Spatial residual blocks combined parallel network for hyperspectral image classification.

ZHANG, B., QING, C., XU, X. and REN, J.

2020

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 22, 2020, accepted April 6, 2020, date of publication April 17, 2020, date of current version May 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2988553

Spatial Residual Blocks Combined Parallel Network for Hyperspectral Image Classification

BUYI ZHANG[1], (Graduate Student Member, IEEE), CHUNMEI QING[1], (Member, IEEE), XIANGMIN XU[1], (Senior Member, IEEE), AND JINCHANG REN[2], (Senior Member, IEEE)

[1]School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China
[2]Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow G1 1XQ, U.K.

Corresponding author: Chunmei Qing (qchm@scut.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61972163, Grant U1801262, Grant 61702192, Grant U1636218, and Grant 61802131.

ABSTRACT In hyperspectral image (HSI) classification, there are challenges of the spatial variation in spectral features and the lack of labeled samples. In this paper, a novel spatial residual blocks combined parallel network (SRPNet) is proposed for HSI classification. Firstly, the spatial residual blocks extract spatial features from rich spatial contexts information, which can be used to deal with the spatial variation of spectral signatures. Especially, the skip connection in spatial residual blocks is conducive to the back-propagation of gradients and mitigates the declining-accuracy phenomenon in the deep network. Secondly, the parallel structure is employed to extract spectral features. Spectral feature learning on parallel branches contains fewer independent connection weighs through parameter sharing. Thus, fewer parameters of the network require a lesser number of training samples. Furthermore, the feature fusion is conducted on the multi-scale features from different layers in the spectral feature learning part. Extensive experiments of three representative HSI data sets illustrate the effectiveness of the proposed network.

INDEX TERMS Feature fusion, hyperspectral image classification, parallel network, spatial residual blocks.

I. INTRODUCTION

Hyperspectral image contains abundant spatial and spectral information, which can provide a lot of useful information for image classification and target detection [1]. With rich spatial and spectral information, it is possible to distinguish the substances which are difficult to be identified in the traditional wide-band remote sensing [2]–[4]. Nowadays, HSI has been successfully used in many fields [5]–[10], such as agriculture science, environmental management, image segmentation, target detection, land-cover mapping, and object recognition. However, there are still some challenges in HSI classification [11]–[14]:

(1) A large number of spectral dimensions cause the curse of dimensionality. Although a large number of spectral dimensions provide a great amount of useful information, there are generally close correlations between spectral bands, especially adjacent ones, which lead to information redundancy. The overall classification performance even declines with the increasing number of bands, forming the so-called Hughes phenomenon.

(2) The number of labeled samples is scarce. With the increase of spectral dimension, more training samples are needed correspondingly. However, the labeled HSI samples are not easy to obtain, which is time-consuming and expensive.

(3) The huge spatial variation in spectral features. Due to the difference of atmospheric conditions at different times, acquisition system states, levels of soil moisture, reflectance and lighting conditions, the obtained spectral curves of the same ground objects may be different, while the different ground objects may present similar spectral curves. Thus, the HSI data in high-dimensional space have the characteristics of nonlinear separability, which called spatial variability.

To better use the abundant information of HSI, plenty of algorithms are proposed, including unsupervised and supervised learning algorithms [15]–[18]. Unsupervised learning algorithms, such as K-means [19], extract information from unlabeled samples according to the sample distribution. Theoretically, K-means can overcome the challenge of scarcity labeled samples. However, the performance of K-means is affected by the choice of clustering centers. Unfortunately, there is no way to ensure the best choice of initial clustering centers. Therefore, K-means is rarely used for the

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

classification of HSI in recent years. Supervised learning algorithms including k-nearest neighbor (KNN), support vector machine (SVM), convolutional neural network (CNN) and so on. These methods use a set of labeled data as input and aim to train a model that can get corresponding output according to the input. KNN is generally considered as the simplest classification method that utilizes the Euclidean distance to compute and evaluate the similarity between the training samples and the testing samples [20], [21]. However, KNN has high computational costs and weak generalization ability when the training set is small. SVM tries to learn the maximum-margin hyperplane among all the samples. Thus, through the hyperplane, the data can be best separated in the high dimensional feature space [22]. SVM has great advantages in solving the task of nonlinear high-dimensional data classification with small samples [23]. However, SVM does not utilize the spatial information of HSI, it is not conducive to the improvement of classification accuracy.

Due to the increase attention to the spatial information of HSI, CNN has become the popular method for the HSI classification [24]–[29]. It has been proved that CNNs can simultaneously extract both spatial and spectral features of HSI to produce precise classification results. As can be seen in [30], a two-channel deep CNN that contains two branches has been proposed. One of the branches is the spectral feature extraction channel, and the other is the spatial feature extraction channel. However, the two-channel network separates spectral feature learning from spatial feature learning, which may lead to the loss of some useful spatial-spectral correlative information during the fusion processing between spectral features and spatial features. To tackle this problem, the 3-D CNN was proposed, which takes raw HSI cube data without any pre-processing or post-processing as input [31]. Such networks can extract spatial information and spectral information in turn, and no longer separate the learning processes of spatial and spectral information. However, using 3-D data as the input increases the computation of deep networks. To solve this problem, [32] designs a parallel network composed of branches with the same structures to extract specific spectral features. Besides, the network contains only one spatial convolutional layer, which further simplifies the structure of the network. However, the spatial feature learning with a shallow layer may lose some useful spatial information, and the simple integration of features in the parallel network also loses some useful spectral information. Although this network is very fast, its classification performance is not very ideal. Besides, with the networks going deeper, the classification performances may become worse, which called the declining-accuracy phenomenon. Thus, [33] designs spectral and spatial residual blocks to learn the spectral information and spatial information, respectively. The residual connection makes gradient backpropagation easier and more stable, and relieve the declining-accuracy phenomenon in deep networks. However, the pooling operation after the spatial residual blocks may lose some useful spatial information. And the classification step uses only the deepest features, which also may lose some useful spatial information.

In order to mitigate the declining-accuracy phenomenon and gain ideal classification performance, this paper proposes a spatial residual blocks combined parallel network (SRPNet) for HSI classification. The major contributions of this paper are summarized and listed below.

(1) The proposed spatial residual blocks can extract abundant spatial features to deal with spatial variability. In addition, the skip connection between two convolutional layers can relieve the declining-accuracy phenomenon in the deep network.

(2) The parallel structure in the spectral feature learning part has fewer trainable parameters, which can reduce the required training samples.

(3) Feature fusion integrates multi-scale features of different layers to gain richer information in the spectral feature learning part, which can improve the classification performance.

(4) Extensive experiments on different cases illustrate that the proposed network is more stable and has competitive performance, especially on the tasks of small training sample numbers.

The rest content of the paper is arranged as hereafter. In part II, there is a detailed description of the proposed SRPNet. Part III reports the setting and results of experiments. Part IV makes the conclusion of the paper.

