Spectral–Spatial Residual Network for Hyperspectral Image Classification: A 3-D Deep Learning Framework

Zilong Zhong, *Student Member, IEEE*, Jonathan Li, *Senior Member, IEEE*, Zhiming Luo, and Michael Chapman

***Abstract*— In this paper, we designed an end-to-end spectral–spatial residual network (SSRN) that takes raw 3-D cubes as input data without feature engineering for hyperspectral image classification. In this network, the spectral and spatial residual blocks consecutively learn discriminative features from abundant spectral signatures and spatial contexts in hyperspectral imagery (HSI). The proposed SSRN is a supervised deep learning framework that alleviates the declining-accuracy phenomenon of other deep learning models. Specifically, the residual blocks connect every other 3-D convolutional layer through identity mapping, which facilitates the backpropagation of gradients. Furthermore, we impose batch normalization on every convolutional layer to regularize the learning process and improve the classification performance of trained models. Quantitative and qualitative results demonstrate that the SSRN achieved the state-of-the-art HSI classification accuracy in agricultural, rural–urban, and urban data sets: Indian Pines, Kennedy Space Center, and University of Pavia.**

***Index Terms*— 3-D deep learning, hyperspectral image classification, spectral–spatial feature extraction, spectral–spatial residual network (SSRN).**

I. Introduction

CLASSIFYING every pixel with a certain land-cover type is the cornerstone of hyperspectral image analysis, which spans a broad range of applications, including image segmentation, object recognition, land-cover mapping, and anomaly detection [1]–[4]. Two major characteristics of hyperspectral imagery (HSI) should be taken into account to obtain discriminative features for HSI classification. First, abundant spectral information, which derives from hundreds of contiguous spectral bands, makes the accurate identification of corresponding ground materials possible [5]. Second, high spatial correlation, which originates from homogeneous areas in HSI, provides complementary information to spectral features for precise mapping [6].

To take advantage of abundant spectral bands, traditional pixelwise HSI classification models mainly concentrate on two steps: feature engineering and classifier training. Feature engineering methods include feature selection (band selection) and feature extraction [7]. The main objectives of feature engineering are to reduce the high dimensionality of HSI pixels and extract the most discriminative features or bands. Next, general-purpose classifiers are trained using the discriminative features obtained from the feature engineering step. Feature extraction approaches usually learn representative features through nonlinear transformation. For example, [8] integrated multiple features derived from different kinds of dimensionality reduction methods to train support vector machine (SVM) classifiers. Unlike feature extraction, feature selection methods try to find the most representative features from raw HSIs without transforming them to retain their physical meaning. For instance, [9] adopted manifold ranking as an unsupervised feature selection method, which chooses the most representative bands for training the classifiers that follow. Moreover, a multitask joint sparse representation-based method [10] integrated band selection method with a smooth prior imposed by the Markov random field. These two band selection-based paradigms used spectral bands from all available pixels for feature selection and can be interpreted as semisupervised learning methods.

On the other hand, there are two ways to incorporate spatial information for HSI classification: spatialized input and postprocessing. The spatialized input methods impose feature engineering step on 3-D cubes obtained from HSI. Many papers suggested that methods expanding input data with more spatial information can improve the classification performance [11], [12]. Among these methods, SVMs are the most commonly used classifiers for HSI classification, because SVMs perform robustly with high-dimensional input data [13], [14]. For example, [15] employed a region-based kernel to extract spectral–spatial features on which the learned SVM classifier identifies the categories of hyperspectral pixels. In contrast, the postprocessing approaches have taken the prior knowledge of smoothness into considera-

Manuscript received March 14, 2017; revised May 28, 2017 and July 25, 2017; accepted August 20, 2017. This work was supported by the China Scholarship Council. *(Corresponding author: Jonathan Li.)*

Z. Zhong is with the Department of Geography and Environmental Management, University of Waterloo, ON N2L 3G1, Canada (e-mail: z26zhong@uwaterloo.ca).

J. Li is with the Department of Geography and Environmental Management, University of Waterloo, ON N2L 3G1, Canada, and also with the Fujian Key Laboratory of Sensing and Computing for Smart City, School of Information Science and Engineering, Xiamen University, Xiamen 361005, China (e-mail: junli@xmu.edu.cn).

Z. Luo is with the Department of Cognitive Science, Xiamen University, Xiamen 361005, China, and also with the Department of Computer Science, University of Sherbrooke, Sherbrooke, QC J1K 2R1, Canada.

M. Chapman is with the Department of Civil Engineering, Ryerson University, Toronto, ON M5B 2K3, Canada.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2017.2755542

tion that neighboring pixels with similar spectral information are likely to belong to the same land-cover categories. For instance, [16] incorporated a probabilistic graphical model as the postprocessing step to improve the classification outcomes of kernel SVMs. Although many works use typical classification frameworks, which are composed of feature extractors followed by trainable classifiers, they suffer from two drawbacks. First, the feature engineering step normally does not generalize well to other scenarios. Second, the de facto one-layer nonlinear transformation (e.g., kernel methods) being applied before the linear classifiers has limited representation capacity to fully utilize the abundant spectral and spatial features.

In the face of these shortcomings of feature engineering-based frameworks, supervised deep learning models have attracted increased attention, due to the fact that the objective functions of deep learning models directly focus on classification in lieu of two independent steps. The fundamental philosophy of deep learning is to let the trained model itself decide which features are more important than other features with fewer constraints imposed by human beings. In other words, deep learning frameworks simultaneously learn feature representation and corresponding classifiers through the training process. Furthermore, multilayer neural networks can extract robust and discriminative features of HSI and outperform SVMs [17], [18]. For example, the stacked autoencoders (SAEs) were used as feature extractors to capture the representative stacked spectral and spatial features with a greedy layerwise pretraining strategy [17]. Similarly, the potential of deep belief networks for HSI classification was explored in [18]. However, both models suffer the same problem of spatial information loss, which is caused by the requirement for 1-D input data.

Recently, convolutional neural networks (CNNs) and their extensions have obtained unprecedented advances in computer vision tasks [19], [20]. Multiple papers have demonstrated that CNNs can deliver the state-of-the-art results using spatialized input for HSI classification [21]–[23]. For example, [23] used CNNs to extract spatial features, which were integrated with spectral features that learned from balanced local discriminant embedding, for HSI classification. However, the input of the CNN models are the three principal components of original HSIs, which means that the spatial feature extraction process still loses some spectral–spatial information. A CNN-based feature extractor was proposed in [21], which can learn discriminative representations from pixel pairs and use a voting strategy to smooth final classification maps. In addition, 3-D CNNs were adopted to extract deep spectral–spatial features directly from raw HSIs and delivered promising classification outcomes [22]. Similarly, [24] further studied 3-D CNNs for spectral–spatial classification using input cubes of HSIs with a smaller spatial size. These models generate thematic maps using an approach that can directly process raw HSIs, whereas the classification accuracy of the CNN models decreases when the network becomes deeper.

To resolve this problem, inspired by [25], we proposed a supervised spectral–spatial residual network (SSRN)[1] with consecutive learning blocks that takes the characteristics of HSI into account. The designed spectral and spatial residual blocks extract discriminative spectral–spatial features from HSI cubes and can be regarded as an extension of convolutional layers in CNNs. The SSRN has a deeper structure than those of 3-D CNNs used in [21]–[24], and contains shortcut connections between every other convolutional layer. Hence, the SSRN can learn robust spectral–spatial representations from original HSIs. Similar to the SSRN, [26] incorporated residual learning with fully convolutional layers to form a contextual CNN. However, this method fails to distinguish spectral features and spatial features. Thus, this paper investigates the effectiveness of two types of residual architecture toward the spectral–spatial feature learning for HSI classification and their robustness in different scenarios.

Compared with a large number of annotated data in computer vision and pattern recognition communities, which play a significant role in the unprecedented success achieved by deep learning models [20], the available amount of training and testing samples in the widely studied HSI data sets is relatively small. Moreover, the unbalanced amounts of differently labeled samples undermine the accuracy of HSI classification. In addition, the input data of SSRN are 3-D cubes of raw HSI, and the multidimensional input data bring more challenges. Therefore, this paper aims to study the generalization ability of the SSRN on HSI data sets with large and small training sizes, high and medium spatial resolution, and various land-cover types with uneven samples for different categories.

