Exploring the Relationship between Center and Neighborhoods: Central Vector oriented Self-Similarity Network for Hyperspectral Image Classification

面向中心向量的自相似网络

Mingsong Li, Yikun Liu, Guangkuo Xue, Yuwen Huang, and Gongping Yang

***Abstract*—To mine the spectral-spatial information of target pixel in hyperspectral image classification (HSIC), convolutional neural network (CNN)-based models widely adopt patch-based input pattern, where a patch represents its central pixel and the neighbor pixels play auxiliary roles in the classification process. However, compared to the central pixel, its neighbor pixels often have different contributions for classification. Although many existing patch-based CNNs could adaptively emphasize the spatial neighbor information, most of them ignore the latent relationship between the center pixel and its neighbor pixels. Moreover, efficient spectral-spatial feature extraction has been a difficult yet vital topic for HSIC. To address the mentioned problems, a central vector oriented self-similarity network (CVSSN) is proposed for HSIC. Specifically, based on two similarity measures, we firstly design an adaptive weight addition based spectral vector self-similarity module (AWA-SVSS) in input space and a Euclidean distance based feature vector self-similarity module (ED-FVSS) in feature space to fully mine the central vector oriented spatial relationships. Besides, a spectral-spatial information fusion module (SSIF) is formulated as a new pattern to fuse the central 1D spectral vector and the corresponding 3D patch for efficient spectral-spatial feature learning of the subsequent modules. Moreover, we implement a channel spatial separation convolution module (CSS-Conv) and a scale information complementary convolution module (SIC-Conv) for efficient spectral-spatial feature learning. Extensive experimental results on four popular HSI data sets demonstrate the effectiveness and efficiency of the proposed method compared with other state-of-the-art methods.**

***Index Terms*—Hyperspectral image classification (HSIC), self-similarity, spectral-spatial information fusion, efficient spectral-spatial feature learning.**

I. Introduction

HYPERSPECTRAL images (HSIs) contain more than hundreds of narrow spectral bands covering the visible, near-infrared (NIR), and shortwave infrared (SWIR) spectrum [1], [2] between 400 and 2500 nm, which is capable of recording the abundant spectral signatures and spatial information of observed scenes. Therefore, HSI is widely applied in various fields, such as urban development, precision agriculture, and environment management [3]–[5]. Moreover, hyperspectral image classification (HSIC), as the core technique of many Earth observation (EO) tasks [6], aims to assign each HSI pixel to a unique semantic label with the help of HSI characteristics according to the set of given land-cover classes.

In the early research in HSIC community, many classic machine learning methods were developed for HSIC task. To be specific, many spectral-based methods exploited rich spectral signatures for HSIC, e.g., principal component analysis (PCA) [7], support vector machine (SVM) [8], and random forest (RF) [9]. Due to the inherent nonlinearity and spectral-spatial characteristics of HSI [10], spatial information was considered for some spectral-spatial methods for discriminative HSIC [11]–[14]. However, the shallow mode of machine learning methods limits the feature extracting ability and the applicability to different HSI scenarios. Recently, attracted by the outstanding feature extraction capability, many hierarchical deep learning models were applied to the challenging HSIC task and obtained promising classification results, including stacked autoencoders (SAEs) [15], recurrent neural networks (RNNs) [16], graph convolutional networks (GCNs) [17], [18], and transformer [19].

With the local perception and parameter sharing characteristics, convolutional neural networks (CNNs) have gained lots of attention and demonstrated their better performance in HSIC task. For instance, a multiscale densely-connected convolutional network (MS-DenseNet) framework was proposed in [20] to sufficiently exploit multiple scales information for HSIC. To fully explore spectral–spatial features, 3D convolution (Conv) operation was employed in a 3D CNN-based model [21]. Zhong *et al.* [22] proposed a supervised spectral–spatial ResNet (SSRN) using 3D residual blocks for spectral–spatial representation learning. A well-designed model named attention-based adaptive spectral–spatial kernel improved residual network (A^2S^2K-ResNet) was presented in [23], which employed the improved 3D ResBlocks and the efficient feature recalibration (EFR) attention mechanism

Manuscript received ** **, ****; revised ** **, ****; accepted ** **, ****. This work was supported in part by the National Natural Science Foundation of China under Grant U1903127, and in part by the TaiShan Industrial Experts Programme under Grant tscy20200303. *(Corresponding author: Gongping Yang.)*

Mingsong Li, Yikun Liu, and Guangkuo Xue are with the School of Software, Shandong University, Jinan 250100, China (e-mail: msli@mail.sdu.edu.cn; liuyk29@163.com; xueguangkuo@mail.sdu.edu.cn)

Yuwen Huang is with the School of Computer, Heze University, Heze 274015, China (e-mail: hzxy_hyw@163.com)

Gongping Yang is with the School of Software, Shandong University, Jinan 250100, China, and also with the School of Computer, Heze University, Heze 274015, China (e-mail: gpyang@sdu.edu.cn).

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

Digital Object Identifier

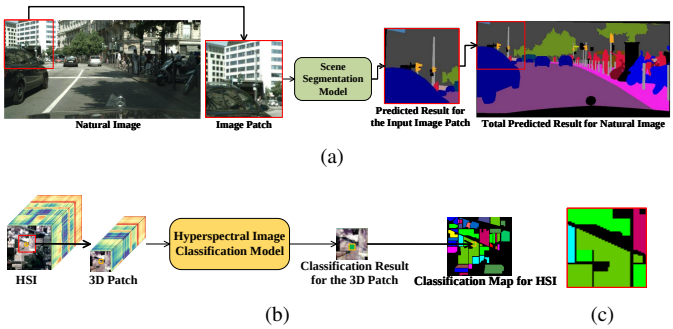

Fig. 1. (a) Patch processing pattern of scene segmentation, image instance from Cityscapes data set [28]. (b) Patch processing pattern of hyperspectral image classification, image instance from Indian Pines (IP) data set. (c) Ground-truth map for all the pixels in the cropped 3D patch.

[24] to boost the classification performance. However, 3D convolutional pattern is accompanied by numerous parameters and a high computational cost, making it difficult to balance performance and efficiency.

Besides 3D convolutional pattern, a lot of works attempted to design parallel dual-branch networks [25]–[27], which learn spectral and spatial features separately and fuse them together. For example, Wang *et al.* [26] proposed an adaptive spectral–spatial multiscale network (ASSMN), consisting of a spectral subnetwork and a spatial subnetwork, to extract multiscale contextual information from spectral and spatial aspects. Zhang *et al.* [27] presented a spectral–spatial self-attention network (SSSAN) in the spectral-spatial dual-branch manner with a spectral self-attention module and a spatial self-attention module. It is worth mentioning that all the above-mentioned dual-branch networks [25]–[27] adopted similar adaptive weight to fuse separate spectral and spatial features. However, with two completely different models in different branches, this spectral-spatial dual-branch pattern faces relatively high model design costs and the tricky spectral-spatial feature fusion problem.

For another side, more and more studies have paid attention to the phenomenon that spatial neighbor pixels of the spectral-spatial patch may have different discriminative abilities. Attention mechanism has made great progress in classic vision tasks, and it is capable of allocating the resource towards the most informative components of the input signal [29]–[32]. Considering above-mentioned issues of patch-based models and the advantage of attention mechanism, many works introduced attention mechanism to highlight the important spatial information for HSIC task. For instance, Sun *et al.* [33] proposed a 3D Conv based spectral-spatial attention network (SSAN) with non-local self-attention block [30] to extract the joint spectral–spatial features for HSIC. Similarly, Zhong *et al.* [19] proposed a spectral–spatial transformer network (SSTN) consisting of spectral and spatial transformer blocks to extract spectral–spatial features. Zhu *et al.* [34] proposed a residual spectral–spatial attention network (RSSAN) with a spectral attention [31] and a spatial attention [32] to refine spectral-spatial feature learning. Apart from directly transferring off-the-shelf attention modules, Hong *et al.* [35] proposed a two-branch attention-aided CNN model that incorporated an original couple of a spectral attention module and a spatial attention module.

Although above-mentioned methods made promising HSI classification performance, the problem is that the patch processing pattern of scene segmentation for classic natural images is different from that of HSIC for HSIs. As shown in Fig. 1 (a), an image patch cropped from the natural image instance is usually as the input of the scene segmentation model, but each pixel in the image patch would be assigned a unique semantic label. The pixels in the image patch are equal and complementary for each other, especially for the inter-adjacent pixels. For comparison, Fig. 1 (b) illustrates the popular patch processing pattern of HSIC, where a patch represents its central pixel and all the neighbor pixels play auxiliary roles. The patch is exploited to assign a semantic label for the central target pixel. Thus, neighbor pixels of the central pixel in the 3D patch provide spatial complementary information for better classification, which is much different from the patch pattern of scene segmentation. As illustrated in Fig.1 (c), when a patch contains the class edge of land-cover, part of the spatial neighbor information in the spatial window would be different from the semantic label of the total patch, i.e., that of the central pixel, and have different contributions in the classification process.

Above all, we claim that it is vital to utilize the central pixel as a standard benchmark to measure different contributions of its neighbor pixels in classification process. However, existing related studies neglected this important insight, because part of them [19], [23], [33], [34] directly transferred and embedded off-the-shelf attention modules [24], [30]–[32] from classic vision tasks into HSIC models, and the other part of them [35], [36] designed original attention modules following the natural image oriented attention mechanism without fully considering the specificity of HSI and patch processing pattern of HSIC. Fortunately, Zhang *et al.* [27] designed a spatial self-attention module based on cosine similarity between the central vector and its neighbor vectors in feature space to explore the spatial feature correlation. However, SSSAN only discussed the spatial feature correlation in feature space. Although the Conv operation keeps spatial translation-equivariant [37], the fact is that the feature vector in feature space is completely distinct from the spectral vector in original input space and part of the spectral signatures may be lost in spectral-spatial learning process. All in all, we argue that the relationship between the central spectral vector and its neighbor spectral vectors in original input space is as significant as that between the central feature vector and its neighbor feature vectors in high-level feature space.

To alleviate the three main above-mentioned drawbacks, we propose a central vector oriented self-similarity network (CVSSN) for HSIC. Specifically, as for original input space, we design an adaptive weight addition based spectral vector self-similarity module (AWA-SVSS) using Euclidean distance similarity measure and cosine angle similarity measure to improve the spatial information representation. Meanwhile, for feature space, we formulate a Euclidean distance based feature vector self-similarity module (ED-FVSS) to enhance the spatial feature representation. Moreover, we propose a spectral-spatial information fusion module (SSIF) for model

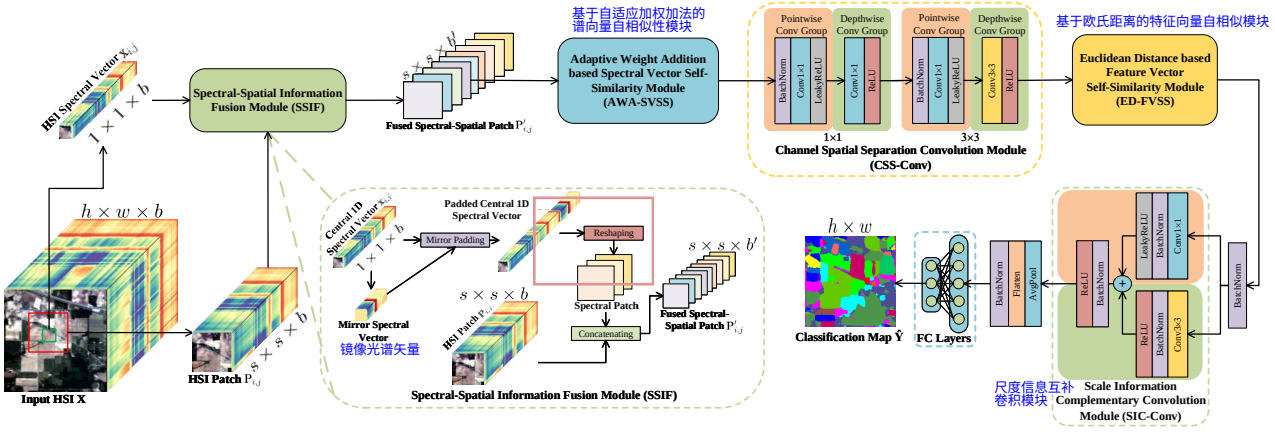

Fig. 2. Framework of the proposed CVSSN, which could be split into a spectral-spatial feature extraction part and a classification part. The former consists of an SSIF module for spectral-spatial information fusion, an AWA-SVSS module for spatial relationship mining in input space, a CSS-Conv module for efficient spectral-spatial feature learning, an ED-FVSS module for spatial relationship mining in feature space, and a SIC-Conv module for the mutual complementation of scale information. The latter part mainly contains a softmax-based fully connected classifier for HSIC.

inputs as a new pattern to fuse the central 1D spectral vector and the corresponding 3D patch for efficient specral-spatial feature learning of the subsequent feature extraction modules in convenient 2D Conv pattern. In addition, in order to keep the model efficiency, we implement a channel spatial separation convolution module (CSS-Conv) and a scale information complementary convolution module (SIC-Conv) as the fundamental modules for efficient spectral-spatial feature learning. To make the proposed model easier to be understood, Fig. 2 illustrates its main framework, in which the Indian Pines data set is taken into account. The major contributions of this article are summarized as follows.

