Deep Collaborative Attention Network for Hyperspectral Image Classification by Combining 2-D CNN and 3-D CNN

Hao Guo, Jianjun Liu, *Member, IEEE*, Jinlong Yang, Zhiyong Xiao, and Zebin Wu, *Senior Member, IEEE*

***Abstract*—Deep learning-based methods based on convolutional neural networks (CNNs) have demonstrated remarkable performance in hyperspectral image (HSI) classification. Most of these approaches are only based on 2-D CNN or 3-D CNN. It is dramatic from the literature that using just 2-D CNN may result in missing channel relationship information, and using just 3-D CNN may make the model very complex. Moreover, the existing network models do not pay enough attention to extracting spectral-spatial correlation information. To address these issues, we propose a deep collaborative attention network for HSI classification by combining 2-D CNN, and 3-D CNN (CACNN). Specifically, we first extract spectral-spatial features by using 2-D CNN, and 3-D CNN, respectively, and then use a "NonLocalBlock" to combine these two kinds of features. This block serves as a typical spatial attention mechanism, and makes salient features be emphasized. Then, we propose a "Conv_Block" that is similar to the lightweight dense block to extract correlation information contained in the feature maps. Finally, we consider a deep multilayer feature fusion strategy, and thereby combine the features of different hierarchical layers to extract the strong correlated spectral-spatial information among them. To test the performance of CACNN approach, several experiments are performed on four well-known HSIs. The results are compared with the state-of-the-art approaches, and satisfactory performance is obtained by our proposed method. The code of CACNN method is available on Dr. J. Liu's GitHub.[1]**

***Index Terms*—Convolutional neural network (CNN), feature extraction, hyperspectral image classification, multilayer feature fusion, spatial attention mechanism.**

I. Introduction

HYPERSPECTRAL images (HSIs) have high spectral resolution with abundant spectral bands, and each spectral band corresponds to an image with a specific wavelength. They are capable of characterizing various properties of objects including physics, chemistry, materials, etc. Therefore, HSI has been widely applied to many fields, such as ecological science, earth exploration, and environmental monitoring [1]–[4]. One of the most important application is to classify every pixel in the HSI.

In the past decades, HSI classification methods have been developed from using the hand-designed feature description to extracting discriminative feature via efficient and automatic ways. In the early stage, typical hand-designed feature classification approaches pay attention to exploring the role of spectral features for enhancing HSI classification performance [5], [6]. Thus, a large number of pixel classification methods, including support vector machine (SVM) [7] [8], multinomial logistic regression [9], [10], sparse representation [11], [12], collaborative representation [13], [14], and so on, have been proposed. In addition, the high dimensionality of HSI easy to produce the Hughes phenomenon [15], so other classification methods focus on dimension–reduction technique, such as principal component analysis (PCA) [16], [17], independent component analysis [18], etc. However, there often exist redundant or even noisy spectral bands in the HSI since the drawback of imaging mechanism and the sensor. To overcome the deficiency of only using spectral bands, many classification methods based on spectral-spatial features are proposed by incorporating spatial-contextual information into classifiers [19]–[23]. For instance, Tarabalka *et al*. [24] proposed a new spectral-spatial approach that combines the results of a pixel-wise SVM classification and the segmentation map obtained by partitional clustering using majority voting. Fauvel *et al*. [25] proposed the method based on the fusion of morphological information and original hyperspectral data. Kang *et al*. [26] proposed a novel spectral-spatial classification framework based on edge-preserving filtering. Due to the limitation of resolution and the complexity of imaging process, different materials often have similar spectra, while the same materials exhibit different spectra.

Recently, deep learning, which automatically and hierarchically extract feature by learning parameters, has attracted increasing attention in many computer vision tasks, such as image classification [27], object detection and tracking [28], semantic segmentation [29], and so on. Chen *et al*. [30] first apply deep learning to HSI classification. With the development of deep learning, many other approaches based on this are employed

Manuscript received June 23, 2020; revised July 26, 2020; accepted August 9, 2020. Date of publication August 14, 2020; date of current version August 28, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61601201, and in part by the Natural Science Foundation of Jiangsu Province under Grant BK20160188. *(Corresponding author: Jianjun Liu.)*

Hao Guo, Jianjun Liu, Jinlong Yang, and Zhiyong Xiao are with the Jiangsu Provincial Engineering Laboratory for Pattern Recognition and Computational Intelligence, Jiangnan University, Wuxi 214122, China (e-mail: guohao1220ily@163.com; liuofficial@163.com; yjlgedeng@163.com; zhiyong.xiao@jiangnan.edu.cn).

Zebin Wu is with the School of Computer Science, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: zebin.wu@gmail.com).

Digital Object Identifier 10.1109/JSTARS.2020.3016739

[1] Online. [Avaialble]: https://github.com/liuofficial

in HSI classification to improve performance [31]–[35]. The representative models based on deep learning include convolutional neural network (CNN) [36], deep belief network [37], and recurrent neural network [38]. Moreover, different structures of CNN models have been investigated, such as 2-D CNN [39], and 3-D CNN [40], [41], due to its remarkable performance gain over the hand-designed features [42]. For combining the advantages of these network structures to improve classification performance, researchers have also proposed many deep networks. For instance, Yang *et al*. [43] proposed a deep CNN with two-branch architecture (Two-CNN) to extract the joint spectral-spatial features from spectral domain as well as spatial domain of HSIs. Fang *et al*. [44] use deep hash neural networks (SPDF-SVM) to expand discrimination among different classes and extract hyperspectral features for HSI classification, simultaneously. Whereas, with the increase of network depth, the classification accuracy of some CNN models decrease. To overcome this drawback, Zhong *et al*. [45] have proposed the spectral-spatial residual network (SSRN) using the identity mapping to connect every other 3-D convolutional layer. Paoletti *et al*. [46] have proposed pyramidal network based on residual blocks (pResNet) that gradually increases the feature map dimension at all convolutional layers while balancing the workload among all units. To obtain complementary spectral-spatial features among different hierarchical layers, Guo *et al*. [47] present discriminative multiple spatial-spectral feature fusion (FFDN). Xiao *et al*. [48] propose a new method with variable convolution for HSI classification. Pan *et al*. [49] use mapping layers to map the input patch into a low-dimensional subspace by multilinear algebra with a convolutional neural for HSI classification. More recently, many studies have shown that HSI classification framework based on deep spectral-spatial features can achieve state-of-the-art results. So the spectral-spatial joint HSI classification methods based on deep network have become the mainstream [50]–[55].

However, most existing methods have inherent limitations because of the need of extracting enough spectral and spatial correlation information in classification models. On the one hand, deep network structures are only based on 2-D CNN, making model easy to miss channel relationship information. Similarly, model only based on 3-D CNN may be very complicated and perform worse for classes having similar textures on many spectral bands. Obviously, 2-D CNN and 3-D CNN have their own technique shortcomings. 2-D CNN focuses on extracing spatial feature information, and 3-D CNN focuses on extracing spectral-spatial feature information. With the fusion of the two feature maps, the new feature map contains more discriminative spectral-spatial information. On the other hand, to address this problem, attention mechanism is applied in computer vision, which makes people pay more attention to the most important component. Recently, several attempts have been made to incorporate attention processing to improve CNN's performance in classification tasks. SE-Net [56] uses global average pooling to compute channelwise attention. Haut *et al*. [57] designed a new visual attention-driven mechanism into a ResNet. Moreover, several attention techniques have been developed for HSI classification. Fang *et al*. [58] proposed a network to apply the spectralwise attention technique in a densely connected 3-D CNN. Dong *et al*. [59] designed a cooperative spectral-spatial attention dense network (CS^{2}ADN) with the dense connection for HSI classification. Attention mechanism has been favored by more and more researchers because of its good effect in HSI classification.

In this article, we propose a deep collaborative attention network for HSI classification by combining 2-D CNN with 3-D CNN (CACNN). In the pre-processing phase, PCA algorithm is used to extract the most informative components on hyperspectral data. In deep network model, we first use 2-D CNN and 3-D CNN to extract spectral-spatial features, respectively, and then combine these two kinds of features with a "NonLocalBlock" [60]. This block is termed as a typical spatial attention mechanism to make salient features be emphasized. Then, we proposed "Conv_Block" which is similar to the light weight dense block to extract correlation information contained in the feature maps. In addition, we consider a deep multilayer feature fusion to extract the strong correlated spatial-spectral information among different hierarchical layers. Finally, the obtained discriminative spatial-spectral features are fed into a 1 × 1 convolution layer to assist classification.

Some of the innovative characteristics of the proposed approach are highlighted as follows.

1) The 3-D CNN and 2-D CNN layers are collaborated for the proposed model in such a way that we will achieve abundant spectral as well as spatial feature maps. By combining of these feature maps, the new feature maps contain rich spectral-spatial correlation information.
2) The NonLocalBlock and Conv_Block are utilized to emphasize spatial correlation features and extract high-level abstract correlation information, separately.
3) To make full use of complementary spectral-spatial features among different hierarchical layers, CACNN adopts a multilayer feature fusion strategy. And the obtained discriminative feature maps are contributed to achieving expected classification results.

The remainder of this article is organized as follows. Section II describes the proposed CACNN and the corresponding algorithms. The experiments are shown in Section III. Finally, Section IV concludes this article with some remarks and future research directions.