II. THE PROPOSED SRPNet

The overall architecture of SRPNet is shown in Figure 1. Assuming that raw HSI data contain n labeled pixels which presented as $\mathbf{X}_{\text{labeled}} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\} \in \Re^{1\times 1\times H}$, and then $\mathbf{Y}_{label} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n\} \in \Re^{1\times 1\times L}$ denotes corresponding labels, where H and L denote the number of spectral bands and land-cover categories, respectively. To make use of the rich spatial information, the pixels surrounding the target are taken into account, forming the 3-D input data cubes. Therefore, the input are $\mathbf{X}_{\text{in}} = \left\{\mathbf{x}'_1, \mathbf{x}'_2, \ldots, \mathbf{x}'_i\right\} \in \Re^{w\times w\times H}$, where $w \times w$ refers the spatial size of input data cubes, and i denotes the number of input samples.

In the SRPNet, there are three main parts. Firstly, the spatial feature learning part extracts spatial features through two spatial residual blocks. Then, the spectral feature leaning part extracts spectral features from divided data through parallel branches. And feature fusion is adapted in this part. Finally, the classification part concatenates all the outputs of all parallel branches and does classification through a softmax activation. To prevent overfitting, there are some batch normalization (BN) and ReLU activation after convolutional layers.

A. PIXEL-WISE MAPPING

In order to reduce the spectral redundancy, a convolutional layer with 1×1 kernel size is used to perform pixel-wise nonlinear mapping on spectral channels. This operation transforms the spectral channels into spectral

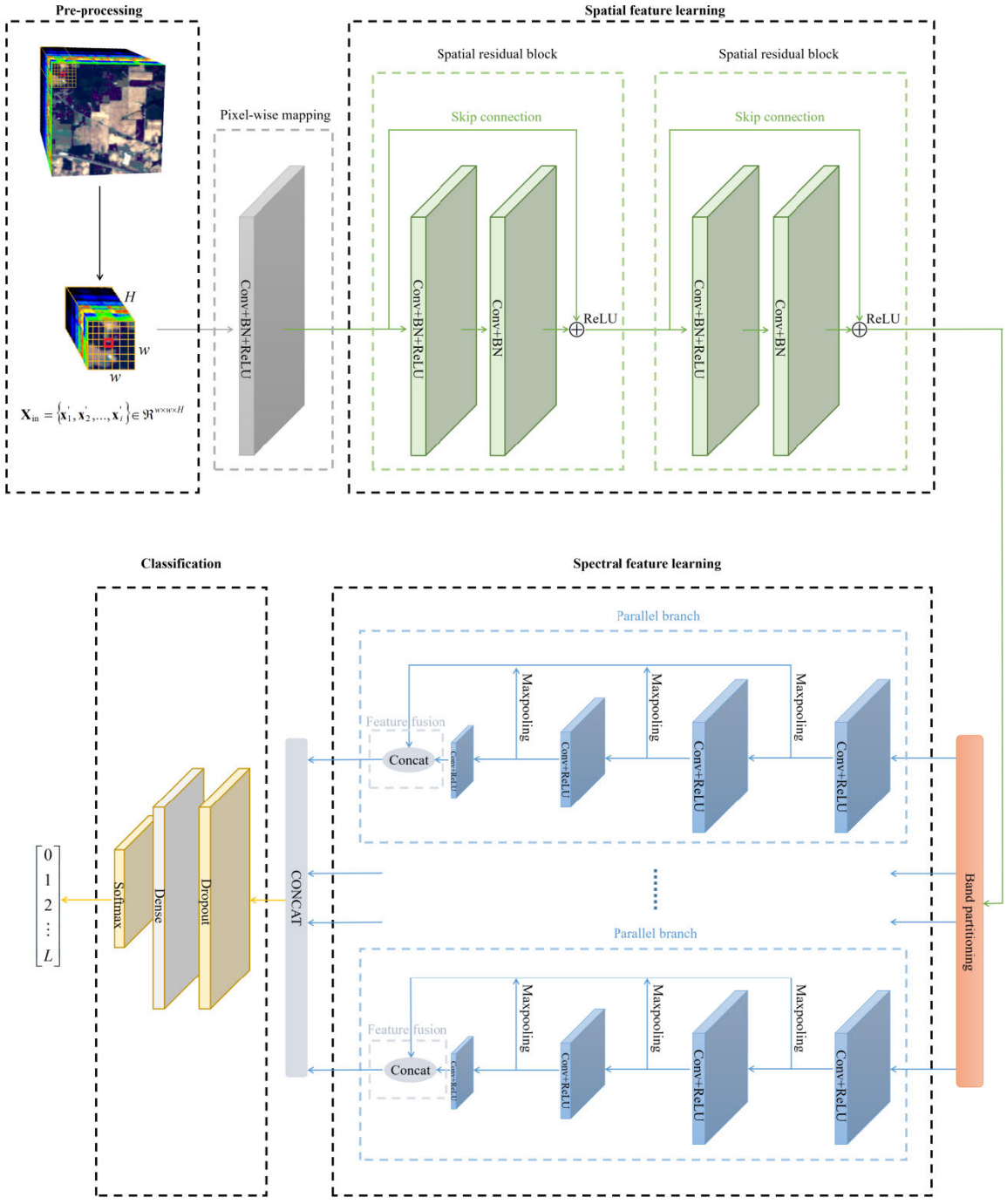

FIGURE 1. Block diagram of the SRPNet architecture.

features. Through this step, this layer gets the output $\mathbf{X}_{out}$ from input $\mathbf{X}_{in}$. The process can be performed as follows

$$\mathbf{x}_i'' = \Phi(\mathbf{x}_i'), \tag{1}$$

$$\mathbf{X}_{out} = \left\{\mathbf{x}_1'', \mathbf{x}_2'', \cdots, \mathbf{x}_i''\right\} \in \Re^{w\times w\times H_{out}}, \tag{2}$$

where $\mathbf{x}_i' = [x_i^{1'}, x_i^{2'}, \ldots, x_i^{H'}]$, $\mathbf{x}_i'' = [x_i^{1''}, x_i^{2''}, \ldots, x_i^{H_{out}''}]$. H_{out} refers the number of output channels along the spectral dimension. $x_i^{h'}, h = 1, 2, \ldots, H$ and $x_i^{h_{out}''}, h_{out} = 1, 2, \ldots, H_{out}$ denote the i-th input and output channels, respectively. $\Phi(\cdot)$ refers non-linear mapping algorithm. Let $w_{h_{out}h}$ denote the weight, $b_{h_{out}}^h$ denote the bias. Thus, the non-linear mapping algorithm can be performed as

$$x_i^{h_{out}''} = \sum_{h=1}^{H} w_{h_{out}h} x_i^{h'} + b_{h_{out}}^h. \tag{3}$$

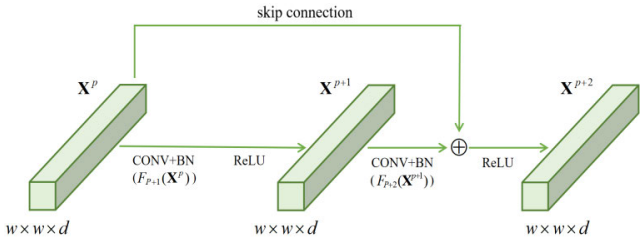

FIGURE 2. The structure of spatial residual block.

