IEEE TGRS-2022-01845

Large Kernel Spectral and Spatial Attention Networks for Hyperspectral Image Classification

Genyun Sun, *Senior Member, IEEE*, Zhaojie Pan, Aizhu Zhang, *Member, IEEE*, Xiuping Jia, *Fellow, IEEE*, Jinchang Ren, *Senior Member, IEEE*, Hang Fu, Kai Yan

***Abstract*—Currently, long-range spectral and spatial dependencies have been widely demonstrated to be essential for hyperspectral image (HSI) classification. Due to the transformer superior ability to exploit long-range representations, the transformer-based methods have exhibited enormous potential. However, existing transformer-based approaches still face two crucial issues that hinder the further performance promotion of HSI classification: 1) treating HSI as 1D sequences neglects spatial properties of HSI, 2) the dependence between spectral and spatial information is not fully considered. To tackle the above problems, a large kernel spectral-spatial attention network (LKSSAN) is proposed to capture the long-range 3D properties of HSI, which is inspired by the visual attention network (VAN). Specifically, a spectral-spatial attention module is first proposed to effectively exploit discriminative 3D spectral-spatial features while keeping the 3D structure of HSI. This module introduces the large kernel attention (LKA) and convolution feed-forward (CFF) to flexibly emphasize, model, and exploit the long-range 3D feature dependencies with lower computational pressure. Finally, the features from the spectral-spatial attention module are fed into the classification module for the optimization of 3D spectral-spatial representation. To verify the effectiveness of the proposed classification method, experiments are executed on four widely used HSI data sets. The experiments demonstrate that LKSSAN is indeed an effective way for long-range 3D feature extraction of HSI.**

***Index Terms*—Deep learning, long-range 3D spectral-spatial feature extraction, spectral-spatial attention, large kernel attention (LKA), convolutional feed-forward (CFF), hyperspectral image (HSI) classification.**

I. Introduction

HYPERSPECTRAL images (HSI) contain hundreds of continuous and narrow spectral bands. Such valuable information can precisely characterize the physical properties and bring great convenience for land object recognition [1]. These characteristics have enabled HSI to be widely applied in environmental protection [2], and land-cover mapping [3], urban development monitoring [4]. HSI classification is one of the most critical tasks in these applications.

Over the few years, various methods have been developed for HSI classification. The earlier classification methods mainly focus on spectral features, i.e., support vector machine (SVM) [5], multiple regression (MLR) [6], and k-nearest neighbors [7]. However, the high dimensionality with a small number of labeled samples may lead to the Hughes phenomenon [8], which usually causes overfitting [9]. Feature selection [10] and feature extraction [11] are persuasive to alleviate these problems. Nevertheless, an increasing number of studies have demonstrated that it is challenging to distinguish confusing objects only utilizing spectral information [12, 13]. Meanwhile, the spatial context of HSI provides complementary features to their abundant spectral correlations for precise recognition [14]. Thus, effectively exploiting spatial and spectral information is essential for HSI classification. The traditional spectral and spatial feature extraction methods are partitioned into two categories [15]: spectral plus spatial feature extraction methods and spectral-spatial feature extraction methods. For spectral plus spatial feature extraction, numerous spatial algorithms, such as morphological operators [16], gabor filters [17], hypergraph structure [18], and markov random fields (MRFs) [19], were employed to extract spatial features. However, the feature-stacking-based methods lead to information redundancy, causing overfitting of classification models [20]. For spectral-spatial feature extraction, spectral and spatial information are jointly exploited to keep the most discriminative features for HSI classification. The typical methods include 3-D discrete wavelets [21], 3-D scattering wavelets [22], and 3-D gabor filters [23]. Nevertheless, these traditional methods extract the features of the original data in a shallow manner, which is hard to dig deep nonlinear spectral-spatial correlation information [24].

Compared with traditional feature extraction, deep learning (DL) can extract more abstract and discriminative features and

Manuscript received xxx; accepted December xxx. Date of publication xxx; date of current version xxx. This work was supported by the National Natural Science Foundation of China (41971292, 42271347), the National Key Research and Development Program (2019YFE0126700), *(Corresponding author: Kai Yan.)*

G. Sun, Z. Pan, A. Zhang, and H. Fu, are with the College of Oceanography and Space Informatics, China University of Petroleum (East China), Qingdao, 266580, China, and also with the Laboratory for Marine Mineral Resources, Qingdao National Laboratory for Marine Science and Technology, Qingdao, 266237, China (e-mail: panzhj99@163.com; genyunsun@163.com; zhangaizhu789@163.com; hangf_upc@163.com).

X. Jia is with the School of Engineering and Information Technology, University of New South Wales at Canberra, Canberra, ACT 2600, Australia. (e-mail: x.jia@adfa.edu.au).

J. Ren is with the School of Computer Sciences, Guangdong Polytechnic Normal University, Guangzhou 510665, China, and also with the National Subsea Centre, Robert Gordon University, Aberdeen AB21 0BH, U.K. (e-mail: jinchang.ren@ieee.org).

Kai Yan is with the Faculty of Geographical Science, Beijing Normal University, Beijing 100875, China, and also with the School of Land Science and Techniques, China University of Geosciences, Beijing 100083, China (e-mail: kaiyan.earthscience@gmail.com).

has been widely used in HSI classification. Recurrent neural networks (RNNs) [25], autoencoders (AEs) [26], deep belief networks (DBNs) [27], fully convolutional networks (FCNs) [28, 29] and convolutional neural networks (CNNs) [30] are the mainstream DL architectures. Among these methods, patch-based CNN is the most popular framework, mainly because it elegantly integrates spectral features with spatial-contextual information from HSI data in its unique local connection. Note that patch-based methods may face problems such as multiple computations and insufficient global information mining. Fortunately, their advantages in the following three aspects allow them to remain in the mainstream. 1) Local feature mining is more convenient. The features related to classification performance mainly exist locally, and spatial partitioning helps to reduce the interference of redundant information. 2) Sample production is more convenient. The model can be optimized based on point samples, which considerably reduces the sample production time in practical applications. 3) The model is more flexible. The end-to-end model is not suitable for the problems of scale diversity and high spatial heterogeneity, while the patch-based method can improve the migration of the model in different geographical environments with different patch sizes.

In recent decades, numerous CNN-based methods have made prosperous progress in HSI classification. In [31], CNNs were used to extract hierarchical deep spatial features. Lee et al. [32] introduced three-dimensional convolutional network (3DCNN) to explore local spectral-spatial contextual interactions simultaneously. While the CNNs enhanced the performance of methods, the extraction of spatial and spectral features is still insufficient. Hence, Xu *et al*. [33] designed a dual-tunnel CNN in which one-dimensional convolutional network (1DCNN) was employed to exploit the spectral features, two-dimensional convolutional network (2DCNN) was comprised to highlight the spatial characteristic. Li *et al*. [34] proposed an automatic clustering-based two-branch network to extract spectral and spatial features. However, the methods based on dual-tunnel or two-branch do not meet the characteristics of HSI spectral-spatial integration and thus inherently overlook the ample spectral-spatial correlation information. To resolve these issues, 3D spectral-spatial feature extraction is required to reveal the 3D inherent structure of HSI [35]. Zhong *et al*. [36] designed an end-to-end spectral-spatial residual network (SSRN) that takes raw 3-D cubes as input data and introduced residual connections [37] for HSI classification. Sellami *et al*. [38] combined band clustering based on spectral clustering with 3DCNN to find informative and distinctive spectral-spatial features. Jia *et al*. [39] adopted spectral-spatial schroedinger eigenmaps (SSSE) and dual-scale convolution to obtain the joint spatial-spectral correlation information.

However, the aforementioned-based CNN methods treat local features equally, which is easy to cause the loss of crucial information. Fortunately, the attention mechanism can alleviate this issue since it can adaptively capture salient parts through constructing the weight map. In [40], a squeeze-and-excitation network (SEnet) was constructed to recalibrate channel-wise feature responses. Based on this structure, spectral attention classification networks were developed and obtained favorable applications [41, 42]. In [43], a convolutional block attention module (CBAM) was proposed, which can learn channel dependencies and spatial connections by spectral and spatial weights. Inspired by CBAM, Ma *et al*. [44] proposed the double-branch multi-attention mechanism network (DBMA) to excavate the spectral and spatial information. Pu *et al*. [45] developed a learnable spectral-spatial attention module (SSAM) to focus on spectral-spatial correlations. Nevertheless, SEnet and CBAM do not fully consider the long-range correlation over local spectral and spatial features. Therefore, a dual attention network (DAN) [46] was proposed, which can effectively mitigate the problem through channel and spatial self-attention modules. Subsequently, [47-49] introduced self-attention to set various weights for extracted long-range spectral-spatial features.

Transformer [50] is a new self-attention-based network and further enhances performance with its remarkable capability of exploiting long-range features. In [51], a transformer was applied for the first time in the HSI classification task. Xue *et al*. [52] proposed a local transformer with a spatial partition restore network (SPRLT-Net) to model locally detailed spatial discrepancies. Similarly, Sun *et al*. [53] constructed a spectral-spatial feature tokenization transformer (SSFTT) method to capture spectral-spatial features and high-level semantic features. More recently, Yang *et al*. [54] developed an interactive learning framework to achieve multi-scale, detail-aware, and space-interactive classification based on different transformer structures. Although the above-mentioned transformer-based approaches have made many efforts for long-range spectral-spatial features extraction and gain better performance, there are still two crucial obstacles that can be summarized: 1) treating HSI as 1D sequences inevitably neglect 3D properties of HSI, 2) the dependence between spectral and spatial information is not fully considered.

To address these two limitations, we propose the following three strategies. 1) We use large kernel 3D convolution to extract spectral and spatial features while maintaining the HSI 3D structure. Compared with the transformer-based methods, the method is more facilitated to exploit the spectral-spatial correlation characterization and can still effectively leverage long-range dependencies. 2) Large kernel convolution causes a massive computational burden. Therefore, we introduce convolutional decomposition to alleviate this problem. 3) The transformer-based methods can adaptively extract salient features by attention mechanism. Hence, we design an attention structure to enhance vital information. Unlike the existing attention mechanism, this structure is encapsulated in a large kernel 3DCNN and can more effectively exploit the long-range spectral-spatial dependencies, which is more in line with HSI.

Based on the above strategies, we proposed a patch-based large kernel spectral-spatial attention network (LKSSAN) to capture long-range 3D spectral-spatial features, which is inspired by Visual Attention Network (VAN) [55]. LKSSAN firstly uses principal component analysis (PCA) to reduce the HSI spectral dimension, followed by image segmentation to generate 3D patches as the input of the model. Then, the spectral-spatial attention module and classification module are

designed to exploit long-range 3D features and generate labels for patch center pixels, respectively. The spectral-spatial attention module follows a modular design, and the basic module is composed of a scale expansion block, several hybrid blocks, and a layer normalization. The hybrid block is the main feature processing structure of LKSSAN and contains two core operations, large kernel attention (LKA) and convolutional feed-forward (CFF). LKA captures long-range spectral-spatial characters by large kernel convolution with a 3D spectral-spatial attention mechanism. The CFF integrates the 3D features along the spatial and spectral dimensions by employing MLP and depth-wise convolution. Both LKA and CFF take the 3D patch as input, which make the spectral-spatial attention module easy to exploit 3D spectral-spatial feature associations, thereby improving the performance of the feature extraction. The classification module is a specially designed simple multilayer perceptron (SMLP), which can take advantage of the features extracted by the spectral-spatial attention module to complete high-precision HSI classification. The three major contributions of this paper are listed as follows.

1) A large kernel spectral-spatial attention network (LKSSAN) is proposed to extract the long-range 3D properties of HSI. In this model, the spectral-spatial attention module and classification module are adopted to emphasize, exploit, and distinguish the long-range spectral-spatial features with lightweight structures, thereby relieving the problems of the 3D feature utilization.

2) Inspired by VAN, the spectral-spatial attention module integrates the attention mechanism and convolutional decomposition into large kernel 3DCNN to exploit long-range dependencies. To further explore the 3D characterization, the module simultaneously introduces the CFF. With this structure, 3D properties can be fully identified with a lower computational burden, which is more suitable for high spatial and spectral resolution 3D HSI.

3) To promote spectral-spatial attention module performance, a patch-based sampling strategy, and a classification module are designed to generate 3D image patches and the labels of land cover. The patch-based sampling strategy helps excavate local spatial and spectral information. The classification module can be perfectly integrated with the spectral-spatial attention module, and thus effectively aggregates and classify features.

The remainder of this article is organized as follows. The related works of the proposed method are presented in Section II. In Section III, we describe the proposed method in detail. The experiments and results are presented and discussed in Section IV. Finally, concluding remarks are provided in Section V.

