a linguistically regularized CNN with gated mechanism," *J. Intell. Fuzzy Syst.*, vol. 36, no. 5, pp. 3971–3980, May. 2019.

[21] Y. Shen *et al.*, "Empirical comparisons of deep learning networks on liver segmentation," *Comput. Mater. Continua*, vol. 62, no. 3, pp. 1233–1247, 2020.

[22] R. Meng, S. G. Rice, J. Wang, and X. Sun, "A fusion steganographic algorithm based on faster R-CNN," *Comput. Mater. Continua*, vol. 55, no. 1, pp. 1–16, May. 2018.

[23] Y. Chen, W. Xu, J. Zuo, and K. Yang, "The fire recognition algorithm using dynamic feature fusion and IV-SVM classifier," *Cluster Comput.*, vol. 22, no. 3, pp. 7665–7675, Mar. 2019.

[24] Y. Song, G. Yang, H. Xie, D. Zhang, and S. Xingming, "Residual domain dictionary learning for compressed sensing video recovery," *Multimedia Tools Appl.*, vol. 76, no. 7, pp. 10083–10096, Jun. 2017.

[25] Y. Chen, J. Xiong, W. Xu, and J. Zuo, "A novel online incremental and decremental learning algorithm based on variable support vector machine," *Cluster Comput.*, vol. 22, no. 3, pp. 7435–7445, Jan. 2019.

[26] G. Licciardi, P. R. Marpu, J. Chanussot, and J. A. Benediktsson, "Linear versus nonlinear PCA for the classification of hyperspectral data based on the extended morphological profiles," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 3, pp. 447–451, May 2011.

[27] E. Blanzieri and F. Melgani, "Nearest neighbor classification of remote sensing images with the maximal margin principle," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 6, pp. 1804–1811, Jun. 2008.

[28] Y. Chen, Z. Lin, and X. Zhao, "Riemannian manifold learning-based k-nearest-neighbor for hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2013, pp. 1975–1978.

[29] W. Li, C. Chen, H. Su, and Q. Du, "Local binary patterns and extreme learning machine for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3681–3693, Jul. 2015.

[30] S. Deng, Y. Xu, Y. He, J. Yin, and Z. Wu, "A hyperspectral image classification framework and its application," *Inf. Sci.*, vol. 299, pp. 379–393, Apr. 2015.

[31] G. Camps-Valls, L. Gomez-Chova, J. Muñoz-Marí, J. Vila-Francés, and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 1, pp. 93–97, Jan. 2006.

[32] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3973–3985, Oct. 2011.

[33] S. Zhou, M. Ke, and P. Luo, "Multi-camera transfer GAN for person re-identification," *J. Vis. Commun. Image Representation*, vol. 59, pp. 393–400, Feb. 2019.

[34] J. Liu, C. Gu, J. Wang, G. Youn, and J.-U. Kim, "Multi-scale multi-class conditional generative adversarial network for handwritten character generation," *J. Supercomput.*, vol. 75, no. 4, pp. 1922–1940, Dec. 2019.

[35] S. He, Z. Li, Y. Tang, Z. Liao, J. Wang, and H. Kim, "Parameters compressing in deep learning," *Comput. Mater. Continua*, vol. 62, pp. 321–336, 2020.

[36] Y. Tu, Y. Lin, J. Wang, and J.-U. Kim, "Semi-supervised learning with generative adversarial networks on digital signal modulation classification," *Comput. Mater. Continua*, vol. 55, no. 2, pp. 243–254, 2018.

[37] X. Zhang, Y. Sun, K. Jiang, C. Li, L. Jiao, and H. Zhou, "Spatial sequential recurrent neural network for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens.*, vol. 11, no. 11, pp. 4141–4155, Nov. 2018.

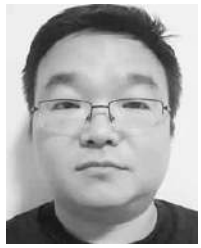
[38] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.

[39] 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.

[40] L. Sun, C. Ma, Y. Chen, H. J. Shim, Z. Wu, and B. Jeon, "Adjacent superpixel-based multi-scale spatial-spectral kernel for hyperspectral classification," *IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens.*, vol. 12, no. 6, pp. 1905–1919, Jun. 2019.

[41] L. Sun *et al.*, "Low rank component Induced spatial-spectral kernel method for hyperspectral image classification," *IEEE Trans. Circuits Syst. Video Technol.*, to be published, doi: 10.1109/TCSVT.2019.2946723.

