DCTN: Dual-Branch Convolutional Transformer Network With Efficient Interactive Self-Attention for Hyperspectral Image Classification

Yunfei Zhou, Xiaohui Huang, *Member, IEEE*, Xiaofei Yang, Jiangtao Peng, *Senior Member, IEEE*, and Yifang Ban, *Senior Member, IEEE*

***Abstract*— Hyperspectral image (HSI) classification is an essential task in remote sensing with substantial practical significance. However, most existing convolutional neural network (CNN)-based classification methods focus only on local spatial features while neglecting global spectral dependencies. Meanwhile, Transformer-based methods exhibit robust capabilities for global spectral feature modeling but struggle to extract local spatial features effectively. To fully exploit the local spatial feature extraction capabilities of CNN-based networks and the global spectral feature extraction capabilities of Transformer-based networks, this article proposes a dual-branch convolutional Transformer method with efficient interactive self-attention (EISA) for HSI classification, namely the dual-branch convolutional transformer network (DCTN), which can aggregate local and global spatial-spectral features fully. Specifically, DCTN includes two core modules: the spatial-spectral fusion projection module (SFPM) and the EISA module. The former utilizes 3-D convolution with adaptive pooling and 2-D group convolution with residual connection to parallel extract fused and grouped spatial-spectral features, respectively. The latter performs EISA across height, width, and spectral dimensions, enabling deep fusion of spatial-spectral features. Extensive experiments on three real HSI datasets demonstrate that the proposed DCTN method outperforms existing classification methods, yielding state-of-the-art classification performance. The code is available at https://github.com/AllFever/DeepHyperX-DCTN for reproducibility.**

***Index Terms*— Convolution neural networks (CNNs), hyperspectral image (HIS) classification, transformer, self-attention mechanism.**

I. Introduction

Hyperspectral images (HSIs) are high-resolution digital images with hundreds or thousands of continuous spectral bands, providing rich spectral information [1]. By capturing reflection information across different spectral bands, HSIs are widely applied in fields such as agriculture [2], environmental monitoring [3], geological exploration [4], and natural resource management [5]. HSI classification, which involves accurately identifying different types of ground objects in HSIs, is a crucial task in remote sensing image processing [6]. Given the extensive spectral coverage in HSIs, they offer a rich and intricate feature space. However, the limited labeled samples for training pose a challenge for discerning diverse land cover types from these high-dimensional data. Resolving this issue requires developing robust and efficient techniques that can leverage available samples and extract discriminative features for precise land cover classification.

With the advancement of deep learning, HSI classification tasks increasingly rely on deep learning techniques. Deep learning methods have demonstrated remarkable outcomes compared to traditional machine learning methods due to their capability of extracting deeper features. Hu et al. [7] proposed a 1D-CNN network architecture to extract deep spectral features for HSI classification. Zhao and Du [8] leveraged 2D-CNN to extract spatial features from HSIs to achieve deep extraction. Song et al. [9] integrated residual connections and feature fusion into multilayer 2D-CNN networks for extracting spatial and spectral features at various levels. He et al. [10] introduced 3D-CNN for HSI classification tasks and proposed a multiscale 3D-CNN network for extracting fused spatial-spectral feature information. Zheng et al. [11] proposed a method combining 2D-CNN and 3D-CNN to extract spatial and spectral features using mixed convolution and covariance difference pooling layers.

Various other network architectures have been utilized for HSI classification tasks. For instance, Chen et al. [12] utilized stacked autoencoders (SAEs) and proposed the Joint Spectral-Spatial Classification Framework to extract spatial-spectral features through layer-by-layer training. In addition, a new classification framework based on deep belief networks (DBNs) was proposed by Chen et al. [13]. Moreover, Mou et al. [14] employed recurrent neural networks

Manuscript received 28 October 2023; revised 14 December 2023; accepted 24 January 2024. Date of publication 8 February 2024; date of current version 16 February 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62062033, Grant 62301174, and Grant 42171351; in part by the Natural Science Foundation of Jiangxi Province under Grant 20232BAB202018; and in part by the Graduate Innovative Special Fund Projects of Jiangxi Province under Grant YC2023-S530. *(Corresponding authors: Xiaohui Huang; Xiaofei Yang.)*

Yunfei Zhou and Xiaohui Huang are with the School of Information Engineering, East China Jiaotong University, Nanchang 330013, China (e-mail: 2022068085404024@ecjtu.edu.cn; hxh016@hotmail.com).

Xiaofei Yang is with the School of Electronic and Communication Engineering, Guangzhou University, Guangzhou 511370, China (e-mail: xiaofeiyang@gzhu.edu.cn).

Jiangtao Peng is with the Hubei Key Laboratory of Applied Mathematics, Faculty of Mathematics and Statistics, Hubei University, Wuhan 430062, China (e-mail: pengjt1982@hubu.edu.cn).

Yifang Ban is with the Division of Geoinformatics, School of Architecture and the Built Environment, KTH Royal Institute of Technology, 11428 Stockholm, Sweden (e-mail: yifang@kth.se).

Digital Object Identifier 10.1109/TGRS.2024.3364143

(RNNs) for HSI classification to extract rich spectral features and analyze hyperspectral sequence data better. Zhu et al. [15] introduced adversarial generation networks (GANs) into HSI classification tasks due to the sparse training samples from limited HSI datasets and presented a new classification method, 3D-GAN, combining the generated adversarial samples and actual training samples to enhance the model's generalization ability. Moreover, researchers have successfully employed graph convolutional neural networks (GCNs) to achieve promising results in HSI classification tasks [16].

Although the methods mentioned-above achieve good performance in some cases, they still have some limitations.

1) While CNN-based networks excel at extracting local spatial features, they have limitations that constrain further performance improvement. CNN-based networks struggle to fully exploit the spatial-spectral information in HSIs. This motivates the development of new deep learning techniques that can overcome the limitations of CNNs for more effective HSI classification.
2) SAE-based networks struggle to achieve robust training and efficient application due to the scarcity of hyperspectral data and the complexity of their training procedure. This motivates the development of other classification techniques that can better utilize the limited samples and achieve simpler end-to-end training.
3) While RNN-based networks possess the ability to model spectral sequence information, they inadequately model spatial information and exhibit insufficient long-range spectral dependencies acquisition capabilities.
4) Effective modeling of long-range spectral dependencies is also challenging for DBN, GAN, and GCN network architectures. Furthermore, limited training samples also result in poor robustness of the models.

Recently, the self-attention mechanism in the Transformer network [17] has achieved great success in natural language processing tasks. The Transformer network comprises two essential components: the encoder and the decoder, with the encoder converting the input sequence into a high-level representation and the decoder utilizing this information to predict the output sequence. The encoder and decoder use self-attention mechanisms to extract contextual information from the entire sequence. Very recently, Dosovitskiy et al. [18] introduced the Transformer network into computer vision and proposed the vision transformer (ViT) for image classification, which segments the input image into 16×16 patches and utilizes the Transformer encoder to encode the position-added vectors of each patch into a high-level representation. ViT segments input images into patches and uses a Transformer encoder to obtain high-level representations. Due to its ability to model long-range dependencies, the Transformer network has been applied to HSI classification [19], [20]. For example, He et al. [21] devised a bidirectional encoded Transformer network, coined HSI-BERT. However, relying solely on the Transformer architecture is challenging to achieve satisfactory results.

In the realm of HSI classification, there is a growing trend toward hybrid approaches that blend both CNN and Transformer architectures. For instance, Mei et al. [19] introduced the group-aware hierarchical transformer (GAHT), which tackles the challenge of excessive dispersion in feature extraction with multihead self-attention by incorporating a novel groupwise pixel embedding module. Zhou et al. [22] proposed an innovative framework known as the mobile 3-D convolution visual transformer (MDvT), leveraging inverted residual structures to reduce model parameters and implementing square blockwise tokenization sequences to boost model performance. Furthermore, Zhang et al. [23] presented the Mixer, a hybrid network that integrates both convolution and Transformer components to extract deep spatial-spectral and spatial features, respectively. Fang et al. [24] introduced a lightweight multiattention convolutional transformer that adeptly fuses spectral and spatial HSI features, achieving efficient classification results even at significantly reduced sample rates.