II. Related Works

In this section, we will provide a brief introduction to Patch-based CNN, Attention mechanism, and the VAN, which are also the crucial techniques of the proposed methods.

A. Patch-based CNN

Based on the input method of HSI data, CNNs mainly has two structures: patch-based CNNs and pixel-based CNNs. The comparison of the patch-based and pixel-based methods is shown in Fig. 1. Patch-based CNNs handle images as a multidimensional input, instead as a single vector and considers the spatial contexts of image pixels explicitly. To facilitate the utilization of spatial information, it uses HSI patches to assist pixel classification.

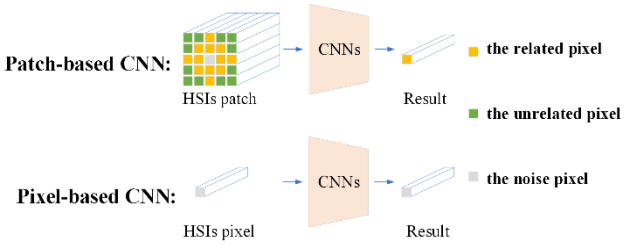

Fig. 1. The patch-based CNN and pixel-based CNN

Specifically, Let $F \in R^{H\times W\times C}$ as the HSI for classification, where H, W, and C denote the length, the width, and the number of bands of F, respectively. Denote the patch size is w, then F is decomposed into some image patches, which number is $H\times W$, and the data patch can be represented as $X \in R^{w\times w\times c}$, where $w\times w$ denotes the size of patch and c denotes the number of bands. After that, the data cube is input into the CNN model for extracting spectral and spatial information and classification. Finally, the model output tensor $T \in R^{1\times s}, T=[t_1,t_2,...,t_i,...,t_{s-1},t_s]$, where s represents the total number of categories in the data set. If the index corresponds to the largest data in the T is i, the classification of the label of center pixel of the patch is i.

B. Attention Mechanism

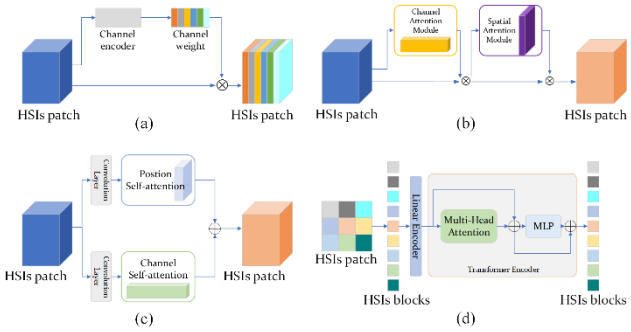

Fig. 2. Overview of currently well-recognized attention models for the HSI classification task, such as (a) SEnet [40], (b) CBAM [43], (c) DAN [46], and (d) Transformer [50].

The attention mechanism originated in computer vision, embedding it into the network can promote the representation capacity of the critical features, and Fig. 2 exhibits the existing mainstream attention structure. Attention models can be divided into spectral attention models and spatial attention models. SEnet [40] is the typical representative of the spectral attention model, which can enhance crucial information on spectral dimension through spectral weight maps. The spatial attention models are often combined with spectral attention structures, and the typical models include convolutional block attention module (CBAM) [43] and dual attention network

(DAN) [46]. CBAM first uses spectral attention to enhance the critical HSI spectral information and then uses spatial attention to exploit the spatial correlation. DAN is a two-branch structure, including a channel attention module (CAM) and position attention module (PAM), which can complete spectral attention and spatial attention, respectively. The association between long-range spectral features helps improve the classification effect, and transformer [50] is better at processing the features than the state-of-the-art attention networks.

The attention models mentioned above have been widely applied in HSI classification [54], but they cannot achieve 3D spectral-spatial attention. The main reason for this situation is that the existing attention structures have a series of deficiencies as follows.

1) Spectral attention models: As an early attention architecture, spectral attention can describe long-range dependencies and boost the discriminability of spectral features, which can enhance the critical spectral information for classification. However, these models have not taken full consideration of the discriminative spatial features, which limits their further development in HSI classification.

2) Spectral-spatial attention models: Unlike spectral attention models, these models combining spectral and spatial attention can facilitate methods to exploit spectral and spatial information. Nevertheless, none of these models can achieve 3D spectral-spatial attention information. For instance, CBAM processes the spectral and spatial features in turn, and DAN exploits spectral and spatial contextual through parallel subnetworks. They all ignore the inherent spectral-spatial correlation information in HSI, which does not meet the characteristics of HSI spectral-spatial integration.

3) Transformer-based models: these methods can excavate the associated information in long-range data and have an advantage in processing HSI data. However, because the transformer originated from NLP, its block idea destroys the spatial structure of HSI.

In summary, none of the mainstream attention-based networks for HSI classification are compatible with the spectral-spatial unity while extracting image features. Therefore, they inherently overlook long-range 3D dependencies regardless of how to combine the spectral and spatial attention structures or what structures of the transformer are employed to mine the image features.

C. Attention Mechanism

VAN is a novel attention-based model that can simultaneously enhance spatial and channel information through a three-dimensional attention map built by large kernel convolution in visual tasks. VAN has a simple hierarchical structure, as overviewed of VAN as shown in Fig. 3. The basic block of VAN first performs down-sampling, and then feature extraction is completed through LKA and CFF [56], and the last step is layer normalization. LKA and CFF are the core modules of VAN. Large kernel convolution means that the computational burden of image classification will be larger. To address this problem, VAN designs a large convolution kernel decomposition method to generate spectral-spatial attention maps. VAN divides a large kernel convolution into three components: a spatial local convolution (depth-wise convolution), a spatial long-range convolution (depth-wise dilation convolution), and a channel convolution (1×1 convolution).

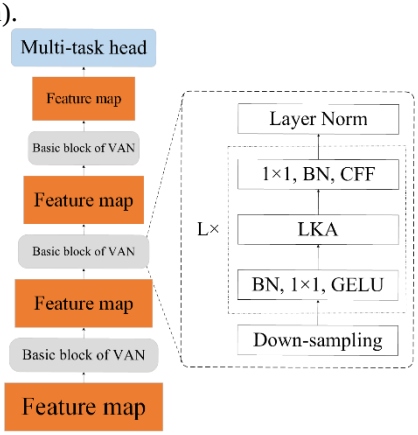

Fig. 3. The original VAN consists of four stages, the scale of the feature map is compressed at each step, and finally uses the multi-task head to complete different downstream tasks.

Suppose F is the input feature map, the operation of using a large convolution kernel to complete the attention can be expressed as:

$$Attention = conv_{(1\times1,1)}(DW_d(DW(F))) \quad (1)$$

$$F^{LKA} = Attention \otimes F \quad (2)$$

where $Attention$ is the spectral-spatial attention map, $con(k\times k, g)$ denotes the two-dimensional convolution function with a kernel size of k and a group value of g, $DW_d(\cdot)$ is depth-wise dilation convolution, $DW(\cdot)$ is depth-wise convolution, and $\otimes$ denotes the element-wise product.

After the LKA, the feature map is processed by CCF. The difference between CFF and feed-forward is that a depth-wise convolution is added between the 1×1 convolution and activation function, which can enhance spatial information in different channels. The processing result of the CFF can be expressed by:

$$F^{CFF} = conv_{(1\times1,1)}(GELU(DW(conv_{(1\times1,1)}(F^{LKA})))) \quad (3)$$

where $GELU(\cdot)$ denotes an activation function called gaussian error linear units (GELU), and the functions that have been introduced above will not be repeated.

Nonetheless, there are still some problems in VAN which hinder its development. 1) For input scale, although VAN perform outstandingly in solving the problem of natural image, it is not suitable for HSI. 2) For feature utilization, VAN can extract high-quality features but cannot achieve classification. However, it is challenging to build a classifier that can fully integrate with feature extraction [57]. Hence, we introduced VAN into the patch-based model, which not only solves the scale problem but also promotes the mining of local spectral and spatial information. In addition, we specially designed simple multilayer perceptron (SMLP) classifier to directly establish the relationship between the obtained features and the classification of center pixel of the patch.

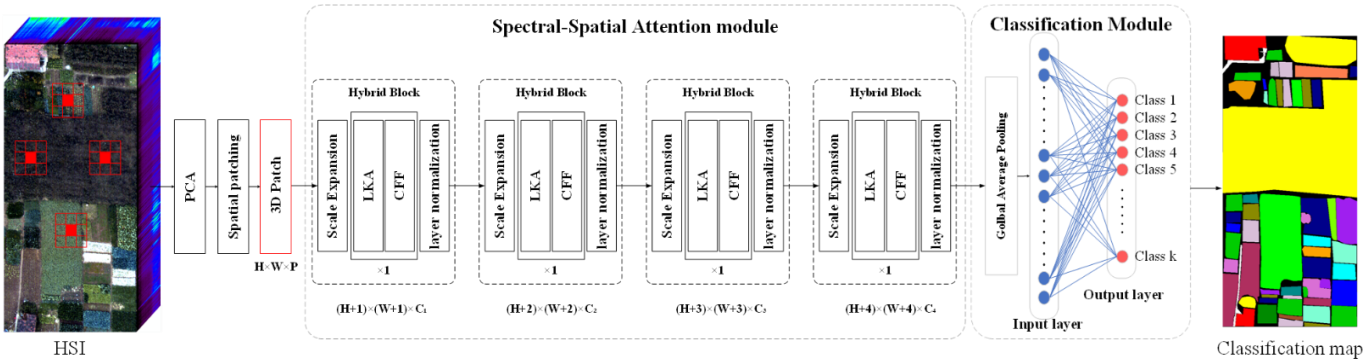

Fig. 4. Overview illustration of the proposed LKSSAN for the HSI classification task. LKSSAN consists of two modules, i.e., spectral-spatial attention module and classification module, and spectral-spatial attention module has two core components: LKA and CCF.

III. LKSSAN FOR HSI CLASSIFICATION

The proposed LKSSAN is a patch-based attention HSI classification network, as shown in Fig. 4, which includes data preparation with spatial patching, spectral-spatial attention module, and classification module. The challenges of DL for HSI classification are: 1) it is difficult to exploit long-range 3D features and 2) these methods that are employed to capture long-range information impose a huge computational burden on the facility. Spatial patching is introduced for generating a large number of 3D patches to be the input of the model, which is more favorable for the model to use the local features indispensable for classification. In the spectral-spatial attention module, the LKA and CFF are introduced to guide the network to be more focused on the most informative long-range 3D features of the input data, which is achieved by adaptively weighting the different pixel blocks. LKA is used to emphasize spectral-spatial cohesion features with lightweight structures by progressively learning 3D feature embedding. CFF is then introduced to adaptively aggregate the spatial and spectral dimensional features. Differing from the feed-forward in the transformer, the CFF can flexibly recover the spatial information of the semantic features. In the classification module, the 3D feature maps with long-range spectral and spatial dependencies extracted by the spectral-spatial attention module are further refined and reinforced through SMLP to generate class probability maps. Apart from that, to facilitate the utilization of the spectral-spatial connections in the multi-level feature maps, a scale expansion block is used to refine the spatial and channel sizes of feature maps in the spectral-spatial attention module.

A. Spectral-Spatial Attention Module

It is essential to explore discriminant 3D spectral-spatial features for more effective HSI classification. Although previous studies have applied spatial contextual information and long-range spectral correlation features for HSI classification, the correlation information between spectral and spatial features is ignored. Fortunately, 3D spectral-spatial feature extraction can reveal the 3D inherent structure of HSI. Hence, we design a spectral-spatial attention module that follows a modular design and can effectively extract 3D spectral-spatial features by a scale expansion block, L_i hybrid blocks, and layer normalization, where i is the number of hybrid blocks of the ith basic block. Let $F_{in} \in R^{(2r+1)\times(2r+1)\times P}$ be the input 3D patch, where r is the number of pixels separating the central pixel from the edge and P denotes the number of the first several principal components (PCs) selected by PCA. Each component of basic block is described in the following.

1) Scale expansion block

To excavate spatial and spectral correlation features, we first feed feature map into a scale expansion block to increase the number of channels and the spatial scale of patch, as shown in Fig.5. We can see that the scale expansion block first expands the spatial scale of the patch by using the spatial depth convolution with a convolution kernel of size 2, and then the channel convolution is used to expand the channel features of the patch. The process can be expressed vividly as follows:

$$A_i = con_{(1\times1,1)}\left(con_{(2\times2,c_{i-1})}\left(F_{i-1}\right)\right) \tag{4}$$

where $A_i \in R^{(2r+1+i)\times(2r+1+i)\times c_i}$ is the feature map after scale expansion, F_{i-1} is the output of i-1th basic block, c_{i-1} is the number of channels of F_{i-1}. When i is 1, the F_{i-1} is F_{in}. According to the above description, we can learn that the scale expansion module is simple and provides a larger feature space for subsequent feature mining. Note that although the convolution can only affect the edge features of 3Dpatch during the scale expansion, the influence range of this edge feature will be effectively expanded after the base block processing.