[42] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.,* 2015, pp. 4959–4962.
[43] W. Zhao and S. Du, "Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Aug. 2016.
[44] 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.
[45] T. Alipourfard, H. Arefi, and S. Mahmoudi, "A novel deep learning framework by combination of subspace-based feature extraction and convolutional neural networks for hyperspectral images classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2018, pp. 4780–4783.
[46] J. Yue, W. Zhao, S. Mao, and H. Liu, "Spectral–spatial classification of hyperspectral images using deep convolutional neural networks," *Remote. Sens. Lett.*, vol. 6, no. 6, pp. 468–477, Dec. 2015.
[47] M. Zhang, W. Li, and Q. Du, "Diverse region-based CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2623–2634, Jun. 2018.
[48] S. Mei, J. Ji, J. Hou, X. Li, and Q. Du, "Learning sensor-specific spatial-spectral features of hyperspectral images via convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4520–4533, Aug. 2017.
[49] Y. Xu, L. Zhang, B. Du, and F. Zhang, "Spectral–spatial unified networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 5893–5909, Oct. 2018.
[50] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2017.
[51] W. Wang, S. Dou, Z. Jiang, and L. Sun, "A fast dense spectral–spatial convolution network framework for hyperspectral images classification," *Remote. Sens.*, vol. 10, no. 7, 2018, Art. no. 1068.
[52] Z. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza, and J. Li, "Visual attention-driven hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 8065–8080, Oct. 2019.
[53] P. Duan, X. Kang, S. Li, and P. Ghamisi, "Noise-robust hyperspectral image classification via multi-scale total variation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 6, pp. 1948–1962, Jun. 2019.
[54] S. Fang, D. Quan, S. Wang, L. Zhang, and L. Zhou, "A two-branch network with semi-supervised learning for hyperspectral classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2018, pp. 3860–3863.
[55] B.-S. Liu and W.-L. Zhang, "Multi-scale convolutional neural networks aggregation for hyperspectral images classification," in *Proc. IEEE Symp. Piezoelect., Acoust. Waves Device Appl.*, Jan. 2019, pp. 1–6.
[56] S. Wu, J. Zhang, and C. Zhong, "Multiscale spectral-spatial unified networks for hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul./Aug. 2019, pp. 2706–2709.
[57] K. Pooja, R. R. Nidamanuri, and D. Mishra, "Multi-Scale Dilated Residual Convolutional Neural Network for Hyperspectral Image Classification," in *Proc. IEEE 10th Workshop Hyperspectral Imag. Signal Process., Evol. Remote Sens.*, Sep. 2019, pp. 1–5.
[58] H. Sun, X. Zheng, X. Lu, and S. Wu, "Spectral-spatial attention network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3232–3245, Nov. 2020.
[59] Y. Gui and G. Zeng, "Joint learning of visual and spatial features for edit propagation from a single image," *Vis. Comput.*, vol. 36, no. 3, pp. 469–482, Jan. 2020.
[60] D. Zhang, T. Yin, G. Yang, M. Xia, L. Li, and X. Sun, "Detecting image seam carving with low scaling ratio using multi-scale spatial and spectral entropies," *J. Vis. Commun. Image Representation*, vol. 48, pp. 281–291, 2017.
[61] Z. Lu *et al.*, "The classification of gliomas based on a pyramid dilated convolution ResNet model," *Pattern Recognit. Lett.*, vol. 133, pp. 173–179, May 2020.
[62] B. Du, Q. Wei, and R. Liu, "An improved quantum-behaved particle swarm optimization for endmember extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 6003–6017, Aug. 2019.
[63] Z. Wu, Y. Xu, and J. J. Liu, "Sparsity-based methods for classification," in *Hyperspectral Image Analysis*, S. Prasad and J. Chanussot, Eds. Cham, Switzerland: Springer, 2020, pp. 233–257.



Zhenyu Lu (Member, IEEE) received the B.Sc. degree in electricity and the M.Sc. degree in information and communication from the Nanjing Institute of Meteorology, Nanjing, China, in 1999 and 2002, respectively, and the Ph.D. degree in optics engineering from the Nanjing University of Science and Technology, Nanjing, in 2008.

He was a Research Associate with the Department of Mathematics and Statistics, University of Strathclyde, Glasgow, U.K., from 2012 to 2013. He is currently a Professor with the School of AI, Nanjing University of Information Science and Technology. He has published seven international journal papers. His current research interests include neural networks, stochastic control, and artificial intelligence.

Bin Xu is currently working toward the master's degree in information and communication engineering in the School of Electronics and Information Engineering, Nanjing University of Information Science and Technology, Nanjing, China.

His research interests include hyperspectral image classification, machine learning, and deep learning.

Le Sun (Member, IEEE) was born in Jiangsu, China, in 1987. He received the B.S. degree from the School of Science, Nanjing University of Science and Technology (NJUST), Nanjing, Jiangsu, China, in 2009, and the Ph.D. degree from the School of Computer Science and Engineering, NJUST, in 2014.

He was doing research in the field of multi-images fusion based on sparse dictionary learning and compressive sensing as a Postdoctor with the Digital Media Laboratory, School of Electronic and Electrical Engineering, Sungkyunkwan University, Korea, from September 2015 to August 2018. He is currently an Associate Professor with NJUST. His research interests include sparse representation, compressive sensing and deep learning, especially in the field of hyperspectral image processing.

Tianming Zhan received the B.S. and M.S. degrees from the School of Math and Statistics, Nanjing University of Information Science and Technology, Nanjing, China, in 2006 and 2009, respectively, and the Ph.D. degree from the School of Computer Science and Engineering, Nanjing University of Science and Technology, in 2013.

He is currently an Associate Professor with the School of Information and Engineering, Nanjing Audit University, Nanjing. His research interests include medical image processing, hyperspectral image processing, machine learning, and data analysis.

Songze Tang received the B.Sc. degree in information and computer science from Anhui Agricultural University, Hefei, China, in 2009, and the Ph.D. degree in computer science and technology from Nanjing University of Science and Technology, Nanjing, China, in 2015.

He is currently an Assistant Professor with the Department of Criminal Science and Technology, Nanjing Forest Police College, Nanjing. His research interests include hyperspectral image processing, and computer simulation.