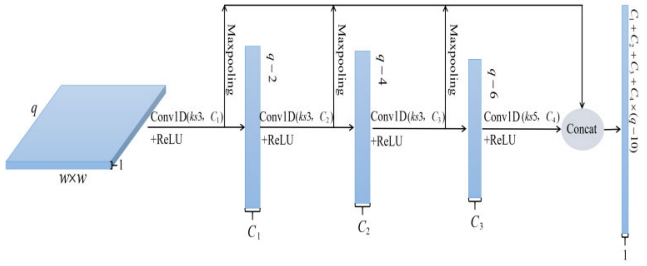

FIGURE 3. The structure of parallel branches, where *ks*3 denotes the kernel size is 3. *ks*5 denotes the kernel size is 5. $C_i, i = 1, 2, 3, 4$ denotes the number of output channels of the *i*-th convolutional layer in parallel branch.

B. SPATIAL RESIDUAL BLOCKS

To extract spatial features, two spatial residual blocks with the 3×3 kernel size are utilized after the convolutional layer. Through skip connection, the gradients of the upper layers can be quickly propagated back to the lower layers, which makes the training process of the model more convenient and stable. Therefore, the skip connection can alleviate the declining-accuracy phenomenon when the network going deeper.

The spatial residual block is shown in Figure 2. The filter bank h^{p+1} in the $(p+1)$-th layer contains d convolutional kernels of size 3×3. The value of d equals to the number of channels of $\mathbf{X}^p$. The spatial size of $\mathbf{X}^{p+1}$ and $\mathbf{X}^{p+2}$ stay at $w \times w$ using padding. The Conv-BN function $F_{P+1}(\mathbf{X}^p)$ and $F_{P+2}(\mathbf{X}^{p+1})$ denote the $(p+1)$-th and the $(p+2)$-th spatial convolutional layer which followed with the batch normalization (BN) $f_{BN}(\cdot)$. Thus, the architecture of spatial residual block is expressed as

$$F_{P+1}(\mathbf{X}^p) = f_{BN}\left(\mathbf{X}^{p*}\mathbf{h}^{p+1} + \mathbf{b}_{p+1}\right), \tag{4}$$

$$\mathbf{X}^{p+1} = R\left(F_{p+1}\left(\mathbf{X}^p\right)\right), \tag{5}$$

$$\mathbf{X}^{p+2} = R\left(\mathbf{X}^p + F_{p+2}\left(\mathbf{X}^{p+1}\right)\right), \tag{6}$$

where $\mathbf{b}_{p+1}$ represents the bias of the $(p+1)$-th layer. $R(\cdot)$ is the ReLU activation function. $\mathbf{X}^{p+1}$ and $\mathbf{X}^{p+2}$ represent the input and output feature cubes of the $(p+1)$-th layer.

C. BAND PARTITIONING

Deep networks often have a surprising amount of parameters. The training process of deep models requires high computing resources. Especially, in the proposed network, to learn more sufficient features, there are a large amount of channels in the spatial feature learning part, which require a lot of computation. To reduce the number of network trainable parameters and optimize the calculation, the parallel network contains parallel branches of parameter sharing is used to extract spectral features in the spectral feature learning part. In addition, fewer trainable parameters require lesser training samples.

Before spectral feature learning, the band partitioning operation is used to evenly divide input data into blocks along spectral dimension channels (shown as d in Figure 2). The band partition operation is expressed as

$$\{B_1, B_2, \ldots, B_{n_b}\} = \Psi(\mathbf{X}''', n_b), \tag{7}$$

where $\mathbf{X}''' = \left\{\mathbf{x}_1''', \mathbf{x}_2''', \cdots \mathbf{x}_i'''\right\} \in \Re^{w \times w \times d}$ refers the output of the spatial feature learning part. $\Psi(\cdot, \cdot)$ is the band partitioning function that splits $\mathbf{X}'''$ into n_b blocks with non-overlapping adjacent channels $\{B_i\}_{i=1}^{n_b}$ of equal width q, and $q = d/n_b$.

D. PARALLEL BRANCHES

After band partitioning operation, n_b parallel branches are used to extract the spectral features of the n_b data blocks $\{B_1, B_2, \ldots, B_{n_b}\}$, respectively. All the parallel branches in the parallel network have the same structure, which includes four 1-D convolutional layers. And the ReLU activation is used after every convolutional layer. The parallel branches share all the parameters and reduce network computing. The structure of one parallel branch is shown in Figure 3.

Furthermore, due to the strong correlation between different layers, hierarchical features can provide useful supplementary information for classification. To make use of the complementary information, the multi-scale features from convolutional layers of the parallel branch are fused. The process of feature fusion can be represented as

$$\mathbf{X}_{fused} = [\mathbf{L}_4, f(\mathbf{L}_3), f(\mathbf{L}_2), f(\mathbf{L}_1)], \tag{8}$$

where $\mathbf{X}_{fused}$ represents the result of feature fusion on the parallel branch. $[\cdot]$ refers the concatenation of outputs from different convolutional layers in the parallel branch. $f(\cdot)$ denotes the maxpooling function based on the channels. $\mathbf{L}_i, i = 1, 2, 3, 4.$ refers the i-th layer of the parallel branch.

Finally, features of every parallel branch are merged for classification.

E. CLASSIFICATION

After spatial and spectral feature learning, all the extracted features are transported to the classification part. Firstly, the dropout strategy is utilized to make the activation value of some neurons stop working with a certain probability through randomness. Dropout enhances the generalization of the model and improves the classification performance of the network. Then, a dense layer is used to extract the correlation between the previously extracted features by nonlinear changes and map features to the output space. Finally, there is a softmax layer used for classification.

F. TRAINING OF THE NETWORK

The model parameters are updated through the back-propagating loss which is expressed as

$$Loss = -\sum_{i=1}^{M}\sum_{l=1}^{L} P_{data}\left(l|\mathbf{x}_i\right)\log\left(P_{model}\left(l|\mathbf{x}_i\right)\right), \quad (9)$$

where M denotes the amount of training samples. L denotes the total number of output ground-truth classes. $P_{data}(class = l|\mathbf{x}_i)$ denotes observed conditional distribution of the i-th sample. $P_{model}(class = l|\mathbf{x}_i)$ denotes model conditional distribution of the i-th sample.

Obviously, the observed conditional distribution $P_{data}(class = l|\mathbf{x}_i)$ satisfies the following distribution

$$P_{data}(class = l|\mathbf{x}_i) = \begin{cases} 1, & if\ |\mathbf{y}_i| = l \\ 0, & otherwise \end{cases} \quad (10)$$

where $\mathbf{y}_i$ denotes the corresponding ground-truth label of $\mathbf{x}_i$.

Thus, loss function could be written as

$$Loss = -\sum_{i=1}^{M}\log\left(P_{model}\left(\mathbf{y}_i|\mathbf{x}_i\right)\right). \quad (11)$$

During backpropagation, the derivative g of formula (11) is required when updating the model parameters. Specifically, the mini-batch which contains m samples is selected from M training data. Then the gradient can be written as

$$g = \frac{\partial Loss}{\partial\theta} = \frac{1}{m}\nabla_\theta\left(-\sum_{i=1}^{m}\log(P_{model}(\mathbf{y}_i|\mathbf{x}_i)\right), \quad (12)$$

where θ denotes the initial parameters of the model, and ∇ refers the derivative operator.