These CNN-Transformer hybrid methods have demonstrated remarkable competitiveness when compared to conventional CNN approaches and other network architectures. This success can be attributed to their ability to combine the local spatial modeling capabilities of CNNs with the long-range dependency modeling prowess of Transformers. Additionally, these methods introduce novel modules and structures to better adapt to the challenges posed by HSI classification tasks. Furthermore, certain methods, such as the one proposed by Zhou et al. [22], have made notable strides in reducing model parameters and enhancing computational efficiency, thereby further improving the overall model performance. Nevertheless, it is important to acknowledge that these approaches still exhibit certain limitations, which can be summarized as follows.

1) As demonstrated in works like [21], networks exclusively relying on Transformers excel in effectively modeling spectral features. However, they fall short of adequately capturing spatial information, thereby limiting the full exploitation of HSI data and hindering the realization of desired outcomes.
2) The majority of networks that combine convolutional modules with Transformers, as seen in [19] and [22], currently incorporate only a partial integration, which limits their ability to fully leverage the advantages of CNNs in enhancing Transformers. This limitation restricts the potential for further improvements in model performance.
3) Certain methods effectively enhance the attention mechanism of the Transformer through the incorporation of CNNs, as demonstrated by [23] and [24]. These approaches have yielded notable results; nevertheless, they fall short in considering the extraction and interaction of information across various dimensions within HSI images, encompassing height, width, and spectral. Consequently, this omission hinders the potential for further enhancing model performance.

To overcome the weaknesses of existing networks, we propose a novel network called the dual-branch convolutional transformer network (DCTN) for HSI classification. DCTN integrates convolutional and Transformer modules to extract both local and global spatial-spectral features. Specifically, we design a spatial-spectral fusion projection module (SFPM) that combines 3-D and 2-D convolutions to encode HSIs, which facilitates comprehensive extraction of local and global spatial-spectral features. Furthermore, we utilize an efficient interactive self-attention (EISA) mechanism that operates across height, width, and spectral dimensions. This enables deep fusion of spatial-spectral features and modeling of long-range dependencies. Moreover, we devise a CNN branch that extracts spatial features, supplementing the network's utilization of local spatial information. The main contributions of this work are summarized as follows.

1) We propose a novel dual-branch network called the DCTN for HSI classification. By integrating the unique strengths of Transformer and CNN, DCTN adeptly exploits spatial-spectral information in HSI, thereby elevating classification performance. It is noteworthy that our approach goes beyond a mere combination of CNN and Transformer; instead, we thoroughly explore the characteristics of HSIs, seamlessly integrating convolution into the Transformer architecture and attention computation.
2) We propose a novel convolution-based EISA mechanism, named EISA. EISA can perform EISA to fuse the height, width, and spectral information for extracting global and local spatial-spectral features more fully.
3) We introduce an innovative SFPM that ingeniously combines 3-D convolution and grouped 2-D convolution. SFPM fully integrates the strengths of 3-D and grouped 2-D convolution to extract complementary spatial-spectral features from the original input of HSIs effectively, providing rich representations for subsequent classification tasks.
4) We conduct comprehensive experiments on three benchmark HSI classification datasets: Pavia University, Indian Pines, and Houston2013. The results showed that our proposed method achieves state-of-the-art results, surpassing the most advanced HSI classification methods currently available.

The rest of this article is structured as follows: Section II introduces the related work in HSI classification. Section III details the proposed method, followed by a more detailed description of the SFPM and EISA modules. In Section IV, we report and analyze the experimental results. Finally, Section V presents a summary of our work.

II. Related Work

HSI classification holds significant importance within the realm of remote sensing. Numerous researchers have dedicated their efforts to this task and presented various methods to enhance classification performance [25], [26], [27], [28], [29]. This section provides a concise summary of relevant works, organized into three parts: methods relying exclusively on CNNs, methods relying exclusively on Transformer networks, and methods combining CNN and Transformer networks.

A. Methods Relying Exclusively on CNNs

Due to the robust modeling capabilities of CNNs for spatial features, numerous CNN-based approaches have been proposed by researchers for HSI classification tasks, yielding promising results. Hu et al. [7] introduced a 1D-CNN method for HSI classification, which consists of a convolutional layer, a max-pooling layer, and two fully connected layers. This network achieved particular effectiveness in extracting spectral features but lacked modeling of spatial features. Recognizing the vital capacity of 2D-CNN in extracting spatial contextual information, He et al. [30] proposed a 2D-CNN method that leveraged multiscale covariance maps (MCMs) to integrate spatial and spectral information. However, due to the limited receptive field of the 2D-CNN convolutional kernels, the 2D-CNN network was still insufficient in extracting fused spatial-spectral feature representations. Because 3D-CNN convolutional kernels possess a 3-D receptive field and are capable of capturing fused spatial-spectral features, Zhong et al. [31] introduced an end-to-end spectral-spatial residual network (SSRN) consisting of two Spectral residual blocks and two Spatial residual blocks, effectively extracting spatial and rich spectral features through a deep -D convolutional network. However, relying solely on 3-D convolutions did not produce satisfactory classification results. To effectively combine the strengths of 3D-CNN and 2D-CNN, Roy et al. [32] proposed a network named HybridSN, which employed a hybrid approach to combine both 3-D and 2-D convolutions. Initially, the HybridSN network employed 3-D convolutions to extract fused spatial-spectral features, followed by using 2-D convolutions to further extract spatial features, thereby comprehensively capturing HSI information. Although the networks mentioned above demonstrated promising outcomes, the high-dimensional nature of HSIs posed challenges in extracting global feature dependencies solely through 2-D and 3-D convolutions, limiting further improvements in classification performance.

B. Methods Relying Exclusively on Transformer Networks

With the pioneering work of Dosovitskiy et al. [18], the Transformer has gained significant traction in the field of computer vision. Researchers have increasingly applied this model to image classification tasks with notable success [33], [34], [35]. Capitalizing on the Transformer's exceptional performance, several methods based on Transformer networks have been devised and employed in HSI classification tasks [36], [37], [38], showcasing promising outcomes. For instance, He et al. [21] introduced HSI-BERT, a bidirectional encoding architecture rooted in the Transformer network, which effectively captures global feature dependencies within spectral sequences through a global receptive field. Similarly, Hong et al. [39] reevaluated the applicability of Transformer networks in HSI classification, offering the SpectralFormer approach. This approach involves grouping input data to discern feature information from adjacent spectra. An adaptive residual connection method was introduced to enhance hierarchical feature propagation. Furthermore, Chen et al. [40] proposed a multilevel visual transformer model comprising four feature extraction stages to facilitate pyramid feature extraction and mitigate computational complexity. Meanwhile, Qing et al. [41] presented SAT-Net, an enhanced Transformer method incorporating designed spectral attention and a multihead self-attention module equipped with trainable embedding vectors. SATNet proficiently extracts spatial-spectral features and establishes remote spectral dependencies. Moreover, He et al. [42] introduced Spa-Spe-TR, a dual-branch Transformer network architecture encompassing the spectral-transformer (Spe-TR) and spatial-transformer (Spa-TR) modules. The Spe-TR module focuses on spectral feature extraction, while the Spa-TR module captures spatial features. Despite the evident advantage of these Transformer-based networks in modeling remote feature dependencies, their capability to extract spatial features remains limited, resulting in the underutilization of HSI data.