The four major contributions of this paper are listed as follows.

1) The designed SSRN adopts residual connections to mitigate the decreasing-accuracy phenomenon and improve the HSI classification accuracy.
2) Two consecutive residual blocks learn spectral and spatial representations separately, through which more discriminative features can be extracted.
3) This paper validates the effectiveness of batch normalization (BN) as a regularization method to improve classification outcomes using unbalanced training samples.
4) The uniform architecture design makes the SSRN a framework that generalizes well in three commonly studied HSI data sets. More importantly, the SSRN achieves the state-of-the-art classification accuracy using limited training data with a fixed spatial size.

The rest of this paper is organized as follows. Section II describes two types of residual block and introduces the detailed architecture of SSRN. The network configuration and experimental results are reported, and some discussions are offered in Section III. Some conclusions are drawn in Section IV.

II. Proposed Framework

Fig. 1 shows the whole deep learning framework of HSI classification based on the SSRN. In this framework, all available annotated data are separated into three groups: training, validation, and testing groups for each data set. Suppose the HSI data set $\boldsymbol{X}$ contains N labeled pixels

[1] https://github.com/zilongzhong/SSRN

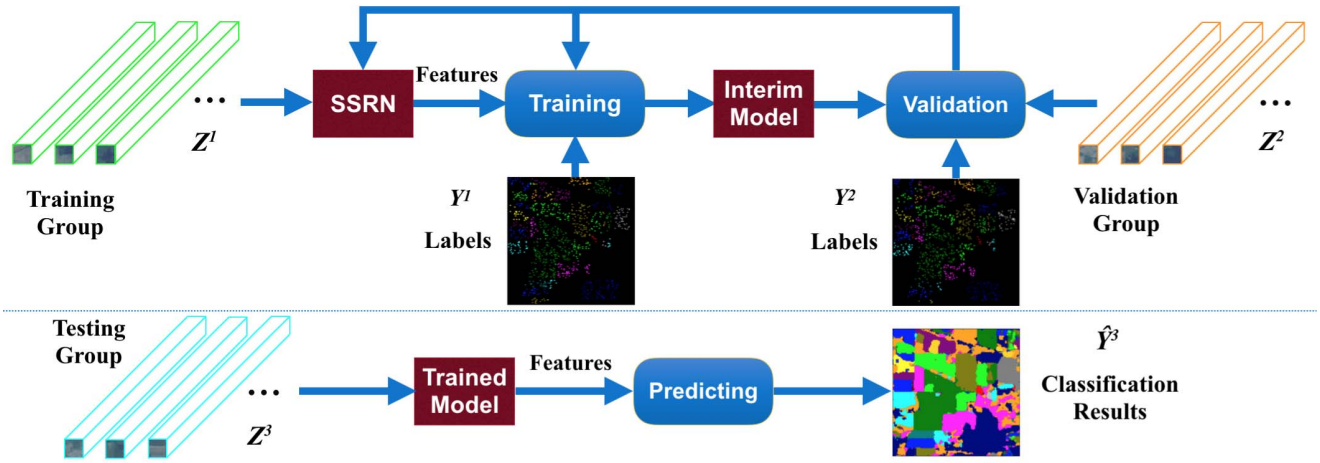


Fig. 1. SSRN-based framework for HSI classification. (Top) Training group $\boldsymbol{Z^1}$ and their corresponding labels are used for updating the parameters of network. The validation group $\boldsymbol{Z^2}$ and their corresponding labels $\boldsymbol{Y^2}$ are used for monitoring the interim models generated in the training stage. (Bottom) Testing group $\boldsymbol{Z^3}$ is employed for assessing the optimal trained network.

$\{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_N}\} \in \mathbb{R}^{1\times1\times b}$ and $\boldsymbol{Y} = \{\boldsymbol{y_1}, \boldsymbol{y_2}, \ldots, \boldsymbol{y_N}\} \in \mathbb{R}^{1\times1\times L}$ is the set of corresponding one-hot label vectors, where b and L represent the numbers of spectral bands and land-cover categories, respectively. Neighboring cubes centered at pixels in $\boldsymbol{X}$ form a new group of data set $\boldsymbol{Z} = \{\boldsymbol{z_1}, \boldsymbol{z_2}, \ldots, \boldsymbol{z_N}\} \in \mathbb{R}^{w\times w\times b}$. To fully utilize the spectral and spatial information provided by HSIs, the proposed networks take cubes of size $w \times w \times b$ from raw data as input, where is the short width of 3-D cubes in training group $\boldsymbol{Z^1}$, validation group $\boldsymbol{Z^2}$, and testing group $\boldsymbol{Z^3}$ in Fig. 1. Their corresponding label vector sets are $\boldsymbol{Y^1}$, $\boldsymbol{Y^2}$, and $\boldsymbol{Y^3}$. For example, the size of HSI cubes for the Indian Pines (IN) data set is $7 \times 7 \times 200$. Therefore, the objective of the training process is to update the parameters of the SSRN until the model can make high-accuracy predictions $\hat{\boldsymbol{Y}}^{\boldsymbol{3}}$ with regard to the ground-truth labels $\boldsymbol{Y^3}$ given the neighboring cubes $\boldsymbol{Z^3}$.

After the architecture of deep learning models is built and the hyperparameters for training are configured, the models are trained for hundreds of epochs using the training group $\boldsymbol{Z^1}$ and their ground-truth label vector set $\boldsymbol{Y^1}$. In this process, the parameters of the SSRN are updated through backpropagating the gradients of the cross-entropy objective function in (1), which represents the difference between predicted label vector $\hat{\boldsymbol{y}} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_L]$ and ground-truth label vector $\boldsymbol{y} = [y_1, y_2, \ldots, y_L]$

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

The validation group $\boldsymbol{Z^2}$ is used for monitoring the training process by measuring the classification performance of interim models, which are intermediate networks generated during the training stage, to select the network with the highest classification accuracy. Finally, the testing group $\boldsymbol{Z^3}$ is employed for assessing the generalizability of the trained SSRN through calculating classification metrics and visualizing thematic maps.

A. Three-Dimensional Convolutional Layer With Batch Normalization

Deep learning models consist of multiple layers of nonlinear neurons that can learn hierarchical representations through a large number of labeled images [19]. CNNs have achieved or surpassed human-level intelligence in several perception tasks [20], [27], because convolutional layers enable CNNs to learn more discriminative features with sparsity constraint.

In this paper, 3-D convolutional layers are adopted as the basic element of the SSRN. In addition, BN [28] is conducted at every convolutional layer in SSRN. This strategy makes the training processing of deep learning models more efficient. As shown in Fig. 2, if the $(k+1)$th 3-D convolutional layer has n^k input feature cubes of size $w^k \times w^k \times d^k$, a convolutional filter bank that contains n^{k+1} convolutional filters of size $a^{k+1} \times a^{k+1} \times m^{k+1}$, and the subsampling strides of (s_1, s_1, s_2) for the convolutional operation, then this layer generates n^{k+1} output feature cubes of size $w^{k+1} \times w^{k+1} \times d^{k+1}$, where the spatial width $w^{k+1} = \lfloor 1 + (w^k - a^{k+1})/s_1 \rfloor$ and the spectral depth $d^{k+1} = \lfloor 1 + (d^k - m^{k+1})/s_2 \rfloor$. The ith output of $(k+1)$th 3-D convolutional layer with BN (CONVBN) can be formulated as

$$\boldsymbol{X_i^{k+1}} = R\left(\sum_{j=1}^{n^k} \hat{\boldsymbol{X}}_j^k * \boldsymbol{H_i^{k+1}} + b_i^{k+1} \right) \tag{2}$$

$$\hat{\boldsymbol{X}}^k = \frac{\boldsymbol{X}^k - E(\boldsymbol{X}^k)}{\mathrm{Var}(\boldsymbol{X}^k)} \tag{3}$$

where $\boldsymbol{X}_j^k \in \mathbb{R}^{w\times w\times d}$ is the jth input feature tensor of the $(k+1)$th layer, $\hat{\boldsymbol{X}}^k$ is the normalization result of batch feature cubes $\boldsymbol{X}^k$ in the kth layer, and $E(\cdot)$ and $\mathrm{Var}(\cdot)$ represent the expectation and variance function of the input feature tensor, respectively. $\boldsymbol{H}_i^{k+1}$ and b_i^{k+1} denote the parameters and bias of the ith convolutional filter bank in the $(k+1)$th layer, $*$ represents the 3-D convolutional operation, and $R(\cdot)$ is the rectified linear unit activation function that sets elements with negative numbers to zero.