Then the model parameters θ is upgraded according to the loss value in every iteration by use Adaptive Moment Estimation(Adam), which is shown in (13) and (14).

$$\Delta\theta = -\varepsilon\frac{\hat{s}}{\sqrt{\hat{r}} + \delta}, \quad (13)$$

$$\theta = \theta + \Delta\theta, \quad (14)$$

where $\hat{s} = s/(1 - p_1^t)$ refers the bias-corrected first moment estimate. $s = \rho_1 s + (1 - \rho_1)g$ refers the biased first moment estimate. $\hat{r} = r/(1 - p_2^t)$ denotes the bias-corrected bias second moment estimate. $r = \rho_2 r + (1 - \rho_2)g^2$ denotes the biased second moment estimate. ε denotes the step size. $\rho_1, \rho_2 \in [0, 1)$ denote the moment estimation of exponential decay rate. t denotes the time step.

The training process is shown in Algorithm 1.

Algorithm 1 Adaptive Moment Estimation of Model

Input: mini-batch size m, maximum number of iterations N, training data cubes $\{\mathbf{X}_i, \mathbf{y}_i\}_{i=1}^{M}$.
1: while iteration $< N$ do
2: Choose one mini-batch $\{\mathbf{X}_i, \mathbf{y}_i\}_{i=1}^{m}$.
3: Calculate the mini-batch loss value by using formula (11).
4: Update the model parameters by using formula (12), (13) and (14).
5: end while
Output: Trained model

III. THE EXPERIMENTAL SETTING AND RESULTS ANALYSIS

A. DATA SETS

To evaluate the classification performance of the presented network, experiments are conducted on three representative HSI data sets: Indian pines data set(IN), The University of Pavia data set(UP), and Salinas data set.

IN: It is a HSI data set of the Indian Pines test site, which captured through the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) in 1992. It contains 16 ground-truth classes and 220 spectral channels, and the wavelengths range from $0.2\mu m$ to $2.4\mu m$. Its spatial size is 145×145, and the resolution is 20 m per pixel.

UP: It is a HSI data set of Northern Italy, which captured through the Reflective Optics System Imaging Spectrometer (ROSIS). It contains 9 ground-truth classes. Its spatial size is 610×340 and the resolution is 1.3m per pixel. After removing absorption bands, there remain 103 bands and the wavelengths range from $0.43\mu m$ to $0.86\mu m$.

Salinas: It is a HSI data set of Salinas Valley, which captured through the AVIRIS. Its spatial size is 512×217 and the resolution is 3.7m per pixel. It contains 224 spectral channels and 16 ground-truth classes.

In the IN data set, since there are only 10249 labeled samples, 20% of the labeled samples are randomly selected for training, and the rest 10% for validation, 70% for testing. As to UP and the Salinas data set, since the labeled samples are more than 40 thousand, 10% of the labeled samples are randomly selected for training, and the rest 10% for validation, 80% for testing. Tables 1-3 list the amounts of labeled samples for training, validation, and testing on three data sets.

B. EXPERIMENTAL SETTING

Different sizes of input patches contain different amounts of information, which may influence the classification performance. To test the impact of different input patch sizes, experiments were set on our model with different input cube sizes. Considering that the overall accuracy (OA) can reflect the overall classification performance of methods, the OA that defined in (15) is used as the evaluation index.

$$OA = \frac{\sum_{i=1}^{L} t_i}{\sum_{i=1}^{L} s_i}, \quad (15)$$

where $t_i, i = 1, 2, \ldots, L$ represents the amount of the test samples that correctly classified in the i-th class, $s_i, i = 1, 2, \ldots, L$ denotes the amount of the test samples in the i-th class. L denotes the number of land-cover categories.

TABLE 1. The assignment of train, VAL and test samples in the IN data set (20%, 10%, 70%).

No	Class	Train	Val	Test
1	Alfalfa	8	4	34
2	Corn-notill	260	130	1038
3	Corn-mintill	166	83	581
4	Corn	48	24	165
5	Grass-pasture	96	48	339
6	Grass-trees	140	70	520
7	Grass-pasture-mowed	6	3	19
8	Hay-windrowed	94	47	337
9	Oats	6	3	11
10	Soybean-notill	190	95	687
11	Soybean-mintill	478	239	1738
12	Soybean-clean	120	60	413
13	Wheat	42	21	142
14	Woods	250	125	890
15	Buildings-Grass-Trees-Drives	78	39	269
16	Stone-Steel-Towers	18	9	66
	Total	2000	1000	7249

TABLE 2. The assignment of train, VAL and test samples in the UP data set (10%, 10%, 80%).

	Class	Train	Val	Test
1	Asphalt	660	660	5311
2	Meadows	1800	1800	15049
3	Gravel	210	210	1679
4	Trees	307	307	2450
5	Painted metal sheets	135	135	1075
6	Bare Soil	500	500	4029
7	Bitumen	133	133	1064
8	Self-Blocking Bricks	360	360	2962
9	Shadows	95	95	757
	Total	4200	4200	34376

TABLE 3. The assignment of train, VAL and test samples in the Salinas data set (10%, 10%, 80%).

No	Class	Train	Val	Test
1	Brocoli_green_weeds_1	201	201	1607
2	Brocoli_green_weeds_1	373	373	2980
3	Fallow	198	198	1580
4	Fallow_rough_plow	140	140	1114
5	Fallow_smooth	268	268	2142
6	Stubble	396	396	3167
7	Celery	358	358	2863
8	Grapes_untrained	1112	1112	9047
9	Soil_vinyard_develop	620	620	4963
10	Corn_senesced_green_weeds	328	328	2622
11	Lettuce_romaine_4wk	107	107	854
12	Lettuce_romaine_5wk	193	193	1541
13	Lettuce_romaine_6wk	92	92	732
14	Lettuce_romaine_7wk	107	107	856
15	Vinyard_untrained	726	726	5816
16	Vinyard_vertical_trellis	181	181	1445
	Total	5400	5400	43329

TABLE 4. OA on IN, UP and Salinas with different patch sizes.

Patch size \ Data set	IN	UP	Salinas
3	97.84	98.59	98.26
5	99.25	99.56	99.36
7	99.68	99.92	99.86
9	99.72	99.93	99.88
11	99.74	99.94	99.90

The results of OA on different data sets with different patch sizes are shown in Table 4. According to Table 4, the larger the patch size is, the better the classification performance is. However, when the patch sizes are equal or larger than 7×7, the improvement of classification performance is negligible. In addition, the larger the patch size is, the more computation of the network is. After multiple comparing, we chose the input patch size of 7×7 for all data sets. Moreover, to make a fair comparison, the input size was fixed as 7×7 for all models.

In addition, since different data sets have different spectral channels, the different number of parallel branches were set for different data sets. There are 10 parallel branches in the proposed network for IN data set classification, and 8 parallel branches for UP, 14 parallel branches for Salinas, respectively. Furthermore, the adapted learning rate set is 3×10^-4.