1) To explore the relationship between the central spectral vector and its neighbor spectral vectors in original input space, we propose an AWA-SVSS module adopting Euclidean distance and cosine angle similarity measures, which directly mines spatial information without spectral signatures loss.

2) To capture the relationship between the central feature vector and its neighbor feature vectors in high-level feature space, we propose a ED-FVSS module, which takes fully advantage of spatial feature information to enhance the spectral-spatial feature representation.

3) For model inputs, we design an SSIF module as a new pattern to fuse the central 1D spectral vector and the corresponding 3D spectral-spatial patch for efficient 2D Conv-based spectral-spatial feature extraction of the subsequent modules, which effectively avoids the computational cost of 3D convolutional pattern and the model design complexity of spectral-spatial dual-branch pattern.

4) To balance the performance and efficiency of the proposed CVSSN, we devise a CSS-Conv module and a SIC-Conv module as the fundamental modules for efficient spectral-spatial feature learning.

The remainder of this article is organized as follows. Section II introduces the proposed CVSSN in detail. Experimental settings, results, and corresponding analyses are reported in Section III. Finally, conclusions and future research direction are drawn in Section IV.

II. Proposed Method

In formal, in a raw HSI $\mathbf{X} \in \mathbb{R}^{h\times w\times b}$, $\mathbf{X}$ is composed of $t = p + q$ pixels, $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_p, \mathbf{x}_{p+1}, ..., \mathbf{x}_t\}$, $\mathbf{X}$=$\mathbf{X}_{\{p\}} \cup \mathbf{X}_{\{q\}}$, and $t = h \cdot w$, where $\mathbf{X}_{\{p\}}$ denotes the set of the first p labeled pixels, and $\mathbf{X}_{\{q\}}$ denotes the set of the remaining q unlabeled pixels. Each pixel $\mathbf{x}_{i,j} \in \mathbb{R}^b$ in $\mathbf{X}_{\{p\}}$ corresponds a class semantic label $y_{i,j} = k$ belonging to the corresponding label set $\mathbf{Y}_{\{p\}} = \{y_1, ..., y_p\}$, where $k = 1, ..., K$. Here, h, w, b, and K represent the height of spatial dimension, the width of spatial dimension, the number of spectral bands, and the number of classes, respectively. $i = 1, ..., h$ and $j = 1, ..., w$ jointly represent the position of a pixel in HSI $\mathbf{X}$. The purpose of HSI classification task is assigning a semantic label $y_{i,j}$ for each pixel $\mathbf{x}_{i,j}$ of HSI $\mathbf{X}$ in accordance with the couple of labeled pixel set $\mathbf{X}_{\{p\}}$ and label set $\mathbf{Y}_{\{p\}}$. To make full use of the joint spectral-spatial information of HSI, the raw 3D HSI $\mathbf{X}$ is divided in a set of small overlapping 3D patches $\mathbf{P}$ in most cases, and each patch $\mathbf{P}_{i,j} \in \mathbb{R}^{s\times s\times b}$ in $\mathbf{P}$ covering the spatial window of size $s \times s$ and all the b spectral bands is centered at pixel $\mathbf{x}_{i,j}$, and the label of $\mathbf{P}_{i,j}$ depends on the label of $\mathbf{x}_{i,j}$.

As shown in Fig. 2, the proposed CVSSN is an end-to-end patch-based HSIC model. As defined above, the input $\mathbf{X} \in \mathbb{R}^{h\times w\times b}$ denotes the raw 3D HSI, the labeled pixel $\mathbf{x}_{i,j}$ acts as an input central 1D spectral vector and $\mathbf{P}_{i,j}$ represents the corresponding input 3D spectral-spatial patch. Moreover, the output $\hat{\mathbf{Y}} \in \mathbb{R}^{h\times w}$ denotes the predicted class label map for $\mathbf{X}$. In particular, CVSSN could be split into a spectral-spatial feature extraction part and a classification part. The former part consists of five different modules, including an SSIF module, an AWA-SVSS module, a CSS-Conv module, an ED-FVSS module, and a SIC-Conv module in sequence. The latter part mainly contains a stand softmax-based fully connected classifier. In the following, we will describe each module of CVSSN in detail.

A. SSIF Module

To avoid the computational cost of 3D convolutional pattern [21]–[23], [33], [38] and the model design complexity of

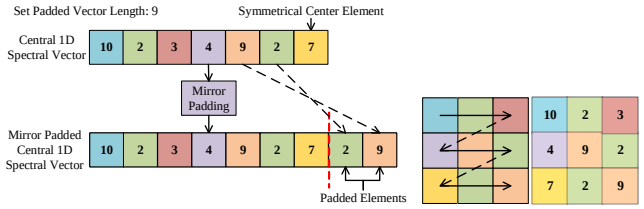

Fig. 3. (a) Mirror padding for a central 1D spectral vector instance. According to the set padded vector length, i.e., 9, the elements 2 and 9 from the central 1D spectral vector are selected to pad the central 1D spectral vector. Besides, the padding elements selection and the padding process are both symmetric about the symmetrical center element 7, i.e., the element of the last band of the central 1D spectral vector. (b) The adopted horizontal row-wise reshaping direction of the padded central 1D spectral vector and the corresponding instance for one of spectral channels maps of the reshaped 3D spectral patch.

spectral-spatial dual-branch pattern [25]–[27], for the inputs of CVSSN, we try a new pattern term as spectral-spatial information fusion module (SSIF) to fuse the central 1D spectral vector and the corresponding 3D patch into a new spectral-spatial patch for efficient spectral-spatial learning of the subsequent feature extraction modules in 2D Conv pattern.

As illustrated in Fig. 2, SSIF firstly reshapes a central 1D spectral vector $\mathbf{x}_{i,j} \in \mathbb{R}^b$ as 3D spectral patch to implement the concatenation between the reshaped 3D spectral patch and the original input 3D spectral-spatial patch $\mathbf{P}_{i,j} \in \mathbb{R}^{s\times s\times b}$. Besides, mirror padding is employed to use partial consecutive elements of the central 1D spectral vector to pad itself for guaranteeing that each channel window of the reshaped 3D spectral patch is filled with elements.

As illustrated in Fig. 3 (a), the core of mirror padding is that the padding elements selection and the padding process both are symmetric about the symmetrical center element, i.e., the element of the last band of the central 1D spectral vector. In addition, the set padded vector length is free of manual setting, which depends on the spectral bands of different HSI data sets and the spatial window size of the cropped 3D patch. Fig. 3 (b) illustrates that the reshaping direction of the central 1D spectral vector is horizontal row-wise [39], [40]. As illustrated in Fig. 3, both of the adopted mirror padding and horizontal row-wise reshaping direction help to maintain the original continuity of the spectral vector. Note that the reshaped 3D spectral patch keeps the same size of spatial dimension window as that of the original input 3D patch.

The operations of the SSIF module could be expressed as follows:

$$\mathbf{P}'_{i,j} = Concat(Reshape(MirrorPad(\mathbf{x}_{i,j})), \mathbf{P}_{i,j}) \quad (1)$$

where $\mathbf{P}'_{i,j} \in \mathbb{R}^{s\times s\times b'}$ denotes the fused spectral-spatial patch, i.e., the module output. b' is the number of spectral bands of the fused spectral-spatial patch $\mathbf{P}'_{i,j}$. $MirrorPad$, $Reshape$, and $Concat$ denote corresponding operations in Fig. 2. Particularly, $Concat$ is along the depth direction of $\mathbf{P}_{i,j}$.

More importantly, when 2D Conv of the subsequent feature modules operates the fused spectral-spatial patch, it would capture the spatial feature from the original spectral channels and exploit local and non-local spectral feature from the newly added channels, and then make an integration for spectral-spatial feature learning. Hence, the proposed SSIF pattern is a new attempt to enable the proposed CVSSN to extract spectral-spatial features using the efficient 2D Conv pattern, which is different from 3D convolutional pattern and spectral-spatial dual-branch pattern.

B. AWA-SVSS Module and ED-FVSS Module

As discussed above, existing studies [19], [23], [27], [33]–[36] employing attention mechanism fail to explore the relationship between the central spectral vector and its neighbor spectral vectors in original input space and high-level feature space. Thus, as illustrated in Figs. 4-5, an adaptive weight addition based spectral vector self-similarity module (AWA-SVSS) and a Euclidean distance based feature vector self-similarity module (ED-FVSS) are specially designed in the proposed CVSSN.

1) AWA-SVSS Module: As mentioned above, AWA-SVSS module is designed for explore contributions of spatial complementary information from different neighbor spectral pixels in original input space. As shown in Fig. 4, the module input is the fused spectral-spatial patch $\mathbf{P}'_{i,j} \in \mathbb{R}^{s\times s\times b'}$, which is fused from SSIF as the first layer transformation. The central spectral vector is denoted as $\hat{\mathbf{x}}^0_{i,j} \in \mathbb{R}^{b'}$, which could be directly cropped from $\mathbf{P}'_{i,j}$. Firstly, two efficient similarity measures, i.e., Euclidean distance similarity measure and cosine angle similarity measure, are considered to calculate corresponding central spectral vector oriented self-similarity representations, $\widehat{\mathbf{E}}_{i,j} \in \mathbb{R}^{s\times s}$ and $\widehat{\mathbf{C}}_{i,j} \in \mathbb{R}^{s\times s}$. In detail,

$$\widehat{\mathbf{E}}_{i,j} = EDSim(\hat{\mathbf{x}}^0_{i,j}, \mathbf{P}'_{i,j}) \quad (2)$$

where $EDSim$ calculates the Euclidean distance similarity matrix $\widehat{\mathbf{E}}_{i,j}$ by computing the Euclidean distance similarity value between the central spectral vector $\hat{\mathbf{x}}^0_{i,j}$ and a neighbor spectral vector $\hat{\mathbf{x}}_{i',j'} \in \mathbb{R}^{b'}$ from the fused spectral-spatial patch $\mathbf{P}'_{i,j}$, i.e.,

$$\begin{aligned} \hat{e}_{i',j'} &= EDSimV(\hat{\mathbf{x}}^0_{i,j}, \mathbf{x}_{i',j'}) \\ &= \frac{1}{1+\parallel \hat{\mathbf{x}}^0_{i,j} - \mathbf{x}_{i',j'} \parallel} \end{aligned} \quad (3)$$

where $\hat{e}_{i',j'}$ denotes the calculated Euclidean distance similarity value between $\hat{\mathbf{x}}^0_{i,j}$ and $\mathbf{x}_{i',j'}$ by $EDSimV$, which is the similarity value located in the position (i', j') of $\widehat{\mathbf{E}}_{i,j}$, $i' = 1, ..., s$ and $j' = 1, ..., s$. Note the similarity value has been normalized as the unified representation that the higher the similarity, the closer the similarity value is to 1.

Similarly, the cosine similarity matrix $\widehat{\mathbf{C}}_{i,j}$ and the cosine similarity value $\hat{c}_{i',j'}$ coordinated at (i', j') are computed as follows:

$$\widehat{\mathbf{C}}_{i,j} = CosSim(\hat{\mathbf{x}}^0_{i,j}, \mathbf{P}'_{i,j}) \quad (4)$$

$$\hat{c}_{i',j'} = CosSimV(\hat{\mathbf{x}}^0_{i,j}, \mathbf{x}_{i',j'}) = \frac{\hat{\mathbf{x}}^0_{i,j} \cdot \mathbf{x}_{i',j'}}{\|\hat{\mathbf{x}}^0_{i,j}\| \|\mathbf{x}_{i',j'}\|} \quad (5)$$

where $CosSim$ calculates $\widehat{\mathbf{C}}_{i,j}$ by computing cosine angle similarity value between $\hat{\mathbf{x}}^0_{i,j}$ and each $\mathbf{x}_{i',j'}$ through $CosSimV$.