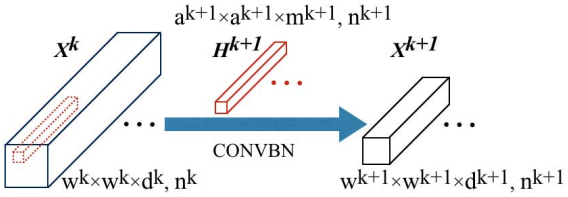


Fig. 2. Three-dimensional CONVBN. The $(k+1)$th layer conducts a 3-D convolution of input feature cubes $\boldsymbol{X}^k$ and a convolutional filter bank $\boldsymbol{H}^{k+1}$ and generates output feature cubes $\boldsymbol{X}^{k+1}$.

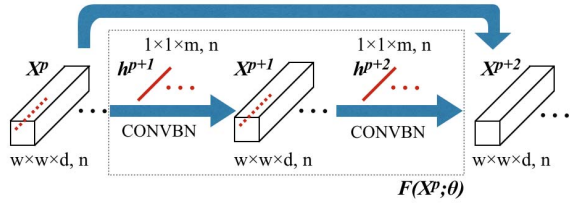


Fig. 3. Spectral residual block for spectral feature learning. This block includes two successive 3-D convolutional layers, and a skip connection directly adds input feature cubes $\boldsymbol{X}^p$ to output feature cubes $\boldsymbol{X}^{p+2}$.

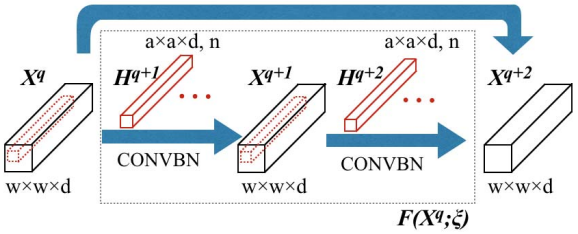


Fig. 4. Spatial residual block for spatial feature learning. This block includes two successive 3-D convolutional layers, and a skip connection directly adds input feature cubes $\boldsymbol{X}^q$ to the output feature cubes $\boldsymbol{X}^{q+2}$.

B. Spectral and Spatial Residual Blocks

Although CNN models have been used for HSI classification and achieved the state-of-the-art results, it is counterintuitive that, after several layers, the classification accuracy decreases with the increase of convolutional layers [22]. This phenomenon stems from the fact that the representation capacity of CNNs is too high compared with the relative small number of training samples with the same regularization settings. However, this decreasing-accuracy issue can be alleviated by adding shortcut connections between every other layer to build residual blocks [25]. To this end, we designed two residual blocks in a general architecture to consecutively extract spectral and spatial features from raw 3-D HSI cubes, owing to the high spectral resolution and high spatial correlation of HSI. As shown in Fig. 3, a residual block can be regarded as an extension of two convolutional layers. This architecture enables gradients in higher layers rapidly propagate back to the lower layers, thereby facilitating and regularizing the model training process.

In the spectral residual blocks, as shown in Fig. 3, convolutional kernels of size $1\times 1\times m$ are used in successive filter banks $\boldsymbol{h}^{p+1}$ and $\boldsymbol{h}^{p+2}$ for pth and $(p+1)$th layers, respectively. At the same time, the spatial size of 3-D feature cubes $\boldsymbol{X}^{p+1}$ and $\boldsymbol{X}^{p+2}$ is kept at $w\times w$ unchanged through a padding strategy, which means that the output feature cubes copy the values from the border area to the padding area after convolutional operation in the spectral dimension. Then, these two convolutional layers build a residual function $F(\boldsymbol{X}^p;\boldsymbol{\theta})$ instead of directly mapping $\boldsymbol{X}^p$ using a skip connection. The spectral residual architecture can be formulated as follows:

$$\boldsymbol{X}^{p+2} = \boldsymbol{X}^p + F(\boldsymbol{X}^p;\boldsymbol{\theta}) \tag{4}$$

$$F(\boldsymbol{X}^p;\boldsymbol{\theta}) = R(\hat{\boldsymbol{X}}^{p+1}) * \boldsymbol{h}^{p+2} + \boldsymbol{b}^{p+2} \tag{5}$$

$$\boldsymbol{X} = R(\hat{\boldsymbol{X}}^{p}) * \boldsymbol{h}^{p+1} + \boldsymbol{b}^{p+1} \tag{6}$$

where $\boldsymbol{\theta} = \{\boldsymbol{h}^{p+1}, \boldsymbol{h}^{p+2}, \boldsymbol{b}^{p+1}, \boldsymbol{b}^{p+2}\}$, $\boldsymbol{X}^{p+1}$ represents the n input 3-D feature cubes of $(p+1)$th layer, $\boldsymbol{h}^{p+1}$ and $\boldsymbol{d}^{p+1}$ denote the spectral convolutional kernels and bias in the $(p+1)$th layer, respectively. In fact, the convolutional kernels $\boldsymbol{h}^{p+1}$ and $\boldsymbol{d}^{p+1}$ are composed of 1-D vectors, which can be regarded as a special case of 3-D convolutional kernels. The output tensor of the spectral residual block also includes n 3-D feature cubes.

In the spatial residual block, as shown in Fig. 4, a focus is primarily placed on the spatial feature extraction using n 3-D convolutional kernels of size $a\times a\times d$ in the successive filter banks $\boldsymbol{H}^{q+1}$ and $\boldsymbol{H}^{q+2}$ for the two successive layers. The spectral depth d of these kernels equals to that of the input 3-D feature cubes $\boldsymbol{X}^q$. The spatial size of feature cubes $\boldsymbol{X}^{q+1}$ and $\boldsymbol{X}^{q+2}$ is kept unchanged at $w\times w$. Thus, the spatial residual architecture can be formulated as follows:

$$\boldsymbol{X}^{q+2} = \boldsymbol{X}^q + F(\boldsymbol{X}^q;\boldsymbol{\xi}) \tag{7}$$

$$F(\boldsymbol{X}^q;\boldsymbol{\xi}) = R(\hat{\boldsymbol{X}}^{q+1}) * \boldsymbol{H}^{q+2} + \boldsymbol{b}^{q+2} \tag{8}$$

$$\boldsymbol{X} = R(\hat{\boldsymbol{X}}^{q}) * \boldsymbol{H}^{q+1} + \boldsymbol{b}^{q+1} \tag{9}$$

where $\boldsymbol{\xi} = \{\boldsymbol{H}^{q+1}, \boldsymbol{H}^{q+2}, \boldsymbol{b}^{q+1}, \boldsymbol{b}^{q+2}\}$, $\boldsymbol{X}^{q+1}$ represents the 3-D input feature volume in the $(q+1)$th layer, and $\boldsymbol{H}^{q+1}$ and $\boldsymbol{b}^{q+1}$ denote the n spatial convolutional kernels in the $(q+1)$th layer, respectively. Compared with their spectral counterparts, the convolutional filter banks in spatial residual blocks comprises 3-D tensors. The output of this block is a 3-D feature volume.

C. Spectral–Spatial Residual Network

Considering HSIs contain one spectral dimension and two spatial dimensions, we proposed a framework that consecutively extracts spectral and spatial features for pixelwise HSI classification. As shown in Fig. 5, the SSRN includes a spectral feature learning section, a spatial feature learning section, an average pooling layer, and a fully connected (FC) layer. Compared with CNN, SSRN alleviated the decreasing-accuracy phenomenon by adding skip connections between every other layer to formulate the hierarchical feature representation layers to consecutive residual blocks. We take the IN data set, the 3-D samples of which have the size of $7\times 7\times 200$, as an example to explain the designed SSRN.