C. Methods Combining CNNs and Transformer Networks

In order to leverage the respective strengths of CNNs in spatial feature extraction and Transformers in handling long-range sequential features, researchers have made concerted efforts to fully exploit the feature extraction capabilities of combining both networks, leading to the proposal of numerous methods for HSI classification tasks. For instance, He et al. [43] introduced the spatial-spectral transformer (SST), which initially employs VGGNet [44] to extract spatial feature representations and feeds these features into a Transformer Encoder to extract spectral feature representations. Sun et al. [45] presented the spectral-spatial feature tokenization transformer (SSFTT), which employs two convolutional mappings to extract shallow features, followed by feature extraction in a Transformer Encoder using a Gaussian-weighted feature tokenizer. Despite demonstrating certain classification performance, these networks merely concatenate CNNs and Transformers without fully capitalizing on the advantages of both networks.

To address this limitation, Ouyang et al. [46] incorporated convolution into the attention mechanism to capture global dependencies of spectrum and space among different tokens, named HybridFormer. Qi et al. [47] introduced a pioneering Global-Local 3-D Convolutional Transformer network by investigating the intricate local spectral relationships within global spectral sequence features and frequency bands. Furthermore, Zhao et al. [48] introduced the convolutional transformer fusion splicing network (CTFSN) which considers local and global perspectives by employing additive and channel superposition fusion methods to capture features expertly. Yang et al. [49] proposed the Interactive Transformer and CNN network integrated with a multilevel feature fusion network (ITCNet) which seamlessly extracts features from diverse perceptual domains and depths by utilizing multilayer Transformers and CNNs, culminating in a multilevel feature fusion process. Despite these methods effectively amalgamating the strengths of Transformers and CNNs while showcasing commendable performance, they do not thoroughly exploit feature information across different dimensions—height, width, and spectral. This neglect results in the oversight of critical interdimensional correlations, ultimately affecting the prospects of further enhancing model performance.

Considering feature extraction across various dimensions, Yang et al. [50] enhances the Transformer's Encoder module by utilizing deep convolutional operations to independently encode spatial-spectral representations along the height, width, and spectral dimensions for HSI classification. In addition, Huang et al. [51] introduced an innovative fusion technique that integrates convolution and transformers by incorporating spatial-spectral attention modules, including height-spatial attention, width-spatial attention, and spectral attention. While both approaches address deep feature extraction across diverse dimensions, they fail to account for feature interactions between these dimensions and lack an effective mechanism for fusing features extracted from these distinct dimensions. This limitation hampers the potential for further enhancements in model performance.

In our work, we propose a novel convolutional Transformer method called DCTN, which comprehensively integrates the advantages of Transformers and CNNs in feature extraction. Our proposed DCTN comprises a dual-branch spatial-spectral feature projection module and a Transformer Encoder with interactive self-attention. The interactive self-attention mechanism we introduce facilitates the extraction of features by incorporating interactions across the height, width, and spectral dimensions, enabling the capture of integrated global and local spatial-spectral features. Additionally, we incorporate a CNN branch to complement the extraction of local spatial features.

III. DCTN With EISA

In this section, we will introduce the DCTN method, focusing on its overall architecture, the SFPM module, and the Encoder sequence with the EISA module.

A. Framework of DCTN

This study presents a novel convolutional Transformer approach for classifying HSI, named DCTN, as shown in Fig. 1. DCTN comprises three key components: the SFPM, the Transformer Branch with EISA, and the CNN Branch. The SFPM module primarily performs shallow feature mapping using 3-D convolution and grouped 2-D convolution. The Transformer branch primarily extracts spatial-spectral features for global and local fusion. The Transformer branch incorporates a well-designed and EISA mechanism, enabling feature extraction across three dimensions: height, width, and spectral. This interactive fusion mechanism ensures the effective fusion of spatial and spectral information. The CNN Branch enhances the extraction of local spatial features by utilizing alternating 2-D standard convolution and 2-D depthwise convolution. This approach effectively captures subtle spectral differences while extracting spatial features.

B. SFPM Module

Fig. 2 shows the SFPM module proposed in this work, which concurrently learns shallow mapping relationships from two branches. The first branch, known as the adaptive 3-D (A3D) convolution branch, extracts fused spatial-spectral information, while the second branch, referred to as the residual grouped 2-D (R2D) Convolution branch, captures grouped local spatial features and spectral differences. Let $X \in R^{C\times S\times H\times W}$ represent a patch of the whole HSI, where

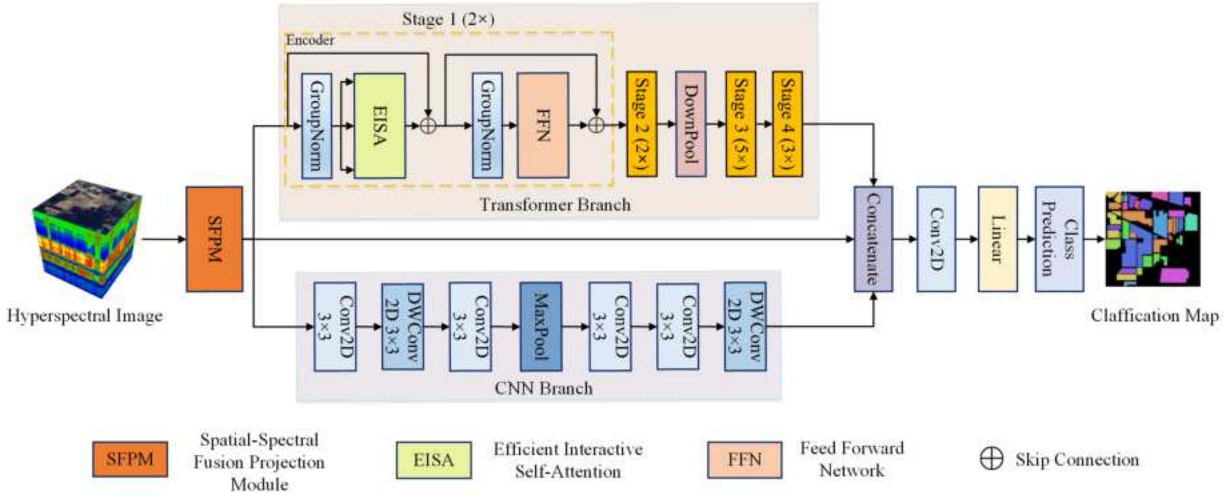


Fig. 1. Overall architecture of DCTN.

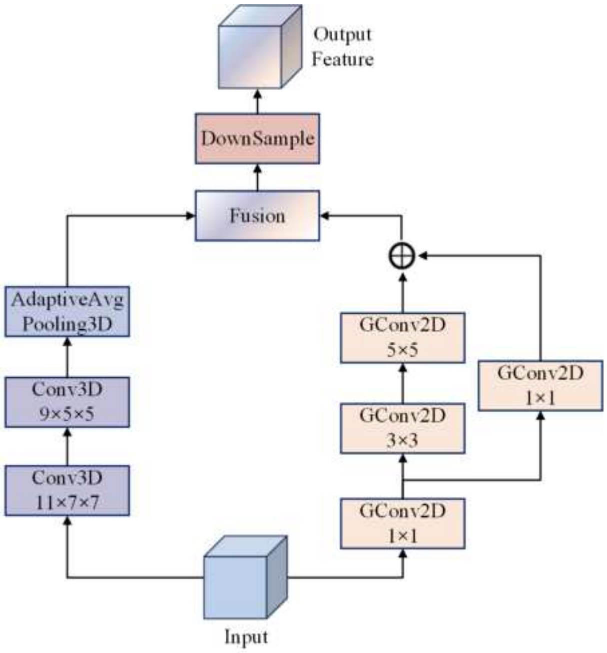


Fig. 2. Overall architecture of the proposed SFPM module.

C denotes the channel dimension, S represents the spectral dimension, and H and W correspond to the height and width dimensions, respectively. The SFPM module can be mathematically expressed as follows:

$$Y_{\text{SFPM}} = \text{DownSample}(\gamma_1 \times \text{A3D}(X) \oplus \gamma_2 \times \text{R2D}(X)) \quad (1)$$

where DownSample denotes a downsampling operation implemented using a 3 × 3 convolution kernel with a stride of 2 × 2 in Conv2D. This downsampling operation is beneficial for subsequent feature extraction. The parameters γ_1 and γ_2 are learnable and used to adjust the weight ratios of the two branches. The symbol $\oplus$ represents elementwise addition, facilitating the merging of features from the two branches. In the subsequent sections, we will provide detailed descriptions of the implementation of each branch.