II. Proposed Framework

The main procedure of the proposed CACNN is shown in Fig. 1, including spectral-spatial feature extraction by combining 2-D CNN and 3-D CNN, NonLocalBlock, Conv_block, deep multilayer feature fusion, and a 1 × 1 convolution layer followed by a softmax function. Generally, the input of the original HSI can be denoted as $\mathbf{X} \in \mathbb{R}^{H\times W\times D}$, the output $\mathbf{Y} \in \mathbb{R}^{H\times W\times C}$ denotes the class probability of each pixel, where H, W, D, and C are indicated as height, width, number of bands, and number of classes, separately. In the CACNN, due to high spectral resolution and hundreds of channels along the spectral dimension, we use PCA algorithm to remove the spectral redundancy in raw HSI data ($\mathbf{X}$). The PCA reduces spectral bands from D to B while maintaining the same spatial dimensions (i.e., width

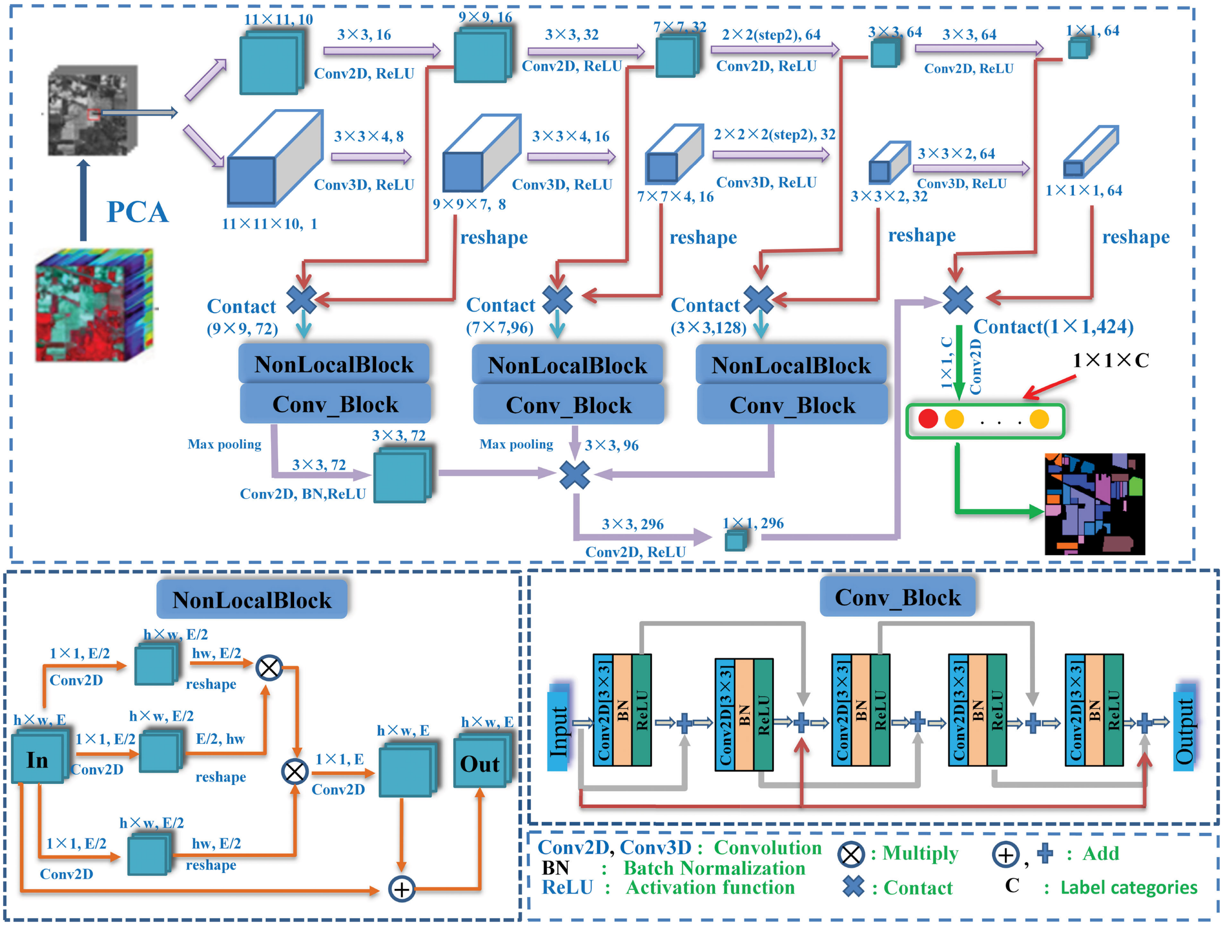


Fig. 1. Overall flowchart of HSI classification based on the CACNN.

W and height H). The low-dimensional $\mathbf{I} \in \mathbb{R}^{H \times W \times B}$ is a new data after PCA. In order to exploit better spectral-spatial features, we design a substructure with 2-D CNN and 3-D CNN. In this substructure, spatial feature information are extracted by 2-D CNN, and meanwhile spectral-spatial contexts are exploited by 3-D CNN. In the proposed framework, batch normalization (BN) [61] and ReLU [62] are added after some convolution layers.

A. Spectral-Spatial Feature Extraction

In order to make the extracted feature maps contain enough spectral-spatial correlation contexts, 2-D and 3-D convolutional operation is designed in CACNN model. As shown in Fig. 1, Conv2D and Conv3D, respectively, denotes 2-D CNN and 3-D CNN. As we all know, the kernel in 2-D CNN is strided cover full spatial dimension convolution happens in fact by computing the sum of the dot product between feature maps and kernel, which generates new feature maps. At spatial position (p, q) in the *j*th feature map of the *i*th layer, the activation value $v_{i,j}^{p,q}$ is generated by the following equation:

$$v_{i,j}^{p,q} = \Phi\left(\sum_{\eta=1}^{n_{l-1}} \sum_{\gamma=-\rho}^{\rho} \sum_{\delta=-\sigma}^{\sigma} w_{i,j,\eta}^{\delta,\gamma} \times v_{i-1,\eta}^{p+\delta,q+\gamma} + b_{i,j}\right) \quad (1)$$

where Φ is the activation function, $w_{i,j}$ and $b_{i,j}$ is weight and bias parameter in the *j*th feature map of the *i*th layer, respectively, n_{l-1} is the number of feature map in (l-1)th layer, $2\rho + 1$ and $2\sigma + 1$ are width and height of kernel.

In 3-D CNN, the input data are convolved with 3-D kernels before going through activation function to produce the feature maps. At spatial position (p, q, u) in the *j*th feature map of the *i*th layer, the activation value $v_{i,j}^{p,q,u}$ is generated as follows:

$$v_{i,j}^{p,q,u} = \Phi\left(\sum_{\eta=1}^{n_{l-1}} \sum_{\theta=-\varepsilon}^{\varepsilon} \sum_{\gamma=-\rho}^{\rho} \sum_{\delta=-\sigma}^{\sigma} w_{i,j,\eta}^{\delta,\gamma,\theta} \right.$$
$$\left. \times\, v_{i-1,\eta}^{p+\delta,q+\gamma,u+\theta} + b_{i,j}\right) \quad (2)$$

where $2\varepsilon + 1$ is the depth of kernel in spectral dimension and other parameters are the same as in (1).

The following part will describe in detail how to make the feature maps contain abundant spectral-spatial correlation information by combining 2-D CNN and 3-D CNN. In terms of layer types, input and output map dimensions, a detailed summary of the proposed model is given in Table I. Before feeding the deep network, we create neighboring patches $P \in \mathbb{R}^{S \times S \times B}$ by

TABLE I
LAYERWISE SUMMARY OF THE PROPOSED CACNN ARCHITECTURE WITH A WINDOWS SIZE 11 × 11 FOR ALL DATASETS

Layer(type)	Input Shape	Kernel Size	Padding	Stride	Filter Size	Output Shape
conv2D_1(Conv2D)	(11,11,10)	(3,3)	(0,0)	(1,1)	16	(9, 9, 16)
conv3D_1(Conv3D)	(11,11,10,1)	(3,3,4)	(0,0,0)	(1,1,1)	8	(9, 9, 7, 8)
reshape_1(Reshape)	(9,9,7,8)					(9, 9, 56)
conv2D_2(Conv2D)	(9,9,16)	(3,3)	(0,0)	(1,1)	32	(7, 7, 32)
conv3D_2(Conv3D)	(9,9,7,8)	(3,3,4)	(0,0,0)	(1,1,1)	16	(7, 7, 4, 16)
reshape_2(Reshape)	(7,7,4,16)					(7, 7, 64)
conv2D_3(Conv2D)	(7,7,32)	(3,3)	(0,0)	(2,2)	64	(3, 3, 64)
conv3D_3(Conv3D)	(7,7,4,16)	(3,3,2)	(0,0,0)	(2,2,2)	32	(3, 3, 2, 32)
reshape_3(Reshape)	(3,3,2,32)					(3, 3, 64)
conv2D_4(Conv2D)	(3,3,64)	(3,3)	(0,0)	(1,1)	64	(1, 1, 64)
conv3D_4(Conv3D)	(3,3,2,32)	(3,3,2)	(0,0,0)	(1,1,1)	64	(1, 1, 1, 64)
reshape_4(Reshape)	(1,1,1,64)					(1, 1, 64)
contact_1(Contact)	(9,9,16) and (9,9,56)					(9, 9, 72)
NonLocalBlock Conv_Block	(9,9,72)					(9, 9, 72)
Max_Pooling	(9,9,72)	(2,2)	(0,0)	(2,2)		(4, 4, 72)
conv2D_5(Conv2D)	(4,4,72)	(2,2)	(0,0)	(1,1)	72	(3, 3, 72)
contact_2(Contact)	(7,7,32) and (7,7,64)					(7, 7, 96)
NonLocalBlock Conv_Block	(7,7,96)					(7, 7, 96)
Max_Pooling	(7,7,96)	(2,2)	(0,0)	(2,2)		(3, 3, 96)
contact_3(Contact)	(3,3,64) and (3,3,64)					(3, 3, 128)
NonLocalBlock Conv_Block	(3,3,128)					(3, 3, 128)
contact_4(Contact)	(3,3,72) and (3,3,96) and (3,3,128)					(3, 3, 296)
conv2D_6(Conv2D)	(3,3,296)	(3,3)	(0,0)	(1,1)	296	(1, 1, 296)
contact_5(Contact)	(1,1,296) and (1,1,64) and (1,1,64)					(1, 1, 424)
conv2D_7(Conv2D)	(1,1,424)	(1,1)	(0,0)	(1,1)	*C*	(1, 1, *C*)
flatten (Flatten)	(1,1,*C*)					(*C*)