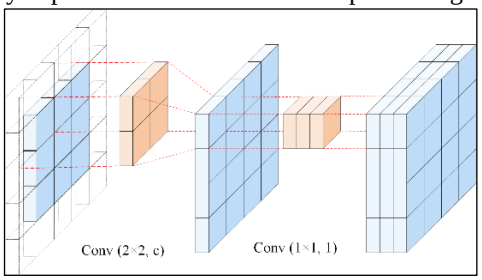

Fig. 5. Flowchart of the proposed scale expansion block.

2) Large Kernel Attention for exploiting 3D features

After the scale expansion block, the A_i is used as the input of hybrid blocks, and large kernel convolution is employed to realize the spectral-spatial attention operation. Fig. 6 shows the structure of the LKA.

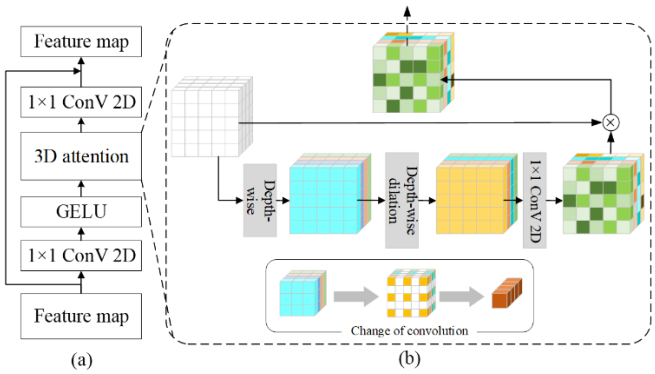

Fig. 6. LKA for spectral-spatial feature learning. (a) Detailed structure of LKA in LKSSAN. (b) Changes of feature map when using LKA.

Specifically, the LKA firstly uses 1×1 convolution to improve the flexibility of the model. Next, GELU is adopted to model the nonlinear features. And then, large kernel convolution decomposition is used to obtain a three-dimensional spectral-spatial attention weight map $W_i \in R^{(2r+1+i)\times(2r+1+i)\times c_i}$ by calculating the spectral and spatial correlation between pixels. This process can be expressed as

$$B_i^j = GELU(conv_{(1\times1,1)}(E_i^{j-1})) \tag{5}$$

$$W_i^j = conv_{(1\times1,1)}(DW_d(DW(B_i^j))) \tag{6}$$

where E_i^{j-1} is the output of CFF of j-1th hybrid block. When j is 1, the E_i^{j-1} is A_i. It can be deduced from (6) that the pixel block in each position of $B_i^j \in R^{(2r+1+i)\times(2r+1+i)\times c_i}$ has corresponding weight value at the same position in W_i^j, and the pixel block of W_i^j can effectively establish dependencies with long-range features by adaptive weighted aggregation of convolution. Then, a weighted matrix is obtained by

$$C_i^j = B_i^j \otimes W_i^j \tag{7}$$

where $\otimes$ denotes the element-wise product, $C_i^j \in R^{(2r+1+i)\times(2r+1+i)\times c_i}$ is the 3D attention. According to equation (7), we can speculate that each pixel block of C_i^j has been fused with neighborhood features based on the same local statistical properties of convolutional modeling by weighting, which will effectively suppress high-frequency noise and reinforce the crucial long-range dependencies.

Finally, a residual connection and 1×1 convolution is performed to obtain the output of LKA $D_i^j \in R^{(2r+1+i)\times(2r+1+i)\times c_i}$. It can be expressed by:

$$D_i^j = A_i^j + conv_{(1\times1,1)}(C_i^j) \tag{8}$$

LKA can capture long-range relationship with slight computational cost and parameters through the decomposition of large kernel convolution. After obtaining long-range relationship, LKA can estimate the importance of a pixel block and generate spectral-spatial attention map. As shown in Fig. 6, LKA combines the advantages of convolution and self-attention, which enable it to take the local contextual spatial information, and large receptive field into consideration and realize 3D spectral-spatial attention. The architecture of LKA is shown in Fig. 6. It consists of two 1×1 convolution, an activation function and a spectral-spatial attention. The output spectral-spatial features of this subnetwork are fed into CFF for further information fusion and extraction.

3) Convolutional Feed-Forward for spectral and spatial features fusion

The mechanism of combining attention with feed-forward has been proven to be an effective strategy in transformer, which can enhance information exchange between long-range information. Unlike transformer, the spectral-spatial attention module replaces feed-forward with CFF, as shown in Fig. 7.

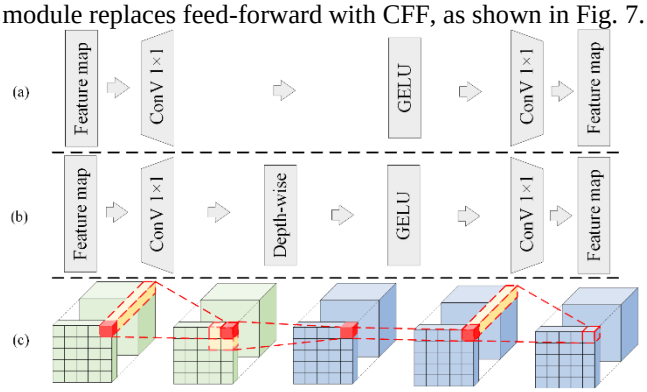

Fig. 7. The structures of Feed-forward and Convolution Feed-forward. (a) Feed-forward, (b) Convolution Feed-forward, (c) An exploded view of Convolution Feed-forward.

The difference between CFF and FF is that CFF encodes the spatial information of each channel by adding depth-wise convolution, which is zero-padding convolution. Therefore, the CFF operation can be expressed as

$$E_i^j = con_{(1\times1,1)}(GELU(DW(con_{(1\times1,1)}(D_i^j)))) + D_i^j \tag{9}$$

where $E_i^j \in R^{(2r+1+i)\times(2r+1+i)\times c_i}$ is the output of CFF. It can be inferred from (9) that CFF realizes the weight fusion of spectral and spatial information of each pixel and capture the local continuity of the input tensor spatial dimension. by combining depth-wise convolution and 1×1 convolution.

Finally, the output of ith basic block F_i is generated by

$$F_i = LN\left(E_i^{L_i}\right) \tag{10}$$

where $LN(\cdot)$ is layer normalization function.

B. Classification Module

After spectral-spatial attention module has performed sufficiently spectral-spatial feature extraction and fusion. To take full advantage of the information, we design SMLP classifier. The SMLP consists of global average pooling and MLP, as shown in Fig.4. The MLP only contains one layer, the input layer is the output layer, and without the hidden layer. The operation process of the SMLP can be expressed as:

$$p = FC(Avg(F_4)) \tag{11}$$

$$Y = \arg\max\left(p\right) \tag{12}$$

where $F_4 \in R^{(2r+5)\times(2r+5)\times c_4}$ is the output of the spectral-spatial attention module, $FC(\cdot)$ and $Avg(\cdot)$ denote fully connected

and global average pooling, respectively. The $\arg\max(\cdot)$ is argmax function, and $p=[p_0, p_1, ..., p_k, ..., p_K]$, where K is the number of class in dataset, p_k is the probability that the central pixel of the 3D-patch belongs to category k. After that, $Y \in \{0,1,2,...,K-1\}$ represents the class of the center pixel. Such a simple classifier can preserve the integrity of the acquired features, which is more helpful for the optimization of the spectral-spatial attention module.

During the model training, we introduce focalloss [58] to suppress the effect of sample imbalance on model training. The focal loss for multiclassification can be calculated using the following formula:

$$t_{ki} = \begin{cases} 1, & k = \hat{y}_i \\ 0, & k \neq \hat{y}_i \end{cases} \tag{13}$$

$$L = -\frac{1}{B}\sum_{i=1}^{B}\sum_{k=0}^{K-1}\left(\alpha_k \times (1-p_k)^{\gamma} \times \log(p_k) \times t_{ki}\right) \tag{14}$$

where α_k denotes the weighting factor that is used to balance the sample distribution, γ is the focus parameter that smoothly adjusts the weight reduction rate of the simple example, and B is the batch size. The $\hat{y}_i$ denotes truth label of center pixel. By means of the loss function L and backpropagation algorithm [59], the model is optimized and the predicted class of each pixel is output.

IV. Experiment And Discussion

In this section, we first introduce the four HSI datasets and then describe the experimental evaluation indicators and hardware configuration. To analyze the performance of LKSSAN, we test the influence of related parameters and key modules on the accuracy of the algorithm, and finally, we will qualitatively and quantitatively analyze the performance of the proposed LKSSAN and state-of-the-art methods on the HSI datasets.

A. Datasets description

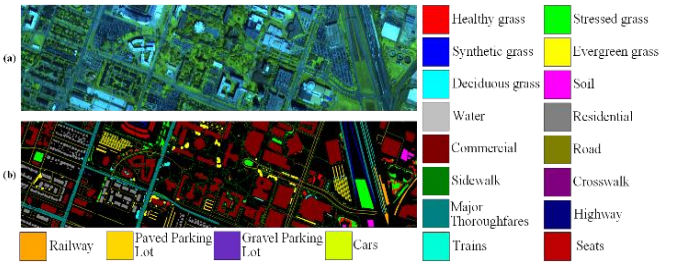

Fig. 8. (a) False-color image, (b) Spatial distribution of ground truth

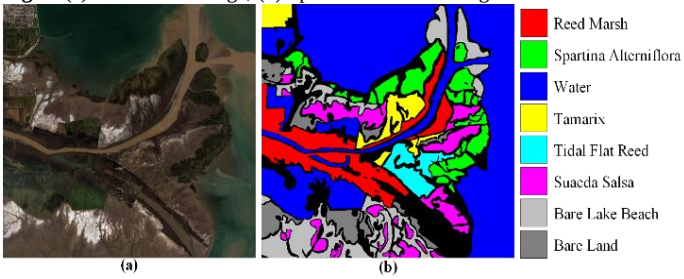

Fig. 9. (a) True-color image, (b) Spatial distribution of ground truth

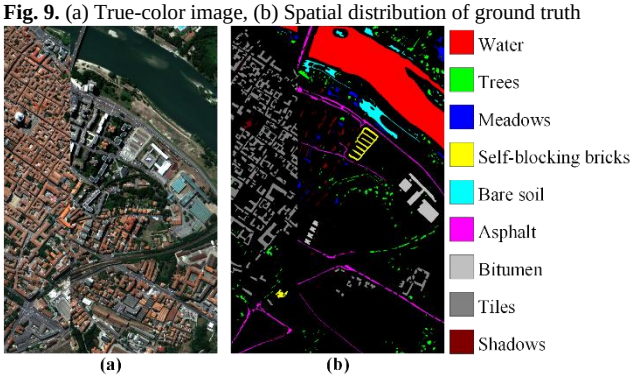

Fig. 10. (a) True-color image, (b) Spatial distribution of ground truth

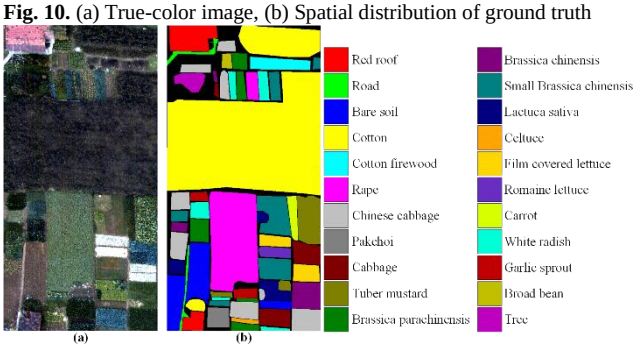

Fig. 11. (a) True-color image, (b) Spatial distribution of ground truth

University of Houston 2018 dataset (UH2018): The UH2018 is captured by the ITRES CASI 1500 sensor over the University of Houston and contains 501015 labeling pixels with 20 ground-truth classes. This database contains consists of 601 × 2385 pixels, 1 m ground sampling distance (GSD), and 48 spectral bands with wavelengths ranging from 380-1050 nm. Table I reports the detailed pixel distributions in each class. The specific ground features of the UH2018 dataset are shown in Fig. 8.

Yellow River Estuary coastal wetland (YRE): The second hyperspectral dataset was acquired by using the Gaofen-5 satellite over the Yellow River Estuary coastal wetland between Bohai Bay and Laizhou Bay. The whole image contains pixels 740 × 761 pixels with 30 m per pixel resolution. The YRE coastal wetland image has 8 classes with 296 spectral bands. The specific ground features of the YRE dataset are shown in Fig. 9 and the fixed number of training and testing samples are detailed in Table II.