C. RESULTS AND ANALYSIS

The proposed SRPNet was compared with the other four excellent performed models: BASS Net [32], SSRN [33], 3D-CNN[34], and SSUN[35]. Moreover, three self-compare experiments were conducted to access the reasonability of spatial residual blocks, parallel structure, and feature fusion. The three experiments were under the situation that the proposed network without feature fusion, without spatial residual blocks and without parallel structure, respectively. In the experiments of the network without feature fusion, the feature fusion on the spectral feature learning part was removed. In the experiments of the network without spatial residual blocks, all the skip connections in spatial residual blocks were

TABLE 5. OA, AA, and k of different models on the IN data set (20% train, 10% VAL, and 70% test).

	BASS Net [32]	SSRN [33]	3D-CNN [34]	SSUN [35]	Without feature fusion	Without residual blocks	Without parallel structure	SRPNet (ours)
OA(%)	98.44	99.25	95.97	99.65	99.38	98.59	99.41	**99.68**
AA(%)	96.02	99.39	96.61	99.42	99.13	98.05	99.56	**99.74**
k	0.9822	0.9914	0.9540	0.9960	0.9929	0.9840	0.9933	**0.9964**
1	50.00	**100.00**	96.77	**100.00**	91.18	97.06	**100.00**	**100.00**
2	97.78	99.90	95.70	**99.91**	99.04	97.50	98.75	99.62
3	98.62	99.31	95.38	99.85	98.79	98.97	**100.00**	**100.00**
4	99.39	97.08	96.43	**100.00**	**100.00**	97.58	**100.00**	**100.00**
5	**100.00**	99.56	96.29	**100.00**	99.41	96.76	99.12	97.64
6	**100.00**	99.42	98.05	**100.00**	99.62	99.81	99.23	**100.00**
7	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
8	**100.00**	99.39	98.20	**100.00**	**100.00**	99.41	99.70	**100.00**
9	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	90.91	**100.00**	**100.00**
10	96.65	99.12	96.46	99.61	**99.85**	97.23	**99.85**	99.71
11	98.79	98.61	95.34	**99.80**	99.25	99.25	99.66	99.66
12	99.27	**99.52**	90.26	99.16	**99.52**	**99.52**	98.31	99.27
13	**100.00**	**100.00**	97.87	98.18	**100.00**	**100.00**	**100.00**	**100.00**
14	99.10	99.89	97.87	99.90	99.44	99.66	99.44	**100.00**
15	96.65	99.64	95.83	96.85	**100.00**	95.17	98.89	**100.00**
16	**100.00**	**100.00**	95.31	97.40	**100.00**	**100.00**	**100.00**	**100.00**

removed. In the experiments of the network without parallel structure, only one spectral learning branch was set in the spatial feature learning part.

Three commonly used evaluation indicators are adopted to evaluate the classification performance, namely OA, average accuracy (AA) that defined in (17), and kappa (k) that defined in (19). AA demonstrates the model classification performance in every category. As for k, it reflects the consistency of classification results, and its values ranging from -1 to 1. The higher the value of k, the better the classification performance is.

$$a_i = \frac{t_i}{s_i}, \tag{16}$$

$$AA = \frac{\sum_{i=1}^{L} a_i}{L}, \tag{17}$$

$$p_e = \frac{\sum_{i=1}^{L} t_i \times s_i}{\sum_{i=1}^{L} s_i \times \sum_{i=1}^{L} s_i}, \tag{18}$$

$$k = \frac{OA - p_e}{1 - p_e}, \tag{19}$$

where $t_i, i = 1, 2, \ldots, L$ represents the amount of the test samples that correctly classified in the i-th class, $s_i, i = 1, 2, \ldots, L$ denotes the amount of the test samples in the i-th class. L denotes the number of land-cover categories.

The experiment results are shown in Tables 5 to 7, and the best result of every row is marked in bold. It can be seen that, in all three experiments, the proposed SRPNet achieved the best classification performance with the highest value of OA, AA, and k. In the three self-compare experiments, the classification results of the network without feature fusion, without residual and without parallel structure are worse than the SRPNet. These results show that the SRPNet achieves the best performance only when all residual blocks, parallel branches, and feature fusion are set.

Figures 4-6 show the visualized results of different methods in IN, UP, and Salinas data set, respectively. Since all the methods perform well, there are only little differences in Figures 4-6. In order to better distinguish the difference between the compared methods and ours, white boxes are utilized to

TABLE 6. OA, AA, and k of differet models on the UP data set (10% train, 10% VAL, and 80% test).

	BASS Net [32]	SSRN [33]	3D-CNN [34]	SSUN [35]	Without feature fusion	Without residual blocks	Without parallel structure	SRPNet (ours)
OA(%)	99.05	99.79	98.87	99.79	99.55	98.91	99.65	**99.92**
AA(%)	98.51	99.66	99.03	99.59	99.31	98.13	99.57	**99.87**
k	0.9847	0.9972	0.9850	0.9972	0.9947	0.9856	0.9953	**0.9989**
1	98.67	**99.92**	98.61	99.87	99.39	98.62	98.53	99.81
2	99.79	99.96	98.89	99.91	99.97	99.91	99.99	**100.00**
3	95.00	98.46	99.00	99.10	98.41	88.29	**99.94**	99.52
4	99.28	99.69	99.75	99.53	98.16	98.50	99.51	**99.88**
5	**100.00**	99.99	**100.00**	99.92	**100.00**	**100.00**	**100.00**	**100.00**
6	99.74	99.94	99.39	99.91	99.87	99.88	99.65	**100.00**
7	97.14	99.82	99.13	98.35	99.68	99.46	**100.00**	**100.00**
8	97.37	99.22	97.16	**99.94**	98.96	98.67	99.83	99.76
9	99.58	**99.95**	99.34	99.76	99.38	99.87	98.68	99.87

TABLE 7. OA, AA, and k of different models on the Salinas data set (10% train, 10% VAL, and 80% test).

	BASS Net [32]	SSRN [33]	3D-CNN [34]	SSUN [35]	Without feature fusion	Without residual blocks	Without parallel structure	SRPNet (ours)
OA(%)	97.92	99.67	98.94	99.73	99.36	98.64	99.53	**99.86**
AA(%)	99.24	99.80	99.40	99.83	99.28	99.43	99.72	**99.91**
k	0.9768	0.9963	0.9882	0.9973	0.9929	0.9849	0.9947	**0.9984**
1	**100.00**	**100.00**	**100.00**	**100.00**	93.42	**100.00**	**100.00**	**100.00**
2	99.97	99.92	99.97	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
3	99.87	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
4	99.90	99.90	99.64	99.84	**100.00**	**100.00**	99.73	99.91
5	99.87	**100.00**	99.62	99.96	**100.00**	99.91	99.95	99.77
6	**100.00**	99.91	**100.00**	**100.00**	99.65	99.94	**100.00**	**100.00**
7	99.93	**100.00**	**100.00**	**100.00**	**100.00**	99.96	**100.00**	**100.00**
8	99.62	98.90	97.88	99.26	**100.00**	95.17	99.27	99.83
9	99.84	**100.00**	**100.00**	**100.00**	99.05	99.98	**100.00**	**100.00**
10	97.78	99.87	99.28	**100.00**	**100.00**	97.88	99.96	99.85
11	99.85	**100.00**	**100.00**	98.97	98.96	**100.00**	99.65	99.78
12	**100.00**	99.61	99.80	**100.00**	99.85	**100.00**	**100.00**	**100.00**
13	99.61	**100.00**	99.73	99.64	**100.00**	99.81	**100.00**	**100.00**
14	**100.00**	**100.00**	98.15	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
15	91.71	99.55	96.33	99.58	**100.00**	98.98	97.99	99.43
16	99.93	**100.00**	99.93	**100.00**	97.53	99.29	99.03	99.86