choosing an $S \times S$ neighborhood of the central pixel from $\mathbf{I}$, centered at the spatial location (α, β) and including B bands.[2]

For spatial correlation feature extraction, 2-D convolution operation is adopted as a basic element of spatial features extraction. As shown in Table I, the spatial extraction section includes four 2-D CNN (conv2D_$\{i\}_{i=1}^{4}$) layers to obtain feature maps with different spatial sizes. For Conv2D_1, Conv2D_2, and Conv2D_4, 3 × 3 spatial kernel with a subsampling stride of (1, 1) are all applied in these convolutional operation. The Conv2D_3 includes 2 × 2 spatial kernel with a stride (2, 2).

For spectral correlation feature extraction, 3-D convolutional operation is applied to capture spectral correlations in spectral dimension. As shown in Table I, four 3-D convolutional layers (conv3D_$\{i\}_{i=1}^{4}$) are utilized to obtain spectral features with different depth. For conv3D_1, conv3D_2, and conv3D_4, $3 \times 3 \times 4$, $3 \times 3 \times 4$, and $3 \times 3 \times 2$ spectral-spatial kernel with a subsampling stride of (1, 1, 1) are utilized in these convolution operation, respectively. The conv3D_3 includes $2 \times 2 \times 2$ spectral-spatial kernel with a stride (2, 2, 2).

We fuse the first three spatial correlation feature maps with spectral-spatial correlation feature maps, which make the fusion feature maps contain ample correlation information. And the fusion of fourth spatial correlation feature map and spectral-spatial correlation feature map is employed to assist classification.

B. Spatial Attention Module (NonLocalBlock)

In Section III, we observed that CACNN model including attention module NonLocalBlock can offer better classification results. However, how to account for the phenomenon that this attention module help CACNN improve classification performance? we conduct thorough research and dive into experiments to seek the answer to this question. Specifically, effectiveness of NonLocalBlock will be explained with theory and experiment in following steps.

In our proposed model, we apply the NonLocalBlock to emphasize correlation information in fused spectral-spatial feature maps. First, denote by $\boldsymbol{X}_{in} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ all pixels in a feature map, where N represents the number of pixels, and each pixel $\boldsymbol{x}_i$ is an E-dimensional vector, where E is the number of feature map channels. The output $\boldsymbol{O}_{\text{out}} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_N\}$. The pairwise similarity between every two feature pixels $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ can be modeled as

$$\begin{aligned} \boldsymbol{o}_i &= \text{softmax}(\phi(\boldsymbol{x}_i)^T \varphi(\boldsymbol{x}_j)) g(\boldsymbol{x}_j) \\ &= \frac{1}{\sum_{\forall j} exp^{(\phi(\boldsymbol{x}_i)^T \varphi(\boldsymbol{x}_j))}} (\phi(\boldsymbol{x}_i)^T \varphi(\boldsymbol{x}_j)) g(\boldsymbol{x}_j) \end{aligned} \tag{3}$$

where $\phi(\cdot)$, $\varphi(\cdot)$, and $g(\cdot)$ all denote a Conv2D with 1×1 spatial kernel of $E/2$ in Fig. 2. By multiplying matrices twice and once normalizing (softmax), we obtain a feature map $h \times w \times E/2$

[2] We set S to 11 and B to 10 in our experiments.

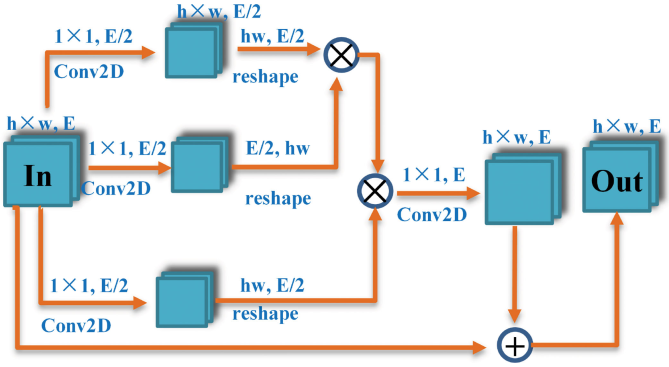


Fig. 2. Overall flowchart of NonLocalBlock.

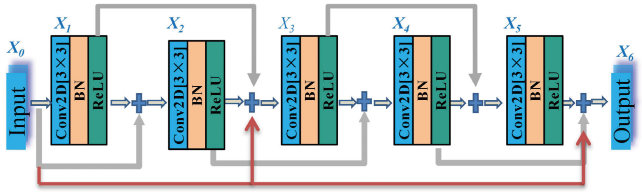


Fig. 3. Conv_Block with five composite layers.

(h and w are width and height of $\boldsymbol{X}_{\text{in}}$, separately). Then, it is transferred to $h \times w \times E$ with Conv2D including 1×1 spatial kernel of E. Finally, we add this feature map with $\boldsymbol{X}_{\text{in}}$ to $\boldsymbol{O}_{\text{out}}$.

Compared with convolution and pooling operation, NonLocalBlock considers weight of all position to establish the relationship between two pixels at a remote distance on the feature maps. And NonLocalBlock considers the correlation between the pixels in the entire feature map while convolution and pooling operation only consider the pixels of kernel size. As shown in Table I, the spatial size of fusion feature maps are $9 \times 9, 7 \times 7$, 3×3, which is not very big and generate a little computational complexity with NonLocalBlock. We confirm the validity of NonLocalBlock on improving classification performance from the theoretical viewpoint.

In Section III, we will confirm the validity of NonLocalBlock on improving classification performance from the experimental viewpoint.

C. Lightweight Dense Block (Conv_Block)

In this subsection, we will elaborate on reasons of using Conv_Block in the CACNN model. Inspired by Dense_Block [62], we design the Conv_Block with a little computational complexity due to only deliver previous features to a subsequent layer with the mathematical addition. It makes the spectral and spatial size keep constant between input and output, and meanwhile strengthens representative features and weakens the nonsignificant information.

In Fig. 3, $\boldsymbol{X}_0$ and $\boldsymbol{X}_6$ denote the input feature map and output feature map. $\{\boldsymbol{X}_i\}_{i=1}^{5}$ denoted feature maps are obtained by a series of Conv2D, BN, and ReLU operation. Each feature map is calculated as follows:

$$\begin{aligned}
\boldsymbol{X}_1 &= C_1^{2D}(\boldsymbol{X}_0)\\
\boldsymbol{X}_2 &= C_2^{2D}(\boldsymbol{X}_0 + \boldsymbol{X}_1)\\
\boldsymbol{X}_3 &= C_3^{2D}(\boldsymbol{X}_0 + \boldsymbol{X}_1 + \boldsymbol{X}_2)\\
\boldsymbol{X}_4 &= C_4^{2D}(\boldsymbol{X}_2 + \boldsymbol{X}_3)\\
\boldsymbol{X}_5 &= C_5^{2D}(\boldsymbol{X}_3 + \boldsymbol{X}_4)\\
\boldsymbol{X}_6 &= \boldsymbol{X}_0 + \boldsymbol{X}_4 + \boldsymbol{X}_5
\end{aligned} \tag{4}$$

where $\{C_i^{2D}(\cdot) | i = 1, \ldots, 5\}$ is defined as consecutive operations of each layer: Conv2D, BN, and ReLU. Such connectivity pattern strongly encourages feature transfer to deep layers. We confirm the validity of Conv_Block on improving classification performance from the theoretical viewpoint.

In Section III, the effectiveness of Conv_Block on improving classification performance will be described in detail.