Pavia Centre dataset (PC): The image was gathered by the reflective optics system imaging spectrometer (ROSIS) sensor, which covers the area of Pavia, northern Italy. The HSI cube comprises 1096 × 715 pixels with 102 wavelength bands in the range of 430-860 nm. As shown in Fig. 10, there exist 9 different classes of land covers with a resolution of 1.3 m per pixel. Meanwhile, the training set with few samples and test set is listed in Table III.

WHU-Hi-HongHu dataset (WH): The WH dataset was recorded by the Headwall Nano-Hyperspec sensor with fragmented plots and various crops. The image size in pixels is 940 × 475, with a spatial resolution of 0.043 m, composed of 270 bands ranging from 400 to 1000 nm. The ground truth of this scene consists of 22 classes, and 0.1% samples per class

was selected to train the networks, and the remaining were used for testing, as listed in Table IV. A three-band true-color composite image and the ground-truth map are shown in Fig. 11.

TABLE I
NUMBERS OF TRAINING AND TEST SAMPLES USED IN UH2018 DATASET.

No.	Land Cover Type	Train	Test	Total
1	Healthy grass	100	9395	9495
2	Stressed grass	100	31830	31930
3	Synthetic grass	100	544	644
4	Evergreen grass	100	13310	13410
5	Deciduous grass	100	4828	4928
6	Soil	100	4383	4483
7	Water	100	138	238
8	Residential	100	39405	39505
9	Commercial	100	223896	223996
10	Road	100	44844	44944
11	Sidewalk	100	32033	32133
12	Crosswalk	100	1417	1517
13	Major Thoroughfares	100	46683	46783
14	Highway	100	9782	9882
15	Railway	100	6827	6927
16	Paved Parking Lot	100	11337	11437
17	Gravel Parking Lot	100	39	139
18	Cars	100	6505	6605
19	Trains	100	5129	5229
20	Seats	100	6690	6790
	Total	2000	499015	501015

TABLE II
NUMBERS OF TRAINING AND TEST SAMPLES USED IN YRE DATASET.

No.	Land Cover Type	Train	Test	Total
1	Reed Marsh	45	44394	44439
2	Spartina Alterniflora	31	30687	30718
3	Water	215	214194	214409
4	Tamarix	17	16294	16311
5	Tidal Flat Reed	13	12233	12246
6	Suaeda Salsa	30	29559	29589
7	Bare Lake Beach	47	46044	46091
8	Bare Land	22	21276	21298
	Total	420	414681	415101

TABLE III
NUMBERS OF TRAINING AND TEST SAMPLES USED IN PC DATASET.

No.	Land Cover Type	Train	Test	Total
1	Water	66	65905	65971
2	Trees	8	7590	7598
3	Asphalt	4	3086	3090
4	Self-blocking bricks	3	2682	2685
5	Bitumen	7	6577	6584
6	Tiles	10	9238	9248
7	Shadows	8	7279	7287
8	Meadows	43	42783	42826
9	Bare soil	3	2860	2863
	Total	152	148000	148152

TABLE IV
NUMBERS OF TRAINING AND TEST SAMPLES USED IN WH DATASET.

No.	Land Cover Type	Train	Test	Total
1	Red roof	15	14026	14041
2	Road	4	3508	3512
3	Bare soil	22	21799	21821
4	Cotton	164	163121	163285
5	Cotton firewood	7	6211	6218
6	Rape	45	44512	44557
7	Chinese cabbage	25	24078	24103
8	Pakchoi	5	4049	4054
9	Cabbage	11	10808	10819
10	Tuber mustard	13	12381	12394
11	Brassica parachinensis	12	11003	11015
12	Brassica chinensis	9	8945	8954
13	Small Brassica chinensis	23	22484	22507
14	Lactuca sativa	8	7348	7356
15	Celtuce	2	1000	1002
16	Film covered lettuce	8	7254	7262
17	Romaine lettuce	4	3006	3010
18	Carrot	4	3213	3217
19	White radish	9	8703	8712
20	Garlic sprout	4	3482	3486
21	Broad bean	2	1326	1328
22	Tree	5	4035	4040
	Total	401	386292	386693

B. Experimental setup

1) Evaluation Metrics

To quantify the classification performance of LKSSAN, the overall accuracy (OA), average accuracy (AA), kappa coefficient (kappa), producer's accuracies (PA) of each land-cover category, and time taken by the model to train (T) are employed as evaluation measures. Moreover, the classification maps obtained by different models are visualized to make a qualitative comparison.

2) Comparison With State-of-the-Art Backbone Networks

To analyze the effectiveness of LKSSAN and the performance of long-range 3D spectral-spatial information for HSI classification, we compare LKSSAN with other existing state-of-the-art methods of various types and structures. The comparison algorithm is divided into three categories, 3D spectral-spatial extraction, two-branch and Transformer-based structures. For the 3D spectral-spatial extraction, we choose SSRN, multi-scale dense networks (MSDN) [60], spectral-spatial unified networks (SSUN) [61], and residual spectral-spatial attention network (RSSAN) [62] for comparison. For the two-branches, we choose adaptive spectral-spatial multiscale network (ASSMN) [63], double-branch dual-attention mechanism network (DBDA) [49], and attention-based adaptive spectral-spatial kernel resnet (A2S2K) [64] to analyze the effectiveness of the LKSSAN spectral-spatial attention module. For the transformer-based models, we use spectral-spatial transformer network (SSTN) [57] and spectral-spatial feature tokenization transformer (SSFTT) [53] to evaluate the performance of LKSSAN in HSI spectral-spatial features processing capability.

3) Implementation Details

All the experiments were implemented on Windows 10 and an 8-GB GPU of Nvidia GeForce GTX 1080. The proposed networks were carried out using the PyTorch platform as the backend. We trained the network for 200 epochs adopt the Adam optimizer [65] with a batch size of 28 and a learning rate of 0.005. To avoid biased estimation, all experiments were conducted with five independent tests, and the average values were reported for all the evaluation metrics.

C. Model Analysis

1) Parameter Sensitivity Analysis

For CNN-based models, the settings of hyperparameters often have a greater impact on model accuracy. Therefore, we analyze the effect of hyperparameters on LKSSAN in this part. LKSSAN has a simple structure and does not perform complex processing on HSI. The number of PCs, the radius of the Patch, the number of training samples, and the combination of value of kernel-padding-dilation on LKA are critical to the performance of LKSSAN. Note that in this paper, except for the UH2018 dataset, the training sample size of each category of training samples selected in the rest of the datasets is taken as 0.1% of the total number of samples in the respective category. In different parameter sensitivity analysis experiments, we set the default values of the above parameters to 10, 4, 0.1%/100, and (7,9,3) when we do not specifically mention parameter settings. Next, we will analyze them in detail.

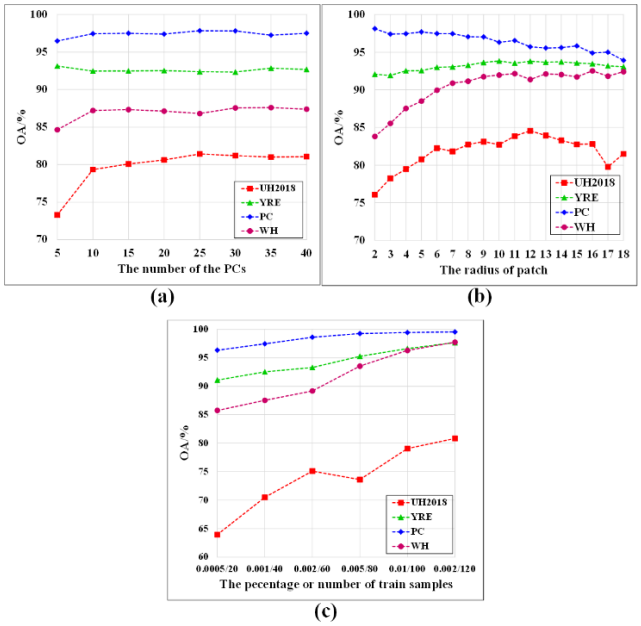

Fig. 12. Effect of (a)The number of PCs, (b) The Size of Patch, (b) The number of training sample on the classification accuracies in the four datasets.

a) Effect of the number of PCs

To reduce the redundancy of the spectral information, we perform principal component analysis on the HSI at the initial stage and select a certain number of principal components (PCs) for subsequent processing. Hence, the number of PCs selected influences the performance of the model to a large extent. Fig. 12 (a) displays the OA versus the number of PCs in different datasets. We can observe that except for the YRE dataset, all other datasets show a rise and then stabilization with the rise in the number of PCs. For this reason, we can simply speculate that only a small number of PCs have an impact on the model performance. In addition, the variation of OA on the YRE dataset is relatively stable which indicates that when the number of PCs is set in the range of {5, 10, 15, 20, 25, 30, 35, 40}, the impact on its performance is rather small. According to the above analysis, we finally set the number of PCs to 10 to balance the performance of the model on each dataset and reduce the operational burden.

b) Effect of the Size of Patch

LKSSAN is a patch-based CNN model, which is essential for the analysis of patch size. Fig. 12 (b) shows the effect of the spatial size of the patch on the OA of the proposed network. A large patch block can contain more spatial contextual information, but a too-large patch size also contains more noise, which adversely affects the spatial features analysis of the center pixel. The key spectral-spatial information of remote sensing images mainly exists locally. Therefore, after the patch-size increases to a critical value, if the patch size continues to increase, the effective information will not increase, and the accuracy will be reduced due to excessive redundant information. Note that the variation of OA in the PC dataset does not conform to the trend of OA in other datasets, instead, it gradually decreases with the increase of patch size in the higher accuracy range. This result further confirms that the perceptual field of the model is not as large as possible, and the images with different resolutions and different scenes are suitable for different patch scales. In Fig. 12 (b), it is clear that 12, 10, 2, and 16 patch size leads to the best results on UH2018, YRE, PC, and WH datasets, respectively. After that, the network performance starts to decrease on these four datasets. Therefore, in the following experiments, the patch size is set to12, 10, 2, and 16 for UH2018, YRE, PC, and WH datasets, respectively.

c) Effect of the Training Percent

Few-shot is an important problem for HSI classification models, and the change of the number of training samples often has a huge impact on the model. Therefore, we analyzed the LKSSAN accuracy under different the number of training samples. In this paper, 100 training samples were taken for each category in the UH2018 dataset in the quantitative analysis phase, and the training samples used in the other datasets were all obtained proportionally based on the total number of samples. Therefore, in order to keep the same sample distribution for different experiments on the same dataset, in this part the variation range of samples for each category on the UH2018 dataset is {20, 40, 60, 80, 100, 120}, while the variation range of training samples on the other datasets is {0.05%, 0.1%, 0.2%, 0.5%, 1%, 2%}. Fig. 12 (c) shows the performance on different the number of training samples. According to the experimental results, we can know that in the simple scenario of the PC dataset, OA of the model can be better than 95% by relying on 0.05% of the training percent. In YRE and PC datasets, if the accuracy is better than 95%, only 1% of the training samples are needed. In addition, the model can achieve better than 70% performance in the UH2018 dataset when only 40 samples of each type are taken, which indicates that the model has a vigorous ability to resist the imbalance of sample distribution. However, the performance on the UH2018 dataset with a training set of 80 is worse than that of 60, which demonstrates that the performance of the model in the small sample dataset with unbalanced sample distribution needs to be improved.

d) Effect of the Kernel-padding-dilation

In LKSSAN, we introduce a convolutional decomposition, which uses depth-wise convolution, depth-wise dilation convolution, and 1×1 convolution instead of large kernel 3D convolution to construct image 3D weights. The kernel size and dilation size of the atrous convolution will directly affect the spatial perceptual field size of LKA. Therefore, in this section, we analyze the effect of different combinations of kernel, padding, and dilation on the experimental results. Table V shows the experimental results on different datasets when the kernel-padding-dilation are set to the parameters in {(5,4,2), (7,6,2), (7,9,3), (9,8,2), (9,12,3)}.

In this part, we set the radius of the patch to 14 to ensure the reasonableness of the experiment. According to Table V, we can intuitively see that different datasets are adapted to different parameters. The convolutional kernels corresponding to the

parameters adapted to different datasets are PC, YRE, UH2018, and WH in descending order, and this order is the same as the order from small to large based on the optimal patch size of each dataset, i.e., the smaller the patch size is, the farther the long-range information is needed. This phenomenon may occur because the feature distribution presents an overall dispersion and a local aggregation in this hypothesis space; meanwhile, the larger the optimal patch size of the image, the larger the dispersion value of its spatial feature set distribution, and the smaller the dispersion value between the elements within different spatial feature sets. Although the optimal values are obtained for different scenes and data with different spatial resolutions under different combinations of kernel and dilation, the discrepancies between the results obtained in different combinations of the same dataset is relatively small. Therefore, there is no harm in eventually setting the kernel-padding-dilation to (7,9,3).