1) A3D Branch: The A3D branch utilizes 3-D convolutions to effectively extract fused spatial-spectral features. Additionally, it incorporates a 3-D adaptive average pooling layer to facilitate size adjustment and information fusion of the feature map by adaptively modifying the pooling area's size. The A3D branch consists of two 3-D convolutional layers and an adaptive average pooling layer. The convolutional kernel sizes for the two 3-D convolutional layers are 11 × 7 × 7 and 9 × 5 × 5, respectively. By utilizing larger convolutional kernels, the model can capture features at different scales, thereby enhancing its understanding of the spatial-spectral feature structures in the input data. The A3D branch can be formulated as follows:

$$\begin{aligned} Y^1_{\text{A3D}} &= \text{Conv3D}_{9\times5\times5}(\text{Conv3D}_{11\times7\times7}(X)) \\ Y_{\text{A3D}} &= \text{Reshape}\big(\text{BN}\big(\text{AvgPool3D}\big(Y^1_{\text{A3D}}\big)\big)\big) \end{aligned} \quad (2)$$

where $\text{Conv3D}_{9\times5\times5}$ represents the 3-D convolution operation with a convolutional kernel size of 9 × 5 × 5. AvgPool3D represents the 3-D adaptive average pooling layer, and BN denotes batch normalization. Furthermore, the data processing performed by the A3D branch converts the input data $X \in R^{C\times S\times H\times W}$ into $Y \in R^{H\times W\times D}$, where D signifies the encoded spectral dimension.

2) R2D Branch: The R2D branch utilizes grouped 2-D convolutions to partition the spectral bands into multiple groups, facilitating the extraction of local spatial features and subtle spectral differences. Moreover, residual connections are employed to retain lower level information, thereby enhancing the branch's capacity for comprehensive spatial perception. This branch incorporates three convolutional layers and one residual layer. We configure the convolutional kernel sizes of the three convolutional layers to 1 × 1, 3 × 3, and 5 × 5, respectively. Furthermore, we employ a 1 × 1 convolutional layer as the residual layer, which is connected to the end of the branch. Prior to performing 2-D convolutional operation, the input $X \in R^{C\times S\times H\times W}$ undergoes a reshaping process to $X' \in R^{H\times W\times D}$. The representation of this branch is as follows:

$$Y^1_{\text{R2D}} = \text{GConv2D}_{1\times1}\big(X'\big)$$

$$
\begin{aligned}
Y_{\mathrm{R2D}}^2 &= \mathrm{GConv2D}_{5\times 5}\left(\mathrm{GConv2D}_{3\times 3}\left(Y_{\mathrm{R2D}}^1\right)\right)\\
Y_{\mathrm{R2D}} &= Y_{\mathrm{R2D}}^2 \oplus \mathrm{GConv2D}_{1\times 1}\left(Y_{\mathrm{R2D}}^1\right)
\end{aligned}
\tag{3}
$$

where $\oplus$ denotes elementwise addition, GConv2D represents grouped 2-D convolution, and the subscript of GConv2D indicates the size of the convolutional kernel. The output Y_{R2D} has dimensions $R^{H\times W\times D}$. It is important to note that batch normalization layers and activation functions are applied after each convolutional layer, although they are omitted here for brevity.

C. Transformer Branch

The Transformer Branch, depicted in Fig. 1, comprises four stages of Encoders and a downsampling layer. A varying number of Encoders characterizes each stage. The purpose of the downsampling layer is to mitigate computational complexity and enhance the model's performance. The Encoder primarily consists of two pivotal modules, namely EISA and feed forward network (FFN), which facilitate the extraction of spatial-spectral features fused globally and locally. Notably, GroupNorm is employed within the Encoder to optimize the utilization of grouped features through intragroup normalization. Assuming the input data Y_{SFPM}, the Encoder can be represented as follows:

$$
\begin{aligned}
Z' &= \mathrm{EISA}(\mathrm{GN}(Y_{\mathrm{SFPM}})) \oplus Y_{\mathrm{SFPM}}\\
Z &= \mathrm{FFN}\left(\mathrm{GN}\left(Z'\right)\right) \oplus Z'
\end{aligned}
\tag{4}
$$

where GN denotes the operation of GroupNorm normalization, $\oplus$ signifies elementwise addition, and Z' denotes the output resulting from the residual connection between the input data Y_{SFPM} and the data processed by EISA. At the same time, Z represents the output resulting from the residual connection between the input data Z' and the data processed by FFN. The subsequent sections will provide a comprehensive explanation of both the EISA module and the FFN module.

1) EISA Module: As shown in Fig. 3, we propose an EISA mechanism to extract features from the height, width, and spectral dimensions. This mechanism facilitates the fusion of global and local spatial-spectral information. Assuming the input data Y_{SFPM}, which has been normalized through GroupNorm, is represented as $Y_{\mathrm{in}} \in R^{H\times W\times D}$, we feed Y_{in} into three branches: the height branch (B_h), the width branch (B_w), and the spectral branch (B_s), for feature extraction. The following description provides details about the data processing in each branch. In the height branch B_h, the input data Y_{in} undergoes local feature mapping via a 2-D convolutional layer, followed by a nonlinear transformation using the GELU activation function. This process yields $Y_h^1 \in R^{H\times W\times D}$. Subsequently, Y_h^1 is passed through AdaptiveAvgPool2d to perform average pooling, resulting in $Y_h^2 \in R^{H\times 1\times 1}$. To facilitate the Conv1D convolutional operation with a sigmoid activation function, we apply the Squeeze operation to compress the dimensions of Y_h^2, followed by dimension transformation. The squeezed feature is then fed into two 1-D convolutional layers, one with a kernel size of 1 and the other with a kernel size of 3, for feature mapping at different scales. Finally, the UnSqueeze, dimension transformation, and expansion operations are applied to the learned feature representation to generate $Y_h^3 \in R^{H\times 1\times 1}$. The learned feature mapping is elementwise multiplied with Y_h^1 to obtain the representation of long-range dependency features in the height dimension, denoted as $Y_h \in R^{H\times W\times D}$. The computations in this branch can be expressed as follows:

$$
\begin{aligned}
Y_h^1 &= g(\mathrm{Conv2D}_{1\times 1}(Y_{in}))\\
Y_h^2 &= \mathrm{AvgPool2D}\left(Y_h^1\right)\\
Y_h^3 &= \mathrm{Conv1D}_{k=3}\left(\mathrm{Conv1D}_{k=1}\left(Y_h^2\right)\right)\\
Y_h &= Y_h^3 \otimes Y_h^1
\end{aligned}
\tag{5}
$$

where g represents the GELU activation function, AvgPool2D denotes the 2-D adaptive average pooling layer, $\mathrm{Conv1D}_{k=1}$ and $\mathrm{Conv1D}_{k=3}$ represent 1-D convolutional operations with kernel sizes of 1 and 3, respectively, and $\otimes$ denotes elementwise multiplication. Note that, for the sake of brevity, we have omitted the dimension transformation operations in the data processing and the activation functions after each 1-D convolution.