D. Deep Multilayer Feature Fusion

To gain high-level discriminative features, we design deep multilayer feature fusion. After NonLocaLBlock and Conv_Block, feature maps contain abundant high-level spectral-spatial correlation information. As shown in Table I, to fuse multilayer feature maps (concat_4), we reduce spatial size of feature maps by Max-Pooling and Conv2D and make them with same spatial size. Moreover, the obtained feature maps are transformed to new feature maps ($1 \times 1 \times 296$) with conv2D so as to better assist classification. The concat_5 denotes fusion of deep spatial features, spectral-spatial features and obtained discriminative features, which is fed into a 1×1 convolutional layer to achieve expected results. The output vector is $\hat{y} = [\hat{y_1}, \hat{y_2}, \ldots, \hat{y_C}]$. And the truth one-hot label $y = [y_1, y_2, \ldots, y_C]$ is the number of land-cover categories. The loss function of CACNN is defined as

$$\mathcal{L} = -\frac{1}{n_\tau} \sum_{k=1}^{n_\tau} [y_k \log(\hat{y_k}) + (1 - y_k) \log(1 - \hat{y_k})] \tag{5}$$

where $\hat{y_k}$ is corresponding predicted labels for the kth training/test batch, y_k is the true label, and n_τ is the size of training/test batch. In order to optimize the $\mathcal{L}$, all parameters are optimized by Adam [63] at the same time.

III. Experimental Results and Analysis

In this section, we first introduce four hyperspectral imagery datasets used in experiments and then evaluate the proposed CACNN approach by a series of experiments. Three widely used quality metrics are utilized to evaluate the performance of classification methods, i.e., the overall accuracy (OA), the average accuracy (AA), and the Kappa coefficient (κ). Before our experiments, quantitative measures are obtained by averaging 20 random sampling runs.

TABLE II
NINE GROUND REFERENCE CLASSES IN THE AVIRIS IP DATASET

NO	Name	Samples
C1	Corn-notill	1428
C2	Corn-mintill	830
C3	Grass/pasture	483
C4	Grass/trees	730
C5	Hay-windrowed	478
C6	Soybeans-notill	972
C7	Soybeans-mintill	2455
C8	Soybean-clean till	593
C9	Woods	1265
Total		9234

TABLE III
NINE GROUND REFERENCE CLASSES IN THE ROSIS UP DATASET

NO	Name	Samples
C1	Asphalt	6631
C2	Meadow	18,649
C3	Gravel	2099
C4	Trees	3064
C5	Metal sheets	1345
C6	Bare soil	5029
C7	Bitumen	1330
C8	Bricks	3682
C9	Shadows	947
Total		42,776

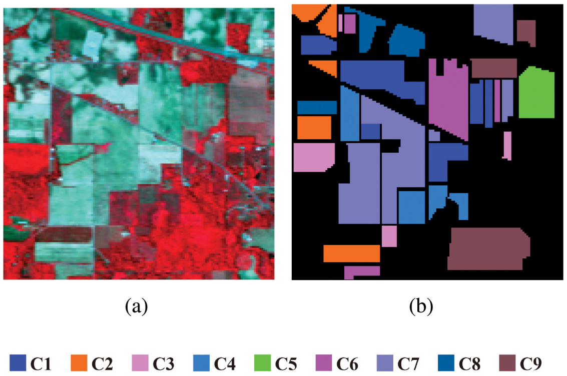

(a) (b)

Fig. 4. AVIRIS IP dataset. (a) RGB composite image of three bands. (b) Ground reference map.

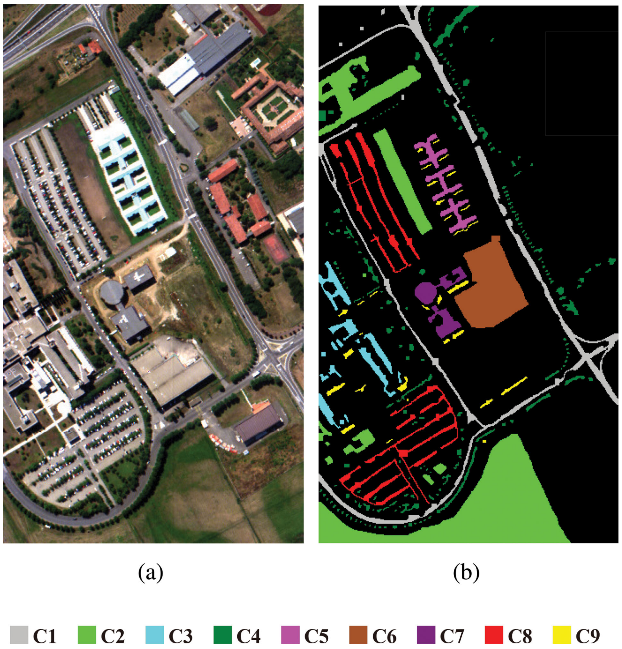

(a) (b)

Fig. 5. ROSIS UP dataset. (a) RGB composite image of three bands. (b) Ground reference map.

A. Hyperspectral Imagery Datasets

To evaluate the performance of the proposed approach, four hyperspectral imagery datasets have been considered in our experiments.

1) The first dataset is the Indian Pines (IP) image acquired by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS). This image, with size of 145 × 145 pixels, contains 220 bands covering the wavelength range of 0.4–2.5 μm, and the spectral and spatial resolutions are 10 nm and 17 m, separately. The ground truth contains 16 classes and some of the classes have fewer label samples. Therefore, 9 classes of samples with the largest number are selected, as shown in Table II. The false-color image and the ground reference map are shown in Fig. 4. In our experiments, there are 200 bands retained by removing 20 water absorption bands.
2) The second dataset is the University of Pavia (UP) image acquired by the Reflective Optics System Imaging Spectrometer (ROSIS). The ROSIS sensor collects 115 bands, covering the wavelength range from 0.43 to 0.86 μm. This image contains 610 × 340 pixels, with very high spatial resolution of 1.3 m per pixel. There are 9 ground reference classes of interests, as shown in Table III. The false-color image and the ground reference map are shown in Fig. 5. In our experiments, we work with 103 spectral bands after removal of noisy bands.
3) The third dataset was also gathered by the AVIRIS sensor over the region of Salinas Valley (SA), CA, USA and with 3.7 m per pixel spatial resolution. The Salinas scene consists of 512 × 217 pixels and contains 16 ground reference classes, as shown in Table IV. The false-color image and the ground reference map are shown in Fig. 6. In the experiments, 204 bands were used after removing 20 water absorption bands.
4) The fourth dataset is the University of Houston (HT) image acquired by an ITRES CASI-1500 sensor, covering a 0.38–1.05 μm spectral range with 48 bands at a 1-m ground sampling distance. This image is one of the multimodal optical remote sensing datasets released by the 2018 Data Fusion Contest of the IEEE Geoscience and Remote Sensing Society (GRSS) [64]. These datasets available on the website[3] were acquired by the National Center for Airborne Laser Mapping (NCALM) at HT on February 16, 2017, covering the HT campus and its surrounding urban areas. For the considered hyperspectral imagery, its original image size is 4172 × 1202 pixels and a subimage with size of 601 × 596 is selected. This subimage contains 12 ground reference classes of interests, as shown in Table V. The false-color image and the ground reference map are shown in Fig. 7.

[3] Online. [Available]: http://hyperspectral.ee.uh.edu/?page_id=1075

TABLE IV
16 Ground Reference Classes in the AVIRIS SA Dataset

NO	Name	Samples
C1	Weeds_1	2009
C2	Weeds_2	3726
C3	Fallow	1976
C4	Fallow_plow	1394
C5	Fallow_smoooth	2678
C6	Stubble	3959
C7	Celery	3579
C8	Grapes	11,271
C9	Soil	6203
C10	Corn	3278
C11	Lettuce 4wk	1068
C12	Lettuce 5wk	1927
C13	Lettuce 6wk	916
C14	Lettuce 7wk	1070
C15	Vinyard untrained	7268
C16	Vinyard trellis	1807
Total		54,129

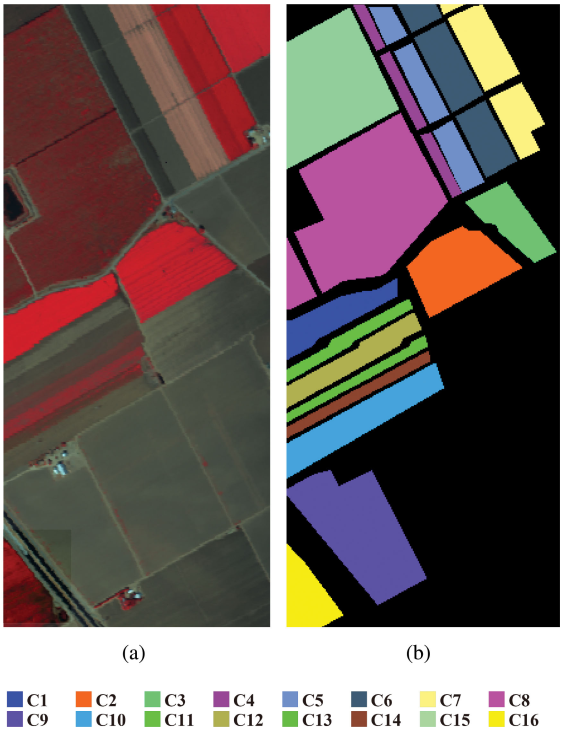

(a) (b)

Fig. 6. AVIRIS SA dataset. (a) RGB composite image of three bands. (b) Ground reference map.