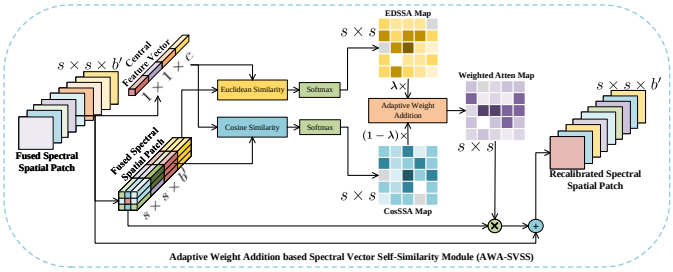

Fig. 4. Structure of the proposed AWA-SVSS module, which employs Euclidean distance similarity measure and cosine angle similarity measure in an adaptive weight addition pattern to mine the central spectral vector oriented spatial relationship in original input space.

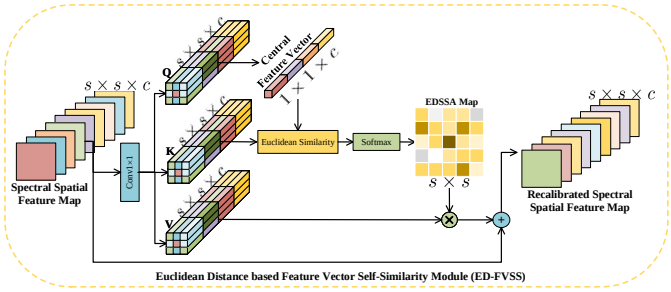

Fig. 5. Structure of the proposed ED-FVSS module, which employs Euclidean distance similarity measure to mine the central feature vector oriented spatial relationship in high-level feature space.

Moreover, the softmax function is used to further normalize and produce the Euclidean distance self-similarity attention map $\widehat{\mathbf{ESS}}_{i,j} \in \mathbb{R}^{s\times s}$ and the cosine self-similarity attention map $\widehat{\mathbf{CSS}}_{i,j} \in \mathbb{R}^{s\times s}$ from the corresponding similarity matrix, respectively. In detail,

$$\widehat{\mathbf{ESS}}_{i,j} = Softmax(\widehat{\mathbf{E}}_{i,j}) = \frac{exp(\hat{e}_{i',j'})}{\sum_{i'=1}^{s}\sum_{j'=1}^{s} exp(\hat{e}_{i',j'})} \tag{6}$$

$$\widehat{\mathbf{CSS}}_{i,j} = Softmax(\widehat{\mathbf{C}}_{i,j}) = \frac{exp(\hat{c}_{i',j'})}{\sum_{i'=1}^{s}\sum_{j'=1}^{s} exp(\hat{c}_{i',j'})} \tag{7}$$

where softmax function guarantees that the sum of all element values of the target self-similarity attention map is 1 and the non-negativity of the self-similarity attention map value, especially for $\widehat{\mathbf{CSS}}_{i,j}$.

Furthermore, we adopt an adaptive weight addition pattern to fuse two self-similarity attention maps for enhancing the spatial information representation, i.e.,

$$\mathbf{SS}^{\lambda}_{i,j} = \lambda \times \widehat{\mathbf{ESS}}_{i,j} + (1-\lambda) \times \widehat{\mathbf{CSS}}_{i,j} \tag{8}$$

where $\mathbf{SS}^{\lambda}_{i,j} \in \mathbb{R}^{s\times s}$ denotes the fused self-similarity attention map. λ is a weighting parameter set 0.5 as the initial value, which could be adaptively optimized during model optimization.

Finally, the recalibrated spectral-spatial patch $\mathbf{X}^{1}_{i,j} \in \mathbb{R}^{s\times s\times b'}$ is produced by

$$\mathbf{X}^{1}_{i,j} = \mathbf{SS}^{\lambda}_{i,j} \otimes \mathbf{P}'_{i,j} + \mathbf{P}'_{i,j} \tag{9}$$

where $\otimes$ represents the element-wise multiplication along the spectral channel dimension. In particular, an element-wise addition operation with $\mathbf{P}'_{i,j}$ is performed to avoid vanishing gradient and network degradation.

To sum up, all the operations of AWA-SVSS module are performed in the input space, which is neglected by existing study [27]. All the neighbor pixels in the module input $\mathbf{P}'_{i,j}$ are adaptively weighted according to the similarity relationship with the central spectral vector $\hat{\mathbf{x}}^{0}_{i,j}$. Since the spectral vector of input space is without spectral information loss, we specially employ two different similarity measures to exploit more discriminative spatial relationship. The experiments on AWA-SVSS module design and performance are discussed in Section III-B-2.

2) ED-FVSS Module: As discussed above, ED-FVSS module is designed to mine the spatial relationship between the central feature vector and its neighbor feature vectors in feature space. As illustrated in Fig. 5, ED-FVSS module takes feature map $\mathbf{X}^{l}_{i,j} \in \mathbb{R}^{s\times s\times c}$ from the lth layer transformation as its module input, where c is the number of channels in feature space. In ED-FVSS module, the $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V} \in \mathbb{R}^{s\times s\times c}$ are firstly generated from $\mathbf{X}^{l}_{i,j}$ by a 1×1 Conv layer. Specifically, the central feature vector $\hat{\mathbf{x}}^{l}_{i,j} \in \mathbb{R}^{c}$ is cropped from $\mathbf{Q}$.

Moreover, Euclidean distance similarity is employed to measure the similarity relationship between $\hat{\mathbf{x}}^{l}_{i,j}$ and each neighbor feature vector of feature map $\mathbf{K}$ to produce central feature vector oriented self-similarity representation $\widetilde{\mathbf{E}}_{i,j} \in \mathbb{R}^{s\times s}$, i.e.,

$$\widetilde{\mathbf{E}}_{i,j} = EDSim(\hat{\mathbf{x}}^{l}_{i,j}, \mathbf{K}) \tag{10}$$

where each Euclidean distance similarity value of $\widetilde{\mathbf{E}}_{i,j}$ is also calculated by Eq. (3). Then corresponding Euclidean distance self-similarity attention map $\widetilde{\mathbf{ESS}}_{i,j} \in \mathbb{R}^{s\times s}$ is generate through softmax function, which is simply expressed as

$$\widetilde{\mathbf{ESS}}_{i,j} = Softmax(\widehat{\mathbf{E}}_{i,j}) \tag{11}$$

Finally, the recalibrated spectral-spatial feature map $\mathbf{X}^{l+2}_{i,j} \in \mathbb{R}^{s\times s\times c}$ is obtained by

$$\mathbf{X}^{l+2}_{i,j} = \widetilde{\mathbf{ESS}}_{i,j} \otimes \mathbf{V} + \mathbf{X}^{l}_{i,j} \tag{12}$$

where the intermediate generated feature map $\mathbf{V}$ is reweighted by $\widetilde{\mathbf{ESS}}_{i,j}$, and the element-wise addition operation with the module input $\mathbf{X}^{l}_{i,j}$ is also performed like as in AWA-SVSS module.

All in all, ED-FVSS module is generally similar to the classic self-attention mechanism [30], [41], and the difference between them lies in the Euclidean distance similarity measure between the central feature vector $\hat{\mathbf{x}}^{l}_{i,j}$ and its neighbor feature vectors. Based on the fact that Conv operation keep spatial translation-equivariant [37], the central spectral vector $\hat{\mathbf{x}}^{0}_{i,j}$ in the fused spectral-spatial patch $\mathbf{P}'_{i,j}$ would be extracted as high-level spectral-spatial feature and further keep the same spatial location in the lth layer transformed feature map. Hence, $\hat{\mathbf{x}}^{l}_{i,j}$ is also the most correlated feature vector with the original $\hat{\mathbf{x}}^{0}_{i,j}$ of feature space. Since it directly pays attention to the most effective relationship of $\hat{\mathbf{x}}^{l}_{i,j}$ to enhance the spectral-spatial feature representation, ED-FVSS module is

more suitable and efficient for spatial feature exploiting in the patch-based CNN model for HSIC task than the classic self-attention mechanism. The experiments on ED-FVSS module design and performance are also discussed in Section III-B-2.

C. CSS-Conv Module and SIC-Conv Module

As illustrated in Fig. 2, based on the mined central vector oriented spatial relationships in input space and feature space, we specially design a channel spatial separation convolution module (CSS-Conv) and a scale information complementary convolution module (SIC-Conv) as two fundamental modules employing 1×1 Conv and 3×3 Conv to extract discriminative spectral-spatial features and maintaining model efficiency.

1) CSS-Conv Module: The depthwise separable convolution [42] splits the standard Conv into a pointwise Conv and a depthwise Conv to reduce computational cost and model parameters. Many works [18], [43], [44] have developed the depthwise separable convolution for HSIC task. Inspired by the spectral–spatial Conv of CEGCN [18], our well-designed CSS-Conv module consists of a 1×1 CSS-Conv and a 3×3 CSS-Conv for efficient spectral-spatial feature learning, and each CSS-Conv is comprised of a pointwise convolutional group (PCG) and a depthwise convolutional group (DCG).

In formal, given the lth layer transformed feature map $\mathbf{X}_{i,j}^{l} \in \mathbb{R}^{s\times s\times b'}$ as the input of k×k CSS-Conv, which denotes a CSS-Conv with k×k Conv layer in the depthwise convolutional group. In the PCG, $\mathbf{X}_{i,j}^{l}$ is firstly normalized by batch normalization (BN) operation $\mathcal{B}$ [45] and then transformed by 1×1 Conv layer with leaky rectified linear unit (LeakyReLU) activation function $\mathcal{LR}$ [46]. In the DCG, $\mathbf{X}_{i,j}^{l+1}$ is directly transformed by k×k Conv layer with rectified linear unit (ReLU) activation function $\mathcal{R}$ [47]. For convenience, $CSS\text{-}Conv_{k\times k}$ denotes the transformations of k×k CSS-Conv, i.e.,

$$\begin{aligned}\mathbf{X}_{i,j}^{l+2} &= CSS\text{-}Conv_{k\times k}(\mathbf{X}_{i,j}^{l}) \\ &= \mathcal{R}(\mathbf{W}_{k\times k}^{l+2} * \mathcal{LR}(\mathbf{W}_{1\times 1}^{l+1} * \mathcal{B}(\mathbf{X}_{i,j}^{l})) + \mathbf{b}^{l+2})\end{aligned} \tag{13}$$

where $\mathbf{X}_{i,j}^{l+2} \in \mathbb{R}^{s\times s\times c}$ represents the generated feature map from k×k CSS-Conv. $\mathbf{W}_{1\times 1}^{l+1}$, $\mathbf{W}_{k\times k}^{l+2}$ and $\mathbf{b}^{l+2}$ are the weight of 1×1 Conv layer, the weight and the bias of k×k Conv layer, respectively, and $*$ represents 2D Conv operator. Note that 1×1 Conv layer of PCG is employed without bias.

Furthermore, the total transformations of CSS-Conv module could be expressed as

$$\mathbf{X}_{i,j}^{l+4} = CSS\text{-}Conv_{3\times 3}(CSS\text{-}Conv_{1\times 1}(\mathbf{X}_{i,j}^{l})) \tag{14}$$

where $\mathbf{X}_{i,j}^{l+4} \in \mathbb{R}^{s\times s\times c}$ denotes the final CSS-Conv module output. The difference between CSS-Conv and the spectral–spatial Conv of CEGCN [18] is ReLU activation used in the DCG, which is a simple yet vital design and the related comparison experiment is presented in Section III-B-3.