TABLE V
THE EFFECTS OF KERNEL-PADDING-DILATION ON LKA. THE BEST ONE IS SHOWN IN BOLD

Kernel-padding-dilation		5-4-2	7-6-2	7-9-3	9-8-2	9-12-3
Houston 2018 dataset	OA/%	84.07	**84.80**	84.06	83.64	83.85
	AA/%	74.30	**85.71**	73.11	72.99	74.73
	Kappa	0.7967	**0.8053**	0.7911	0.7926	0.7973
YRE coastal wetland	OA/%	93.64	93.23	**93.72**	93.09	93.11
	AA/%	90.48	88.89	**90.15**	89.80	89.55
	Kappa	0.9085	0.9025	**0.9091**	0.9004	0.9004
Pavia Centre dataset	OA/%	94.93	95.70	95.61	96.26	**96.43**
	AA/%	87.73	90.73	88.10	92.05	**90.78**
	Kappa	92.79	0.9389	0.9375	0.9470	**0.9494**
WHU-Hi-HongHu dataset	OA/%	**92.59**	90.19	92.03	92.36	90.94
	AA/%	**84.45**	83.12	82.56	84.20	80.30
	Kappa	**0.9060**	0.9012	0.8991	90.34	0.8854

2) Ablation Study

This paper proposes the LKSSAN method, and the technical contributions include: the scale expansion block to expand both spatial and spectral scales of 3D patch; the LKA for long-range 3D representation learning; the CFF to assist LKA to further exploit spatial and spectral information. In this part, we investigate how these structures in the LKSSAN affects the classification performance in YRE and PC datasets. For that, we conduct extensive ablation experiments on the datasets to verify the effectiveness of these components in LKSSAN for HSI classification, and the detailed classification results with different structures are shown in Table VI. Note that data input and classification are the basic modules of LKSSAN, and good performance of the model is the best proof for them, so we do not analyze the performance of these two modules in detail in this paper.

Table VI indicates that the spatial expansion increases accuracy significantly by 1.01% of OA on the YRE dataset, by 0.5% of OA on the PC dataset. Furthermore, we can observe that the spectral expansion can improve higher accuracy compared to the spatial expansion. The aforementioned results show that although the scale expansion structure is simple, it can facilitate the model to extract spectral and spatial information by expanding the spectral and spatial scales of the data. Moreover, when the LKA is removed, the OA of the model are reduced by 1.7% and 0.8% on the YRE and PC datasets, respectively, which reflects the importance of jointly extraction of spectral and spatial information. Meanwhile, we can notice that when LKA loses the assistance of CFF, the OA values will drop to 90.89% and 96.98%, respectively, while CFF without LKA also fails to obtain excellent performance. This indicates that LKA and CFF can achieve mutual reinforcement, which consequently justifies the combination of these two structures.

In addition, to verify the effect of convolutional decomposition on the performance of LKSSAN, we replace the weight construction part of LKA with the corresponding 3D convolution. Since the default kernel and dilation of depth-wise dilation convolution are 7 and 3, respectively, the spatial and channel kernels of the corresponding 3D convolution are set to 19 and c_i, respectively, where c_i denotes the dimension of the feature map in the *i*th base module. Obviously, in the YRE dataset, the OA of LKSSAN is inferior to that of 3D-LKSSAN OA, while in the PC dataset, the OA of LKSSAN is better than that of 3D-LKSSAN. Although LKSSAN and 3D-LKSSAN have their advantages in different datasets, the mutual advantages are small or even negligible, which indicates that the convolutional decomposition can effectively replace the large kernel 3D convolution to exploit the long-range 3D spectral-spatial features. Moreover, Table VI presents the size of parameters required by the model when the two models take the same size of input for the PC dataset. It can be found that the size of the parameters of LKSSAN is much smaller than that of 3D-LKSSAN, which reflects that convolutional decomposition can effectively reduce the model parameters. In summary, although large kernel 3D convolution is effective in extracting long-range 3D features, it has high computational pressure, and the convolutional decomposition approach can maintain the model performance to the maximum while significantly reducing the number of model parameters.

TABLE VI
ABLATION ANALYSIS OF THE PROPOSED LKSSAN WITH A COMBINATION OF DIFFERENT MODULES ON THE YRE AND PC DATASETS. THE BEST ONE IS SHOWN IN BOLD

Framework		No scale expansion	No spatial expansion	No-LKA	No-CFF	3D-LKSSAN	LKSSAN
YRE coastal wetland	OA/%	90.40	91.52	90.83	90.89	**92.82**	92.53
	AA/%	85.50	85.32	85.61	84.11	**86.97**	86.67
	Kappa	0.8705	0.8822	0.8812	0.8688	**0.8964**	0.8921
Pavia Centre dataset	OA/%	94.91	97.26	96.96	96.98	97.64	**97.76**
	AA/%	76.51	91.10	90.58	90.74	91.58	**91.64**
	Kappa	0.9277	0.9653	0.9611	0.9514	0.9665	**0.9680**
Parameter size		——	——	——	——	21.57M	**0.74M**

D. Comparison With State-of-the-Art Methods

In this section, to evaluate the performance of LKSSAN, we qualitatively and quantitatively compare LKSSAN with other existing state-of-the-art methods. All these methods are implemented using open source code with optimal parameters, as described in the corresponding references. Furthermore, for fair comparison, all methods are trained and tested on the same sample number, as listed in Tables I-IV.

1) Results on the UH2018 Dataset

The UH2018 dataset shows urban scenes with more feature classes and is mainly used to verify the performance of the model for refined classification in urban scenarios. Table VII shows the mean and standard deviation of each accuracy metric obtained by each algorithm after five experiments on the

UH2018 dataset. As shown in Table VII, LKSSAN produces the best OA and kappa. Specifically, the two-branch algorithms obtain the best result among the compared algorithms, while the 3D algorithms obtain the worst performance. In the two-branch networks, ASSMN yields the lowest OA value, but it has the highest number of optimal PAs. This phenomenon may occur because the UH2018 dataset has the same number of training samples for all types of features, which leads to it hard for ASSMN to adequately explore the characteristics of land cover with more validation samples. Among the compared algorithms, DBDA and SSSFTT have higher OA values of 81.36% and 80.42%, respectively. By contrast, LKSSAN obtained the finest OA with 84.55%, which is superior to them by 3.19% and 4.13%, respectively. The three algorithms that obtain the highest AA are ASSMN, SSUN, and LKSSAN, but the OA and kappa of ASSMN and SSUN are lower, which indicates that they extracted fewer representative features. In addition, the kappa of LKSSAN is 0.8035, which achieves significant improvements ranging from 0.0473 to 0.2914. Moreover, although LKSSAN has an absolute advantage in accuracy metrics, its training time is longer, which indirectly reflects the necessity of convolutional decomposition.

Fig. 13 displays the classification maps of all methods for visual performance estimation. It can be seen that the results of MSDN, RSSAN, and A2S2K are seriously affected by noise, and the commercial misclassification phenomenon is more serious in the images of SSRN, SSUN, ASSMN, SSTN, and SSFTT. Although DBDA performs best in the comparison algorithms, its classification map shows serious confusion between paved parking lot and car. Compared with the comparison algorithms, LKSSAN obtains a smooth classification map with optimal visualization.

2) Results on the YRE Dataset

Among all experimental datasets, the YRE dataset has the widest spectral coverage, the highest number of spectra, and the lowest spatial resolution, and is mainly used to verify the performance of the model for coastal wetland classification based on satellite images.

Table VIII shows the experimental results of each algorithm on the YRE dataset. According to Table VIII, we can find that all the algorithms obtain excellent performance on this dataset. The SSUN requires the shortest training time and its OA is 93.03%. The OA of SSRN exceeds the OA of SSUN by 0.33%, but its training time is about 200 times longer than the training time of SSUN. In the two-branch algorithms, DBDA obtains the best performance with the least training time and the best accuracy. In the transformer-based algorithms, the difference in OA between SSTN and SSFTT is smaller, but SSTN takes more training time. Although the dominant algorithms in different categories all obtain good performance, their results have large variance values and the differences in each accuracy are large. By contrast, LKSSAN obtains the optimal OA and the best kappa, which directly proves that LKSSAN can alleviate the imbalanced training data problems and indirectly reflects the advantages of the 3D spectral-spatial extraction method in remote sensing target recognition based on satellite images.

Fig. 14 shows the experimental maps of each algorithm on the YRE dataset. The results demonstrate that SSRN, MSDN, DBDA, A2S2K, and SSFTT are severely affected by strip noise, while RSSAN, ASSMN, and SSTN maintain details well, but the tamarix in their upper left of classification maps are more severely affected by noise. The 3D feature extraction-based model SSUN has good detail retention, but the suaeda salsa at the bottom is miss-classified, and LKSSAN is more balanced for all kinds of feature recognition, but there is still a weak banding phenomenon. In summary, SSUN and LKSSAN obtain the best visualization results.

TABLE VII

QUANTITATIVE PERFORMANCE OF DIFFERENT CLASSIFICATION METHODS IN TEAMS OF OA, AA, AND KAPPA, AS WELL AS THE ACCURACY FOR EACH CLASS ON THE UH2018 DATASET. THE BEST ONE IS SHOWN IN BOLD.

Class No.	3D spectral-spatial extraction				two-branch			Transformer		LKSSAN
	SSRN	MSDN	SSUN	RSSAN	ASSMN	DBDA	A2S2K	SSTN	SSFTT	
1	79.18±2.567	71.68±5.484	**93±2.695**	62.94±4.09	92.43±3.174	79.9±2.266	81.62±2.55	67.41±7.085	63.14±6.018	61.83±2.29
2	**94.83±0.504**	88.3±0.653	87.22±2.251	85.81±0.837	89.29±2.616	85.52±0.818	93.45±0.629	91.35±2.005	90.91±1.589	90.13±1.129
3	97.16±1.414	58.76±9.709	**100±0**	48.08±16.183	**100±0**	81.02±18.519	83.36±5.229	79.4±13.883	63.22±25.202	94.38±6.295
4	81.77±0.983	86.34±3.719	94.85±1.117	83.31±1.972	**98.48±0.643**	88.16±1.596	87.52±0.658	82.32±4.016	83.94±3.309	74.51±0.854
5	49.89±8.358	34.8±7.495	90.03±2.431	32.34±3.092	**93.63±1.907**	91.6±2.367	49.05±5.729	17.41±1.603	56.26±7.416	59.53±2.442
6	78.92±5.524	45.53±12.551	99.13±0.515	49.92±12.102	**99.84±0.204**	98.61±1.177	84.9±4.022	51.72±16.541	87.56±1.583	96.56±1.513
7	88.32±4.842	26.29±16.331	100±0	6.23±2.111	**100±0**	100±0	64.25±4.194	26.61±22.381	84.11±17.646	57.11±9.181
8	62.32±2.147	62.26±4.683	80.41±3.85	64.57±3.429	**92.79±1.28**	69.78±1.321	75.24±0.76	63.93±4.496	71.99±3.934	82.65±1.111
9	98.15±0.153	97.05±0.522	60.13±1.851	95.39±0.726	73.13±1.512	94.7±0.265	98.11±0.135	97.17±0.305	98.09±0.532	**98.56±0.15**
10	53.03±2.675	53.9±2.406	52.48±5.741	36.33±0.82	53.63±2.645	64.22±2.213	54.51±3.031	50.75±2.945	70.91±2.992	**75.54±0.486**
11	56.52±0.762	47.98±3.237	46.98±7.61	40.61±1.732	55.42±2.834	54.46±1.368	54.03±2.418	**55.42±1.916**	50.36±4.688	47.22±0.406
12	8.48±0.989	5.1±0.485	60.21±9.503	3.1±0.389	**80.48±2.075**	0±0	7.82±1.039	5.08±1.276	9.11±0.643	12.25±1.169
13	76.51±1.835	59.14±3.153	45.87±7.223	50.21±1.781	64.57±1.292	61.02±0.714	70.73±2.683	67.13±3.312	81.29±1.957	**83.85±0.091**
14	58.7±4.663	47.91±6.08	90.68±2.638	47.51±4.306	**98.5±0.734**	87.22±4.398	58.8±3.514	57.24±6.098	67.09±5.603	83.25±1.457
15	91.28±2.756	59.68±17.733	98.04±1.119	53.04±2.907	**99.23±0.532**	83.47±2.313	83.84±3.021	46.86±5.147	96.04±1.535	97.07±2.999
16	78.2±2.591	58.6±7.044	91.07±2.74	49.28±4.069	**95.62±1.964**	64.66±1.022	77.93±1.914	57.41±10.165	88.16±1.939	88.33±2.599
17	54.09±18.569	2.82±1.911	100±0	2.51±1.282	**100±0**	0±0	12.35±6.303	4.22±2.635	29.46±10.013	17.39±7.359
18	58.67±2.086	26.89±8.566	91.28±0.836	27.02±2.374	**97.52±0.584**	83.11±22.554	51.33±1.542	42.72±7.004	57.24±3.686	92.58±2.982
19	69±3.934	39.16±15.726	97.71±1.538	50.66±4.857	**99.86±0.114**	95.49±2.059	74.96±6.851	48.47±8.473	70.77±8.519	96.96±1.592
20	75.61±3.934	51.96±9.037	97.47±1.687	55.1±5.9	**100±0.007**	96.69±0.924	67.07±3.784	73.6±6.085	75.67±3.616	87.48±6.53
OA (%)	78.06±0.391	58.09±1.956	65.94±0.895	61.22±1.333	72.34±1.18	81.36±0.284	77.63±0.593	67.32±2.269	80.42±1.238	**84.55±0.311**
AA (%)	71.03±0.579	64.52±2.028	**83.83±0.267**	47.127±1.746	75±0.379	73.98±1.016	66.55±0.734	54.31±1.843	69.77±2.165	74.86±0.816
kappa	0.7277±0.004	0.5121±0.014	0.5931±0.009	0.5403±0.013	0.7309±1.337	0.7562±0.368	0.7226±0.007	0.6089±0.023	0.7543±0.014	**0.8035±0.004**
Time/s	3203.58	1827.27	**76.19**	4785.74	2969.93	732.48	3983.24	3258.96	425.37	3602.39

TABLE VIII

QUANTITATIVE PERFORMANCE OF DIFFERENT CLASSIFICATION METHODS IN TEAMS OF OA, AA, AND KAPPA, AS WELL AS THE ACCURACY FOR EACH CLASS ON THE YRE DATASET. THE BEST ONE IS SHOWN IN BOLD.