TABLE V
12 Ground Reference Classes in the ITRES CASI-1500 HT Dataset

NO	Name	Samples
C1	Healthy grass	1453
C2	Stressed grass	5071
C3	Artificial turf	684
C4	Evergreen trees	6435
C5	Deciduous trees	2955
C6	Residential buildings	8063
C7	Roads	7039
C8	Sidewalks	12,497
C9	Major thoroughfares	17,901
C10	Paved parking lots	4156
C11	Cars	1101
C12	Stadium seats	6824
Total		73,579

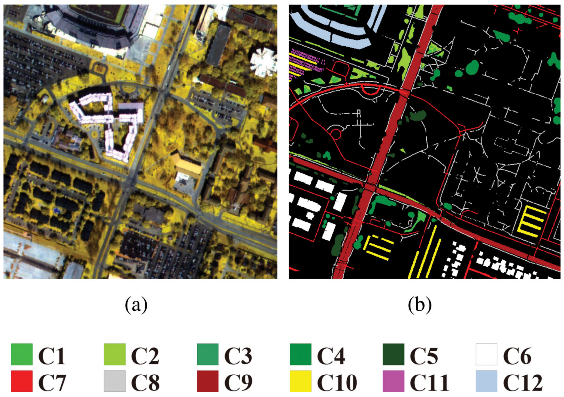

(a) (b)

Fig. 7. ITRES CASI-1500 HT dataset. (a) RGB composite image of three bands. (b) Ground reference map.

B. *Experimental Settings*

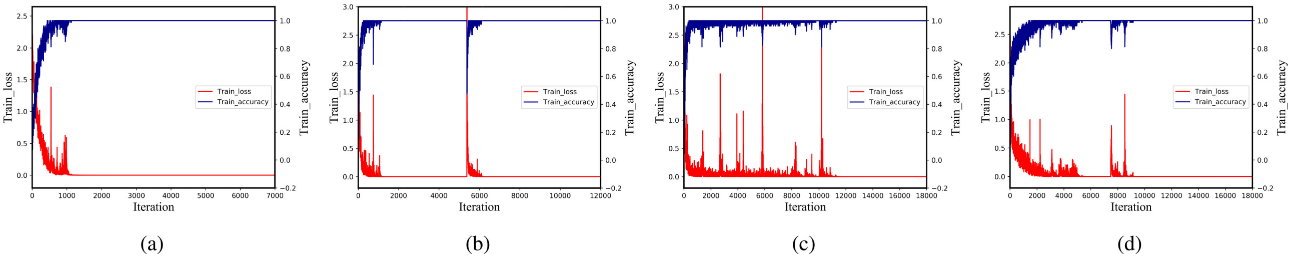
In this subsection, we will introduce the parameters used in our experiments. The learning rate is initialized by 0.0012 and decays every 5000 iterations by multiplying 0.99, which is to avoid the loss oscillating near the optimal value. Batch sizes are set to be 80 and 2000 for training and test, separately. We use a neighborhood window (S is fixed to 11) for 10 principal components (B is fixed to 10) as the input of CACNN for four datasets. As shown in Fig. 8, we will select the appropriate number of iterations for each data set based on training loss and accuracy. So the CACNN model is performed on IP, PU, SA, and HT datasets with 2000, 8000, 12000, and 10000 iterations for training.

C. *Experiments With the Data Sets*

Several related methods are compared to show the performance of the proposed CACNN approach. These methods can be divided into two categories: traditional machine learning based methods and deep learning-based methods. One is the classical SVM using Gaussian radial basis function kernel. The other includes five representative methods based on deep learning, Two-CNN, SPDF-SVM, 2-D CNN, 3-D CNN, SSRN, and pResNet. They are used as the comparison methods since these deep networks all use spectral-spatial techniques to extract high-level features.

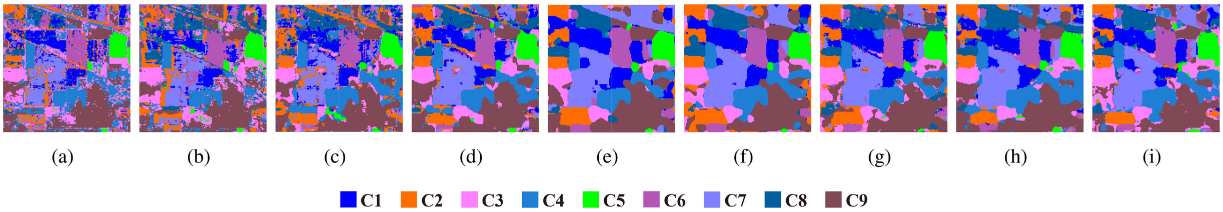
Four different experiments are employed to demonstrate the performance of the proposed method as follows.

1) In our first experiment, the proposed CACNN method is compared with the standard SVM, Two-CNN, 2-D CNN, 3-D CNN, SPDF-SVM, SSRN, pResNet, and FFDN classification methods using a training set made up of 200 available labels per class and use the rest as test sets for the IP, UP, SA, and HT datasets. In all the mentioned deep learning methods except Two-CNN, all datasets are processed with PCA algorithm and then 11 × 11 window and 10 spectral bands are used as input.
2) In our second experiment, OA obtained by the CACNN is compared with other spectral-spatial methods, in particular Two-CNN, 2-D CNN, 3-D CNN, SSRN, pResNet, and



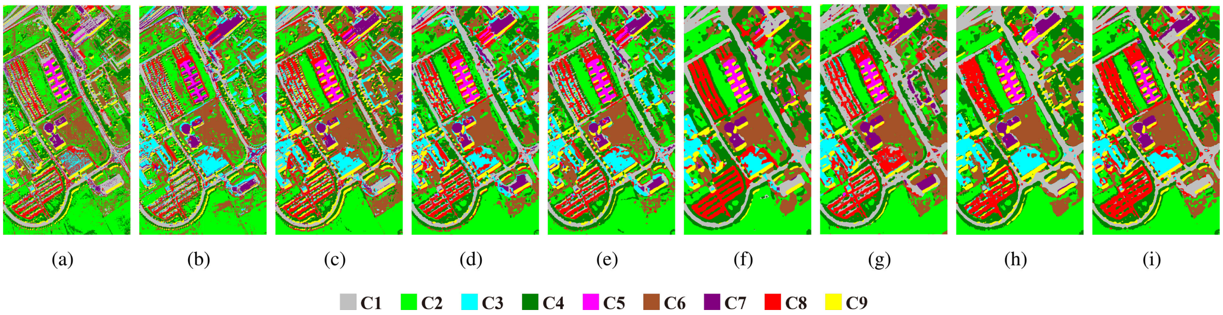
(a) (b) (c) (d)

Fig. 8. Training loss and accuracy as a function of the number of iterations for IP, UP, SA, and HT datasets. (a) For IP. (b) For UP. (c) For SA. (d) For HT.



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Fig. 9. AVIRIS IP dataset. (a) SVM. (b) Two-CNN. (c) 2-D CNN. (d) 3-D CNN. (e) SPDF-SVM. (f) SSRN. (g) pResNet. (h) FFDN. (i) CACNN.



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Fig. 10. ROSIS UP dataset. (a) SVM. (b) Two-CNN. (c) 2-D CNN. (d) 3-D CNN. (e) SPDF-SVM. (f) SSRN. (g) pResNet. (h) FFDN. (i) CACNN.

FFDN by considering different training percentages on IP, UP, SA, and HT datasets. Specifically, we use 1%, 3%, 5%, 8%, and 10% training data and set the input patch size of $11 \times 11 \times 10$ for 2-D CNN, 3-D CNN, SSRN, pResNet, FFDN, and CACNN.

3) In our third experiment, first, we design a comparative experiment to explain why we use lightweight_dense_block (Conv_Block) instead of original_dense_block. Then, to demonstrate that NonLocalBlock and Conv_Block could improve classification performance in CACNN approach, we design this experiment to validate.

4) In our fourth experiment, in term of training time, network parameters and FLOPs, the experiment compares many different deep learning approaches.

D. Results and Analysis

1) Experiment 1: Figs. 9–12 show classification maps obtained by different methods associated with the corresponding each dataset. From Figs. 9–12, it can be seen that the classification obtained by SVM is not satisfactory since some noisy estimations are still visible. Among deep learning methods, 3-D CNN performs better than 2-D CNN for the reason that the former can extract high-level spatial-spectral feature than the latter. Compared with 2-D CNN and 3-D CNN, the classification maps obtained by SPDF-SVM, SSRN, pResNet, and FFDN are better since deep convolution with good feature extraction method are applied. The best classification map is obtained by CACNN because it considers correlation of spatial-spectral features and hierarchical features between convolution layers. In these methods, original methods including Two-CNN, SSRN, and pResNet aim at extracting spectral-spatial feature using the different patch size and bands as input. To make the test fair, we apply PCA on all compared methods (beside Two-CNN) in the same manner to reduce spectral bands. For the classification map of SSRN, pResNet, FFDN, and CACNN, it is not easy to find out which classification map is best. For red region (Grapes category) of Fig. 11, We can see that the CACNN method handles this category better. Similarity, for gray region (Sidewalks category) and red region (Major thoroughfares category) of Fig. 12, more accurate classification to this two categories is obtained by CACNN.