2) SIC-Conv Module: As illustrated in Fig. 2, we design the SIC-Conv to further extract spectral-spatial features through a 1×1 convolutional scale branch and a 3×3 convolutional scale branch. Formally, note the lth layer transformed feature map $\mathbf{X}_{i,j}^{l} \in \mathbb{R}^{s\times s\times c}$ as the input of SIC-Conv module. To be specific, 1×1 convolutional scale branch consists of 1×1 Conv layer, BN layer, and LeakyReLU function in sequence, and 3×3 convolutional scale branch is comprised of 3×3 Conv layer, BN layer, and ReLU function. Then the element-wise addition operation is employed to fuse the different features of two scale branches to implement the mutual complementation of scale information. The total transformations of SIC-Conv module could be formulated as

$$\begin{aligned}\mathbf{X}_{i,j}^{l+1} &= SIC\text{-}Conv(\mathbf{X}_{i,j}^{l}) \\ &= \mathcal{R}(\mathcal{B}(\mathcal{LR}(\mathcal{B}(\mathbf{W}_{1\times 1}^{l+1} * \mathbf{X}_{i,j}^{l})) + \mathcal{R}(\mathcal{B}(\mathbf{W}_{3\times 3}^{l+1} * \mathbf{X}_{i,j}^{l}))\end{aligned} \tag{15}$$

where $\mathbf{X}_{i,j}^{l+1} \in \mathbb{R}^{s\times s\times c}$ denotes the feature map generated by SIC-Conv module. $\mathbf{W}_{1\times 1}^{l+1}$ and $\mathbf{W}_{3\times 3}^{l+1}$ represent the weight of the corresponding Conv layer in two scale branches. Note that we do not use bias in both Conv layers due to the following BN operation.

D. Final Classification of CVSSN

As mentioned above, the spectral-spatial feature extraction part of CVSSN is comprised of an SSIF module, an AWA-SVSS module, a CSS-Conv module, an ED-FVSS module, and a SIC-Conv module in order. For the classification part, a global average pooling layer is adopted to transform the $s\times s\times c$ extracted spectral-spatial feature map to the $1\times 1\times c$ feature vector. Then a fully connected layer with a softmax function is adopted for HSIC, where softmax function outputs the class probability vector $\hat{\mathbf{y}}_{i,j} = [p_1, ..., p_K] \in \mathbb{R}^K$. Furthermore, the classic cross-entropy loss is employed to train and optimize the proposed CVSSN, which is formulated as

$$\mathcal{L} = -\frac{1}{M}\sum_{m=1}^{M}\sum_{k=1}^{K} In(y_m = k) log(\hat{y}_m(p_k)) \tag{16}$$

where M is the number of samples in a minibatch sample set, y_m is the ground truth of the mth HSI patch sample $\mathbf{P}_m$ in the current minibatch sample set, $In(y_m = k)$ is the indicator function (if $y_m = k$, $In(y_m = k)$ is 1; else, $In(y_m = k)$ is 0), $\hat{y}_m(p_k)$ is the softmax output probability of $\mathbf{P}_m$ belonging to the kth class.

III. EXPERIMENTS

A. Data Sets Description and Evaluation Metrics

In order to fairly demonstrate the effectiveness and efficiency of the proposed CVSSN, we conduct extensive experiments on four popular HSI data sets, i.e., Indian Pines (IP)[1], Kennedy Space Center (KSC)[1], University of Pavia (UP)[1], and University of Houston 13 (UH)[2].

The IP data set was gathered by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) [1] sensor over the Indian Pines agricultural test site in North-western Indiana in June 1992, which contains 145×145 pixels with a spatial resolution of 20 m per pixel. After removing 20 water absorption and low signal-noise ratio (SNR) bands, the remaining 200

[1]IP, KSC, and UP data sets are available from http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes#anomaly_detection

[2]UH data set is available from https://hyperspectral.ee.uh.edu/?page_id=459

TABLE I
NUMBER OF TRAINING, VALIDATION, AND TESTING SAMPLES FOR IP, KSC, UP, AND UH DATA SETS

No.	Indian Pines (IP) (10%)				Kennedy Space Center (KSC) (10%)				University of Pavia (UP) (5%)				University of Houston (UH) (fixed ratio)			
	Class	Train.	Val.	Test.	Class	Train.	Val.	Test.	Class	Train.	Val.	Test.	Class	Train.	Val.	Test.
1	Alfalfa	4	1	41	Scrub	76	7	678	Asphalt	331	33	6267	Grass-healthy	198	3	1050
2	Corn-notill	142	14	1272	Willow-swamp	24	2	217	Meadows	932	93	17624	Grass-stressed	190	3	1061
3	Corn-mintill	83	8	739	Cp-hammock	25	2	229	Gravel	104	10	1985	Grass-synth	192	3	502
4	Corn	23	2	212	Cp/O-hammock	25	2	225	Trees	153	15	2896	Tree	188	3	1053
5	Grass-pasture	48	4	431	Slash-pine	16	1	144	Painted-m-s	67	6	1272	Soil	186	3	1053
6	Grass-trees	73	7	650	Oak/B-hammock	22	2	205	Bare-soil	251	25	4753	Water	182	3	140
7	Grass-pasture-m	3	1	16	HW-swamp	10	1	94	Bitumen	66	6	1258	Residentiald	196	3	1069
8	Hay-windrowed	47	4	427	Graminoid-marsh	43	4	384	Self-block-b	184	18	3480	Commercial	191	3	1050
9	Oats	3	1	16	Spartine-marsh	52	5	463	Shadows	47	4	896	Road	193	3	1056
10	Soybean-notill	97	9	866	Cattail-marsh	40	4	360					Highway	191	3	1033
11	Soybean-mintill	245	24	2186	Salt-marsh	41	4	374					Railway	181	3	1051
12	Soybean-clean	59	5	529	Mud-flats	50	5	448					Parking-lot1	192	3	1038
13	Wheat	20	2	183	Water	92	9	826					Parking-lot2	184	3	282
14	Woods	126	12	1127									Tennis-court	181	3	244
15	Buildings-g-t	38	3	345									Running-track	187	3	470
16	Stone-steel-t	9	1	83												
	Total	1020	98	9131	Total	516	48	4647	Total	2135	210	40431	Total	2832	45	12152

bands with 10249 labeled pixels within a wavelength range of 400-2500 nm are adopted for analysis. The ground truth is designated into 16 classes, and the number of samples in some classes is highly imbalanced.

The KSC data set, collected by AVIRIS [1] instrument over the Kennedy Space Center in the USA in 1996, contains 512×614 pixels with a spatial resolution of 18 m per pixel and a wavelength ranging from 400 to 2500 nm. After removing water absorption and low SNR bands, 176 bands are used for the analysis. The KSC data set consists of 13 upland and wetland classes with 5211 labeled pixels.

The UP data set was captured by Reflective Optics Spectrographic Imaging System (ROSIS) [48] during a flight campaign, over Pavia, Northern Italy, 2002. It is composed of 610×340 pixels with a spatial resolution of 1.3 m per pixel and 103 spectral bands covering from 430 to 860 nm after removing 12 noisy bands. The UP data set contains 42776 labeled pixels of 9 urban classes.

The UH data set was acquired by the ITRES-CASI (Compact Airborne Spectrographic Imager) 1500 sensor over the University of Houston campus and its neighboring urban area in June 2012. The data is composed of 349 × 1905 pixels with 144 spectral channels ranging from 364 to 1046 nm and a spatial resolution of 2.5 m per pixel. Additionally, ground-truth reference was subdivided into spatially disjoint subsets for training and testing, which includes 15 mutually exclusive urban land-cover classes with 15029 labeled pixels.

In addition, the detailed class information of each data set is uniformly reported in Table I. About 10%, 1%, and 89% labeled pixels of total labeled pixels per class are randomly selected as training, validation, testing set for IP and KSC scenes, while about 5%, 0.5%, and 94.5% for the UP scenario. As for the UH data set, based on the fixed data set division, 3 samples per class in the testing set are randomly selected as the validation set.

To quantitatively compare the classification performance of different methods and modules from various aspects, four common evaluation metrics, i.e., overall accuracy (OA), average accuracy (AA), kappa coefficient ($\mathcal{K}$), and accuracy of each class (AEC), are employed in the following experiments. Particularly, OA is the ratio of the number of correctly classified labeled samples to the number of total samples in the testing set. AA is the mean of the accuracies of all the land-cover classes. Kappa comprehensively measures the consistency between classification result and ground truth. AEC is the accuracy of each category, which is particularly useful for imbalanced data. Additionally, for quantitatively efficiency analysis, We take training time (T_{train}) and testing time (T_{test}) to jointly evaluate the running time of each method. Model parameters (Params) and floating-point operations (FLOPs) are adopted to measure model complexity and computational cost of each method, respectively.

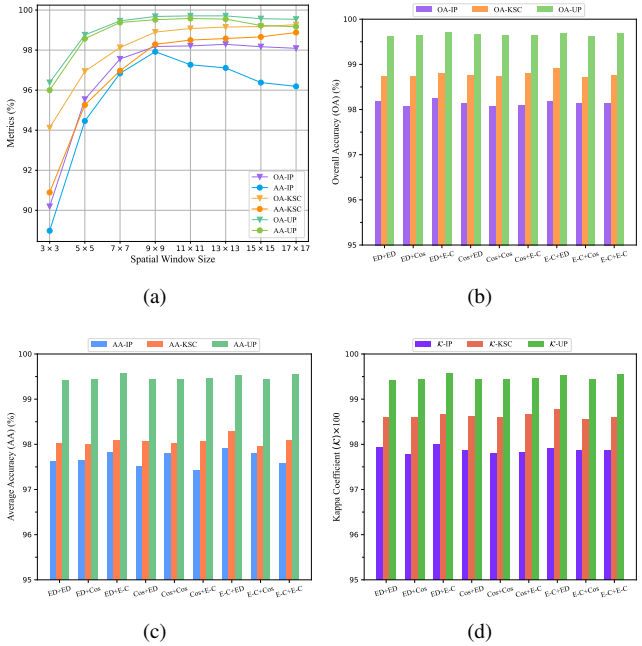

Fig. 6. Setting experiments on IP, KSC, and UP data sets. (a) OAs and AAs of CVSSN with different spatial window sizes. (b)-(d) Classification results of CVSSN with different similarity measure combinations in SVSS and FVSS modules. (b) OAs. (c) AAs. (d) $\mathcal{K}$s.

B. Experimental Settings and Model Design

All experiments are performed on a workstation with an Intel Core CPU i9-10900K with 64 GB RAM and a single Nvidia GeForce RTX 3090 GPU with 24 GB GPU memory. The software environment is the operating system of Ubuntu

TABLE II
CLASSIFICATION RESULTS OF CVSSN WITH DIFFERENT LEAKYRELU AND RELU COMBINATIONS FOR 1×1 CONV AND 3×3 CONV IN CSS-CONV AND SIC-CONV MODULES ON IP AND KSC DATA SETS

Data Set	Metric	LR+LR	LR+R	R+LR	R+R
IP	OA (%)	**98.19±0.23**	98.18±0.27	98.17±0.19	98.19±0.34
	AA (%)	97.54±0.85	**97.92±0.75**	97.29±0.75	97.42±1.14
	$\mathcal{K} \times 100$	**97.94±0.26**	97.92±0.30	97.91±0.21	97.92±0.39
KSC	OA (%)	98.76±0.27	**98.90±0.30**	98.83±0.24	98.82±0.27
	AA (%)	98.08±0.38	**98.29±0.45**	98.15±0.36	98.15±0.38
	$\mathcal{K} \times 100$	98.62±0.30	**98.78±0.33**	98.69±0.26	98.69±0.30

18.04.5 LTS 64 bit, and our proposed CVSSN is implemented by Python-3.8.5 with pytorch-1.8.1 framework. For the CVSSN training, the Adam [49] optimizer is employed, and batch size, learning rate, and the number of training epochs are set to 32, 0.001, and 100, respectively. In addition, the whole process is repeated ten times to record the average accuracy and the standard deviation. Besides, all the data sets are pre-processed by band-wise de-Mean and variance standardization. Following the standard patch processing pattern of HSIC task, the zero padding operation is adopted to pad $(s-1)/2$ zero pixels out of each edge pixel of HSI before cropping 3D patch for each HSI pixel. For clear comparisons, **boldface** highlights the best results while underline the second in the following suitable table results.

1) Spatial Window Size: For patch-based HSIC models, spatial window size of the 3D patch influences how much the spatial neighbor information the patch contains, and a larger spatial window size also means that more complicated mixed pixels would be mined in feature extraction, which has an impact on the model performance. Therefore, a corresponding experiment is conducted to determine the most suitable spatial window size for three main data sets from a set of spatial window sizes, which is set from 3×3 to 17×17 with an interval of 2-pixel size, and the classification results evaluated by OA and AA are illustrated in Fig. 6 (a). The overall trends of classification accuracy are rising at first and then decreasing with the increase of the spatial window size, which is especially evident on the IP scene. When the spatial size is larger than 9×9, the accuracy curves show slow growth or even decrease. Considering that AA result is especially important for the class-imbalanced IP data set and larger spatial window size is accompanied by more computational cost. Therefore, the spatial window size is set as 9×9 for more competitive AA result on IP data set and more acceptable computational cost of the proposed model for all the considered data sets in all the following experiments.