The spectral feature learning section includes two convolutional layers and two spectral residual blocks. In the first convolutional layer, 24 $1\times 1\times 7$ spectral kernels with a

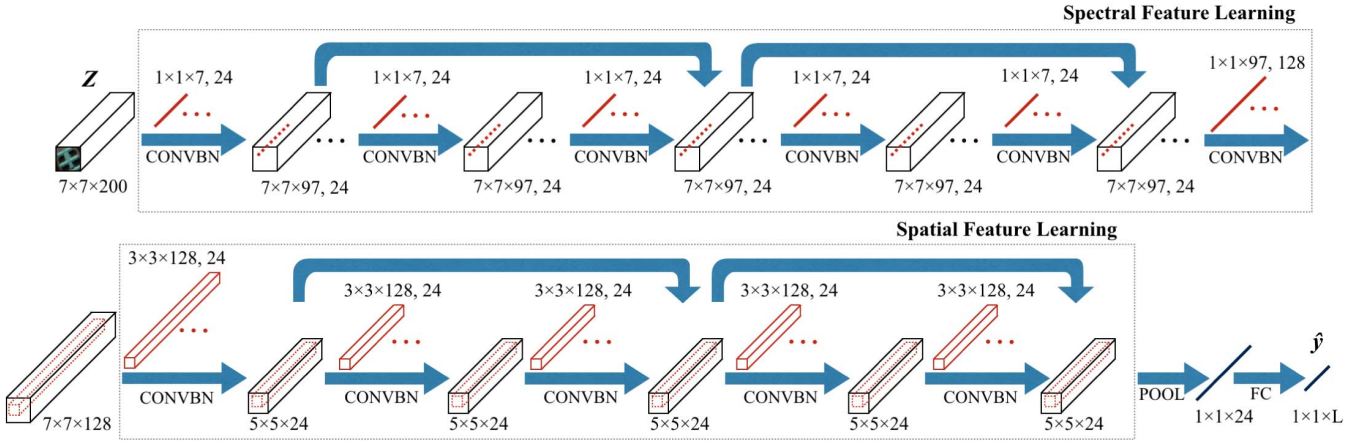


Fig. 5. SSRN with a $7 \times 7 \times 200$ input HSI volume. The network includes two spectral and two spatial residual blocks. An average pooling layer and an FC layer transform a $5 \times 5 \times 24$ spectral–spatial feature volume into a $1 \times 1 \times L$ output feature vector $\hat{y}$.

subsampling stride of (1, 1, 2) convolves the input HSI volume to generate 24 $7 \times 7 \times 97$ feature cubes. Because the raw input data contain rich and redundant spectral information, $1 \times 1 \times 7$ vector kernels are used in these blocks. This layer reduces the high dimensionality of input cubes and extracts low-level spectral features of HSI. Then, two consecutive spectral residual blocks, which contain four convolutional layers and two identity mappings, use 24 $1 \times 1 \times 7$ vector kernels at each convolutional layers to learn deep spectral representation. In the spectral residual blocks, all convolutional layers use padding to keep the sizes of output feature cubes the same as input. Following the spectral residual blocks, the last convolutional layer in this learning section, which includes 128 $1 \times 1 \times 97$ spectral kernels for keeping discriminative spectral features, convolves the 24 7×7 feature tensors to produce a 7×7 feature volume as input for spatial feature learning section.

The spatial feature learning section extracts discriminative spatial features using successive 3-D convolutional filter banks, where the kernels have the same depth as the input 3-D feature volume. The section comprises a 3-D convolutional layer and two spatial residual blocks. The first convolutional layer in this section reduces the spatial size of input feature cubes and extract low-level spatial features with 24 $3 \times 3 \times 128$ spatial kernels, resulting an output $5 \times 5 \times 24$ feature tensor. Then, similar to their spectral counterparts, the two spatial residual blocks learn deep spatial representation with four convolutional layers, all of which use 24 $3 \times 3 \times 24$ spatial kernels and keep the sizes of feature cubes unchanged.

After the above two feature learning sections, an average pooling layer (POOL) transforms the extracted $5 \times 5 \times 24$ spectral–spatial feature volume to a $1 \times 1 \times 24$ feature vector. Next, an FC layer adapts the SSRN to HSI data set according to the number of land-cover categories and generates an output vector $\hat{\boldsymbol{y}} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_L]$. The total numbers of trainable parameters (about 360 000) for the SSRN are much larger than the available training data in the three hyperspectral data sets, which means that the network possesses enough capacity to learn the feature representations of HSI but also tend to overfit the training sets. Therefore, BN and dropout [29] are investigated as regularization strategies to further improve the classification performance of SSRN.

III. Results and Discussion

In this section, we introduced the three HSI data sets, specified the model configuring process, and evaluated the proposed methods using classification metrics, such as overall accuracy (OA), average accuracy (AA), and kappa coefficient (κ). We adopted the IN, Kennedy Space Center (KSC), and University of Pavia (UP) data sets for assessing the classification performance of the SSRN framework in the cases of unbalanced training data, a small number of training samples, and high spatial resolution. In all three cases, we ran experiments for ten times with randomly selected training data and reported the mean and standard deviation of main classification metrics.

A. Experimental Data Sets

The IN data set, gathered by Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) in 1992 from Northwest Indiana, includes 16 vegetation classes and has 145×145 pixels with a resolution of 20 m by pixel. Once the 20 bands corrupted by water absorption effects have been discarded, the remaining 200 bands are adopted for analysis and range from 400 to 2500 nm.

The KSC data set, collected by AVIRIS in Florida in 1996, contains 13 upland and wetland classes and has 512×614 pixels with a resolution of 18 m by pixel. Once the bands with low signal-to-noise ratio have been removed, the remaining 176 bands are used for assessment and range from 400 to 2500 nm.

The UP data set, acquired by Reflective Optics System Imaging Spectrometer in Northern Italy in 2001, contains nine urban land-cover types and has 610×340 pixels with a resolution of 1.3 m by pixel. Once the noisy bands have been discarded, the remaining 103 bands are employed for evaluation and ranges from 430 to 860 nm.

TABLE I
TRAINING, VALIDATION, AND TESTING NUMBERS IN THE IN DATA SET

No.	Class	Train.	Val.	Test.
1	Alfalfa	10	1	35
2	Corn-notill	286	131	1011
3	Corn-mintill	166	83	581
4	Corn	48	22	167
5	Grass-pasture	97	42	344
6	Grass-tree	146	69	515
7	Grass-pasture-mowed	6	3	19
8	Hay-windrowed	96	55	327
9	Oats	4	4	12
10	Soybean-notill	195	94	683
11	Soybean-mintill	491	264	1700
12	Soybean-clean	119	56	418
13	Wheat	41	26	138
14	Woods	253	136	876
15	Buildings-Grass-Trees	78	34	274
16	Stone-Steel-Towers	19	5	69
	TOTAL	2055	1025	7169

TABLE II
TRAINING, VALIDATION, AND TESTING NUMBERS IN THE KSC DATA SET

No.	Class	Train.	Val.	Test.
1	Scrub	153	78	530
2	Willow swamp	49	29	165
3	CP hammock	52	28	176
4	Slash pine	51	31	170
5	Oak/Broadleaf	33	18	110
6	Hardwood	46	22	161
7	Swap	21	4	80
8	Graminoid marsh	87	45	299
9	Spartina marsh	104	39	377
10	Cattail marsh	81	40	283
11	Salt marsh	84	39	296
12	Mud flats	101	61	341
13	Water	186	87	654
	TOTAL	1048	521	3642

TABLE III
TRAINING, VALIDATION, AND TESTING NUMBERS IN THE UP DATA SET

No.	Class	Train.	Val.	Test.
1	Asphalt	664	670	5297
2	Meadows	1865	1810	14974
3	Gravel	210	241	1648
4	Trees	307	333	2424
5	Metal Sheets	135	134	1076
6	Bare Soil	503	500	4026
7	Bitumen	133	133	1046
8	Bricks	369	363	2950
9	Shadows	95	97	755
	TOTAL	4281	4281	34214

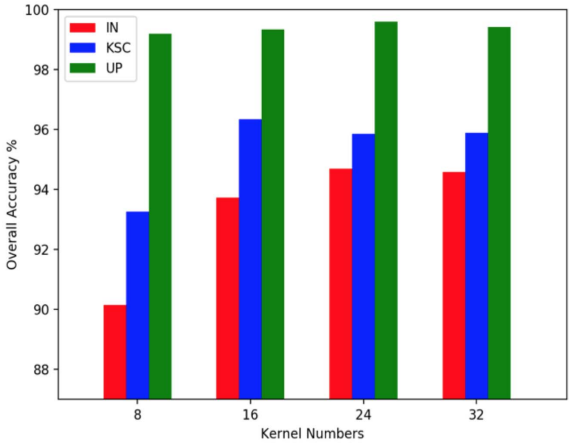


Fig. 6. OA (%) of SSRNs with different kernel numbers in the IN, KSC, and UP data sets.