Class No.	3D spectral-spatial extraction				two-branch			Transformer		LKSSAN
	SSRN	MSDN	SSUN	RSSAN	ASSMN	DBDA	A2S2K	SSTN	SSFTT	
1	91.41±1.721	82.43±6.428	92.95±1.966	85.08±1.356	91.14±2.788	92.64±1.027	**93.29±0.906**	90.84±1.801	92.89±0.408	90.57±0.375
2	95.77±1.094	91.07±2.016	94.67±1.721	91.68±3.86	94.46±2.057	91.80±0.355	95.42±1.15	97.14±1.226	95.22±1.062	**97.14±0.379**
3	98.61±0.397	98.2±0.957	**98.78±0.367**	98.41±0.341	98.72±0.285	**99.61±0.044**	99.34±0.142	98.95±0.2	97.97±0.219	99.29±0.061
4	81.22±4.456	62.23±7.295	73.71±2.831	67.51±4.78	68.47±9.776	**89.10±1.615**	84.6±0.629	81.38±9.39	74.25±3.042	84.73±0.878
5	80.85±2.467	55.22±13.739	71.8±4.918	78.8±6.463	69.53±6.337	68.16±1.2	88.83±2.728	86.93±6.208	76.54±3.201	**89.49±1.806**
6	79.17±2.052	55.2±7.631	72.64±5.787	71.55±2.918	77.56±5.42	84.44±1.853	77.14±1.601	71.94±9.313	77.61±0.595	**85.73±1.686**
7	85.4±1.342	76.69±5.844	**90.83±2.417**	79.11±3.554	88.14±5.595	86.89±1.669	84.91±1.285	84.03±4.437	87.37±1.03	81.58±0.959
8	94.49±2.106	81.12±5.557	93.11±2.095	86.71±3.781	88.6±6.146	**96.49±0.298**	95.09±1.715	96.34±1.469	91.1±0.551	90.87±0.795
OA (%)	93.36±0.004	80.29±1.792	93.03±0.216	90.04±0.591	92.34±1.237	94.04±0.002	92.97±0.302	92.67±0.817	92.64±0.276	**94.39±0.240**
AA (%)	88.36±0.009	86.34±1.243	86.06±0.933	82.36±1.020	84.58±3.157	88.64±0.002	89.83±0.51	88.44±1.534	86.62±0.628	**90.97±0.503**
kappa	0.9040±0.006	0.7527±0.014	0.8995±0.003	0.8561±0.008	0.8896±	0.9143±0.003	0.9031±0.004	0.8942±0.012	0.8935±0.406	**0.9191±0.003**
Time/s	4211.08	1212.23	22.34	739.81	649.45	288.20	6879.30	756.46	89.15	682.41

3) Results on the PC Dataset

The PC dataset shows urban scenes with fewer categories and stitched areas and is mainly used to verify the model's classification effect in multi-scale coarse-grained cases. Table IX shows the results of all the algorithms on the PC dataset. The results show that OAs of all the algorithms except RSSAN and MSDN obtained better than 93%, which may indicate the high spectral quality of the image and low classification difficulty. The PA of LKSSAN was better than 80%, which speculates the model can resist sample imbalance. In addition, although DBDA obtained the highest AA and kappa, the differences with LKSSAN are smaller. Fortunately, the training speed of LKSSAN is better than that of DBDA, so LKSSAN produces the best quantitative metrics based on the overall performance.

Fig. 15 shows the classification map of each algorithm. To facilitate the qualitative evaluation, we enlarge the white box area in the classification graphs. The classification maps of MSDN, RSSAN, ASSMN, DBDA, and SSFTT cannot effectively reflect the spatial distribution of vegetation in this region. According to Fig.15(a), it can be found that the non-residential area contains a large amount of meadows and bare soil except for trees, but SSRN and SSUN have a serious phenomenon of misclassifying these two classes as trees. Based

on the visualization results of the white box field, we can deduce that joint mining of long-range spectral and spatial information helps to alleviate the scenarios with poor spatial regularity of feature distribution. Moreover, compared with LKSSAN, A2S2K misclassifies asphalt and shadows into tiles in the left splicing area. Additionally, LKSSAN can still obtain smooth images in the region without labels indicating that the algorithm can effectively alleviate the spatial autocorrelation issue. In summary, LKSSAN has optimal visualization results.

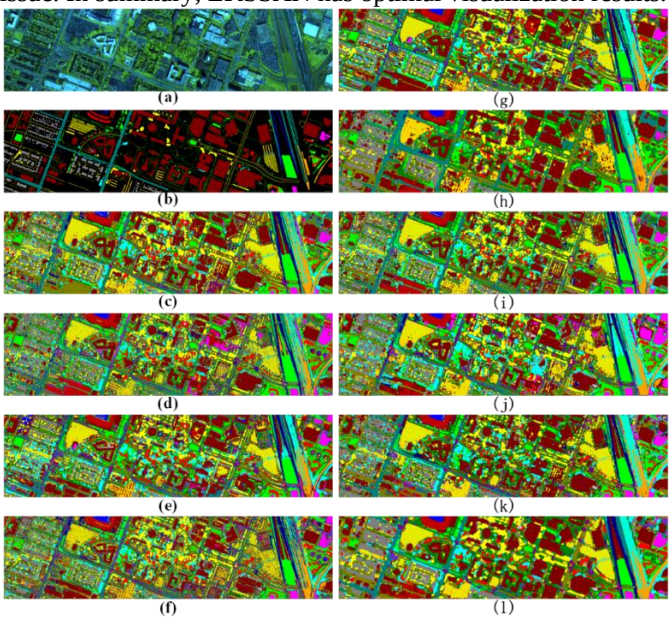

Fig. 13. Classification maps of different methods on UH2018 dataset. (a) False-color image of UH2018. (b)GT. (c) SSRN. (d) MSDN. (e) SSUN. (f) RSSAN. (g) ASSMN. (h) DBDA. (i) A2S2K.(j)SSTN. (k)SSFTT. (l)LKSSAN.

TABLE IX

QUANTITATIVE PERFORMANCE OF DIFFERENT CLASSIFICATION METHODS IN TEAMS OF OA, AA, AND KAPPA, AS WELL AS THE ACCURACY FOR EACH CLASS ON THE PC DATASET. THE BEST ONE IS SHOWN IN BOLD.

Class No.	3D spectral-spatial extraction				two-branch			Transformer		LKSSAN
	SSRN	MSDN	SSUN	RSSAN	ASSMN	DBDA	A2S2K	SSTN	SSFTT	
1	99.99±0.006	97.81±0.514	99.85±0.162	89.31±7.058	99.91±0.083	99.98±0.05	99.98±0.026	99.94±0.062	99.79±0.14	**100±0.004**
2	97.3±1.128	90.91±3.092	91.41±4.852	58.4±20.017	86.71±5.563	94.81±1.356	92.52±3.73	**97.95±1.405**	97.23±1.451	95.08±0.49
3	77.89±9.848	68.35±7.191	49.16±19.044	43.49±21.271	61.26±10.666	**89.21±1.111**	86.28±4.31	75.55±13.574	64.01±4.313	82.29±1.873
4	65.04±5.289	64.53±12.599	34.24±26.843	16.35±13.682	75.47±8.393	77.5±3.113	69.93±5.336	70.55±7.703	77.06±8.312	**87.63±1.906**
5	95.24±3.515	54.59±8.153	63.27±12.26	22.56±4.326	76.86±5.544	**98.8±0.677**	92.4±2.762	94.12±4.226	93.4±4.318	86.51±0.674
6	93.59±1.223	79.07±3.843	92.26±2.523	36.28±10.047	92.54±2.718	93.11±0.638	98.12±1.081	91.76±1.496	86.47±1.913	**96.05±0.368**
7	99.96±0.05	75.85±6.693	80.8±2.274	38.58±12.917	84.39±3.555	98.59±1.722	93.78±1.023	99.15±0.878	**99.15±0.803**	98.69±0.172
8	99.94±0.076	84.52±2.698	97.06±1.298	62.07±5.042	98.39±0.732	99.06±0.173	99.68±0.2	99.29±0.6	99.33±0.159	**99.97±0.007**
9	99.51±0.475	98.76±0.619	83.48±10.166	33.25±18.694	81.06±3.726	99.81±0.273	99.36±0.825	**99.7±0.392**	96.09±1.496	96.27±0.06
OA (%)	97.72±0.376	83.67±0.895	93.01±1.141	74.57±1.291	94.94±0.297	98.1±0.202	97.77±0.306	97.34±0.680	96.74±0.026	**98.12±0.043**
AA (%)	92.05±1.376	88.71±0.628	76.84±4.862	44.48±6.899	84.07±0.85	**94.54±0.665**	92.45±0.899	92.00±1.695	90.28±1.137	93.61±0.277
kappa	0.9678±0.005	0.7938±0.020	0.9005±0.017	0.6048±0.024	0.9280±0.004	**0.9736±0.229**	0.9684±0.004	0.9624±0.010	0.9539±0.004	0.9735±0.001
Time/s	253.43	688.33	**13.94**	182.27	238.55	423.16	380.52	259.59	40.11	202.00

TABLE X

QUANTITATIVE PERFORMANCE OF DIFFERENT CLASSIFICATION METHODS IN TEAMS OF OA, AA, AND KAPPA, AS WELL AS THE ACCURACY FOR EACH CLASS ON THE WH DATASET. THE BEST ONE IS SHOWN IN BOLD.