2) Similarity Measures: In the proposed SVSS and FVSS modules, we mine central vector oriented spatial relationships by measuring the similarity between the central vector and its neighbor vectors in input space and high-level feature space. Here, a corresponding experiment is set to discuss the similarity measure combinations based on the discussed Euclidean distance similarity measure, cosine angle similarity measure, and adaptive weight addition based self-similarity measure, which are termed as 'ED','Cos', and 'E-C', respectively. Fig. 6 (b)-(d) illustrate the classification results of CVSSN adopting different similarity measure combinations in SVSS and FVSS modules on IP, KSC, and UP data sets. For instance, 'ED+ED' represents that both SVSS and FVSS modules use Euclidean distance similarity measure.

TABLE III
VITAL SPECTRAL CHANNEL VALUES IN THE SSIF MODULE FOR ALL THE FOUR DATA SETS

Spectral Channel	IP	KSC	UP	UH
b	200	176	103	144
$\hat{b}$	243	243	162	162
b'	203	179	105	146

TABLE IV
DETAILED STRUCTURES OF THE PROPOSED CVSSN

Module	Layer Setting		Input-Output Channel		Spatial Size	
	1D V	**3D P**	**1D V**	**3D P**	**1D V**	**3D P**
SSIF Module	MirrorPad		[b,$\hat{b}$]	[b, b]	[1]	[9, 9]
	Reshape		[$\hat{b}$, b'-b]	[b, b]	[9, 9]	[9, 9]
	Concatenate		[b'-b, b'-b]	[b, b]	[9, 9]	[9, 9]
	Fused 3D Patch		**Fused 3D Patch**		**Fused 3D Patch**	
AWA-SVSS Module	EDSim	CosSim	[b', b']		[9, 9]	
	Softmax	Softmax	[b', b']		[9, 9]	
	Adaptive Weight Addition		[b', b']		[9, 9]	
	Element-wise-multiplication		[b', b']		[9, 9]	
	Element-wise-addition		[b', b']		[9, 9]	
CSS-Conv Module	BN+1×1 Conv+LeakyReLU		[b', 128]		[9, 9]	
	1×1 Conv+ReLU		[128, 128]		[9, 9]	
	BN+1×1 Conv+LeakyReLU		[128, 128]		[9, 9]	
	3×3 Conv+ReLU		[128, 128]		[9, 9]	
ED-FVSS Module	1×1 Conv		[128, 128]		[9, 9]	
	EDSim+Softmax		[128, 128]		[9, 9]	
	Element-wise-multiplication		[128, 128]		[9, 9]	
	Element-wise-addition		[128, 128]		[9, 9]	
SIC-Conv Module	1 × 1 Conv	3 × 3 Conv	[128, 128]		[9, 9]	
	BN+LeakyReLU	BN+ReLU	[128, 128]		[9, 9]	
	Element-wise-addition		[128, 128]		[9, 9]	
	BN+ReLU		[128, 128]		[9, 9]	
Classifier	AvgPool		[128, 128]		[1, 1]	
	BN+Flatten		[128, 128]		-	
	Linear		[128, K]		-	

[1] The detailed spectral channels in the SSIF Module for four discussed data sets are illustrated in Table III in Section III-B2.
[2] For convenience, '1D V' and '3D P' represent the central 1D spectral vector and the 3D spectral-spatial patch in the SSIF module, respectively.

It is easy to find that the performance on the UP data set is relatively stable and vice versa on the IP and KSC scenarios with different similarity combinations, which is in that UP contains richer category samples while IP and KSC are smaller data sets, even the IP data set is class-imbalanced. Thus, it is harder to achieve promising classification accuracies in minor classes than in major classes for IP and KSC data sets. It is noteworthy that the combinations of 'ED' and 'E-C', i.e., 'ED+E-C' and 'E-C+ED', yield superior performance on the whole three data sets, especially on the IP data set. Finally, we choose the 'E-C+ED' combination to formulate AWA-SVSS and ED-FVSS modules considering its most competitive performance on all the three data sets, especially on the IP scene.

3) Feature Extraction Module Design: In the proposed CVSSN, we perform efficient spectral-spatial feature learning mainly through CSS-Conv and SIC-Conv modules. Based on our experimental observation, the PCG of CSS-Conv and the 1×1 convolutional scale branch of SIC-Conv are both based on 1×1 Conv to perform Conv operations across spectral channels and feature channels, which produces slightly more

TABLE V
CLASSIFICATION RESULTS OF DIFFERENT METHODS ON THE IP DATA SET USING 10% LABELED DATA FOR TRAINING

Class	RF [9]	SVM [8]	ContextNet [38]	RSSAN [34]	SSTN [19]	SSAN [33]	SSSAN [27]	SSAtt [35]	A^2S^2K-ResNe [23]	CVSSN
1	71.29±22.17	58.37±20.71	97.40±5.04	93.81±7.54	92.90±9.88	87.36±14.29	91.73±8.33	**97.48±5.36**	93.72±8.45	96.33±4.76
2	68.69±1.84	77.82±1.99	77.95±4.41	82.22±6.13	94.35±3.58	86.40±3.43	92.34±2.85	93.58±2.69	**98.22±1.25**	97.48±1.22
3	75.85±3.00	78.74±3.90	80.95±4.15	84.88±7.19	97.59±1.66	85.93±3.82	97.09±1.30	96.42±1.38	97.85±1.93	**99.17±0.54**
4	57.39±5.43	68.77±5.66	83.16±5.31	77.88±11.61	95.80±4.76	89.55±6.11	93.93±4.13	96.28±3.80	94.95±4.16	**98.43±3.15**
5	85.61±3.04	89.39±2.89	93.80±2.28	93.49±2.26	94.45±2.55	94.55±2.42	96.88±0.79	97.23±1.16	**98.27±1.48**	98.22±1.02
6	81.31±1.52	86.68±1.50	93.62±3.12	92.48±2.47	97.90±2.22	95.21±2.65	98.79±1.62	98.81±1.30	99.56±0.49	**99.58±0.92**
7	32.95±42.12	79.56±10.04	92.15±14.21	91.11±10.02	76.29±9.91	89.99±10.92	91.72±10.65	**97.73±4.07**	94.57±6.00	93.59±9.25
8	85.16±1.69	91.19±1.98	94.77±3.11	95.01±2.41	98.69±1.35	96.56±1.55	98.04±1.65	98.98±0.88	98.54±1.54	**99.91±0.15**
9	53.33±44.44	66.69±26.41	90.80±12.29	82.35±15.62	75.24±12.68	82.86±21.64	85.38±15.03	91.87±9.21	92.24±9.75	**99.41±1.76**
10	72.71±4.41	73.76±2.48	86.86±3.72	87.10±5.68	91.40±3.40	90.69±2.44	95.74±1.40	94.82±2.27	96.39±1.77	**98.05±0.89**
11	69.53±1.53	73.99±1.52	87.58±1.57	87.22±3.63	95.64±2.79	90.19±2.04	95.65±1.23	95.94±1.49	**98.18±0.67**	98.14±1.12
12	62.60±5.79	78.26±4.56	73.13±4.65	74.39±5.96	94.96±2.33	81.99±3.31	89.12±4.97	92.09±0.19	96.78±1.69	**97.26±0.85**
13	85.40±4.75	91.26±4.88	97.94±2.20	92.18±5.70	98.82±1.16	96.68±4.45	**99.89±0.22**	99.57±0.58	99.16±1.73	99.62±0.85
14	89.63±0.97	90.73±1.46	95.95±1.26	94.68±2.03	97.87±1.33	95.95±0.15	97.94±0.93	97.66±1.06	**98.73±0.78**	98.32±0.86
15	63.48±5.91	77.08±4.09	89.85±3.45	84.83±4.03	90.95±2.92	90.62±2.76	91.47±2.69	94.90±1.34	94.51±1.87	**95.60±1.72**
16	97.25±2.28	97.31±3.26	94.27±5.05	92.40±6.05	94.32±3.89	90.21±6.89	97.24±2.30	93.74±4.63	96.67±2.87	**97.63±2.08**
OA (%)	75.07±0.81	80.19±0.51	87.15±1.35	87.51±3.26	95.31±1.01	90.57±1.51	95.42±0.97	95.93±0.71	97.82±0.33	**98.18±0.27**
AA (%)	72.02±3.86	79.97±2.41	89.39±1.65	87.88±2.76	92.95±1.10	90.30±2.18	94.56±1.62	96.07±0.83	96.77±0.87	**97.92±0.75**
$\mathcal{K} \times 100$	71.22±0.92	77.22±0.61	85.29±1.55	85.71±3.74	94.65±1.15	89.22±1.74	94.78±1.11	95.35±0.82	97.51±0.37	**97.92±0.30**

TABLE VI
CLASSIFICATION RESULTS OF DIFFERENT METHODS ON THE KSC DATA SET USING 10% LABELED DATA FOR TRAINING

Class	RF [9]	SVM [8]	ContextNet [38]	RSSAN [34]	SSTN [19]	SSAN [33]	SSSAN [27]	SSAtt [35]	A^2S^2K-ResNe [23]	CVSSN
1	91.07±1.44	91.15±1.31	98.92±0.63	98.33±1.44	99.55±0.56	98.58±1.08	99.27±0.81	98.67±1.98	98.93±1.15	**99.78±0.52**
2	77.66±6.21	89.27±3.73	82.03±9.15	93.82±6.11	92.82±5.40	90.56±2.47	95.76±3.77	95.64±3.04	96.85±2.74	**98.01±1.54**
3	87.72±4.70	69.73±5.62	75.34±4.86	80.51±5.15	92.28±6.34	75.36±6.37	91.04±4.34	87.82±2.62	87.57±4.27	**97.61±1.72**
4	59.52±4.25	48.94±3.33	70.45±5.64	74.76±9.49	80.58±6.45	70.93±4.72	80.07±8.09	82.47±7.10	86.38±5.41	**91.96±6.29**
5	70.39±6.20	70.71±10.19	64.84±5.92	70.76±10.38	77.11±12.69	69.62±6.18	84.05±7.22	85.45±6.60	89.34±4.04	**96.24±3.81**
6	62.07±6.91	71.96±7.49	86.85±0.49	85.46±4.89	92.59±7.15	84.98±4.38	91.10±5.53	91.37±6.53	94.15±4.42	**96.07±2.94**
7	73.56±3.22	75.95±4.12	89.14±5.39	93.37±5.57	94.46±5.52	85.12±8.76	93.80±4.15	96.50±5.10	94.80±3.50	**99.17±1.56**
8	76.51±4.68	88.34±3.78	91.20±2.82	97.12±1.73	96.79±2.53	94.74±2.71	97.74±1.60	97.83±1.75	97.19±1.38	**99.56±0.63**
9	84.56±2.58	89.19±2.66	95.55±2.16	97.75±1.78	98.11±1.60	95.26±2.87	**99.87±0.20**	99.12±1.28	99.17±0.85	99.68±0.57
10	88.85±6.46	97.22±2.91	95.15±2.48	96.46±2.53	99.02±0.85	96.33±2.82	99.70±0.74	99.36±1.16	98.90±2.41	**100.00±0.00**
11	97.62±1.87	96.51±1.18	98.80±1.33	99.04±1.37	99.26±1.05	99.04±1.31	99.87±0.40	99.38±1.33	99.55±0.60	**100.00±0.00**
12	91.97±2.77	93.40±2.32	94.93±3.15	94.91±3.00	98.60±1.46	96.07±2.79	99.42±0.80	99.13±0.83	99.10±1.35	**99.63±0.61**
13	98.36±0.62	99.81±0.21	99.60±0.78	99.71±0.45	99.57±0.72	99.70±0.54	**100.00±0.00**	99.89±0.29	99.79±0.50	**100.00±0.00**
OA (%)	86.36±1.00	88.20±0.42	92.33±1.14	94.22±1.49	96.12±0.54	93.16±1.37	96.88±0.68	96.74±0.71	97.11±0.64	**98.90±0.30**
AA (%)	81.53±1.54	83.24±1.26	87.91±1.95	90.92±2.37	93.90±1.19	88.95±2.24	94.75±1.00	94.82±1.14	95.52±1.08	**98.29±0.45**
$\mathcal{K} \times 100$	84.79±1.11	86.84±0.47	91.45±1.27	93.57±1.66	95.68±0.60	92.39±1.52	96.53±0.76	96.37±0.79	96.79±0.71	**98.78±0.33**

negative feature values overall than 3×3 Conv. Furthermore, considering that LeakyReLU has a small slope to reserve small nonzero gradients for negative values to avoid the dying ReLU problem, we further explore the different LeakyReLU and ReLU combinations for 1×1 Conv and 3×3 Conv in CSS-Conv and SIC-Conv modules in Table II, where 'LR+R' means the combination of 1×1 Conv with LeakyReLU and 3×3 Conv with ReLU in both modules. It is obvious that 'LR+R' outperforms any other combinations by considerable performance boost, especially for AAs. Therefore, we finally combine 1×1 Conv with LeakyReLU and 3×3 Conv with ReLU both in CSS-Conv and SIC-Conv modules.