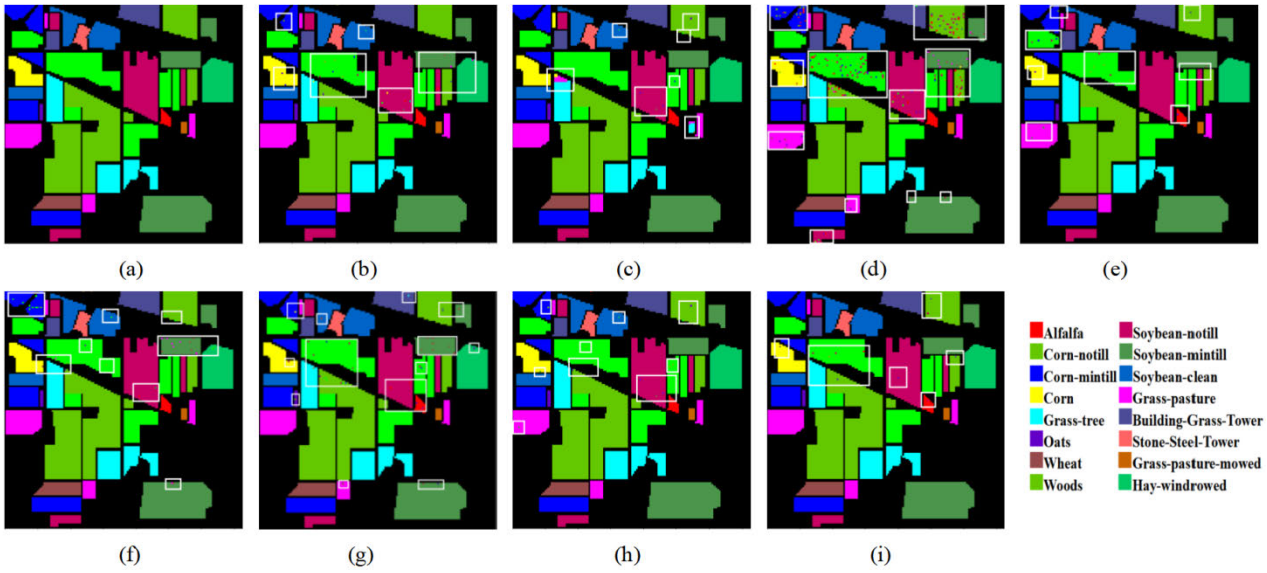

FIGURE 4. Visualized results of different methods for IN (a) Ground truth labels. (b)–(i) Visualized results of BASS Net [32], SSRN [33], 3D-CNN [34], SSUN [35], without feature fusion, without residual blocks, without parallel branches and SRPNet (20% training, 10% validation, and 70% testing).

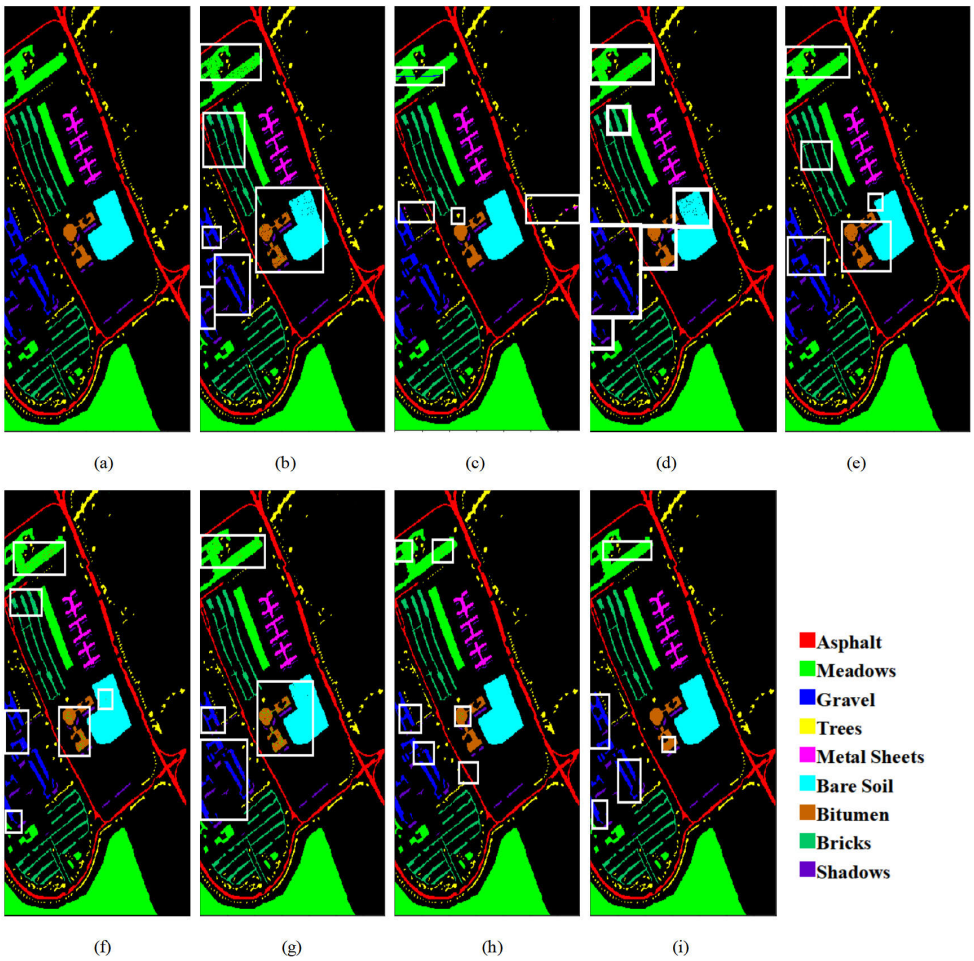

FIGURE 5. Visualized results of different methods for UP. (a) Ground truth labels. (b)–(i) Visualized results of BASS Net [32], SSRN [33], 3D-CNN [34], SSUN [35], without feature fusion, without residual blocks, without parallel branches and SRPNet (10% training, 10% validation, and 80% testing).