In the width branch B_w, the input data Y_{in} is first subjected to local feature mapping through a Conv2D layer with the GELU activation function, resulting in $Y_w^1 \in R^{H\times W\times D}$. After dimension transformation, Y_w^1 is processed by the adaptive average pooling layer to obtain $Y_w^2 \in R^{W\times 1\times 1}$. To enable interaction and fusion with the height dimension features, we use the Concatenate operation to combine Y_h^2 and Y_w^2, yielding fused height-width features. The concatenated features are then subjected to dimension compression and transformation, followed by two 1-D convolutional layers with kernel sizes of 1 and 3, respectively, to perform feature mapping and dimension processing. This process generates $Y_w^3 \in R^{W\times 1\times 1}$. The learned feature representation is then through dimension processing and elementwise multiplied with Y_w^1 to obtain the representation of long-range dependency features $Y_w \in R^{H\times W\times D}$ in the height-width fusion. This branch can be represented as follows:

$$
\begin{aligned}
Y_w^1 &= g(\mathrm{Conv2D}_{1\times 1}(Y_{in}))\\
Y_w^2 &= \mathrm{AvgPool2D}\left(Y_w^1\right)\\
Y_w^3 &= \mathrm{Conv1D}_{k=3}\left(\mathrm{Conv1D}_{k=1}\left(\mathrm{Concat}\left(Y_h^2, Y_w^2\right)\right)\right)\\
Y_w &= Y_w^3 \otimes Y_w^1
\end{aligned}
\tag{6}
$$

where Concat represents the Concatenate operation on the feature representation produced by Y_h^2 and Y_w^2, and the meanings of other symbols are consistent with the definitions provided above.

In the spectral branch B_s, the input data Y_{in} undergoes local spatial feature mapping using a Conv2D layer with the GELU activation function, resulting in $Y_s^1 \in R^{H\times W\times D}$. The dimension of learned feature representation is then transformed, and Y_s^1 is passed through the adaptive average pooling layer, yielding $Y_s^2 \in R^{D\times 1\times 1}$. To enable interaction and fusion of the height, width, and spectral features, we employ the Concatenate operation to concatenate Y_h^2, Y_w^2, and Y_s^2. The concatenated features are then input into two 1-D convolutional layers with activation functions for global feature mapping and dimension processing, resulting in $Y_s^3 \in R^{D\times 1\times 1}$. Finally, the learned fused features are dimensionally processed, and elementwise

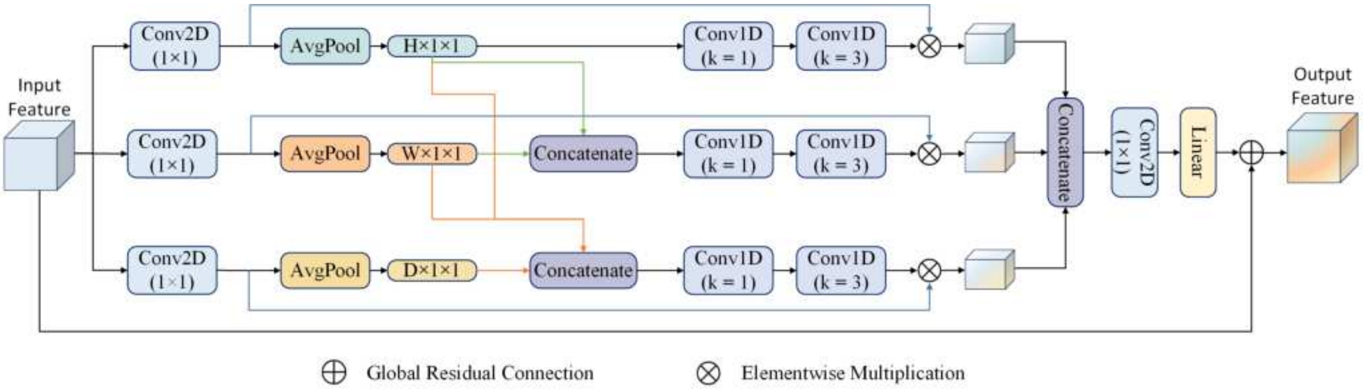


Fig. 3. Architecture of the EISA module.

multiplied with Y_s^1 to obtain the representation of long-range dependency features fused with height, width, and spectral information, denoted as $Y_s \in R^{H\times W\times D}$. This branch can be represented as follows:

$$\begin{aligned}
Y_s^1 &= g(\text{Conv2D}_{1\times1}(Y_{\text{in}}))\\
Y_s^2 &= \text{AvgPool2D}\big(Y_s^1\big)\\
Y_s^3 &= \text{Conv1D}_{k=3}\big(\text{Conv1D}_{k=1}\big(\text{Concat}\big(Y_h^2, Y_w^2, Y_s^2\big)\big)\big)\\
Y_s &= Y_s^3 \otimes Y_s^1.
\end{aligned}\tag{7}$$

After feature mapping in the three branches, representations of features fused from different dimensions are learned. To fully integrate the feature mappings from all three branches, we employ the Concatenate operation to combine the output features from the three branches, followed by a Conv2D and Linear layer for feature fusion. To prevent overfitting to some extent, we use a global residual connection to connect the input feature Y_{in}, resulting in the final output. The fusion process can be represented as follows:

$$\begin{aligned}
Y_{\text{out}}^1 &= \text{Concat}(\alpha_h \times Y_h, \alpha_w \times Y_w, \alpha_s \times Y_s)\\
Y_{\text{out}} &= \text{Linear}\big(\text{Conv2D}_{1\times1}\big(Y_{\text{out}}^1\big)\big) \oplus Y_{\text{in}}
\end{aligned}\tag{8}$$

where $\oplus$ represents the global residual connection, α_h, α_w, and α_s are learnable parameters that control the weight proportions of the height, width, and spectral branches, respectively.

In summary, the EISA module enhances the feature representation in HSI classification tasks by facilitating the interaction and fusion of information across various dimensions. Consequently, the resulting feature representation incorporates global spatial-spectral information and retains intricate details of local features. When this mechanism is absent, the model risks losing correlations among different dimensions, causing information loss. This interaction mechanism is valuable as it fully harnesses correlations across diverse dimensions, enhancing the model's capacity to comprehend and model the data's inherent structure.

2) Feed Forward Network: The FFN module is primarily utilized to perform nonlinear transformations, enhancing the model's representational capacity and performance. As depicted in Fig. 4, the FFN consists of three convolutional layers: two Conv2D layers with 1 × 1 kernels and one DWConv2D layer with a 3 × 3 kernel. Assuming the output of the input data Y_{out} after being normalized through GroupNorm is denoted as $Y_{\text{Fin}} \in R^{H\times W\times D}$. Initially, the Y_{Fin} is fed into the Conv2D layer for feature mapping, followed by a nonlinear operation using the GELU activation function. Subsequently, the DWConv2D layer performs grouped feature mapping on each input channel, followed by GELU activation, and then another Conv2D layer for mapping to yield the final output. This module can be represented as

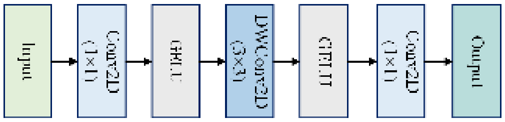


Fig. 4. Structure of the FFN module.

$$Y_{\text{FFN}} = \text{Conv2D}_{1\times1}(\text{DWConv2D}_{3\times3}(\text{Conv2D}_{1\times1}(Y_{\text{in}}))).\tag{9}$$

D. CNN Branch

The CNN branch is explicitly designed to complement the extraction of local spatial features. As illustrated in Fig. 1, this branch primarily consists of four Conv2D layers, two DWConv2D layers, and one MaxPooling layer. We can extract local spatial features more effectively by combining depthwise convolution with regular convolution. Notably, a BatchNorm layer and GELU activation function follow each convolutional layer. Let $Y_{\text{SFPM}} \in R^{H\times W\times D}$ represent the input feature after being mapped by the SFPM module. Initially, dimensionality reduction and deep feature mapping are performed using Conv2D and DWConv2D. Subsequently, Maxpool2D is applied to downsample the feature map obtained through Conv2D convolution and extract salient features. Two consecutive Conv2D layers are utilized to extract spatial information further. Finally, the DWConv2D layer conducts depthwise convolution mapping to obtain the output of this branch. This branch can be expressed as

$$\begin{aligned}
Y_{\text{CNN}}^1 &= \text{DWConv2D}(\text{Conv2D}(Y_{\text{SFPM}}))\\
Y_{\text{CNN}}^2 &= \text{MaxPool}\big(\text{Conv2D}\big(Y_{\text{CNN}}^1\big)\big)\\
Y_{\text{CNN}} &= \text{DWConv2D}\big(\text{Conv2D}\big(\text{Conv2D}\big(Y_{\text{CNN}}^2\big)\big)\big)
\end{aligned}\tag{10}$$

where MaxPool represents the 2-D max pooling layer, and the kernel size for each convolution operation is 3 × 3. We have omitted the formulas for BatchNorm and the activation function to simplify the formula representation.