TABLE IV
OA (%) OF SSRN WITH DIFFERENT REGULARIZERS

SSRN	IN	KSC	UP
None	96.41 ± 0.51	97.75 ± 0.54	98.97 ± 0.17
Dropout	95.83 ± 0.52	96.37 ± 0.89	99.02 ± 0.19
BN	97.73 ± 0.42	98.96 ± 0.23	99.42 ± 0.13
Both	97.76 ± 0.38	99.02 ± 0.31	99.59 ± 0.08

In the IN and KSC data sets, 20%, 10%, and 70% of the labeled data are randomly assigned to training, validation, and testing groups, respectively. In the UP data sets, the ratio is 10%:10%:80%. In addition, all input data of three HSI data sets are standardized to mean value with unit variance. Tables I–III list the numbers of three groups for all data sets.

B. Framework Setting

After designing the SSRN framework, we configured the training process that updates the parameters of 3-D filter banks through backpropagating the gradients of the cost function. Next, we analyzed four factors that control the training process and classification performance of the trained SSRN. The four factors are the learning rate, the kernel number of convolutional layers, the regularization method, and the spatial size of the input cubes. Because the training sets are small, we set the batch size to 16 and adopted the RMSProp optimizer [30] to harness the training process. In the training process of each configuration, the models with the highest classification performance in validation groups were preserved, and the reported results were generated by these optimal models.

First, learning rates control the learning step for each training iteration. Specifically, inappropriate learning rate settings will lead to divergence or slow convergence. Therefore, we used the grid search method and ran each experiment for 200 epochs to find the optimum learning rate from $\{0.01, 0.003, 0.001, 0.0003, 0.0001, 0.00003\}$ for each data set. Based on the classification outcomes, the optimum learning rates for IN, KSC, and UP data sets are 0.0003, 0.0001, and 0.0003, respectively.

Second, the kernel numbers of convolutional filter banks decide the representation capacity and computational consumption of SSRN. As shown in Fig. 5, the proposed network has the same kernel number in each convolutional layer of the spectral and spatial residual blocks. We assessed different kernel numbers from 8 to 32 in an interval of 8 in each convolutional layer to find a general framework. As shown in Fig. 6, the models with 24 kernels in each convolutional

TABLE V
OA (%) OF SSRN WITH DIFFERENT INPUT SIZES

Spatial Size	IN	KSC	UP
3×3	75.83 ± 0.14	92.38 ± 0.99	96.81 ± 0.24
5×5	92.83 ± 0.66	96.99 ± 0.55	98.72 ± 0.17
7×7	97.81 ± 0.34	99.01 ± 0.31	99.54 ± 0.11
9×9	98.68 ± 0.29	99.51 ± 0.25	99.73 ± 0.15
11×11	98.70 ± 0.21	99.57 ± 0.54	99.79 ± 0.08

TABLE VI
CLASSIFICATION RESULTS OF DIFFERENT METHODS FOR THE IN DATA SET

	SVM	SAE	CNN	CNNL	SPA	SPC	SSRN
OA(%)	81.67 ±0.65	85.47 ±0.58	97.41 ±0.43	95.78 ±0.71	98.01 ±0.37	90.68 ±0.75	**99.19 ±0.26**
AA(%)	79.84 ±3.37	86.34 ±1.14	97.39 ±0.56	95.67 ±1.23	98.15 ±0.56	92.00 ±2.84	**98.93 ±0.59**
$\kappa \times 100$	78.76 ±0.77	83.42 ±0.66	97.05 ±0.49	95.18 ±0.81	97.73 ±0.42	89.36 ±0.86	**99.07 ±0.30**
1	96.78	81.82	100.0	96.17	98.71	83.15	97.82
2	78.74	82.16	97.27	95.31	97.60	86.81	99.17
3	82.26	77.54	98.00	95.31	98.27	87.34	99.53
4	99.03	68.11	92.81	88.58	96.36	91.32	97.79
5	93.75	94.36	99.25	99.24	98.67	97.54	99.24
6	85.96	94.45	99.52	98.72	99.69	97.88	99.51
7	40.00	94.70	97.58	96.13	97.92	89.33	98.70
8	91.80	94.36	99.00	98.58	99.26	90.85	99.85
9	0	82.56	96.95	96.32	100.0	100.0	98.50
10	86.00	81.28	95.38	94.35	97.48	81.92	98.74
11	70.94	84.47	97.72	96.28	98.16	91.68	99.30
12	74.73	83.77	97.13	93.07	95.84	85.14	98.43
13	99.04	96.42	99.65	98.01	99.59	99.72	100.0
14	94.29	92.27	97.95	96.62	98.34	97.44	99.31
15	85.11	80.63	92.30	90.90	96.67	93.43	99.20
16	96.78	81.82	100.0	96.17	97.89	83.15	97.82

filter bank achieved the highest classification accuracy in the IN and UP data sets, and the model with 16 kernels obtained the best performance in the KSC data set. These results are acquired in 200-epoch training processes for each setting in three data sets.

Third, given there are more parameters than training samples and deep learning models tend to overfit training data, BN and a 50% dropout can be used for regularizing the training process. Hence, we evaluated the models without regularization method, with dropout, with BN, and with both dropout and BN under the same condition for 200-epoch training. As shown in Table IV, the BN outperforms the dropout in terms of mean overall classification accuracy. More importantly, the SSRN performs the best when using both regularization strategies in all three HSI data sets.

Fourth, to evaluate the influence of the spatialized input, we tested the proposed models with respect to the input cubes of different spatial sizes. Table V shows that the proposed

TABLE VII
CLASSIFICATION RESULTS OF DIFFERENT METHODS FOR THE KSC DATA SET

	SVM	SAE	CNN	CNNL	SPA	SPC	SSRN
OA(%)	80.29 ±0.58	92.99 ±0.82	97.08 ±0.47	95.45 ±0.45	98.63 ±0.38	97.90 ±0.49	**99.61 ±0.22**
AA(%)	65.64 ±0.86	89.76 ±1.25	95.09 ±0.70	92.56 ±0.99	97.81 ±0.64	96.56 ±0.69	**99.33 ±0.57**
$\kappa \times 100$	77.98 ±0.65	92.18 ±0.91	96.74 ±0.53	94.93 ±0.50	98.47 ±0.42	97.66 ±0.55	**99.56 ±0.25**
1	92.16	93.04	99.00	98.47	99.40	99.11	99.70
2	86.16	92.04	98.48	95.20	99.18	99.19	99.88
3	42.55	85.59	92.16	87.53	95.39	92.60	99.00
4	67.69	72.12	81.84	73.35	93.45	85.49	98.26
5	0	82.20	85.38	77.21	95.70	89.63	99.03
6	54.71	83.15	90.96	90.26	96.27	95.94	99.43
7	0	76.46	93.21	89.63	95.19	96.38	97.03
8	65.12	94.10	98.21	97.28	98.67	98.09	99.54
9	67.82	94.57	99.04	98.05	99.43	99.53	99.70
10	93.40	98.91	99.85	99.40	99.96	99.96	99.96
11	100.0	98.39	98.89	98.72	99.63	99.86	99.80
12	83.75	96.42	99.43	98.63	99.31	99.51	100.0
13	100.0	99.83	99.79	99.48	99.89	99.97	100.0