4) Structures of the Proposed CVSSN: In particular, Table III records the vital spectral channel values in the SSIF module for each data set, which are determined by the original spectral channel of different data sets and the spatial window size of the input 3D patch $\mathbf{P}_{i,j}$. In particular, $\hat{b}$ denotes the set padded vector length discussed in Section II-A and Fig. 3. The detailed structures of CVSSN are summarized in Table IV.

C. Comparison of Classification Performance

To fairly evaluate performance and efficiency of the proposed model, we comprehensively compare the proposed CVSSN with two classic machine learning methods, i.e., RF [9] and SVM with radial basis function (RBF) kernel [8], and seven representative state-of-the-art deep learning models, i.e., ContextNet [38][3], RSSAN [34][4], SSTN [19][5], SSAN [33], SSSAN [27], SSAtt [35][6], and A^2S^2K-ResNet [23][7], where SSAN and SSSAN are reproduced with PyTorch framework according the corresponding paper and source code. The selected compared methods comprehensively cover popular manners and patterns including classic machine learning manner, 2D CNN-based pattern, 3D CNN-based pattern, spectral-spatial dual-branch pattern, two-sub-network structure, residual learning manner, and various attention mechanisms. For fair comparisons, all the methods take the same experiment settings mentioned above.

1) Quantitative Accuracy Analysis: For the detailed quantitative accuracy analysis, Tables V-VII report the average classification results and their corresponding standard deviations (ten runs results) on IP, KSC, and UP data sets, respectively. Totally, CVSSN achieves the most outstanding performance compared with the others on all the three data sets. Specifically, our proposed CVSSN model outperforms the second best method A^2S^2K-ResNet by 0.36%, 1.15%, and 0.41% for OA, AA, and $\mathcal{K}$ on the IP data set, and 0.06%, 0.03%, and 0.07% on UP data set. More importantly, compared with A^2S^2K-ResNet model on the KSC data set, our model yields 1.79%, 2.77%, and 1.99% improvements on OA, AA,

[3]https://github.com/eecn/Hyperspectral-Classification
[4]https://github.com/liereranniu/RSSAN-Hyperspectral-Image
[5]https://github.com/zilongzhong/SSTN
[6]https://github.com/weecology/DeepTreeAttention
[7]https://github.com/suvojit-0x55aa/A2S2K-ResNet

TABLE VII
CLASSIFICATION RESULTS OF DIFFERENT METHODS ON THE UP DATA SET USING 5% LABELED DATA FOR TRAINING

Class	RF [9]	SVM [8]	ContextNet [38]	RSSAN [34]	SSTN [19]	SSAN [33]	SSSAN [27]	SSAtt [35]	A^2S^2K-ResNe [23]	CVSSN
1	89.84±1.34	88.60±1.72	96.96±0.74	96.07±0.86	98.03±1.36	97.22±0.92	99.07±0.48	97.84±0.76	99.53±0.38	**99.54±0.21**
2	88.00±0.42	91.39±0.38	99.22±0.26	99.55±0.12	98.87±2.57	99.62±0.20	99.77±0.11	99.76±0.14	99.90±0.08	**99.92±0.05**
3	73.76±2.27	82.95±1.37	92.74±2.20	93.97±3.02	89.55±17.07	95.67±1.72	98.43±1.07	96.84±1.61	98.99±0.76	**99.59±0.33**
4	93.33±1.01	96.61±0.44	99.33±0.49	99.05±0.68	96.38±3.32	**99.54±0.21**	99.53±0.34	99.42±0.54	99.48±0.41	99.43±0.40
5	97.15±1.39	99.08±0.61	99.70±0.41	98.18±1.35	99.59±0.23	99.62±0.59	99.68±0.60	99.31±1.38	**99.80±0.17**	99.63±0.33
6	87.48±1.42	94.98±0.57	98.40±0.58	98.49±0.49	96.92±8.33	98.91±0.62	99.39±0.30	98.78±0.49	99.75±0.20	**99.83±0.08**
7	81.76±2.99	87.91±4.94	95.96±1.67	95.13±1.53	99.19±1.62	96.26±2.28	99.43±0.44	97.16±2.37	99.66±0.69	**99.74±0.23**
8	77.60±2.11	79.24±2.56	93.40±1.14	92.43±0.83	96.85±1.04	94.29±1.41	97.73±0.76	95.70±1.17	98.60±0.54	**98.88±0.56**
9	99.88±0.09	**99.96±0.05**	99.57±0.44	97.90±2.25	96.07±1.60	99.44±0.70	99.81±0.29	99.86±0.22	99.70±0.56	99.08±0.86
OA (%)	87.39±0.24	90.42±0.41	97.88±0.19	97.74±0.28	97.15±3.46	98.39±0.27	99.34±0.15	98.73±0.34	99.62±0.10	**99.68±0.06**
AA (%)	87.64±0.52	91.19±0.57	97.25±0.36	96.75±0.60	96.83±2.91	97.84±0.45	99.20±0.22	98.30±0.54	99.49±0.22	**99.52±0.17**
$\mathcal{K} \times 100$	82.93±0.32	87.08±0.57	97.18±0.26	97.00±0.37	96.22±4.59	97.87±0.36	99.13±0.20	98.32±0.45	99.50±0.13	**99.57±0.09**

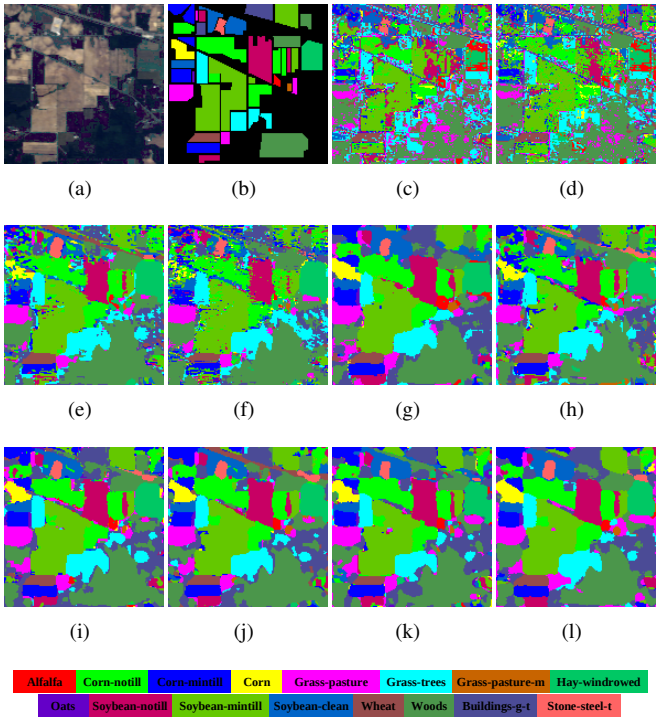

Fig. 7. Classification maps for the IP data set. (a) Three-band false-color composite image. (b) Ground-truth map. (c) RF. (d) SVM. (e) ContextNet. (f) RSSAN. (g) SSTN. (h) SSAN. (i) SSSAN. (j) SSAtt. (k) A^2S^2K-ResNet. (l) CVSSN.

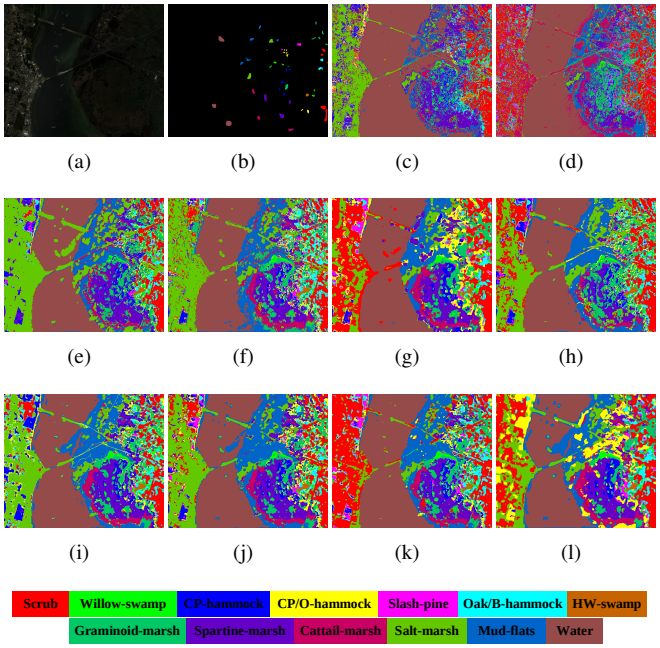

Fig. 8. Classification maps for the KSC data set. (a) Three-band false-color composite image. (b) Ground-truth map. (c) RF. (d) SVM. (e) ContextNet. (f) RSSAN. (g) SSTN. (h) SSAN. (i) SSSAN. (j) SSAtt. (k) A^2S^2K-ResNet. (l) CVSSN.

and $\mathcal{K}$, which shows the remarkable performance obtained by the proposed CVSSN on this KSC scenario. For the accuracy of each class on the three data sets, the proposed model achieves the best categorical accuracy on 9 classes of total 16 classes on the IP data set, 12 classes of total 13 classes on the KSC data set, and 6 classes of total 9 classes on the UP data set.

As for classic methods, RF and SVM only exploit spectral information and present limited performances on all the three scenes. In terms of deep learning methods, ContextNet, RSSAN, and SSAN achieve similar enhancements compared to the two traditional methods for extracting and refining spectral-spatial features. Then SSTN, SSSAN, and SSAtt further improve the classification results by the global spatial correlation mining of transformer, the well-designed spatial and spectral attention dense blocks, and the original two-branch spectral-spatial attention structure, respectively. Furthermore, only A^2S^2K-ResNet achieves the classification results close to those of our model. But as mentioned above, the proposed CVSSN still outperforms A^2S^2K-ResNet with different performance improvements on all the three data sets. Additionally, the training time consumed by A^2S^2K-ResNet is much more than that of our model. Thus, the classification compared results implicitly indicate that it is worthwhile and vital to mine central vector oriented spatial relationships in input space and high-level feature space.

2) Qualitative Accuracy Analysis: For qualitative evaluation, classification maps produced by corresponding methods on IP, KSC, and UP scenes are visualized from Figs. 7-9. Additionally, the three-band false-color composite image and the ground-truth map of each data set are also exhibited for intuitive comparison. In general, the visualizations of different methods on different data sets are in line with the corresponding statistical results recorded in Tables V-VII.