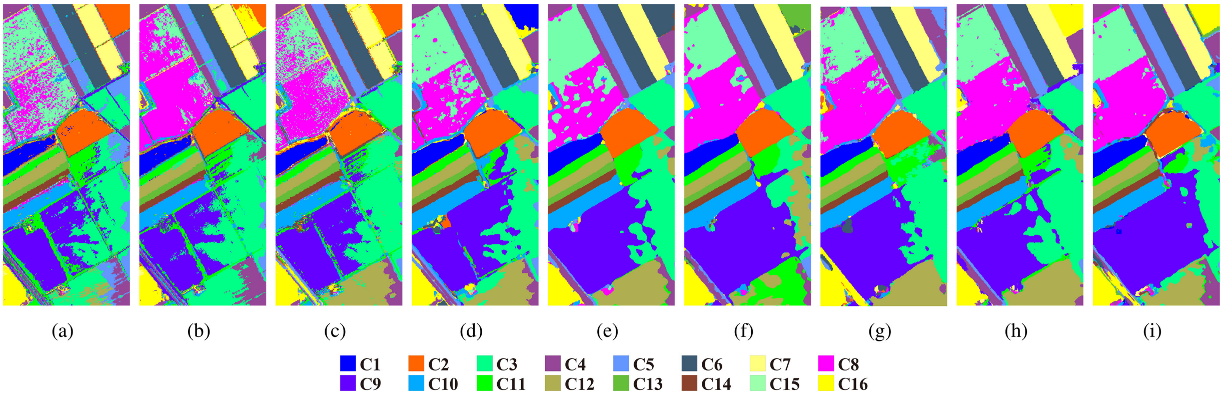


Fig. 11. AVIRIS SA dataset. (a) SVM. (b) Two-CNN. (c) 2-D CNN. (d) 3-D CNN. (e) SPDF-SVM. (f) SSRN. (g) pResNet. (h) FFDN. (i) CACNN.

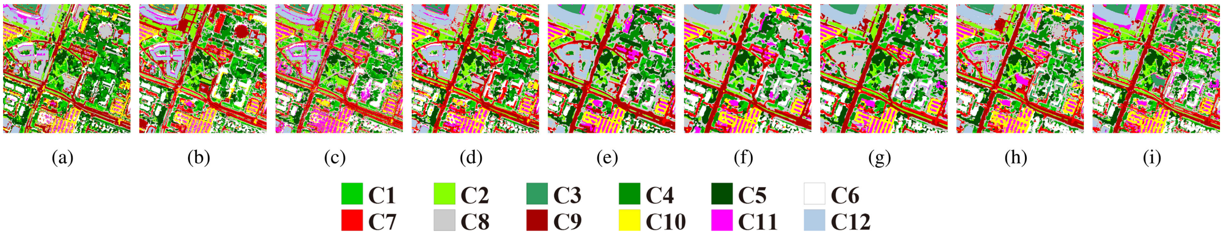


Fig. 12. ITRES HT dataset. (a) SVM. (b) Two-CNN. (c) 2-D CNN. (d) 3-D CNN. (e) SPDF-SVM. (f) SSRN. (g) pResNet. (h) FFDN. (i) CACNN.

TABLE VI
CLASSIFICATION ACCURACY (%) OF DIFFERENT METHODS ON THE IP DATASET

Class	SVM	Two-CNN	2-D CNN	3-D CNN	SPDF-SVM	SSRN	pResNet	FFDN	CACNN
C1	63.35	73.79	72.90	86.03	83.71	95.81	94.28	95.09	96.00
C2	70.64	66.07	72.17	91.61	97.62	99.58	99.19	98.98	99.14
C3	94.66	92.40	92.79	97.16	99.65	99.61	100	99.56	99.61
C4	98.80	99.49	98.66	99.40	99.43	99.91	99.55	99.67	99.53
C5	99.66	100	99.60	99.87	99.64	100	100	99.89	99.95
C6	74.91	80.73	81.70	92.05	97.02	96.85	97.98	97.98	98.10
C7	58.75	61.81	65.82	81.50	83.90	93.53	93.55	94.20	95.17
C8	65.79	81.97	70.94	93.18	95.93	99.08	99.80	99.53	99.25
C9	94.14	99.05	95.16	99.12	96.06	99.78	99.62	99.25	99.07
OA	73.39	77.80	78.28	89.90	91.07	96.88	96.70	96.96	**97.38**
AA	80.08	83.93	83.30	93.33	94.77	97.24	98.21	98.24	**98.42**
$\kappa \times 100$	68.82	73.87	74.37	87.99	89.37	96.26	96.08	96.36	**96.86**

TABLE VII
CLASSIFICATION ACCURACY (%) OF DIFFERENT METHODS ON THE UP DATASET

Class	SVM	Two-CNN	2-D CNN	3-D CNN	SPDF-SVM	SSRN	pResNet	FFDN	CACNN
C1	81.87	80.06	84.18	90.94	90.20	98.45	98.54	98.24	99.09
C2	81.81	84.24	93.45	93.99	91.84	97.90	99.00	98.90	99.16
C3	82.99	84.66	85.83	91.83	92.58	96.99	95.13	98.07	99.19
C4	95.48	96.14	95.68	96.98	98.95	98.36	98.93	97.82	98.76
C5	99.58	99.76	99.76	99.91	100	99.91	100	99.93	99.90
C6	86.73	82.82	94.43	94.99	96.21	97.48	98.44	99.46	99.26
C7	93.76	89.70	90.40	95.43	93.36	100	99.91	99.79	99.77
C8	82.96	82.65	88.58	93.52	95.40	99.27	98.08	98.52	99.03
C9	99.93	99.89	98.84	99.75	99.87	99.55	99.87	99.69	99.77
OA	84.66	85.00	91.69	94.01	93.35	98.19	98.67	98.78	**99.17**
AA	89.46	88.88	92.35	95.26	95.38	98.02	98.65	98.93	**99.33**
$\kappa \times 100$	80.03	80.38	88.96	92.02	91.19	97.57	98.21	98.36	**98.89**

TABLE VIII
CLASSIFICATION ACCURACY (%) OF DIFFERENT METHODS ON THE SA DATASET

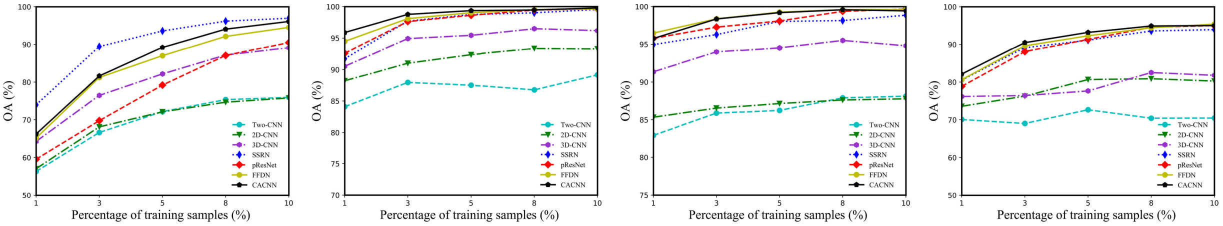
Class	SVM	Two-CNN	2-D CNN	3-D CNN	SPDF-SVM	SSRN	pResNet	FFDN	CACNN
C1	97.84	92.76	93.33	99.04	99.67	99.95	99.98	99.96	100
C2	94.38	95.77	98.41	99.78	100	100	100	99.96	99.93
C3	86.71	85.61	97.89	99.09	99.32	99.94	99.99	99.61	99.79
C4	99.41	97.58	97.67	99.30	99.58	99.91	99.96	99.88	99.79
C5	92.98	93.60	98.62	99.56	97.21	99.57	98.81	99.80	99.83
C6	99.33	97.87	98.02	99.51	100	100	100	99.77	99.96
C7	98.93	95.25	96.70	98.87	99.97	100	99.96	99.80	99.89
C8	55.30	69.26	70.51	81.29	86.14	87.13	91.47	93.77	95.69
C9	94.29	93.83	97.25	98.70	99.83	100	99.99	99.75	99.76
C10	85.61	84.72	89.25	94.56	99.16	99.39	99.64	99.43	99.26
C11	89.98	87.67	93.53	97.70	99.42	99.99	100	99.98	99.86
C12	99.77	94.16	95.45	98.63	99.94	100	100	99.95	99.96
C13	97.91	95.90	95.47	98.40	100	100	100	99.85	99.92
C14	91.03	93.84	94.41	98.48	100	99.98	100	99.90	99.89
C15	71.02	60.49	67.14	84.50	86.86	94.06	93.43	96.63	97.20
C16	91.66	80.89	86.80	98.01	100	99.99	99.97	99.86	99.75
OA	82.69	83.12	86.31	92.87	94.90	96.31	97.15	98.04	**98.55**
AA	90.39	88.70	91.90	96.59	97.94	98.44	98.95	99.24	**99.41**
$\kappa \times 100$	80.75	81.20	84.74	92.05	94.29	95.88	96.81	97.81	**98.38**

TABLE IX
CLASSIFICATION ACCURACY (%) OF DIFFERENT METHODS ON THE HT DATASET

Class	SVM	Two-CNN	2-D CNN	3-D CNN	SPDF-SVM	SSRN	pResNet	FFDN	CACNN
C1	91.93	89.23	92.94	93.50	93.52	92.49	95.32	93.79	94.51
C2	86.93	87.11	86.74	85.81	88.54	90.07	87.56	89.67	90.33
C3	100	98.08	97.92	99.55	99.85	100	100	99.96	99.87
C4	93.84	90.63	92.24	93.43	96.70	98.80	98.61	97.72	97.66
C5	91.48	91.42	90.20	93.54	95.59	98.24	98.67	96.97	97.83
C6	83.76	78.49	86.48	86.67	88.43	93.88	95.05	95.71	95.48
C7	21.88	47.51	54.84	49.52	64.10	61.16	70.96	75.33	77.59
C8	35.32	47.44	59.80	55.36	67.03	65.41	68.98	71.55	75.15
C9	49.06	49.83	61.25	65.86	71.87	84.34	80.48	82.62	84.03
C10	73.23	71.89	80.85	82.07	89.52	96.92	97.63	95.39	96.74
C11	54.01	69.42	86.64	86.83	95.75	95.62	96.84	97.57	98.06
C12	86.94	76.07	93.88	93.81	99.76	99.84	99.73	99.52	99.53
OA	61.99	66.36	73.90	74.04	80.73	84.63	85.28	86.57	**87.89**
AA	72.36	76.07	81.98	82.16	87.55	85.85	90.82	91.32	**92.23**
$\kappa \times 100$	56.99	61.90	70.23	70.32	77.90	82.27	83.06	84.54	**86.04**