IV. Experiments

In this section, we comprehensively evaluated our proposed method using three real datasets: Pavia University, Indian Pines, and Houston2013. The section is structured as follows: datasets description, experimental setup, analysis of comparative experimental results, ablation experiments, and parameter experiments. Detailed descriptions of each component are provided subsequently.

A. Datasets Description

1) Pavia University Dataset: The dataset used in this study was obtained from the University of Pavia, Italy, and it was collected using the ROSIS sensor. The HSIs in the dataset include 610 × 340 pixels, with a spectral range of 0.43–0.86 μm. In order to improve the quality of the dataset, 12 bands affected by water absorption were removed, resulting in 103 bands for image classification. The dataset consists of nine different ground feature categories.

2) Indian Pines Dataset: This dataset was collected in 1992 using AVIRIS sensors in northwestern India, USA, and consists of the images with a spatial size of 145 × 145 pixels. The spectral bands range from 0.4 to 2.5 μm, with 20 water absorption bands removed, resulting in 200 remaining spectral bands. The dataset includes ground truths for 16 classes.

3) Houston2013 Dataset: The dataset used in this study was obtained using the CASI-1500 sensor over the University of Houston and its surroundings in Texas, USA. This dataset includes the images with a spatial size of 949 × 1905 pixels and 144 spectral bands. The dataset consists of 15 classification categories. It is important to note that the Huston 2013 dataset used in this study is a cloud-free version provided by the GRSS Data Fusion Competition.

B. Experimental Setup

1) Parameters Setting: In our experiment, we allocated 10% samples for training set across all three datasets. The remaining samples were utilized for testing. The experiments were conducted on a machine equipped with an NVIDIA RTX 2080Ti GPU, utilizing the PyTorch [52] deep learning framework. We perform the experiments ten times and present the results as mean ± standard deviation to avoid disturbance caused by initialization. We employed the Adam optimizer with an initial learning rate of 10^{-4} to perform gradient descent. The batch was set to 32, and the number of epochs was set to 200. For our proposed DCTN method, patch sizes were configured on the Pavia University, Indian Pines, and Houston2013 datasets: 5 × 5, 9 × 9, and 15 × 15, respectively.

2) Evaluation Metrics: We employed two widely used evaluation metrics for the quantitative evaluation of the DCTN method and the comparative methods: overall accuracy (OA) and the Kappa coefficient (K). Additionally, to conduct a quantitative analysis, we visualized the classification results of the three datasets.

3) Comparison Methods: To validate the effectiveness of our proposed DCTN method, we extensively compared it with state-of-the-art and classical methods. These methods include Mou et al. [14], an RNN-based method; 2D-CNN [53], 3D-CNN [54], HybridSN [32], CNN-based methods; ViT [18], a Transformer-based method; and methods that combine convolution and Transformer, namely SSFTT [45], morphFormer [55], HiT [50], and SS-TMNet [51]. For consistency, the batch size for all comparative methods was set to 32, and the number of epochs was set to 200. The patch sizes for the comparative methods were determined based on the optimal sizes provided in the original papers. Specifically, Mou used a 1 × 1 patch size on all three datasets, morphFormer utilized an 11 × 11 patch size across the same datasets, and SSFTT employed patch sizes of 13 × 13, 13 × 13, and 9 × 9 for the Pavia University, Indian Pines, and Houston 2013 datasets, respectively. All other comparative methods employed a default patch size of 15 × 15. The following paragraphs will provide a detailed introduction to the comparative methods employed in our experiments.

1) Mou [14] utilizes a Recurrent Layer with multiple improved GRU modules to analyze hyperspectral pixels as sequential data.
2) 2D-CNN [53] consists of three layers of 2-D convolution, three MaxPooling layers, and two fully connected layers. It is utilized for extracting spatial features from HSIs.
3) 3D-CNN [54] employs eight layers of 3-D convolution and one fully connected layer to extract spatial-spectral features using a deep network.
4) HybridSN [32] combines both 2-D and 3-D convolutions in its network design. It employs three 3-D convolution layers to extract spatial-spectral features, followed by one 2-D convolution layer to extract spatial information.
5) ViT [18] divides the input image into patches, applies positional encoding, and feeds them into the Transformer Encoder sequence to learn long-range spectral features.
6) SSFTT [45] performs PCA dimensionality reduction on the input data. It employs a Gaussian-weighted feature tokenizer for feature transformation and feeds the data into the Transformer Encoder sequence for feature encoding.
7) morphFormer [55] utilizes Transformer Encoders combined with morphological convolution operations and attention mechanisms to extract spatial-spectral features.
8) HiT [50] employs a designed spectral A3D convolution projection module for shallow feature mapping, which is then fed into Transformer Encoders with embedded Depthwise convolution and Pointwise convolution to extract spatial-spectral features.
9) SS-TMNet [51] initially employs a multiscale 3-D convolution module to extract shallow spatial-spectral features. It then feeds these features into Transformer Encoders with height spatial attention, width spatial attention, and spectral attention for global feature extraction.

C. Analysis of Comparative Experiments

In this section, we conducted experiments on three datasets: Pavia University, Indian Pines, and Houston2013. The purpose was to compare various methods and demonstrate the effectiveness and generalization capability of our proposed

TABLE I

COMPARATIVE EXPERIMENTAL RESULTS ON THE PAVIA UNIVERSITY DATASET

Class	**Methods**									
#	**Mou**	**2D-CNN**	**3D-CNN**	**HybridSN**	**ViT**	**SSFTT**	**morphFormer**	**HiT**	**SS-TMNet**	**DCTN**
1	90.32±0.41	96.37±0.12	94.38±0.82	95.30±0.57	92.92±0.58	97.09±0.04	97.60±0.06	95.14±0.28	96.11±0.24	**98.57±0.09**
2	95.77±0.14	92.70±0.05	92.58±0.10	92.65±0.06	91.13±0.21	93.79±0.04	94.80±0.04	92.53±0.08	92.67±0.08	**97.66±0.07**
3	75.34±0.69	93.44±0.42	89.34±1.78	90.68±1.44	82.35±1.41	94.82±0.33	**95.78±0.21**	89.91±1.29	92.35±0.66	90.37±1.03
4	94.63±0.46	97.52±0.18	96.60±0.44	97.30±0.24	95.80±0.47	97.33±0.24	97.88±0.14	97.15±0.17	96.46±0.49	**98.69±0.33**
5	99.80±0.15	99.97±0.05	99.83±0.12	99.93±0.08	99.69±0.21	99.79±0.12	99.79±0.11	99.91±0.07	99.66±0.16	**99.99±0.02**
6	85.96±0.53	**99.99±0.01**	99.66±0.24	99.77±0.18	94.74±0.86	99.97±0.06	99.97±0.03	99.38±0.23	99.91±0.09	99.04±0.36
7	71.43±2.52	99.46±0.27	93.34±2.57	96.51±1.55	90.72±1.34	**99.98±0.03**	99.92±0.10	95.79±1.50	99.05±0.57	98.86±0.30
8	82.87±0.67	99.21±0.26	96.52±0.83	97.25±1.22	94.43±0.58	99.50±0.22	**99.58±0.14**	97.39±0.57	98.31±0.38	95.04±0.40
9	99.44±0.20	99.57±0.26	97.81±2.25	99.61±0.37	97.79±0.96	98.54±0.57	99.20±0.32	99.47±0.25	98.02±0.76	**99.95±0.06**
OA(%)	91.14±0.21	92.05±0.07	90.95±0.43	91.44±0.28	88.92±0.31	93.18±0.07	94.29±0.06	91.28±0.21	91.74±0.12	**96.57±0.14**
K(%)	88.19±0.27	89.84±0.09	88.44±0.55	89.06±0.35	85.81±0.40	91.25±0.09	92.63±0.07	88.85±0.27	89.44±0.16	**95.49±0.19**