To be specific, for the difficult class-imbalanced IP data set, there are many salt-and-pepper noises on the classification maps of RF and SVM due to the absence of spatial feature extraction. Similar noise-filled maps are obtained from

TABLE VIII
COMPARISONS ON MODEL EFFICIENCY OF DIFFERENT METHODS ON IP, KSC, AND UP DATA SETS

Data Set	Metric	RF [9]	SVM [8]	ContextNet [38]	RSSAN [34]	SSTN [19]	SSAN [33]	SSSAN [27]	SSAtt [35]	A^2S^2K-ResNe [23]	CVSSN
IP	$T_{\text{train}}(\text{s})$	0.74	0.05	46.44	39.31	48.19	385.65	304.46	50.95	792.74	**25.42**
	$T_{\text{test}}(\text{ms})$	117.89	960.54	2.45	2.35	2.98	2.67	19.22	3.43	2.78	**2.21**
	Params (M)	-	-	1.211	0.148	**0.020**	87.020	0.314	0.596	0.371	0.261
	FLOPs (M)	-	-	84.79	7.87	**1.64**	7062.96	20.68	15.04	170.45	21.03
KSC	$T_{\text{train}}(s)$	0.28	0.01	23.51	20.73	24.90	163.19	162.07	28.42	383.31	**14.96**
	$T_{\text{test}}(\text{ms})$	52.97	143.54	**1.78**	2.36	2.84	2.56	17.38	3.48	2.81	2.01
	Params (M)	-	-	1.072	0.131	**0.019**	68.093	0.304	0.579	0.333	0.247
	FLOPs (M)	-	-	76.05	7.29	**1.54**	5515.38	18.921	13.92	149.66	20.00
UP	$T_{\text{train}}(\text{s})$	1.21	0.06	89.73	79.46	92.43	268.05	634.38	108.24	1327.58	**53.65**
	$T_{\text{test}}(\text{ms})$	417.66	2682.61	**1.77**	2.28	2.91	2.63	18.00	3.42	2.82	1.85
	Params (M)	-	-	0.703	0.095	**0.016**	25.024	0.275	0.532	0.221	0.253
	FLOPs (M)	-	-	49.52	5.56	**1.25**	1982.81	13.12	10.51	87.30	20.43

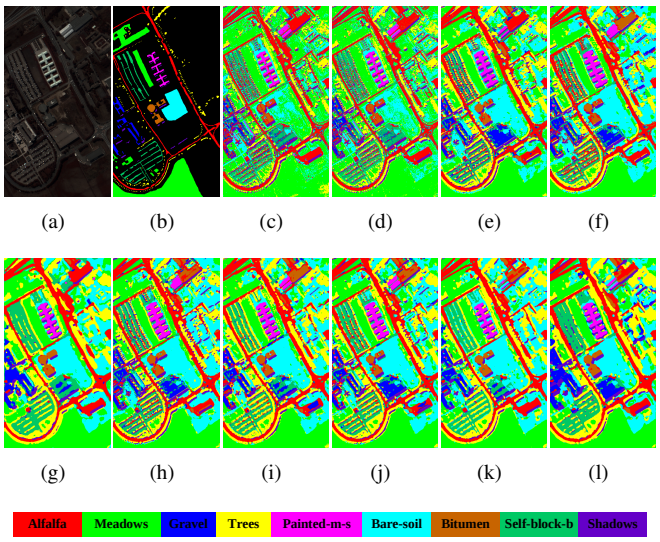

Fig. 9. Classification maps for the UP data set. (a) Three-band false-color composite image. (b) Ground-truth map. (c) RF. (d) SVM. (e) ContextNet. (f) RSSAN. (g) SSTN. (h) SSAN. (i) SSSAN. (j) SSAtt. (k) A^2S^2K-ResNet. (l) CVSSN.

ContextNet, RSSAN, and SSAN for the insufficient feature extraction, and SSTN, SSSAN, SSAtt, A^2S^2K-ResNet, and our model exhibit smoother and clearer visualization results. The pixel identification results produced by CVSSN are more consistent with corresponding ground-truth maps and composite images of three data sets. The edge pixels and mixer pixels on class boundaries are better classified by CVSSN benefiting from the sufficient mining of central vector oriented spatial relationships.

3) Efficiency Analysis: To compare different methods from model efficiency aspect, running time, model complexity, and computational cost of different methods are also recorded in Table VIII. In terms of running time, the classic methods, RF and SVM usually need more time in model testing stage compared with model training stage, while deep learning models show the opposite time-consuming pattern. It is noteworthy that the proposed CVSSN always runs the fastest in model training process among all the considered deep learning models. Although the test time of all deep models is on millisecond (ms) magnitude, the proposed model is also superior on all the three HSI data sets. It obtains the first rank on the IP scene and the second place on both KSC and UP data sets, which is slightly weaker than ContextNet. As the closest method to the classification performance of our model, A^2S^2K-ResNet always requires the most time for model training, which is at least 25 times than that consumed by our proposed CVSSN. In addition, SSAN and SSSAN are also accompanied by larger time consumption due to the total 3D Conv-based and the dense-block-based model structures, respectively.

From Table VIII, the parameters and FLOPs of SSTN always are the least owing to the efficient self-attention model structure. In contrast, SSAN shows the largest demand for parameters memory and the highest computational cost for model complexity on all the three data sets due to the 3D Conv-based feature extraction module. In particular, A^2S^2K-ResNet is also accompanied by the second largest computational cost according to its FLOPs records, which is also consistent with its huge training time consumption. For comparison, the proposed CVSSN shows promising model complexity and low computational cost with 0.261M, 0.247M, and 0.253M in Params and 21.03M, 20.00M, and 20.43M in FLOPs on IP, KSC, and UP data sets, respectively, which are only behind those of SSTN and RSSAN.

4) Summary: To sum up, our proposed CVSSN achieves remarkable HSI classification performance compared with the other nine methods in the following three aspects: 1) the best quantitative accuracy in OA, AA, and $\mathcal{K}$ on all the three data sets; 2) the most realistic visual classification maps on global class maps and local class boundaries; 3) the excellent model efficiency with the outstanding running time, the promising model complexity, and the low computational cost. Generally, the superiority of CVSSN demonstrates the significance and importance of mining the central vector oriented spatial relationships and performing efficient spectral-spatial feature learning.

D. Comparison with Popular Attention Modules

In order to evaluate the effects and superiorities of the proposed AWA-SVSS and ED-FVSS modules, we separately replace one of the two proposed modules with popular spatial attention modules, including the spatial attention module (SAM) of CBAM [32], the position attention module (PAM) of DAN [41], the spatial attention module (SAM) of SSAtt [35], the non-local block (NLB) [30] used in SSAN [33], the spatial self-attention module (SSAM) proposed from SSSAN [27] and

TABLE IX
CLASSIFICATION RESULTS OF DIFFERENT METHODS ON THE DISJOINT UH DATA SET USING OFFICIAL FIXED LABELED SAMPLES AS TRAINING SET

Metric	RF [9]	SVM [8]	ContextNet [38]	RSSAN [34]	SSTN [19]	SSAN [33]	SSSAN [27]	SSAtt [35]	A^2S^2K-ResNe [23]	CVSSN
OA (%)	72.90±0.03	76.75±0.02	74.08±1.08	75.33±1.37	80.81±0.93	77.40±1.08	81.19±0.78	79.44±0.58	**82.76±1.34**	82.55±0.47
AA (%)	75.53±0.02	78.49±0.02	76.24±1.31	77.83±1.79	83.04±1.38	79.90±1.33	83.61±1.88	82.08±0.75	**86.65±1.44**	85.64±0.98
$\mathcal{K} \times 100$	70.89±0.03	74.97±0.02	71.98±1.17	73.35±1.47	79.30±1.00	75.56±1.18	79.69±0.83	77.78±0.62	**81.38±1.41**	81.15±0.50
$T_{\mathrm{train}}(\mathrm{s})$	1.97	0.11	120.08	102.20	130.70	631.06	811.87	148.73	1930.65	**71.82**
$T_{\mathrm{test}}(\mathrm{ms})$	150.00	1743.77	**1.87**	3.16	3.09	2.82	18.08	3.94	2.76	1.93

TABLE X
MODULE COMPARISONS ON IP AND KSC DATA SETS FOR THE CENTRAL VECTOR ORIENTED SELF-SIMILARITY MODULES

Data Set	AWA-SVSS	ED-FVSS	OA (%)	AA (%)	$\mathcal{K} \times 100$
IP	✓	✓	**98.18±0.27**	**97.92±0.75**	97.92±0.30
	CBAM-SAM [32]	✓	98.09±0.23	97.76±0.64	97.82±0.26
	DAN-PAM [41]	✓	98.03±0.37	97.16±0.90	97.76±0.42
	STAtt-SAM [35]	✓	98.12±0.19	97.24±0.81	97.86±0.22
	SSAN-NLB [30]	✓	97.95±0.22	97.04±1.07	97.66±0.25
	SSSAN-SSAM [27]	✓	98.11±0.29	97.66±0.90	**97.93±0.32**
	SGEM [50]	✓	98.13±0.24	97.26±0.87	97.86±0.27
	✓	CBAM-SAM [32]	98.18±0.28	97.29±1.19	97.93±0.32
	✓	DAN-PAM [41]	97.73±0.18	96.81±1.38	97.41±0.20
	✓	STAtt-SAM [35]	97.91±0.37	96.84±1.67	97.62±0.42
	✓	SSAN-NLB [30]	97.92±0.30	97.17±0.88	97.62±0.34
	✓	SSSAN-SSAM [27]	98.13±0.22	97.30±1.02	97.86±0.25
	✓	SGEM [50]	98.12±0.27	97.23±1.04	97.86±0.31
KSC	✓	✓	**98.90±0.30**	**98.29±0.45**	**98.78±0.33**
	CBAM-SAM [32]	✓	98.75±0.32	98.00±0.50	98.61±0.36
	DAN-PAM [41]	✓	98.29±0.42	97.26±0.73	98.10±0.47
	STAtt-SAM [35]	✓	98.39±0.55	97.79±0.79	98.20±0.61
	SSAN-NLB [30]	✓	98.26±0.37	97.31±0.66	98.06±0.41
	SSSAN-SSAM [27]	✓	98.87±0.40	98.13±0.67	98.74±0.45
	SGEM [50]	✓	98.78±0.37	98.05±0.55	98.64±0.41
	✓	CBAM-SAM [32]	98.57±0.40	97.76±0.63	98.40±0.44
	✓	DAN-PAM [41]	97.72±0.65	96.05±1.31	97.47±0.72
	✓	STAtt-SAM [35]	98.40±0.49	97.39±0.81	98.22±0.54
	✓	SSAN-NLB [30]	98.27±0.42	97.15±0.73	98.07±0.46
	✓	SSSAN-SSAM [27]	98.62±0.38	97.79±0.64	98.46±0.42
	✓	SGEM [50]	98.75±0.33	98.09±0.48	98.61±0.37

TABLE XI
ABLATION STUDY ON IP AND KSC DATA SETS

Data Set	SSIF	SVSS	FVSS	CSS	SIC	OA (%)	AA (%)	$\mathcal{K} \times 100$
IP	✓	✓	✓	✓	✓	**98.18±0.27**	**97.92±0.75**	**97.92±0.30**
	✗	✓	✓	✓	✓	98.12±0.29	97.59±0.59	97.85±0.33
	✓	✗	✓	✓	✓	98.14±0.26	97.61±0.52	97.88±0.30
	✓	✓	✗	✓	✓	98.09±0.34	97.48±0.69	97.82±0.39
	✗	✗	✗	✓	✓	98.13±0.24	97.29±0.64	97.87±0.27
	✓	✓	✓	✗	✓	97.43±0.62	97.02±0.92	97.06±0.71
	✓	✓	✓	✓	✗	97.82±0.27	96.62±1.46	97.51±0.30
	✓	✓	✓	✗	✗	86.49±0.83	90.13±1.52	84.53±0.99
KSC	✓	✓	✓	✓	✓	**98.90±0.30**	**98.29±0.45**	**98.78±0.33**
	✗	✓	✓	✓	✓	98.90±0.45	98.26±0.53	98.78±0.53
	✓	✗	✓	✓	✓	98.76±0.37	98.07±0.52	98.62±0.41
	✓	✓	✗	✓	✓	98.73±0.52	98.01±0.77	98.58±0.57
	✗	✗	✗	✓	✓	98.72±0.41	97.90±0.69	98.57±0.46
	✓	✓	✓	✗	✓	98.25±0.29	97.00±0.46	98.05±0.32
	✓	✓	✓	✓	✗	98.52±0.26	97.72±0.46	98.35±0.29
	✓	✓	✓	✗	✗	94.60±0.77	91.03±0.88	93.99±0.86

spatial group-wise enhance module (SGEM) [50]. As summarized in Table IX, both AWA-SVSS and ED-FVSS modules present obvious performance improvements compared with various spatial attention modules. Different spatial attention modules embedded into CVSSN influence the entire efficiency of the corresponding model to different degrees. In particular, SAN proposed from SSSAN and SGEM also yield relative competitive results for capturing spatial feature correlations based on cosine angle similarity and exploring the spatial similarity based on global-local statistical feature, respectively. In fact, when AWA-SVSS module is replaced with SSAN of SSSAN, it is an attempt for SSAN of SSSAN to explore the spatial similarity in original input space by mining the feature similarity representation.

E. Ablation Study

To further explore and validate the contributions of different modules of the proposed CVSSN model, ablation experiments are conducted on two more difficult data sets, IP and KSC. In detail, we conduct CVSSN without a certain module to explore the effect of the corresponding module from the five modules of the spectral-spatial feature extraction part. Besides, the combinations, i.e., 'SVSS+FVSS+SSIF' and 'CSS+SIC', are also validated, respectively.