TABLE VIII
CLASSIFICATION RESULTS OF DIFFERENT METHODS FOR THE UP DATA SET

	SVM	SAE	CNN	CNNL	SPA	SPC	SSRN
OA(%)	90.58 ±0.47	94.25 ±0.18	98.85 ±0.15	98.64 ±0.20	99.25 ±0.08	98.88 ±0.22	**99.79 ±0.09**
AA(%)	92.99 ±0.36	93.34 ±0.39	98.40 ±0.30	98.13 ±0.35	98.99 ±0.27	98.40 ±0.27	**99.66 ±0.17**
$\kappa \times 100$	87.21 ±0.70	92.35 ±0.25	98.47 ±0.20	98.20 ±0.26	99.00 ±0.12	98.52 ±0.30	**99.72 ±0.12**
1	87.24	94.59	98.98	98.29	99.25	99.01	99.92
2	89.93	96.44	99.45	99.50	99.58	99.81	99.96
3	86.48	84.57	96.04	94.54	98.06	95.46	98.46
4	99.95	97.37	99.58	99.28	99.76	99.54	99.69
5	95.78	99.60	99.39	99.94	99.50	99.84	99.99
6	97.69	93.39	99.70	99.50	99.74	99.18	99.94
7	95.44	88.57	97.18	96.82	97.87	98.15	99.82
8	84.40	85.66	95.73	95.54	97.44	94.65	99.22
9	100.0	99.88	99.56	99.74	99.74	99.99	99.95

SSRNs perform robustly for different spatial sizes if these sizes are equal to or larger than 7×7, because the SSRN learns discriminative spatial features of input data. In all three data sets, the classification results increase with the spatial size of input cubes. The important role of spatial context that this experiment demonstrated is in accordance with results in other publications [3], [15]. Considering the larger input sizes lead to higher classification accuracy, we fixed the spatial size of input HSI data to make a fair comparison between different classification methods.

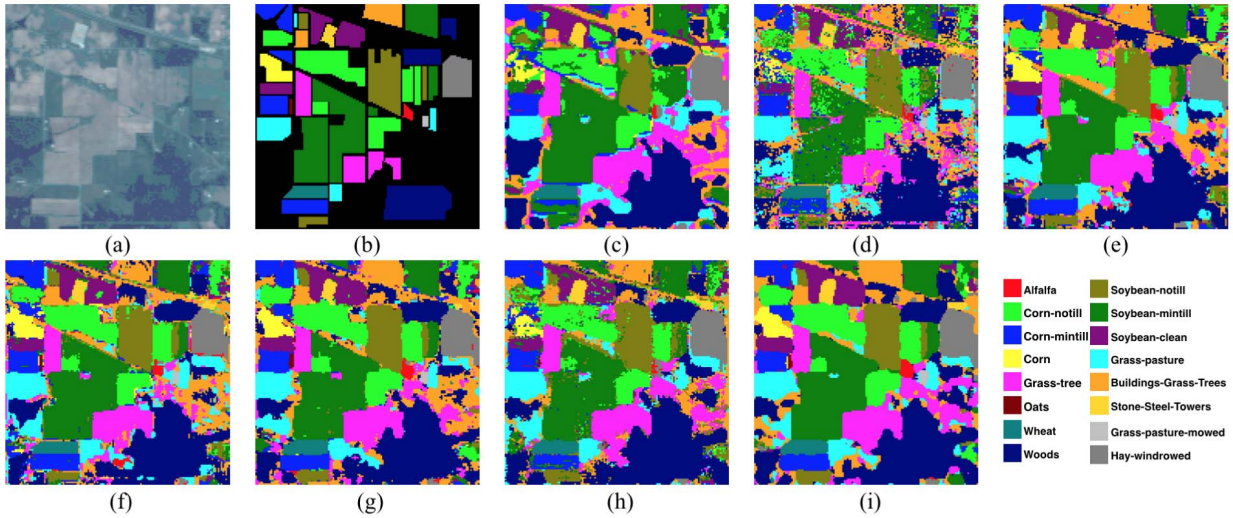


Fig. 7. Classification results of the best models for the IN data set. (a) False color image. (b) Ground-truth labels. (c)–(i) Classification results of SVM, SAE, CNN, CNNL, SPA, SPC, and SSRN.

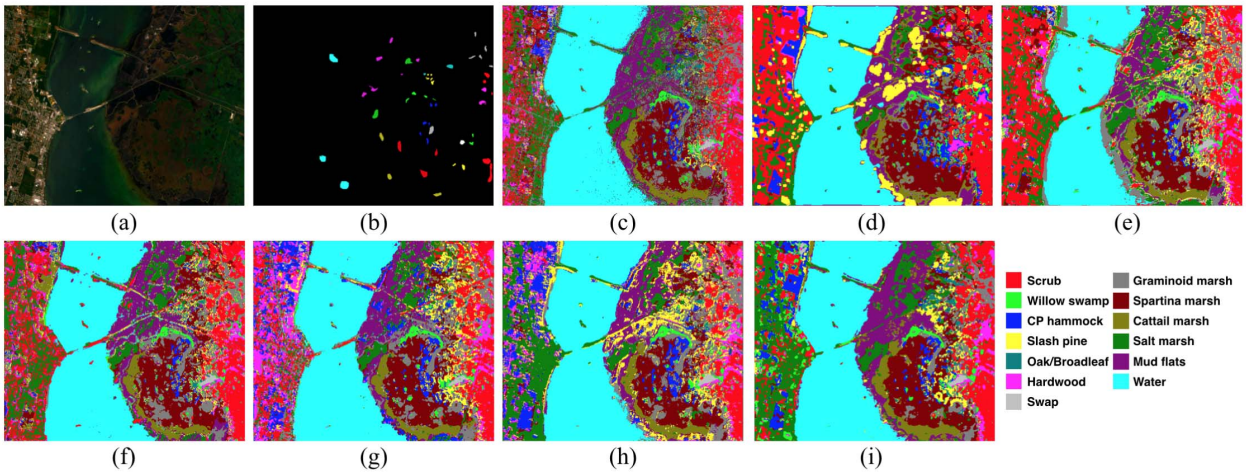


Fig. 8. Classification results of the best models for the KSC data set. (a) False color image. (b) Ground-truth labels. (c)–(i) Classification results of SVM, SAE, CNN, CNNL, SPA, SPC, and SSRN.

C. Classification Results

We compared the SSRN with the kernel SVM [31] and state-of-the-art deep learning models, such as SAE [17] and 3-D CNN [22]. To demonstrate the effectiveness of the spectral and spatial residual blocks in the proposed framework, we also tested the networks that only contain the spectral feature learning part (SPC) and the ones that only contain spatial feature learning part (SPA). Moreover, we evaluated the longer versions of 3-D CNN (denote as CNNL) generated from the SPA models without skip connections to study the effect of the designed spatial residual architecture on the decreasing-accuracy phenomenon [22]. To make a fair comparison, we set the same input volume size of $7 \times 7 \times b$ for all methods and tuned these competitors to their optimal settings. We randomly selected 20%, 20%, and 10% labeled 3-D HSI cubes as training groups for IN, KSC, and UP data sets, respectively.

Tables VI–VIII report the OAs, AAs, kappa coefficients, and the classification accuracies of all classes for HSI classification. In all three cases, the SSRN achieved the highest classification accuracy and lower standard deviation than the 3-D CNN. For example, in the KSC data set, SSRN (99.61%) delivered a roughly 2.5% increase of mean overall classification accuracy compared with CNN (97.08%). All deep learning methods generated obviously better outcomes than the kernel SVM. In all three data sets, the classification results of CNNL were worse than those of CNN. On the other hand, the SPA performed better than the CNN. These outcomes showed the proposed spatial residual structures mitigate the declining-accuracy phenomenon. Furthermore, the SSRN constantly performed better than the SPA, because the spectral residual blocks learned spectral representations that are complementary to spatial features. Although there are few training samples for oats and grass-pasture-mowed classes in the IN data set, the SSRN classified the testing data with higher than the 98% mean classification accuracy. These results validated the robustness of the designed models in the face of difficult conditions.