Tables VI–IX present classification results for IP, UP, SA, and HT datasets, corresponding to our first experiment. As shown in Tables VI–IX, it summaries the global and class-specific accuracies of these methods mentioned above. From the table, it can be seen that other methods using spectral-spatial features yield higher classification accuracies when compared with SVM. For the SPDF-SVM and 3-D CNN, they are implemented by extracting spatial-spectral features, which is helpful to improve the classification performance. Therefore, the accuracies of them are more than 2-D CNN and Two-CNN. SSRN, pResNet, and FFDN are recent state-of-the-art spectral-spatial classification methods. Among these methods, CACNN gives the highest global accuracies. Compared with SSRN, pResNet, and FFDN, most classification accuracies of each dataset are very close. However, for Grapes (C8) of Table VIII, Sidewalks (C8) and Major thoroughfares (C9) of Table IX, CACNN approach gets top classification accuracy. The classification results from Tables VI–IX and Figs. 9–12 confirm the validity of CACNN.

2) Experiment 2: This experiment is designed to analyze the classification performance with different training percentages. Fig. 13 shows results obtained in our second experiment, where different training percentages are tested using IP, UP, SA, and HT datasets. In particular, Two-CNN, 2D-CNN, SSRN, pResNet, FFDN, and CACNN are tested considering 1%, 3%, 5%, 8%, and 10% of the labeled data for training. Specifically, in IP dataset, OA obtained by CACNN is only weaker than the best result got by SSRN. However, as the number of training labels increases, OA obtained by CACNN tends to be got by SSRN.



(a)

(b)

(c)

(d)

Fig. 13. OA as a function of the number of training samples for IP, UP, SA, and HT dataset. (a) For IP. (b) For UP. (c) For SA. (d) For HT.

TABLE X

COMPARSIONS OF THE CACNN METHOD USING ORIGINAL_DENSE_BLOCK AND LIGHTWEIGHT_DENSE_BLOCK, INVOLVING PARAMETERS, CONSUMING TIME (MINS), AND OVERALL ACCURACY OA(%) OVER FOUR DATASETS

Approach	Parameters	IP		UP		SA		HT	
		Times	OA	Times	OA	Times	OA	Times	OA
Original_dense_block	33,470,317	4.74	97.28	35.27	99.21	27.71	97.87	23.86	88.33
Lightweight_dense_block	2,359,797	1.84	97.37	6.58	99.17	9.46	98.55	8.36	87.89

TABLE XI

CLASSIFICATION ACCURACY (%) OF DIFFERENT METHODS ON FOUR DATASETS, "USE_NO," "USE_NONLOCAL," "USE_CONV_BLOCK," AND "USE_BOTH" DENOTE PROPOSED CACNN MODEL WITHOUT NONLOCALBLOCK AND CONV_BLOCK, PROPOSED CACNN MODEL ONLY USING NONLOCALBLOCK (WITHOUT CONV_BLOCK), PROPOSED CACNN MODEL ONLY UTILIZING CONV_BLOCK (WITHOUT NONLOCALBLOCK), PROPOSED CACNN MODEL INCLUDING THIS TWO MODULES, RESPECTIVELY

Approach	IP			UP			SA			HT		
	OA	AA	$\kappa\times100$	OA	AA	$\kappa\times100$	OA	AA	$\kappa\times100$	OA	AA	$\kappa\times100$
Use_no	92.22	95.06	90.72	98.41	98.43	97.87	97.82	99.09	97.56	85.04	90.48	82.80
Use_NonLocal	96.35	97.77	95.63	98.72	98.98	98.28	98.30	99.29	98.10	86.75	91.48	84.76
Use_Conv_Block	95.38	97.11	94.47	98.76	98.89	98.34	98.01	99.19	97.78	85.59	90.85	83.41
Use_both	97.38	98.42	96.86	99.17	99.33	98.89	98.55	99.41	98.38	87.89	92.23	86.04

Next, compared with other approaches in UP, SA, and HT datasets, it can be seen that our proposed CACNN achieves a good classification result. From Fig. 13, classification results confirm that the CACNN approach has stronger generalization ability.

3) Experiment 3: There are two parts in this experiment. The training set is made up of 200 labeled data per class and the input patch size is 11 × 11 with 10 spectral bands. The first part is to illustrate the advantages of using lightweight_dense_block in CACNN with the experiment. In Table X, "Lightweight_dense_block" and "Original_dense_block" separately denote in CACNN using lightweight_dense_block and original_dense_block. From the Table X, it can be seen that there is little difference in OA between the two approaches on four datasets, but "Original_dense_block" takes far more time and network parameters than "Lightweight_dense_block." So we select to add lightweight_dense_block (Conv_Block) in CACNN to ensure the stability of OA while reducing network parameters and time consumption. The other part is to validate the efficiency of NonLocalBlock and Conv_Block in CACNN model by the experiment. In Section II, we have verified effectiveness of this two Blocks from theoretical viewpoint. In the following part, we will certify the validity of this two blocks with the experiment.

TABLE XII

TRAINING TIME(MINS) ON FOUR DATASETS BY DEEP LEARNING METHODS: TWO-CNN, 2-D CNN, 3-D CNN, SSRN, PRESNET, FFDN, AND CACNN

Datasets	Two-CNN	2D-CNN	3D-CNN	SSRN	pResNet	FFDN	CACNN
IP	3.03	1.06	1.26	5.37	4.62	4.28	1.84
UP	4.56	1.60	2.02	8.04	5.50	5.06	6.58
SA	6.63	1.80	2.14	6.97	5.98	5.97	9.64
HT	8.90	2.46	2.69	11.03	6.58	6.05	8.36

In Table XI, "Use_no," "Use_NonLocal," "Use_Conv_Block," and "Use_both" denote proposed CACNN model without NonLocalBlock and Conv_Block, proposed CACNN model only using NonLocalBlock (without Conv_Block), proposed CACNN model only utilizing Conv_Block (without NonLocalBlock), proposed CACNN model including these two modules, respectively. By compared "Use_no" with "Use_NonLocal," it can be seen that accuracies of including NonLocalBlock experiment are higher than without NonLocalBlock. Then, we can achieve the similar results by comparing results of the other two experiments. According to comparison experiments mentioned above, we confirm the validity of NonLocalBlock on improving classification performance from the experimental viewpoint.

TABLE XIII
PARAMETERS AND FLOPs ON IP DATASET BY DEEP LEARNING METHODS: TWO-CNN, 2-D CNN, 3-D CNN, SSRN, pRESNET, FFDN, AND CACNN

Method	Two-CNN	2D-CNN	3D-CNN	SSRN	pResNet	FFDN	CACNN
Parameters	2,132,520	14,464	65,161	54,769	387,512	587,017	2,359,797
FLOPs	2,731,200	111,744	1,744,137	28,448,412	13,234,890	3,904,553	21,523,317

From Table XI, by compared "Use_no" with "Use_Conv_Block" and "Use_NonLocal" with "Use_both," it is obvious that the Conv_Block as an integral part of CACNN to improve classification performance. According to the comparison experiment mentioned above, the effectiveness of Conv_Block on improving classification performance is verified from experiments.

4) Experiment 4: This experiment is designed to demonstrate the CACNN on computational efficiency. All experiments are conducted with python language and tensorflow framework, and results are demonstrated on a PC equipped with an Intel Core i5 with 2.8 GHz, memory 8G, and Nvidia GeForce GTX 1060 3G graphics card. Table XII demonstrates the training time by deep learning methods on four datasets. As shown in XII, due to fewer iterations on IP dataset, CACNN uses less training time. In addition, CACNN consumes moderate training time on UP dataset, while consumes more training time on SA and HT datasets due to use more number of iterations. From the above analysis, CACNN is moderate on computational efficiency. Parameters and FLOPs on IP dataset by deep learning methods are shown in Table XIII. CACNN using deep network and several branches results in more network parameters and FLOPs.

IV. CONCLUSION

This article has proposed a deep collaborative attention network for HSI classification by combining 2-D CNN and 3-D CNN. This novel approach consists of three main steps, using 2-D CNN and 3-D CNN to extract spectral-spatial information, utilizing NonLocalBlock to emphasize spatial correlation features and Conv_block to extract high-level abstract correlation information, and applying hierarchical layer feature fusion to get discriminative features. And it has obtained better classification accuracies and strong generalization performance. We explore and discover that NonLocalBlock and Conv_block exploited in CACNN model is beneficial to improve classification performance. Although results obtained by the proposed approach are very encouraging, further enhancements such as extracting more efficient spectral-spatial correlation features and exploring the fusion strategy with more generalization ability should be pursued in future developments.