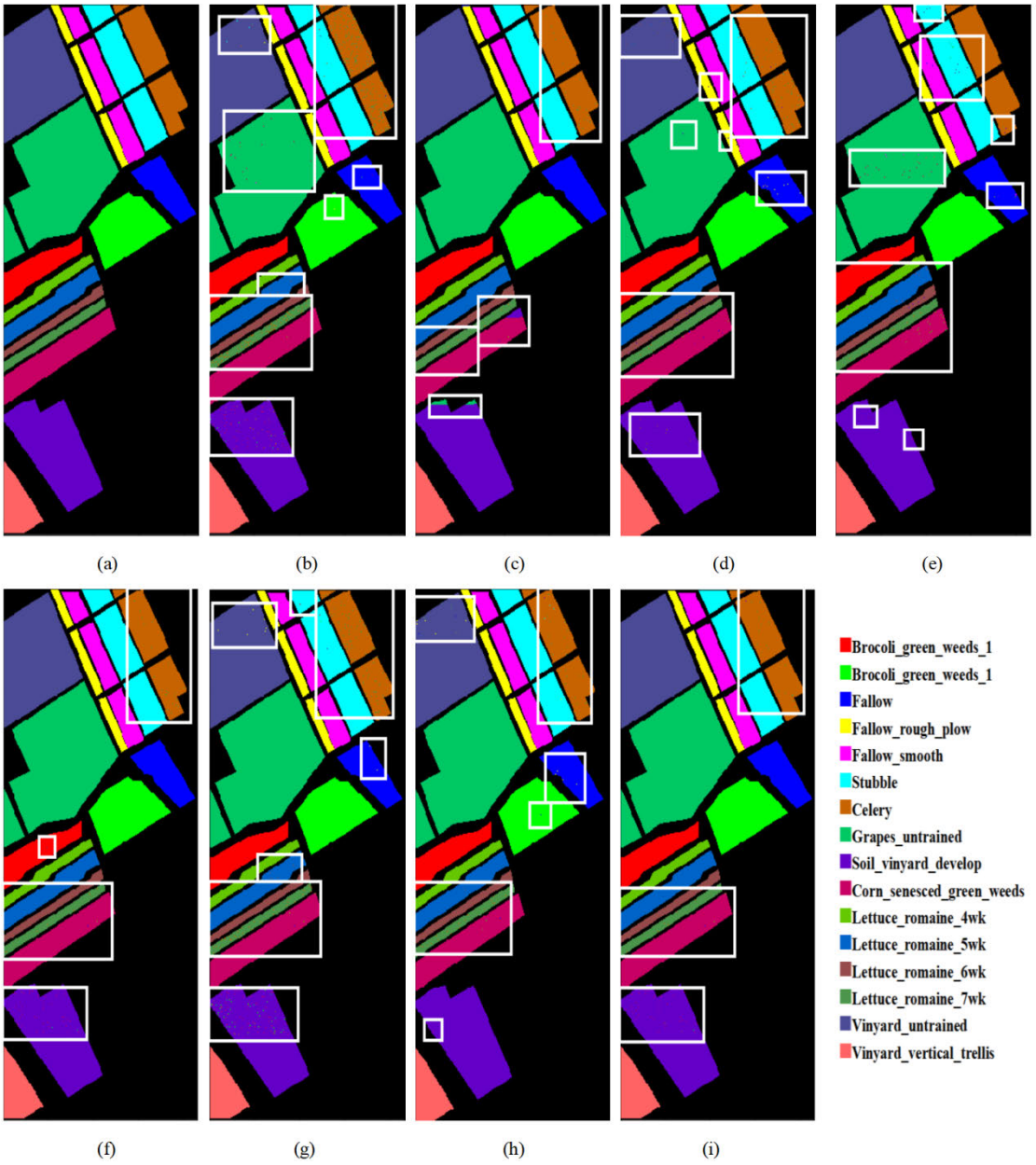

FIGURE 6. Visualized results of different methods for the Salinas. (a) Ground truth labels. (b)–(i) Visualized results of BASS Net [32], SSRN [33], 3D-CNN [34], SSUN [35], without feature fusion, without residual blocks, without parallel branches and SRPNet (10% for training, 10% for validation, and 80% for testing).

illustrate where the pixels were misclassified. It can be seen from the Figures 4-6, the samples misclassified by SRPNet are the fewest.

As shown in Tables 8 to 10, the more the training samples we have, the better the classification performance is. Especially for the UP and Salinas data sets which contain more labeled samples than the IN data set, the classification accuracy is close to 100% when the ratio of training samples is 20%. In addition, it is worth pointing that when the proportion of training samples is 20%, the classification accuracy of SSRN and SSUN on the Salinas data set is slightly higher than ours. Actually, it is predictable. With the increase of training samples, deeper networks usually show better performance. However, when the proportion of training samples increased from 10% to 20%, there is little improvement in classification accuracy. However, for HSI, the acquisition of labeled samples is expensive and time-consuming, which puts forward requirements for HSI classification network with fewer labeled training samples. Therefore, there is no need to pay a high cost for little improvement. That's why we chose 10% training samples as the primary comparison experiment on UP and Salinas data sets. Through those tables, it can be

TABLE 8. OA on IN data set with different training data.

Training data ratio \ Network	BASS Net [32]	SSRN [33]	3D-CNN [34]	SSUN [35]	ours
3%	83.58	91.78	73.32	89.46	**92.57**
5%	89.80	95.22	81.66	93.96	**96.22**
10%	96.25	98.49	87.74	98.45	**98.58**
20%	98.44	99.25	95.97	99.65	**99.68**

TABLE 9. OA on UP data set with different training data.

Training data ratio \ Network	BASS Net [32]	SSRN [33]	3D-CNN [34]	SSUN [35]	ours
3%	96.89	99.12	96.13	98.45	**99.46**
5%	98.19	99.63	97.39	99.44	**99.67**
10%	99.05	99.79	98.87	99.79	**99.92**
20%	99.49	**99.97**	99.15	**99.97**	**99.97**

TABLE 10. OA on Salinas data set with different training data.

Training data ratio \ Network	BASS Net [32]	SSRN [33]	3D-CNN [34]	SSUN [35]	ours
3%	95.45	98.15	95.80	98.24	**98.92**
5%	97.64	99.00	97.24	99.05	**99.32**
10%	98.30	99.67	98.94	99.76	**99.86**
20%	99.02	**99.98**	99.63	**99.98**	99.94

seen that the proposed SRPNet can obtain better performance under different training sample proportions. This demonstrates that our model is more stable and has competitive performance, especially on the tasks of small training sample numbers.

IV. CONCLUSION

In this paper, a novel spatial residual blocks combined parallel network is proposed for HSI classification. Spatial residual blocks are presented to mitigate the decreasing-accuracy phenomenon. In the spatial feature learning part, there are a larger number of channels, which can extract more specific features. In the spectral feature learning part, the parallel branches with the same structure and shared parameters are proposed to decrease the trainable parameters and optimize the computation. Furthermore, Feature fusion is utilized to integrate multi-scale features of different layers to gain richer information for improving classification performance. Extensive experiments on three representative HSI data sets demonstrate that the proposed SRPNet is more stable and has the competitive performance.

REFERENCES

[1] L. Zhang, L. Zhang, D. Tao, X. Huang, and B. Du, "Hyperspectral remote sensing image subpixel target detection based on supervised metric learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 4955–4965, Aug. 2014.

[2] 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.