Class No.	3D spectral-spatial extraction				two-branch			Transformer		LKSSAN
	SSRN	MSDN	SSUN	RSSAN	ASSMN	DBDA	A2S2K	SSTN	SSFTT	
1	93.83±1.723	91.17±3.903	86.92±7.856	80.98±5.284	87.15±4.95	**97.59±0.486**	95.73±2.435	90.57±5.339	94.78±1.422	97.39±0.972
2	75.95±7.203	78.97±8.667	72.62±9.972	56.34±13.385	64.29±7.632	**87.16±2.122**	66.49±6.412	67.33±3.887	63.86±15.786	39.46±5.202
3	78.59±2.919	72.38±12.025	89.78±0.729	72.25±1.803	90.5±2.083	85.19±0.45	82.21±2.419	84.71±4.019	81.15±6.114	**92.71±1.334**
4	94.02±1.095	77.65±11.302	**99.73±0.078**	93.36±2.256	97.5±2.407	96.53±0.247	95.9±0.862	97.54±0.271	97.05±1.663	98.71±0.357
5	70.89±9.09	49.36±27.614	31.61±16.081	31.73±16.984	67.13±12.725	84.79±4.607	59.17±10.152	**87.48±6.989**	74.15±9.827	83.8±6.504
6	85.87±0.999	76.75±7.483	91.4±0.284	74.71±5.528	90±3.506	93.76±0.69	91.33±1.399	93.31±2.534	85.58±6.266	**97.92±0.888**
7	65.08±4.729	51.32±2.776	71.93±8.448	50.63±1.859	73.29±3.868	80.42±0.96	74.74±5.757	77.41±2.367	86.16±8.214	**85.02±2.352**
8	57±6.913	35.92±12.482	15.02±1.989	12.76±7.097	5.91±1.848	77.34±9.535	29.76±7.363	50.52±1.681	66.57±8.799	**81.63±5.092**
9	93.52±1.847	82.63±8.207	87.71±5.262	77.67±10.555	78.4±7.923	**99.18±0.275**	96.36±1.517	95.36±5.035	95.69±0.957	90.93±0.881
10	80.33±2.961	61.95±10.02	60.21±10.479	38.62±8.398	49.8±7.753	69.98±4.702	79.57±2.291	79.8±8.987	83.42±8.63	**83.84±2.558**
11	62.47±7.179	27.51±3.535	26.82±8.199	25.61±8.752	19.7±7.582	79.71±2.278	67.15±7.005	75.65±7.039	63.12±11.362	**82.29±2.618**
12	54.18±9.036	60.03±8.138	39.29±15.083	32.46±5.452	37.43±12.866	66.45±2.092	55.74±5.087	63.78±5.329	67.42±18.05	**75.28±4.59**
13	68.07±4.086	58.73±16.576	80.76±4.213	51.74±2.578	69.22±3.468	69.94±2.296	70.14±3.146	80.1±4.652	68.55±4.342	**92.45±1.779**
14	78.19±4.089	51.98±8.341	1.7±1.687	40.75±1.466	52.16±8.378	**90.23±6.218**	90.3±3.143	84.17±3.083	61.87±16.061	89.67±1.382
15	81.78±21.791	60±48.99	36.12±29.922	18.54±17.105	14.96±15.01	**99.65±0.383**	70.08±23.264	97.81±3.39	79.01±10.654	58.13±13.151
16	74.47±3.905	71.32±5.11	79.32±15.421	59.17±7.126	74.22±9.055	92.02±1.802	82.56±3.115	**96.95±2.231**	87.98±4.939	91.03±1.358
17	49.53±6.147	55.12±8.468	34.7±17.68	27.45±4.261	37.29±18.412	**93.53±1.602**	66.81±11.972	69.32±9.268	86.67±4.357	79.78±3.794
18	72.3±4.283	39.63±22.022	28.42±10.436	14.11±3.86	15.91±10.599	96.66±1.08	61.14±10.433	69.78±8.043	76.54±21.816	**96.34±2.315**
19	68.94±10.23	58.39±11.919	50.09±10.615	53.32±17.232	40.1±6.391	84.47±1.65	91.34±3.668	**94.81±1.859**	82.54±7.58	80.26±2.755
20	64±9.052	73.18±18.604	34.23±9.471	28.44±19.011	37.51±21.193	90.1±2.643	77.67±12.665	76.38±9.814	65.57±13.196	**94.81±1.618**
21	43±7.408	16.49±18.3	1.7±1.687	6.75±2.098	6.58±6.593	10.16±16.472	23.6±7.067	11.14±6.764	64.65±19.741	**97.66±7.813**
22	47.11±14.285	41.11±24.764	50.46±13.127	49.09±20.742	21.48±7.01	88.51±2.007	58.63±7.936	74.07±6.546	70.14±14.34	**98.1±1.622**
OA (%)	83.69±0.337	60.75±11.107	82.34±0.664	74.22±2.227	79.69±2.131	89.84±0.297	86.30±0.416	89.36±0.345	86.32±1.610	**92.52±0.114**
AA (%)	70.87±2.355	71.53±6.097	55.16±1.819	45.29±6.047	51.39±3.45	83.33±1.514	72.11±0.2.002	78.09±0.627	77.38±1.687	**84.42±1.034**
kappa	0.7909±0.004	0.5870±0.060	0.7753±0.009	0.6686±0.030	0.7420±2.75	0.8704±0.004	0.8254±0.006	0.8649±0.440	0.8260±2.129	**0.9054±0.001**
Time/s	3444.91	1533.66	**22.99**	746.24	611.83	1954.41	5593.38	682.05	88.19	2595.43

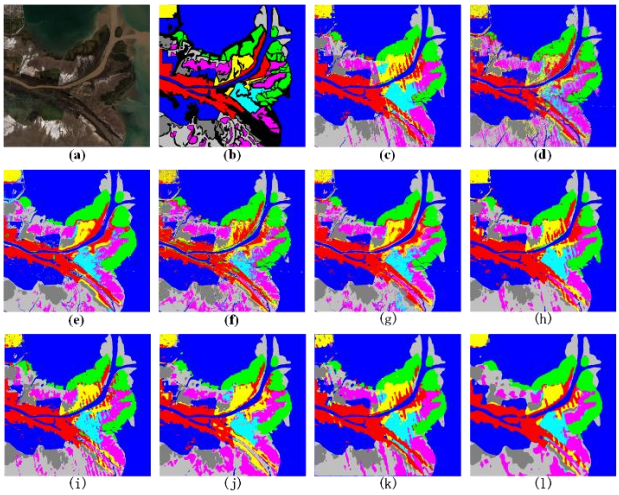

Fig. 14. Classification maps of different methods on YRE dataset. (a) True-color image of YRE. (b)GT. (c) SSRN. (d) MSDN. (e) SSUN. (f) RSSAN. (g) ASSMN. (h) DBDA. (i) A2S2K.(j)SSTN. (k)SSFTT. (l)LKSSAN.

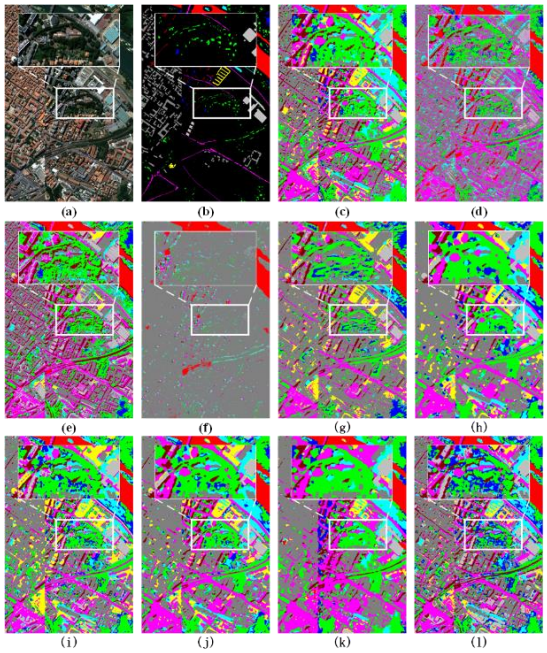

Fig. 15. Classification maps of different methods on PC dataset. (a) True-color image of PC. (b)GT. (c) SSRN. (d) MSDN. (e) SSUN. (f) RSSAN. (g) ASSMN. (h) DBDA. (i) A2S2K.(j)SSTN. (k)SSFTT. (l)LKSSAN.

4) Results on the WH Dataset

The WH dataset is ultra-high spatial resolution hyperspectral data acquired by UAV with numerous agricultural classes, and each type of class provides only 0.1% of its all label for model training. In this section, we use the WH dataset to validate the performance of the models for fine-grained agricultural classification using high spatial and spectral resolution in the small sample case. The experimental results of all models on the WH dataset are shown in Table X. Obviously, LKSSAN has a tremendous advantage with an OA that is 31.77% to 2.68% higher than the comparison algorithm. Specifically, the transformer-based algorithm obtains the best performance, which indicates the advantage of long-range information in refined classification. DBDA and A2S2K have higher OA and longer training times than the 3D spectral-spatial extraction-

based algorithms. In contrast, LKSSAN achieves optimal performance with the highest number of optimal PAs, the best OA, AA, and kappa.

Fig. 16 shows the classification maps of each model for model qualitative analysis. Based on Fig. 16, we can see that MSDN, RSSAN, ASSMN, and A2S2K are most affected by noise, and SSRN, SSUN, SSTN, and SSSFTT have severe feature confusion in the upper left region of their classification maps. Although he classification map of DBDA has the best visualization, with the GT map as the benchmark, we can identify the best image quality obtained by LKSSAN. By comparing the 3D extraction models as SSRN, SSUN, and RSSAN with the two-branch models such as DBDA and A2S2K, we can find that although the 3D extraction models perform poorly overall, the edges between local objects in the classification map are better maintained. Together with the performance of LKSSAN on the WH dataset, we can infer that 3D feature extraction is essential for the fine classification of agriculture.

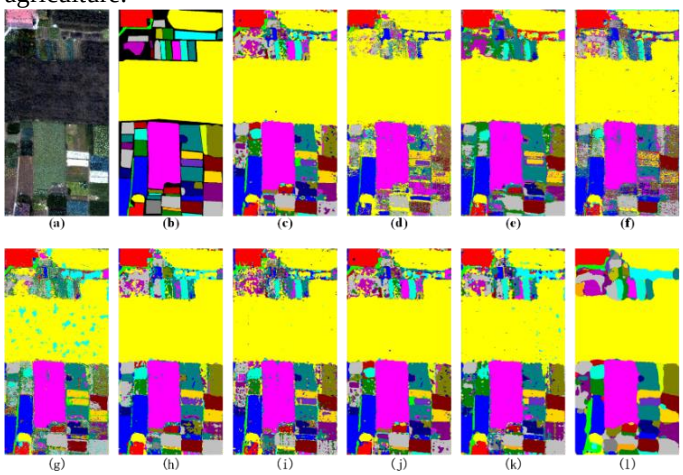

Fig. 16. Classification maps of different methods on WH dataset. (a) False-color image of WH. (b)GT. (c) SSRN. (d) MSDN. (e) SSUN. (f) RSSAN. (g) ASSMN. (h) DBDA. (i) A2S2K. (j)SSTN. (k)SSFTT. (l)LKSSAN

TABLE XI
Trainable Parameters of Different DL-based Methods on YRE datasets. The best one is shown in bold.

Patch size	3D spectral-spatial extraction				two-branch			Transformer		LKSSAN
	SSRN	MSDN	SSUN	RSSAN	ASSMN	DBDA	A2S2K	SSTN	SSFTT	
Parameters	0.5114M	1.5374M	0.3117M	**0.0234M**	3.0245M	0.5565M	0.5205M	0.0257M	0.1527M	0.7427M

5) Model Complexity Analysis

To evaluate the complexity of the proposed LKSSAN method, we list the trainable parameters of different DL-based methods with the same patch size on YRE datasets in Table XI. According to Table XI, we can notice that RSSAN has the least parameters, but the lightweight structure causes performance loss. In the two-branch networks, A2S2K and DBDA have fewer parameters. The transformer-based methods have fewer parameters and poorer performance than the two-branch models. In addition, LKSSAN has the best performance on each dataset, but it requires more parameters compared to the other methods. Fortunately, it is small that the difference between the parameters of LKSSAN and the state-of-the-art algorithm. The result confirms the feasibility and value of LKSSAN in practical applications to some extent.

V. CONCLUSION

In this article, we have proposed a large kernel spectral-spatial attention network (LKSSAN) to exploit the long-range 3D dependency. Instead of the transformer, the proposed method can excavate the long-rang features by large kernel 3D convolution, which can effectively address the challenges in the field of HSI classification. To emphasize and model the critical 3D features, we are inspired by VAN and design a spectral-spatial attention module that contains two crucial structures (i.e., LKA and CFF). The LKA extracts long-range 3D dependencies from the 3D patch expanded by scale expansion block through attention and convolution decomposition, and the CFF facilitates the LKA and exploits the more abstract 3D semantic representation. Moreover, to adequately utilize the long-rang 3D information, the classification module fuses the information and obtains the final classification map by SMLP.

Experiments with SSRN, MSDN, SSUN, RSSAN, ASSMN, DBDA, A2S2K, SSTN, and SSFTT on UH2018, YRE, PC, and WH datasets demonstrate the superior performance of LKSSAN. The reasons are that the spectral-spatial attention module can effectively excavate the long-range 3D spectral-spatial features, and the well-designed SMLP can perfectly match the spectral-spatial attention module to realize the efficient integration and utilization of information. However, it should be mentioned that, although the LKSSAN can obtain high classification accuracies with various complex scenes, its ability in addressing insufficient training data problems and reducing computational burden still needs to be improved. In the future, we will introduce more lightweight structures to optimize HSI classification speed and build a semi-supervised model to alleviate the insufficient training data issue.

REFERENCES

[1] L. He, J. Li, C. Liu, and S. Li, "Recent Advances on Spectral–Spatial Hyperspectral Image Classification: An Overview and New Guidelines," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1579-1597, 2018.

[2] B. Zhang, D. Wu, L. Zhang, Q. Jiao, and Q. Li, "Application of hyperspectral remote sensing for environment monitoring in mining areas," *Environmental Earth Sciences*, vol. 65, pp. 649-658, 2012.

[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, 2013.