TABLE II

COMPARATIVE EXPERIMENTAL RESULTS ON THE INDIAN PINES DATASET

Class	**Methods**									
#	**Mou**	**2D-CNN**	**3D-CNN**	**HybridSN**	**ViT**	**SSFTT**	**morphFormer**	**HiT**	**SS-TMNet**	**DCTN**
1	31.28±11.84	83.41±4.85	59.12±19.14	34.68±26.05	50.11±10.28	**95.31±4.14**	93.19±4.91	80.64±8.22	87.48±8.15	72.29±11.04
2	72.76±1.68	92.06±1.02	68.94±3.90	67.37±20.83	65.46±2.57	**97.10±1.01**	96.33±1.06	86.18±4.40	88.56±2.34	95.70±1.57
3	55.39±2.30	78.55±1.27	56.48±4.81	44.89±22.79	52.57±2.58	85.94±0.55	85.95±1.57	69.94±4.99	76.50±3.23	**88.93±1.40**
4	47.20±6.37	84.17±2.68	42.42±10.92	34.47±25.45	57.92±7.50	90.00±3.16	90.63±3.37	75.63±5.39	82.19±3.92	**92.77±2.77**
5	85.59±2.77	84.64±2.45	79.25±4.81	55.75±24.99	52.76±5.08	89.33±1.47	89.31±2.69	75.26±2.71	81.71±3.49	**92.04±1.28**
6	93.19±0.92	96.98±1.13	94.78±2.04	81.17±21.38	79.49±2.52	98.51±0.75	98.30±0.61	94.79±1.60	97.76±0.92	**98.64±0.68**
7	50.16±17.71	77.66±7.68	29.28±19.57	13.61±16.56	43.72±16.49	**83.61±14.18**	81.12±14.71	73.03±18.03	72.09±16.30	73.76±9.76
8	93.37±0.81	94.16±0.71	92.98±1.32	75.92±26.44	89.41±2.32	97.38±0.26	**99.68±0.38**	93.09±0.78	94.39±0.47	97.75±0.95
9	33.62±14.20	64.30±12.20	48.46±22.24	32.78±29.91	31.35±13.48	**91.39±8.90**	82.29±14.81	59.99±16.58	68.66±17.26	76.70±15.73
10	66.05±1.98	88.50±1.01	74.63±3.50	52.51±32.21	61.48±2.95	93.33±1.13	93.01±1.03	85.34±3.43	87.19±1.98	**93.57±1.21**
11	72.82±1.14	92.40±0.67	81.21±1.85	80.16±9.91	72.26±1.33	**95.71±0.55**	95.65±0.59	89.73±2.47	90.70±1.63	95.46±0.53
12	60.66±2.77	84.97±1.89	58.86±8.24	49.05±25.87	51.64±2.99	91.32±1.57	92.32±1.41	76.38±7.70	81.85±3.97	**94.68±1.39**
13	94.23±2.25	95.68±3.16	97.37±2.07	70.88±25.35	86.61±3.46	98.87±1.47	99.27±0.68	95.57±1.90	97.18±3.02	**99.89±0.18**
14	92.56±0.75	97.28±0.51	94.81±1.05	89.57±10.91	88.50±1.37	**99.22±0.27**	98.59±0.60	94.53±1.25	96.21±0.89	98.49±0.40
15	61.43±3.35	65.66±1.31	46.77±8.42	29.38±13.56	44.94±3.84	73.33±1.57	76.11±1.45	58.84±7.65	63.92±3.78	**77.76±1.60**
16	84.57±2.65	84.49±4.60	55.33±16.41	30.91±30.87	48.29±12.06	89.28±4.17	92.80±2.70	86.10±6.24	87.73±3.15	**96.80±1.98**
OA(%)	75.27±0.77	85.98±0.42	74.20±2.41	67.26±13.98	66.21±0.89	91.11±0.40	91.98±0.57	82.13±2.65	84.67±1.25	**92.85±0.41**
K(%)	71.57±0.87	84.22±0.46	70.55±2.84	62.21±16.96	61.65±0.98	89.94±0.45	90.91±0.64	79.77±3.02	82.66±1.41	**91.87±0.47**

TABLE III

COMPARATIVE EXPERIMENTAL RESULTS ON THE HOUSTON2013 DATASET

Class	**Methods**									
#	**Mou**	**2D-CNN**	**3D-CNN**	**HybridSN**	**ViT**	**SSFTT**	**morphFormer**	**HiT**	**SS-TMNet**	**DCTN**
1	95.49±0.92	98.07±0.75	96.80±1.32	97.74±0.72	95.82±0.84	98.40±0.90	97.89±0.45	96.99±0.87	97.60±0.64	**98.86±0.50**
2	96.28±0.68	98.49±0.51	96.53±1.24	97.54±0.91	96.03±0.98	98.66±0.82	**99.48±0.21**	97.69±0.52	98.44±0.56	99.33±0.22
3	**99.97±0.05**	99.85±0.26	98.86±0.93	99.21±1.00	98.15±0.65	99.68±0.26	99.86±0.13	99.28±0.69	99.50±0.23	99.74±0.29
4	96.50±0.97	98.75±0.33	96.22±1.75	98.35±0.84	95.25±0.79	98.48±0.67	98.08±0.69	97.45±0.64	97.26±0.96	**99.21±0.41**
5	97.76±0.71	98.17±0.58	96.55±0.49	96.72±1.11	96.10±0.87	98.64±0.07	**99.32±0.23**	97.49±0.61	98.19±0.33	98.65±0.12
6	97.19±2.86	96.75±2.33	83.29±5.35	93.85±2.61	73.81±5.75	97.67±2.67	97.56±1.12	88.74±3.62	93.67±2.33	**98.75±1.07**
7	83.06±0.99	97.43±0.61	92.15±1.44	93.78±1.73	91.16±1.34	97.90±0.64	97.51±0.94	93.05±1.08	94.54±1.03	**98.20±0.45**
8	67.91±1.94	94.51±1.28	86.63±3.40	90.60±2.20	88.58±1.38	97.53±0.95	97.18±1.03	91.24±1.97	95.74±1.35	**98.22±0.69**
9	78.28±1.83	95.99±0.89	89.57±1.69	89.02±4.05	88.71±1.77	**97.83±1.23**	96.90±0.81	90.64±1.84	94.29±1.32	97.66±0.56
10	72.09±2.47	96.90±1.14	89.98±3.19	92.31±3.74	90.39±1.23	**99.08±0.83**	97.88±1.02	92.39±1.92	96.91±0.81	98.89±0.49
11	76.74±1.04	97.64±0.89	89.87±1.81	91.84±3.23	91.15±1.53	98.39±0.55	97.83±0.94	93.28±1.55	94.94±0.72	**98.51±0.33**
12	71.20±2.04	97.40±0.88	88.76±2.19	91.47±3.03	87.13±1.52	**99.27±0.47**	97.62±1.32	90.72±2.25	96.50±1.00	98.96±0.22
13	54.00±5.40	94.95±1.57	90.01±1.89	92.38±1.39	74.81±4.09	96.63±1.84	97.21±0.81	88.52±2.33	93.42±1.61	**97.68±1.01**
14	95.64±1.02	99.58±0.73	96.54±1.71	96.06±2.52	95.13±1.43	99.86±0.44	99.81±0.45	97.13±1.62	99.88±0.19	**100.00±0.00**
15	98.25±0.40	98.95±0.66	98.01±1.08	96.02±2.05	94.69±2.24	99.19±0.69	**99.62±0.46**	98.40±1.06	98.98±0.58	99.24±0.88
OA(%)	84.91±0.51	97.07±0.41	92.36±1.40	93.90±1.70	91.28±0.69	98.07±0.27	97.98±0.36	93.94±1.02	96.22±0.35	**98.31±0.16**
K(%)	83.68±0.55	96.84±0.44	91.75±1.51	93.41±1.84	90.58±0.74	97.92±0.29	97.82±0.39	93.45±1.10	95.92±0.38	**98.17±0.17**

DCTN method. The analysis of the results is presented in three parts: quantitative analysis, qualitative analysis, and complexity analysis.