Figs. 7–9 visualize the classification results of the best trained models in three data sets, along with the false color images of original HSI and their corresponding ground-truth maps. In all three cases, the qualitative comparison between different methods is in line with the quantitative comparison in

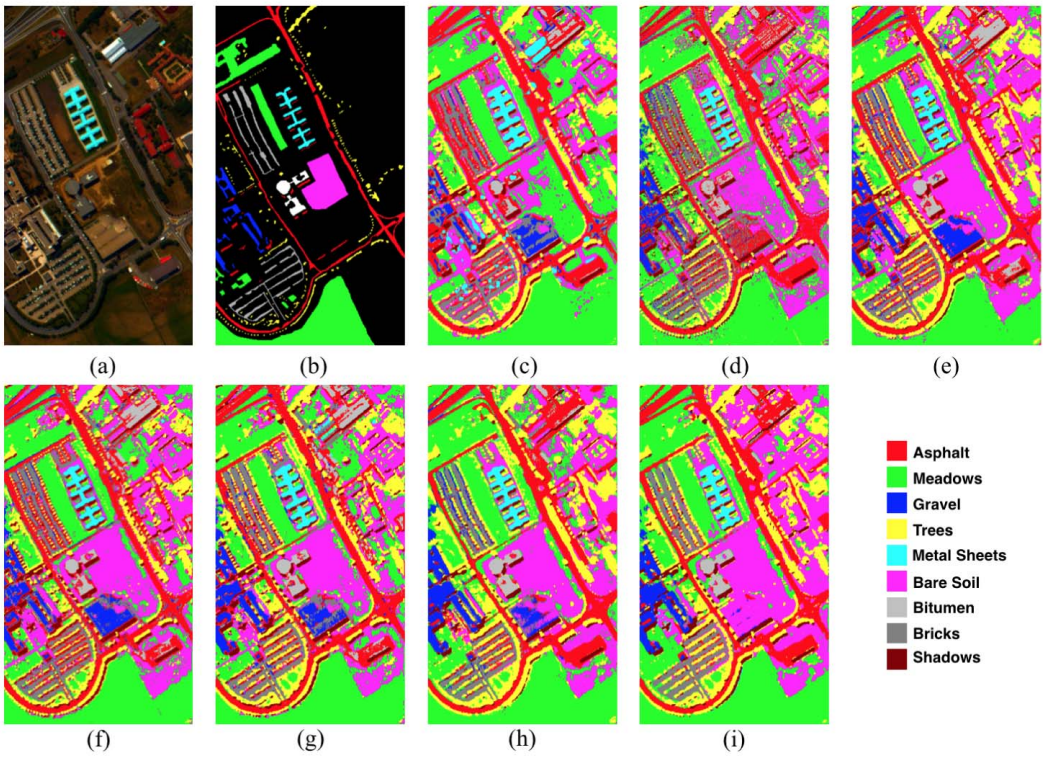


Fig. 9. Classification results of the best models for the UP data set. (a) False color image. (b) Ground-truth labels. (c)–(i) Classification results of SVM, SAE, CNN, CNNL, SPA, SPC, and SSRN.

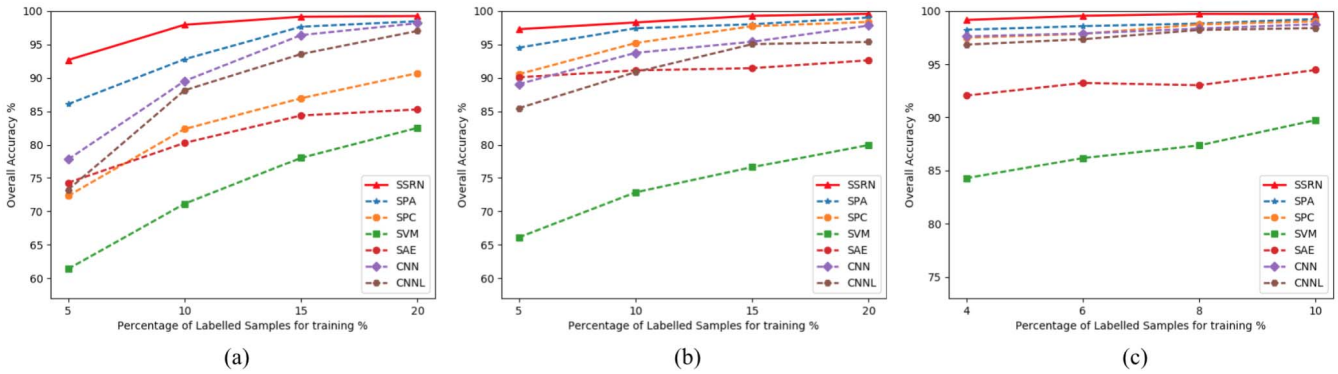


Fig. 10. OA of different methods with different training data percentages. (a) IN data set. (b) KSC data set. (c) UP data set.

Tables VI–VIII. The SPC generated classification maps with great noise. The SPA generated smoother results, but still some dot noises exist in some classes. For example, the SPA reduced the speckles in the Wheat class of IN data set and the Bare Soil class of UP data set. Compared with other methods, the SSRN delivered the most accurate and smooth classification maps for all three HSIs, because the SSRN learned discriminative spectral and spatial features consecutively.

To test the robustness and generalizability of the proposed SSRN toward different numbers of training samples, 5%, 10%, 15%, and 20% labeled samples were randomly chosen as training data for IN and KSC data sets, and 4%, 6%, 8%, and 10% for the UP data set. In Fig. 10, the overall accuracies of different classifiers using different numbers of training data are illustrated. For a small number of training samples, when the SVM generated inferior OA, the SSRN still produced high classification accuracy, and it is more obvious that the SSRN performs the best than other methods, because the SSRN extract more discriminative features than other methods. For a large number of training samples, the SSRN still generates the best classification outcomes in all three HSI data sets, but the improvements are not that clear, simply because the classification accuracy is very high (higher than 99% OA).

To further validate the effectiveness of residual blocks for mitigating the accuracy-decreasing phenomenon, SSRN models with varying residual blocks were constructed for classifying 3-D HSI data. We tested SSRNs with from two to five blocks and treated spectral and spatial residual blocks

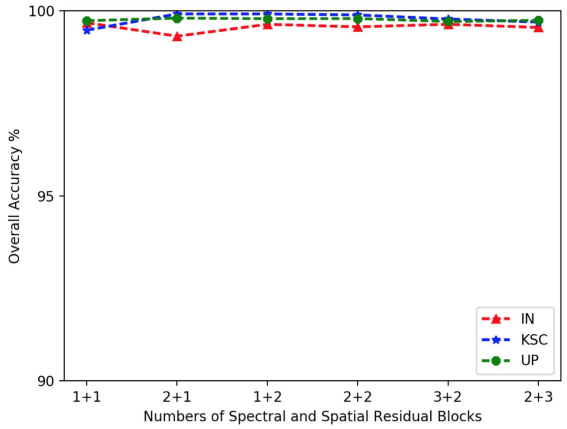


Fig. 11. OA of spectral–spatial neural networks with varying layers and combinations of residual blocks. The $x + y$ formation in the horizontal axis denotes an SSRN with x spectral and y spatial residual blocks.

differently using the same settings as in Tables VI–VIII. In Fig. 11, the overall classification accuracy differences between the deeper SSRNs and their shallow-layer counterparts are negligible. Therefore, in contrast to the obvious accuracy-decreasing effects reported in [17] and [22], the consistent HSI classification performance of SSRNs with varying layers demonstrated that the residual connections mitigate the decreasing-accuracy effects in other deep learning models.

The training and testing times provide a direct measure of computational efficiency for the SSRN. All experiments were conducted on an MSI GT72S laptop with the GTX 980M graphical processing unit (GPU). Table IX lists the training and testing times of the SSRN and other deep learning models. As presented in Table IX, the training times of the spectral section part (SPC) are 5–10 times longer than its spatial counterpart (SPA), because the spectral residual blocks preserved abundant features and kept the spatial size unchanged. In other words, the spectral residual blocks in the SSRN require a larger amount of computational power than their spatial counterparts. The SSRN takes 6–10 times longer for training than the CNN, which means that the SSRN is more computationally expensive than the CNN. Fortunately, the adoption of GPU has largely alleviated the extra computational costs and reduced the training times.