ACKNOWLEDGMENT

The authors would like to thank NCALM and the Hyperspectral Image Analysis Laboratory at HT for providing the ITRES CASI-1500 HT dataset, and the Image Analysis and Data Fusion Technical Committee of the IEEE GRSS for supporting the annual Data Fusion Contest. The authors would also like to thank the anonymous reviewers and editors for their constructive comments on this article.

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] X. Kang, S. Li, L. Fang, M. Li, and J. A. Benediktsson, "Extended random walker-based classification of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 1, pp. 144–153, Jan. 2015.

[3] X. Yang and Y. Yu, "Estimating soil salinity under various moisture conditions: An experimental study," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2525–2533, May 2017.

[4] D. Shen, J. Liu, Z. Xiao, J. Yang, and L. Xiao, "A twice optimizing net with matrix decomposition for hyperspectral and multispectral image fusion," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 4095–4110, 2020.

[5] L. Chang, M. Yong, X. Mei, C. Liu, and J. Ma, "Hyperspectral image classification with robust sparse representation," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 5, pp. 641–645, May 2016.

[6] N. Huang and L. Xiao, "Hyperspectral image clustering via sparse dictionary-based anchored regression," *IET Image Process.*, vol. 13, no. 2, pp. 261–269, 2019.

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

[8] 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.

[9] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4085–4098, Nov. 2010.

[10] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 809–823, Mar. 2012.

[11] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification via kernel sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 217–231, Jan. 2013.

[12] J. Liu, Z. Wu, Z. Wei, L. Xiao, and L. Sun, "Spatial-spectral kernel sparse representation for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 6, pp. 2462–2471, Dec. 2013.

[13] W. Li and Q. Du, "Joint within-class collaborative representation for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2200–2208, Jun. 2014.

[14] J. Liu, Z. Wu, L. Xiao, and H. Yan, "Learning multiple parameters for kernel collaborative representation classification," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2019.2962878.

[15] K. Abend, T. Harley, B. Chandrasekaran, and G. Hughes, "Comments on "on the mean accuracy of statistical pattern recognizers" by Hughes, g. f." *IEEE Trans. Inf. Theory*, vol. 15, no. 3, pp. 420–423, May 1969.

[16] G. Licciardi, P. R. Marpu, J. Chanussot, and J. A. Benediktsson, "Linear versus nonlinear PCA for the classification of hyperspectral data based on the extended morphological profiles," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 3, pp. 447–451, May 2012.

[17] J. Xia, J. Chanussot, P. Du, and X. He, "(Semi-) supervised probabilistic principal component analysis for hyperspectral remote sensing image classification," *IEEE J. Sel. Topics Appl. Earth Observations Remote Sens.*, vol. 7, no. 6, pp. 2224–2236, Jun. 2014.

[18] A. Villa, J. A. Benediktsson, J. Chanussot, and C. Jutten, "Hyperspectral image classification with independent component discriminant analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 12, pp. 4865–4876, Dec. 2011.

[19] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proc. IEEE*, vol. 101, no. 3, pp. 652–675, Mar. 2013.

[20] J. Liu, Z. Wu, J. Li, A. Plaza, and Y. Yuan, "Probabilistic-kernel collaborative representation for spatial-spectral hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 4, pp. 2371–2384, Apr. 2016.

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

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

[23] J. Liu, Z. Wu, L. Xiao, J. Sun, and H. Yan, "Generalized tensor regression for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 2, pp. 1244–1258, Feb. 2020.

[24] Y. Tarabalka, J. A. Benediktsson, and J. Chanussot, "Spectral-spatial classification of hyperspectral imagery based on partitional clustering techniques," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 8, pp. 2973–2987, Aug. 2009.

[25] 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.

[26] 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.

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

[28] S. Ren, K. He, R. Girshick, and S. Jian, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[29] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2014, pp. 3431–3440.

[30] Y. Chen, Z. Lin, Z. Xing, W. Gang, 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.

[31] J. Yue, W. Zhao, S. Mao, and H. Liu, "Spectral-spatial classification of hyperspectral images using deep convolutional neural networks," *Remote Sens. Lett.*, vol. 6, no. 4–6, pp. 468–477, 2015.

[32] Y. Li, H. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sens.*, vol. 9, no. 1, pp. 67–84, 2017.

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

[34] 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.

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

[36] S. Mei, J. Ji, J. Hou, X. Li, and Q. Du, "Learning sensor-specific spatial-spectral features of hyperspectral images via convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4520–4533, Aug. 2017.

[37] 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.

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

[39] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2015, pp. 4959–4962.

[40] Y. Li, H. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sens.*, vol. 9, 2017, Art. no. 67.

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

[42] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 25, no. 2, pp. 1097–1105, 2012.

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

[44] L. Fang, Z. Liu, and W. Song, "Deep hashing neural networks for hyperspectral image feature extraction," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 9, pp. 1412–1416, Sep. 2019.

[45] 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.

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

[47] H. Guo, J. Liu, Z. Xiao, and L. Xiao, "Deep CNN-based hyperspectral image classification using discriminative multiple spatial-spectral feature fusion," *Remote Sens. Lett.*, vol. 11, no. 9, pp. 827–836, 2020.

[48] Q. Liu, L. Xiao, J. Yang, and J. C. Chan, "Content-guided convolutional neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2020.2974134.

[49] R. Li, Z. Pan, Y. Wang, and P. Wang, "A convolutional neural network with mapping layers for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3136–3147, May 2020.

[50] G. Abdi, F. Samadzadegan, and P. Reinartz, "Spectral-spatial feature learning for hyperspectral imagery classification using deep stacked sparse autoencoder," *J. Appl. Remote Sens.*, vol. 11, no. 4, 2017, Art. no. 042604.

[51] 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.

[52] L. Wei, G. Wu, Z. Fan, and D. Qian, "Hyperspectral image classification using deep pixel-pair features," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 844–853, Feb. 2016.

[53] L. Zhu, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Generative adversarial networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5046–5063, Sep. 2018.

[54] M. E. Paoletti *et al.*, "Capsule networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2145–2160, Apr. 2019.

[55] Z. Li *et al.*, "Deep multilayer fusion dense network for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 1258–1270, 2020.

[56] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[57] J. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza, and J. Li, "Visual attention-driven hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 8065–8080, Oct. 2019.

[58] B. Fang, Y. Li, H. Zhang, and J. Chan, "Hyperspectral images classification based on dense convolutional networks with spectral-wise attention mechanism," *Remote Sens.*, vol. 11, no. 2, 2019, Art. no. 159.

[59] Z. Dong, Y. Cai, Z. Cai, X. Liu, Z. Yang, and M. Zhuge, "Cooperative spectral-spatial attention dense network for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, to be published, doi: 10.1109/LGRS.2020.2989437.

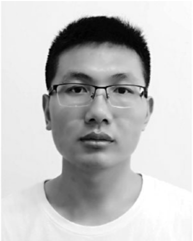
[60] X. Wang, R. B. Girshick, A. Gupta, and K. He, "Non-local neural networks," *IEEE/CVF Conf. Comput. Vis. Pattern Recogn.*, pp. 7794–7803, 2018.

[61] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Int. Mach. Learn.*, 2015, pp. 1–11.

[62] G. Huang, W. Liu, Zhuang, and Kilian, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 2261–2269.

[63] D. Kingma and J. Ba, "Adam: A method for stochastic optimisation," in *Proc. Int. Conf. Learn. Representations*, 2014, pp. 1–15.

[64] Y. Xu *et al.*, "Advanced multi-sensor optical remote sensing for urban land use and land cover classification: Outcome of the 2018 IEEE GRSS data fusion contest," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 6, pp. 1709–1724, Jun. 2019.



Hao Guo received the B.S. degree in industry engineering from Huaiyin Institute of Technology, Huaiyin, China, in 2017. He is currently working toward the M.D. degree in computer science and technology with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China.

His research interests include deep learning and hyperspectral image classification.

Jianjun Liu (Member, IEEE) received the B.S. degree in applied mathematics and the Ph.D. degree in pattern recognition and intelligence system from Nanjing University of Science and Technology, Nanjing, China, in 2009 and 2014, respectively.

He is currently an Associate Professor with the School of Internet of Things Engineering, Jiangnan University, Wuxi, China. Since 2018, he has been a Postdoctoral Researcher with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong. His research interests are in the areas of hyperspectral image classification, superresolution, spectral unmixing, sparse representation, computer vision, and pattern recognition.

Jinlong Yang received the Ph.D. degree in pattern recognition and intelligence system from Xidian University, Xi'an, China, in 2012.

He is currently an Associate Professor with Jiangnan University, Wuxi, China. His current research interests include target tracking, information fusion, and signal processing.

Zhiyong Xiao received the Ph.D. degree in optics and image processing from the Ecole Centrale Marseille, Marseille, France, in 2013.

He is currently an Associate Professor with Jiangnan University, Wuxi, China. His current research interests include image processing, computer vision, and pattern recognition.

Zebin Wu (Senior Member, IEEE) received the B.S. and Ph.D. degrees in computer science from Nanjing University of Science and Technology, Nanjing, China, in 2003 and 2008, respectively.

He is currently a Professor with the School of Computer Science, Nanjing University of Science and Technology, Nanjing, China. His research interests include hyperspectral image processing, high-performance computing, and computer simulation.