[3] J. Li, X. Huang, P. Gamba, J. M. Bioucas-Dias, L. Zhang, J. A. Benediktsson, and A. Plaza, "Multiple feature learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1592–1606, Mar. 2015.

[4] C. Chen, W. Li, H. Su, and K. Liu, "Spectral-spatial classification of hyperspectral image based on kernel extreme learning machine," *Remote Sens.*, vol. 6, no. 6, pp. 5795–5814, 2014.

[5] L. Liang, L. Di, L. Zhang, M. Deng, Z. Qin, S. Zhao, and H. Lin, "Estimation of crop LAI using hyperspectral vegetation indices and a hybrid inversion method," *Remote Sens. Environ.*, vol. 165, pp. 123–134, Aug. 2015.

[6] M. T. Eismann and R. C. Hardie, "Application of the stochastic mixing model to hyperspectral resolution enhancement," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 9, pp. 1924–1933, Sep. 2004.

[7] X. Huang and L. Zhang, "An adaptive mean-shift analysis approach for object extraction and classification from urban hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 12, pp. 4173–4185, Dec. 2008.

[8] P. Zhong, Z. Gong, S. Li, and C.-B. Schonlieb, "Learning to diversify deep belief networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3516–3530, Jun. 2017.

[9] J. Li, P. R. Marpu, A. Plaza, J. M. Bioucas-Dias, and J. A. Benediktsson, "Generalized composite kernel framework for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 9, pp. 4816–4829, Sep. 2013.

[10] J. Zabalza, J. Ren, J. Zheng, H. Zhao, C. Qing, Z. Yang, P. Du, and S. Marshall, "Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imaging," *Neurocomputing*, vol. 185, pp. 1–10, Apr. 2016.

[11] R. Kemker and C. Kanan, "Self-taught feature learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2693–2705, May 2017.

[12] F. Cao, Z. Yang, J. Ren, W.-K. Ling, H. Zhao, and S. Marshall, "Extreme sparse multinomial logistic regression: A fast and robust framework for hyperspectral image classification," *Remote Sens.*, vol. 9, no. 12, p. 1255, 1255.

[13] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 6, pp. 1351–1362, Jun. 2005.

[14] F. Cao, Z. Yang, J. Ren, W.-K. Ling, H. Zhao, M. Sun, and J. A. Benediktsson, "Sparse representation-based augmented multinomial logistic extreme learning machine with weighted composite features for spectral-spatial classification of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6263–6279, Nov. 2018.

[15] T. Qiao, Z. Yang, J. Ren, P. Yuen, H. Zhao, G. Sun, S. Marshall, and J. A. Benediktsson, "Joint bilateral filtering and spectral similarity-based sparse representation: A generic framework for effective feature extraction and data classification in hyperspectral imaging," *Pattern Recognit.*, vol. 77, pp. 316–328, May 2018.

[16] L. Fang, N. He, S. Li, A. J. Plaza, and J. Plaza, "A new spatial-spectral feature extraction method for hyperspectral images using local covariance matrix representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3534–3546, Jun. 2018.

[17] Q. Liu, F. Zhou, R. Hang, and X. Yuan, "Bidirectional-convolutional LSTM based spectral-spatial feature learning for hyperspectral image classification," *Remote Sens.*, vol. 9, no. 12, p. 1330, 1330.

[18] F. Tong, H. Tong, J. Jiang, and Y. Zhang, "Multiscale union regions adaptive sparse representation for hyperspectral image classification," *Remote Sens.*, vol. 9, no. 9, p. 872, 2017.

[19] Y. Liu and Y. Wang, "Classification of hyperspectral image based on K-means and structured sparse coding," in *Proc. 3rd Int. Conf. Inf. Sci. Control Eng. (ICISCE)*, Jul. 2016, pp. 248–251.
[20] W. Li, Q. Du, F. Zhang, and W. Hu, "Collaborative-representation-based nearest neighbor classifier for hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 2, pp. 389–393, Feb. 2015.
[21] B.-C. Kuo, J.-M. Yang, T.-W. Sheu, and S.-W. Yang, "Kernel-based KNN and Gaussian classifiers for hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2008, pp. 1006–1008.
[22] L. Gao, J. Li, M. Khodadadzadeh, A. Plaza, B. Zhang, Z. He, and H. Yan, "Subspace-based support vector machines for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 2, pp. 349–353, Feb. 2015.
[23] J. A. Gualtieri and S. Chettri, "Support vector machines for classification of hyperspectral data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2000, pp. 813–815.
[24] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep learning based feature selection for remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 11, pp. 2321–2325, Nov. 2015.
[25] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[26] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 844–853, Feb. 2017.
[27] 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.
[28] 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.
[29] C. Qing, J. Ruan, X. Xu, J. Ren, and J. Zabalza, "Spatial-spectral classification of hyperspectral images: A deep learning framework with Markov random fields based modelling," *IET Image Process.*, vol. 13, no. 2, pp. 235–245, Feb. 2019.
[30] J. Yang, Y.-Q. Zhao, and J. C.-W. Chan, "Learning and transferring deep joint spectral-spatial features for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4729–4742, Aug. 2017.
[31] Y. Li, H. Zhang, and Q. Shen, "Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sens.*, vol. 9, no. 1, p. 67, 2017.
[32] A. Santara, K. Mani, P. Hatwar, A. Singh, A. Garg, K. Padia, and P. Mitra, "BASS Net: Band-adaptive spectral-spatial feature learning neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5293–5301, Sep. 2017.
[33] 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. 2018.
[34] 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.
[35] 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, May 2018.

BUYI ZHANG (Member, IEEE) received the B.E. degree from the Nanjing University of Posts and Telecommunications. She is currently pursuing the master's degree in signal and information processing with the South China University of Technology. Her research interests include machine learning, computer vision, deep learning, and their application on hyperspectral image classification.

CHUNMEI QING (Member, IEEE) received the B.Sc. degree in information and computation science from Sun Yat-sen University, China, in 2003, and the Ph.D. degree in electronic imaging and media communications from the University of Bradford, U.K., in 2009. Then, she worked as a Postdoctoral Researcher with the University of Lincoln, U.K. Since 2013, she has been an Associate Professor with the School of Electronic and Information Engineering, South China University of Technology (SCUT), Guangzhou, China. Her main research interests include image/video processing, computer vision, pattern recognition, and machine learning.

XIANGMIN XU (Senior Member, IEEE) received the Ph.D. degree from the South China University of Technology, Guangzhou, China. He is currently a Full Professor at the School of Electronic and Information Engineering, South China University of Technology, Guangzhou. His current research interests include image/video processing, human–computer interaction, computer vision, and machine learning.

JINCHANG REN (Senior Member, IEEE) received the B.Eng. degree in computer software, the M.Eng. degree in image processing, and the D.Eng. degree in computer vision from Northwestern Polytechnical University (NWPU), China, and the Ph.D. degree in electronic imaging and media communication from Bradford University, U.K. He is currently with the Department of Electronic and Electrical Engineering, University of Strathclyde. His research interests focus mainly on visual computing and multimedia signal processing, especially on semantic content extraction for video analysis and understanding and hyperspectral imaging.