D. Discussion

The experimental outcomes validate the effectiveness of the SSRN framework. It is worth noting that different deep learning models usually prefer different hyperparameters, which poses a challenge for deploying these models. However, the classification performance of the SSRN with different settings is stable according to experiment results. Compared with traditional feature engineering-based machine learning methods (e.g., kernel SVM), deep learning models have four advantages: 1) automatic feature extraction; 2) hierarchical nonlinear transformation; 3) objective functions that directly focus on classification in lieu of two independent steps; and 4) the ability to utilize computational hardware (especially GPU) efficiently.

TABLE IX
TRAINING AND TESTING TIMES OF DIFFERENT MODELS FOR THREE HSI DATA SETS

		IN	KSC	UP
SAE	Train.(m)	3.5	2.6	12.8
	Test.(s)	2.0	0.8	7.7
CNN	Train.(m)	11.4	4.1	17.0
	Test.(s)	3.1	1.2	8.6
SPA	Train.(m)	10.9	5.4	26.3
	Test.(s)	3.0	1.5	14.5
SPC	Train.(m)	100.5	28.7	123.2
	Test.(s)	21.3	8.9	65.6
SSRN	Train.(m)	106.0	41.1	105.5
	Test.(s)	17.2	4.4	34.5

Three major differences exist between SSRNs and other deep learning models (e.g., SAE and CNN). First, the SSRN adopts residual connections that improve the classification accuracy and make deep learning models much easier to train. Second, the SSRN treats spectral features and spatial features separately in two consecutive blocks, through which more discriminative features can be extracted. Third, owing to BN operation at each convolutional layer, we only need hundreds of iterations for training the SSRN instead of hundreds of thousands in [24].

Three main factors influence the HSI classification performance of supervised deep learning models: 1) the number of training samples; 2) the spatial size of input data; and 3) the representative capacity of the designed models. Because the SSRN obtained very high classification accuracy for relatively few land-cover categories, we did not employ data augmentation [22] to further boost the classification performance of the SSRN despite a small number of training samples. Given a fixed model, the more data used for training, and the more information these data contain, the higher classification accuracy deep learning models can generate. Therefore, to make a fair comparison, we need to test different models under the same number of training samples and the same size for each input sample.

IV. CONCLUSION

In this paper, we have presented a supervised 3-D deep learning framework for spectral–spatial representation learning and HSI classification. The designed SSRN, which contains consecutive spectral and spatial residual blocks, has alleviated the decreasing-accuracy phenomenon. The experiment results demonstrated that the SSRN performs consistently with the highest classification accuracy for all three types of HSI data sets with different challenges. It is worth noting that this network has delivered robust classification performance using small as well as large numbers of uneven training samples. In addition, the BN strategy regularized the training process and improved the classification accuracy. Finally, the SSRN achieved the state-of-the-art results with the limited labeled 3-D cubes as training data in three cases and can easily be generalized to other remote-sensing scenarios because of their uniform structural design and deep feature learning capacity.

The essence of deep learning models is learning the representation of input data automatically without feature engineering, because the models themselves can extract discriminative features given appropriate architectural designs and training process settings. Moreover, these hyperparameter settings depend on the number of training samples and the spatial size of each sample. In the cases of HSI classification, one prominent challenge is the shortage of annotations. Thus, this paper counters this obstacle with the proposed spectral–spatial residual architecture that takes both abundant spectral signatures and spatial contexts into account.

It is suggested that the deep learning methods need a significant amount of labeled data for training [21]. However, the experimental results have demonstrated that the proposed models, which have a spectral–spatial residual architecture and an appropriate regularization strategy, perform vigorously with large numbers as well as a limited numbers of training samples. Also, according to the sensitivity test results, the proposed network can extract more discriminative spatial features with larger input cubes, and simply expanding the sizes of input data will increase the classification accuracy. In other words, HSI classification models using training samples with more spatial information tend to have an advantage over the ones using training data with less spatial information. Therefore, we advocate that the spatial size of input HSI data should be the same when comparing different classification methods. Considering the consistent performance in three widely studied HSI cases, we believe that the SSRN still can outperform other machine learning competitors for HSI classification under the same comparison standards in other cases.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] L. Zhang, Y. Zhong, B. Huang, J. Gong, and P. Li, "Dimensionality reduction based on clonal selection for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 4172–4186, Dec. 2007.

[6] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral–spatial classification of hyperspectral data using loopy belief propagation and active learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 2, pp. 844–856, Feb. 2013.

[7] X. Jia, B.-C. Kuo, and M. M. Crawford, "Feature mining for hyperspectral image classification," *Proc. IEEE*, vol. 101, no. 3, pp. 676–697, Mar. 2013.

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

[9] Q. Wang, J. Lin, and Y. Yuan, "Salient band selection for hyperspectral image classification via manifold ranking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1279–1289, Jun. 2016.

[10] Y. Yuan, J. Lin, and Q. Wang, "Hyperspectral image classification via multitask joint sparse representation and stepwise MRF optimization," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2966–2977, Dec. 2016.

[11] M. Khodadadzadeh, J. Li, A. Plaza, H. Ghassemian, J. M. Bioucas-Dias, and X. Li, "Spectral–spatial classification of hyperspectral data using local and global probabilities for mixed pixel characterization," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 10, pp. 6298–6314, Oct. 2014.

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

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

[14] M. Pal and G. M. Foody, "Feature selection for classification of hyperspectral data by SVM," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 5, pp. 2297–2307, May 2010.

[15] J. Peng, Y. Zhou, and C. L. P. Chen, "Region-kernel-based support vector machines for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 9, pp. 4810–4824, Sep. 2015.

[16] Y. Tarabalka, M. Fauvel, J. Chanussot, and J. A. Benediktsson, "SVM- and MRF-based method for accurate classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 4, pp. 736–740, Oct. 2010.

[17] 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.

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

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1106–1114.

[20] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

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

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

[27] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.

[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[29] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[30] T. Tieleman and G. E. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA, Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–31, 2012.

[31] B. Waske, S. van der Linden, J. A. Benediktsson, A. Rabe, and P. Hostert, "Sensitivity of support vector machines to random feature selection in classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 7, pp. 2880–2889, Jul. 2010.

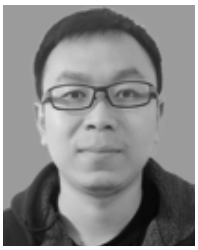
Zilong Zhong (S'15) received the M.Eng. degree in electronic and communication engineering from Lanzhou University, Lanzhou, China, in 2014. He is currently pursuing the Ph.D. degree in remote sensing with the University of Waterloo, Waterloo, ON, Canada.

His research interests include computer vision, deep learning, probabilistic graphical models, and their applications in the context of high-dimensional remotely sensed data.

Jonathan Li (M'00–SM'11) received the Ph.D. degree in geomatics engineering from the University of Cape Town, Cape Town, South Africa.

He is currently a Professor with the Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON, Canada. He is also with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Information Science and Engineering, Xiamen University, Xiamen, China. He has co-authored more than 300 publications, more than 150 of which were published in refereed journals, including the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, ISPRS *Journal of Photogrammetry and Remote Sensing*, and *Remote Sensing of Environment*. His research interests include information extraction from mobile LiDAR point clouds and from earth observation images.

Dr. Li is the Chair of the ISPRS WG I/2 on LiDAR, Airborne and Spaceborne Optical Sensing (2016–2020) and the ICA Commission on Sensor-driven Mapping (2015–2019). He is an Associate Editor of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS and JSTARS.



Zhiming Luo received the B.Sc. degree in cognitive science from Xiamen University, Xiamen, China, in 2011. He is currently pursuing the Ph.D. degree in computer science with Xiamen University and the University of Sherbrooke, Sherbrooke, QC, Canada.

His research interests include traffic surveillance video analytics, computer vision, and machine learning.

Michael Chapman received the Ph.D. degree in photogrammetry from Laval University, Quebec City, QC, Canada.

He is currently a Professor of geomatics with the Department of Civil Engineering, Ryerson University, Toronto, ON, Canada. He has co-authored over 160 technical papers. His research interests include algorithms and processing methods for airborne sensors using GPS/INS geometric processing of digital imagery in industrial environments, terrestrial imaging systems for transportation infrastructure mapping, and algorithms and processing strategies for metrology applications.