[4] Z. Wu, J. Sun, Y. Zhang, Z. Wei, and J. Chanussot, "Recent Developments in Parallel and Distributed Computing for Remotely Sensed Big Data Processing," *Proceedings of the IEEE*, vol. 109, no. 8, pp. 1282-1305, 2021.

[5] S. Zhong, C. I. Chang, and Y. Zhang, "Iterative Support Vector Machine for Hyperspectral Image Classification." pp. 3309-3312.

[6] M. Pal, "Multinomial logistic regression-based feature selection for hyperspectral data," *Int J Appl Earth Obs Geoinf*, vol. 14, no. 1, pp. 214-220, 2012.

[7] B. Tu, J. Wang, X. Kang, G. Zhang, X. Ou, and L. Guo, "KNN-Based Representation of Superpixels for Hyperspectral Image Classification," *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 11, no. 11, pp. 4032-4047, 2018.

[8] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Transactions on Information Theory*, vol. 14, no. 1, pp. 55-63, 1968.

[9] G. Piatetsky-Shapiro, X. E. Bosch, and T. Jung, "High-Dimensional Data Analysis : The Curses and Blessings of Dimensionality."

[10] C. Rodarmel, and J. Shan, "Principal Component Analysis for Hyperspectral Image Classification."

[11] X. Jia, B.-C. Kuo, and M. M. Crawford, "Feature Mining for Hyperspectral Image Classification," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 676-697, 2013.

[12] J. Li, X. Huang, and L. Tu, "WHU-OHS: A benchmark dataset for large-scale Hersepctral Image classification," *Int J Appl Earth Obs Geoinf,* vol. 113, 2022.

[13] F. Luo, L. Zhang, X. Zhou, T. Guo, Y. Cheng, and T. Yin, "Sparse-Adaptive Hypergraph Discriminant Analysis for Hyperspectral Image Classification," *IEEE Geosci. Remote. Sens. Lett.,* vol. 17, no. 6, pp. 1082-1086, 2020.

[14] L. Fang, S. Li, X. Kang, and J. A. Benediktsson, "Spectral–Spatial Classification of Hyperspectral Images With a Superpixel-Based Discriminative Sparse Model," *IEEE Trans. Geosci. Remote Sens.,* vol. 53, no. 8, pp. 4186-4201, 2015.

[15] H. Liu, W. Li, X. G. Xia, M. Zhang, C. Z. Gao, and R. Tao, "Central Attention Network for Hyperspectral Imagery Classification," *IEEE Trans Neural Netw Learn Syst,* vol. PP, Mar 10, 2022.

[16] J. Plaza, A. J. Plaza, and C. Barra, "Multi-Channel Morphological Profiles for Classification of Hyperspectral Images Using Support Vector Machines," *Sensors,* vol. 9, no. 1, pp. 196-218, 2009.

[17] S. Miaohong, and G. Healey, "Hyperspectral texture recognition using a multiscale opponent representation," *IEEE Trans. Geosci. Remote Sens.,* vol. 41, no. 5, pp. 1090-1095, 2003.

[18] R. Ji, Y. Gao, R. Hong, Q. Liu, D. Tao, and X. Li, "Spectral-Spatial Constraint Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.,* vol. 52, no. 3, pp. 1811-1824, 2014.

[19] B. Zhang, S. Li, X. Jia, L. Gao, and M. Peng, "Adaptive Markov Random Field Approach for Classification of Hyperspectral Imagery," *IEEE Geosci. Remote. Sens. Lett.,* vol. 8, no. 5, pp. 973-977, 2011.

[20] X. Kang, S. Li, L. Fang, and J. A. Benediktsson, "Intrinsic Image Decomposition for Feature Extraction of Hyperspectral Images," *IEEE Trans. Geosci. Remote Sens.,* vol. 53, no. 4, pp. 2241-2253, 2015.

[21] Y. Qian, M. Ye, and J. Zhou, "Hyperspectral Image Classification Based on Structured Sparse Logistic Regression and Three-Dimensional Wavelet Texture Features," *IEEE Trans. Geosci. Remote Sens.,* vol. 51, no. 4, pp. 2276-2291, 2013.

[22] Y. Y. Tang, Y. Lu, and H. Yuan, "Hyperspectral Image Classification Based on Three-Dimensional Scattering Wavelet Transform," *IEEE Trans. Geosci. Remote Sens.,* vol. 53, no. 5, pp. 2467-2480, 2015.

[23] Z. Zhu, S. Jia, S. He, Y. Sun, Z. Ji, and L. Shen, "Three-dimensional Gabor feature extraction for hyperspectral imagery classification using a memetic framework," *Information Sciences,* vol. 298, pp. 274-287, 2015/03/20/, 2015.

[24] Y. Xu, B. Du, F. Zhang, and L. Zhang, "Hyperspectral image classification via a random patches network," *ISPRS J. Photogramm. Remote Sens.,* vol. 142, pp. 344-357, 2018.

[25] R. Hang, Q. Liu, D. Hong, and P. Ghamisi, "Cascaded Recurrent Neural Networks for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.,* vol. 57, no. 8, pp. 5384-5394, 2019.

[26] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep Learning-Based Classification of Hyperspectral Data," *IEEE J Sel Top Appl Earth Obs Remote Sens,* vol. 7, no. 6, pp. 2094-2107, 2014.

[27] Y. Chen, X. Zhao, and X. Jia, "Spectral–Spatial Classification of Hyperspectral Data Based on Deep Belief Network," *IEEE J Sel Top Appl Earth Obs Remote Sens,* vol. 8, no. 6, pp. 2381-2392, 2015.

[28] X. Hu, Y. Zhong, X. Wang, C. Luo, J. Zhao, L. Lei, and L. Zhang, "SPNet: Spectral Patching End-to-End Classification Network for UAV-Borne Hyperspectral Imagery With High Spatial and Spectral Resolutions," *IEEE Trans. Geosci. Remote Sens.,* vol. 60, pp. 1-17, 2022.

[29] Z. Zheng, Y. Zhong, A. Ma, and L. Zhang, "FPGA: Fast Patch-Free Global Learning Framework for Fully End-to-End Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.,* vol. 58, no. 8, pp. 5612-5626, 2020.

[30] A. Romero, C. Gatta, and G. Camps-Valls, "Unsupervised Deep Feature Extraction for Remote Sensing Image Classification," *IEEE Trans. Geosci. Remote Sens.,* vol. 54, no. 3, pp. 1349-1362, 2016.

[31] G. Cheng, Z. Li, J. Han, X. Yao, and L. Guo, "Exploring Hierarchical Convolutional Features for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.,* vol. 56, no. 11, pp. 6712-6722, 2018.

[32] H. Lee, and H. Kwon, "Going Deeper With Contextual CNN for Hyperspectral Image Classification," *IEEE Trans Image Process,* vol. 26, no. 10, pp. 4843-4855, Oct, 2017.

[33] X. Xu, W. Li, Q. Ran, Q. Du, L. Gao, and B. Zhang, "Multisource Remote Sensing Data Classification Based on Convolutional Neural Network," *IEEE Trans. Geosci. Remote Sens.,* vol. 56, no. 2, pp. 937-949, 2018.

[34] Y. Li, Q. Xu, W. Li, and J. Nie, "Automatic Clustering-Based Two-Branch CNN for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.,* vol. 59, no. 9, pp. 7803-7816, 2021.

[35] M. Imani, and H. Ghassemian, "An overview on spectral and spatial information fusion for hyperspectral image classification: Current trends and challenges," *Information Fusion,* vol. 59, pp. 59-83, 2020.

[36] 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, 2018.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.

[38] A. Sellami, A. Ben Abbes, V. Barra, and I. R. Farah, "Fused 3-D spectral-spatial deep neural networks and spectral clustering for hyperspectral image classification," *Pattern Recognition Letters,* vol. 138, pp. 594-600, 2020.

[39] S. Jia, Z. Lin, M. Xu, Q. Huang, J. Zhou, X. Jia, and Q. Li, "A Lightweight Convolutional Neural Network for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.,* vol. 59, no. 5, pp. 4150-4163, 2021.

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

[41] L. Mou, and X. X. Zhu, "Learning to Pay Attention on Spectral Domain: A Spectral Attention Module-Based Convolutional Network for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.,* vol. 58, no. 1, pp. 110-122, 2020.

[42] J. Li, R. Cui, B. Li, R. Song, Y. Li, Y. Dai, and Q. Du, "Hyperspectral Image Super-Resolution by Band Attention Through Adversarial Learning," *IEEE Trans. Geosci. Remote Sens.,* vol. 58, no. 6, pp. 4304-4318, 2020.

[43] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module." pp. 3-19.

[44] W. Ma, Q. Yang, Y. Wu, W. Zhao, and X. Zhang, "Double-Branch Multi-Attention Mechanism Network for Hyperspectral Image Classification," *Remote Sens.,* vol. 11, no. 11, 2019.

[45] C. Pu, H. Huang, and L. Yang, "An attention-driven convolutional neural network-based multi-level spectral–spatial feature learning for hyperspectral image classification," *Expert Syst. Appl.,* vol. 185, 2021.

[46] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual Attention Network for Scene Segmentation," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 3141-3149.

[47] H. Sun, X. Zheng, X. Lu, and S. Wu, "Spectral–Spatial Attention Network for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.,* vol. 58, no. 5, pp. 3232-3245, 2020.

[48] X. Zhang, G. Sun, X. Jia, L. Wu, A. Zhang, J. Ren, H. Fu, and Y. Yao, "Spectral–Spatial Self-Attention Networks for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.,* vol. 60, pp. 1-15, 2022.

[49] R. Li, S. Zheng, C. Duan, Y. Yang, and X. Wang, "Classification of Hyperspectral Image Based on Double-Branch Dual-Attention Mechanism Network," *Remote Sens.,* vol. 12, no. 3, 2020.

[50] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *ArXiv,* vol. abs/2010.11929, 2020.

[51] D. Hong, Z. Han, J. Yao, L. Gao, B. Zhang, A. Plaza, and J. Chanussot, "SpectralFormer: Rethinking Hyperspectral Image Classification With Transformers," *IEEE Trans. Geosci. Remote Sens.,* vol. 60, pp. 1-15, 2022.

[52] Z. Xue, Q. Xu, and M. Zhang, "Local Transformer with Spatial Partition Restore for Hyperspectral Image Classification," *IEEE J Sel Top Appl Earth Obs Remote Sens,* pp. 1-1, 2022.

[53] L. Sun, G. Zhao, Y. Zheng, and Z. Wu, "Spectral–Spatial Feature Tokenization Transformer for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.,* vol. 60, pp. 1-14, 2022.

[54] J. Yang, B. Du, and L. Zhang, "From center to surrounding: An interactive learning framework for hyperspectral image classification," *ISPRS J. Photogramm. Remote Sens.,* vol. 197, pp. 145-166, 2023/03/01/, 2023.

[55] M.-H. Guo, C. Lu, Z.-N. Liu, M.-M. Cheng, and S. Hu, "Visual Attention Network," *ArXiv,* vol. abs/2202.09741, 2022.

[56] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "PVT v2: Improved baselines with Pyramid Vision Transformer," *Computational Visual Media,* vol. 8, no. 3, pp. 415-424, 2022/09/01, 2022.

[57] Z. Zhong, Y. Li, L. Ma, J. Li, and W.-S. Zheng, "Spectral–Spatial Transformer Network for Hyperspectral Image Classification: A Factorized Architecture Search Framework," *IEEE Trans. Geosci. Remote Sens.,* vol. 60, pp. 1-15, 2022.

[58] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection." pp. 2999-3007.

[59] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature,* vol. 323, pp. 533-536, 1986.

[60] C. Zhang, G. Li, and S. Du, "Multi-Scale Dense Networks for Hyperspectral Remote Sensing Image Classification," *IEEE Trans. Geosci. Remote Sens.,* vol. 57, no. 11, pp. 9201-9222, 2019.

[61] Y. Xu, L. Zhang, B. Du, and F. Zhang, "Spectral–Spatial Unified Networks for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.,* vol. 56, no. 10, pp. 5893-5909, 2018.

[62] M. Zhu, L. Jiao, F. Liu, S. Yang, and J. Wang, "Residual Spectral–Spatial Attention Network for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.,* vol. 59, no. 1, pp. 449-462, 2021.

[63] D. Wang, B. Du, L. Zhang, and Y. Xu, "Adaptive Spectral–Spatial Multiscale Contextual Feature Extraction for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.,* vol. 59, no. 3, pp. 2461-2477, 2021.

[64] S. K. Roy, S. Manna, T. Song, and L. Bruzzone, "Attention-Based Adaptive Spectral–Spatial Kernel ResNet for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.,* vol. 59, no. 9, pp. 7831-7843, 2021.

[65] D. P. Kingma, and J. Ba, "Adam: A Method for Stochastic Optimization," *CoRR,* vol. abs/1412.6980, 2014.