As recorded in Table XI, with the single absence of the AWA-SVSS module, the ED-FVSS module, and the SSIF module, there are varying degrees of classification performance degradation, especially on AAs. Considering that AA is the mean of the accuracies of all the land-cover categories representing the overall discriminative ability of the model on each category, we can infer that the contributions of the AWA-SVSS module, the ED-FVSS module, and the SSIF module mainly focus on enhancing the fine-grained feature extraction capability based on the CSS-Conv and SIC-Conv modules. When considering the 'SVSS+FVSS+SSIF' modules combination, the corresponding results illustrate distinct decrease on different data sets compared with those of the complete CVSSN model, which further validates the contributions of the three proposed modules. Furthermore, CSS-Conv and SIC-Conv modules demonstrate their irreplaceable roles in efficient spectral-spatial feature extraction. When CVSSN is without CSS-Conv and SIC-Conv modules, the other modules would not play their expected functions owing to insufficient spectral-spatial feature learning, which results in cliff-like descents on both data sets.

F. Comparison Results on the Disjoint UH Data Set

In this subsection, the spatial disjoint UH data set is employed for comprehensive performance comparison. As reported in Table X, the classification performances of most discussed methods are in line with the results on above-discussed data sets. Specifically, the considered two classic machine learning methods exhibit robustness for concentrated exploiting of spectral signatures. A^2S^2K-ResNet and the proposed CVSSN achieve better performances than any other discussed methods, and A^2S^2K-ResNet slightly outperforms CVSSN on three metrics, especially on AAs. However, time consumption is still a serious problem for A^2S^2K-ResNet.

G. Impact of Training Sample Proportion

To further explore the robustness of all the discussed ten different methods, we also investigate the classification

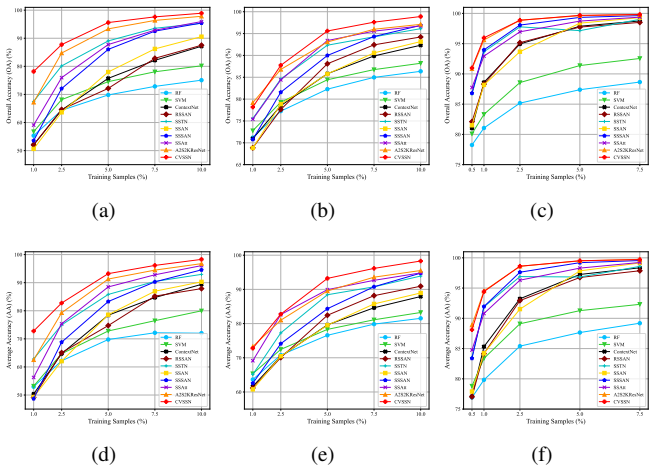

Fig. 10. OAs and AAs of different methods on IP, KSC, and UP data sets with varying amounts of training samples. (a) OAs on IP data set. (b) OAs on KSC data set. (c) OAs on UP data set. (d) AAs on IP data set. (e) AAs on KSC data set. (f) AAs on UP data set.

performances of different methods with different proportions of labeled samples for model training on all the three data sets. As illustrated in Fig. 10, the overall trend of the curve for each method is rising with the percentage of training samples increasing. In detail, RF and SVM still show limited performances with different amounts of training samples on all the three data sets. ContextNet, RSSAN, and SSAN exhibit close growing trends and classification results, which are ahead of the two classic methods. Furthermore, SSTN, SSSAN, and SSAtt achieve similar competitive performances compared with most part of discussed methods. In particular, SSSAN shows outstanding performance under all the training sample proportions on the UP data set. Though A^2S^2K-ResNet achieves promising classification results in most cases, CVSSN always leads the top rank with obvious improvements, especially clear enhancements than any other discussed methods on the KSC scene.

IV. Conclusion

In HSIC task, most patch-based CNNs ignore the latent relationships between the central vector and its neighbor vectors in original input space and high-level feature space. Moreover, how to perform efficient spectral-spatial feature learning is a difficult yet vital topic. In this article, a CVSSN is proposed for HSIC. Specifically, the AWA-SVSS module and the ED-FVSS module are firstly designed for mining the central vector oriented self-similarity spatial relationship in original input space and high-level feature space, respectively. Besides, the SSIF module as a new pattern fuses the central 1D spectral vector and the corresponding 3D spectral-spatial patch for efficient spectral-spatial feature learning of the subsequent modules. Moreover, both CSS-Conv module and SIC-Conv module play vital roles in extracting discriminative spectral-spatial feature and maintaining model efficiency. Experimental results and analysis illustrate that the proposed CVSSN model achieves excellent performance and outstanding efficiency.

One focus of future work will be to pursue more interpretable and efficient mechanisms to mine spectral-spatial information for HSIC. In addition, we will optimize the model structure and pay more attention to spectral domain to improve the robustness of the proposed model to different types of HSI data scenarios.

Acknowledgments

The authors would like to thank the Hyperspectral Image Analysis group and the NSF Funded Center for Airborne Laser Mapping (NCALM) at the University of Houston for providing the UH data set used in this study, and the IEEE GRSS Data Fusion Technical Committee for organizing the 2013 Data Fusion Contest. The authors also would like to thank Assistant Professor Xiangtao Zheng and Dr. Xuming Zhang for providing the source tensorflow code of SSAN [33] and the part of source keras code of SSSAN [27], respectively.

References

[1] G. Vane, R. O. Green, T. G. Chrien, H. T. Enmark, E. G. Hansen, and W. M. Porter, "The airborne visible/infrared imaging spectrometer (aviris)," *Remote Sens. Environ.*, vol. 44, no. 2-3, pp. 127–143, 1993.

[2] S. Zhang, H. Huang, and Y. Fu, "Fast parallel implementation of dual-camera compressive hyperspectral imaging system," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 11, pp. 3404–3414, 2018.

[3] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 652–675, 2012.

[4] R. Ribeiro, G. Cruz, J. Matos, and A. Bernardino, "A data set for airborne maritime surveillance environments," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 9, pp. 2720–2732, 2017.

[5] J. Lei, X. Li, B. Peng, L. Fang, N. Ling, and Q. Huang, "Deep spatial-spectral subspace clustering for hyperspectral image," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 7, pp. 2686–2697, 2020.

[6] A. F. Goetz, G. Vane, J. E. Solomon, and B. N. Rock, "Imaging spectrometry for earth remote sensing," *Science*, vol. 228, no. 4704, pp. 1147–1153, 1985.

[7] X. Kang, X. Xiang, S. Li, and J. A. Benediktsson, "Pca-based edge-preserving features for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7140–7151, 2017.

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

[9] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 492–501, 2005.

[10] P. Ghamisi, N. Yokoya, J. Li, W. Liao, S. Liu, J. Plaza, B. Rasti, and A. Plaza, "Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 37–78, 2017.

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

[12] J. Fan, T. Chen, and S. Lu, "Superpixel guided deep-sparse-representation learning for hyperspectral image classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 11, pp. 3163–3173, 2017.

[13] L. Sun, C. Ma, Y. Chen, Y. Zheng, H. J. Shim, Z. Wu, and B. Jeon, "Low rank component induced spatial-spectral kernel method for hyperspectral image classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 10, pp. 3829–3842, 2019.

[14] H. Liu, Y. Jia, J. Hou, and Q. Zhang, "Global-local balanced low-rank approximation of hyperspectral images for classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 4, pp. 2013–2024, 2021.

[15] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2094–2107, 2014.

[16] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, 2017.

[17] S. Wan, C. Gong, P. Zhong, S. Pan, G. Li, and J. Yang, "Hyperspectral image classification with context-aware dynamic graph convolutional network," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 597–612, 2020.

[18] Q. Liu, L. Xiao, J. Yang, and Z. Wei, "Cnn-enhanced graph convolutional network with pixel-and superpixel-level feature fusion for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 10, pp. 8657–8671, 2020.

[19] 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.*, 2021.

[20] J. Xie, N. He, L. Fang, and P. Ghamisi, "Multiscale densely-connected fusion networks for hyperspectral images classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 1, pp. 246–259, 2020.

[21] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, 2016.

[22] 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, 2017.

[23] 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.

[24] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Eca-net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, June 2020.

[25] S. Hao, W. Wang, Y. Ye, T. Nie, and L. Bruzzone, "Two-stream deep architecture for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2349–2361, 2017.

[26] 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, 2020.

[27] 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.

[28] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3213–3223.

[29] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

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

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

[32] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 3–19.

[33] 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, 2019.

[34] 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, 2020.

[35] R. Hang, Z. Li, Q. Liu, P. Ghamisi, and S. S. Bhattacharyya, "Hyperspectral image classification with attention-aided cnns," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 2281–2293, 2020.

[36] X. Mei, E. Pan, Y. Ma, X. Dai, J. Huang, F. Fan, Q. Du, H. Zheng, and J. Ma, "Spectral-spatial attention networks for hyperspectral image classification," *Remote Sensing*, vol. 11, no. 8, p. 963, 2019.

[37] T. Cohen and M. Welling, "Group equivariant convolutional networks," in *International Conference on Machine Learning*, 2016, pp. 2990–2999.

[38] Lee, Hyungtae and Kwon, Heesung, "Going deeper with contextual cnn for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4843–4855, 2017.

[39] H. Sun, X. Zheng, and X. Lu, "A supervised segmentation network for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 30, pp. 2810–2825, 2021.

[40] W. Zhou, S.-i. Kamata, Z. Luo, and H. Wang, "Multiscanning strategy-based recurrent neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, 2021.

[41] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3146–3154.

[42] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1251–1258.

[43] Y. Cui, J. Xia, Z. Wang, S. Gao, and L. Wang, "Lightweight spectral–spatial attention network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2021.

[44] Y. Dong, Q. Liu, B. Du, and L. Zhang, "Weighted feature fusion of convolutional neural network and graph attention network for hyperspectral image classification," *IEEE Trans. Image Process.*, 2022.

[45] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[46] A. L. Maas, A. Y. Hannun, A. Y. Ng *et al.*, "Rectifier nonlinearities improve neural network acoustic models," in *International Conference on Machine Learning*, vol. 30, no. 1. Citeseer, 2013, p. 3.

[47] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *International Conference on Machine Learning*, 2010, pp. 807–814.

[48] B. Kunkel, F. Blechinger, R. Lutz, R. Doerffer, and H. van der Piepen, "Rosis (reflective optics system imaging spectrometer)-a candidate instrument for polar platform missions," *Optoelectronic Technologies for Remote Sensing from Space*, vol. 868, pp. 134–141, 1988.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[50] X. Li, X. Hu, and J. Yang, "Spatial group-wise enhance: Improving semantic feature learning in convolutional networks," *arXiv preprint arXiv:1905.09646*, 2019.

Mingsong Li received the B.Eng. degree in software engineering from Shandong University, Jinan, China, in 2021, where he is currently pursuing the M.S. degree in artificial intelligence.

His research interests include hyperspectral image analysis, machine learning, and computer vision.

Yikun Liu received the B.S. degree from Huazhong Agriculture University, Wuhan, China, in 2019, and the M.S. degree from Shandong University, Jinan, China, in 2022. Currently, he is pursuing the Ph.D. degree at Shandong University.

His research interests include remote sensing and machine learning.

Guangkuo Xue recieved the B.S. degree from Beijing Information Science and Technology University, Beijing, China, in 2020. He is currently pursuing the master's degree with the School of Software, Shandong University, Jinan, China.

His research interests include remote sensing, computer vision, and deep learning.

Yuwen Huang received the Ph.D. degree in computer science and technology from Shandong University, Jinan, China, in 2021. He is an associate professor in the School of Computer, Heze university.

His research interests include biometrics and machine learning.

Gongping Yang received his Bachelor degree in Computer and Application, Master and Ph.D. degrees in Computer Software and Theory from Shandong University, Jinan, China, in 1992, 2001, and 2007, respectively. Since 2013, he has been a Professor in School of Software at Shandong University, Jinan, China. At present, he is a Senior member of CCF and CAAI, also serve as the Machine Learning Technical Committee of CAAI and Artificial Intelligence and Pattern Recognition Technical Committee of CCF. During the past several years, he has served as a program committee member/organization/program chair of several conferences.

His current research interests include pattern recognition, computer vision, and biometric recognition.