1) Quantitative Analysis: Tables I–III summarize the quantitative analysis results obtained from the Pavia University, Indian Pines, and Houston2013 datasets. These tables include the evaluation metrics OA and Kappa of the results for each class in the datasets. Notably, bold data highlights the best-performing results within each class.

Our proposed DCTN method achieved the best results across all three datasets. On the Pavia University dataset, the OA metric reached 96.57% ± 0.14%, while on the Indian Pines dataset, it achieved 92.85% ± 0.41%. Moreover, on the Houston2013 dataset, the OA metric reached an impressive 98.31% ± 0.16%. It is worth noting that our experimental results were averaged over ten runs and presented as mean ± standard deviation. Our DCTN method achieves higher classification accuracy compared to the state-of-the-art and classical methods, with the lowest standard deviation. These results demonstrate the effectiveness and stability of our proposed method.

Table I illustrates that our proposed DCTN method attained superior results on the Pavia University dataset, achieving OA and Kappa metrics of 96.57% and 95.49%, respectively.

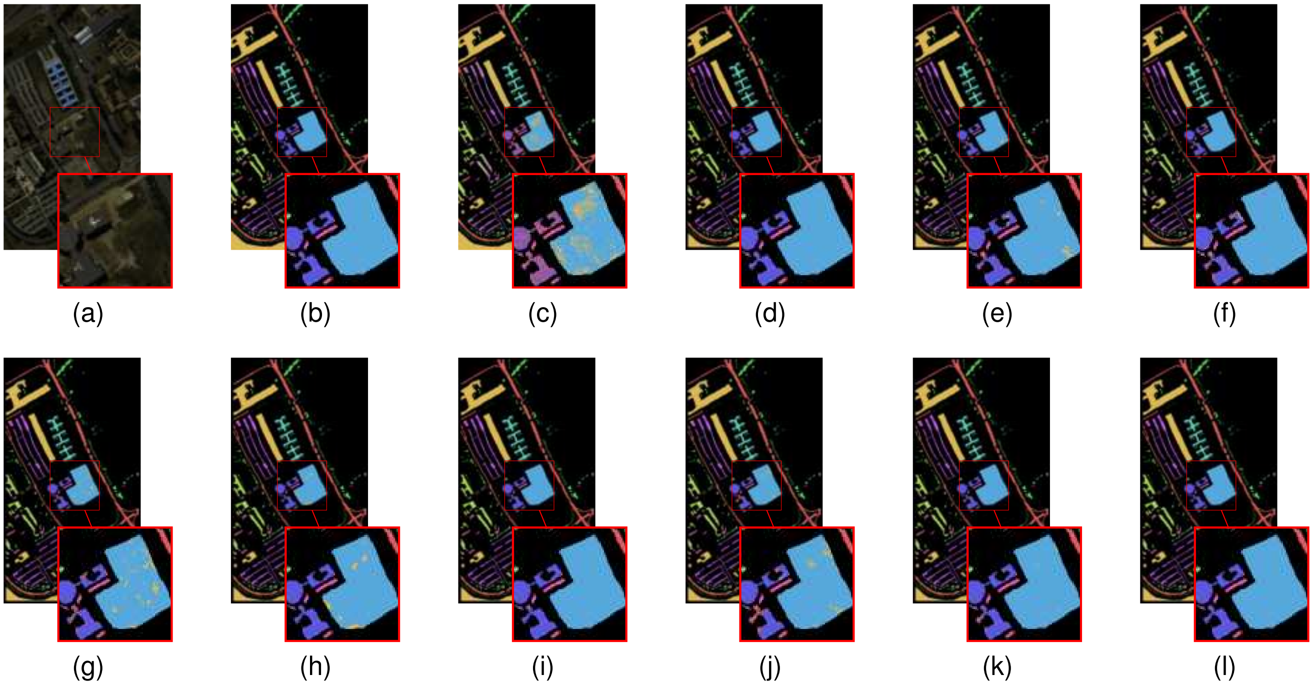


Fig. 5. Visualization of classification maps on the Pavia University dataset. (a) Original image, (b) ground truth, (c) Mou, (d) 2D-CNN, (e) 3D-CNN, (f) HybridSN, (g) ViT, (h) SSFTT, (i) morphFormer, (j) HiT, (k) SS-TMNet, and (l) DCTN(ours).

In contrast, the OA and Kappa accuracies of the SSFTT method were 93.18% and 91.25%, respectively, exhibiting a decrease of 3.39% and 4.24% compared to our DCTN. One factor contributing to the efficacy of our proposed method is the deeper integration of CNN and Transformer advantages within our DCTN. In contrast to the SSFTT method's simplistic merging of the two networks, we introduced a meticulously crafted convolutional-based Transformer encoder, encompassing an efficient self-attention mechanism based on convolution and a group convolution-based FFN module. This methodology facilitates a more comprehensive capturing of local and global spatial-spectral features in HSIs.

In the experiments conducted on the Indian Pines dataset Table II, our proposed DCTN method achieved an OA of 92.85% and a Kappa value of 91.87%, surpassing the performance of the other comparative methods. In contrast, the ViT method exhibited the lowest performance, with an OA of only 66.21%. These results could be attributed to the exclusive emphasis of ViT on extracting long-range spectral features while neglecting spatial information modeling, resulting in unsatisfactory performance. Additionally, the HybridSN method displayed significantly lower performance on the Indian Pines dataset, with an OA of only 67.26%. This could be attributed to the ineffective combination of 3-D and 2-D convolutions in HybridSN, which relies on simple concatenation and fails to model long-range spectral dependencies adequately. In contrast, our proposed DCTN method effectively integrates 2-D and 3-D convolutions using the SFPM module, and the convolutional Transformer Encoder sequence captures global spectral and spatial dependencies, leading to excellent performance.

In Table III, DCTN also achieved the best results with OA and Kappa values of 98.31% and 98.17% on the Houston2013 dataset. Moreover, our proposed method achieved perfect results for class 14 (Tennis Court). In comparison, the OA and Kappa values of the HiT method on this dataset are merely 93.94% and 93.45%, respectively, lagging behind our method by 4.37% and 4.72%. The results can be ascribed to HiT neglecting the extraction of interactive information between different dimensions of HSIs during encoding, resulting in inadequate information utilization. Conversely, our DCTN method integrates an EISA mechanism into the Transformer architecture. It considers information extraction across various dimensions and granularities, aiming for a more comprehensive feature representation.

In summary, our proposed DCTN method consistently achieved the best and most stable results across the majority of classification categories in all three datasets, thereby demonstrating the robustness and effectiveness of our proposed DCTN method.

2) Qualitative Analysis: To conduct the qualitative analysis, we visualized the classification results obtained on the Pavia University, Indian Pines, and Houston2013 datasets, as illustrated in Figs. 5–7. It is worth noting that, to present the visual results clearly, we enlarged the local regions in the classification maps, which are indicated by the red rectangular boxes in the figures.

The DCTN method has showcased superior performance across all three datasets, exhibiting minimal noise and delivering highly accurate classification results when viewed overall. One potential explanation lies in the effective integration of convolution and Transformer within our DCTN method, enabling it to possess robust capabilities in extracting both local and global features. Conversely, the Mou approach yields the least favorable results, generating substantial noise. The possible reason for this result is that Mou lacks spatial

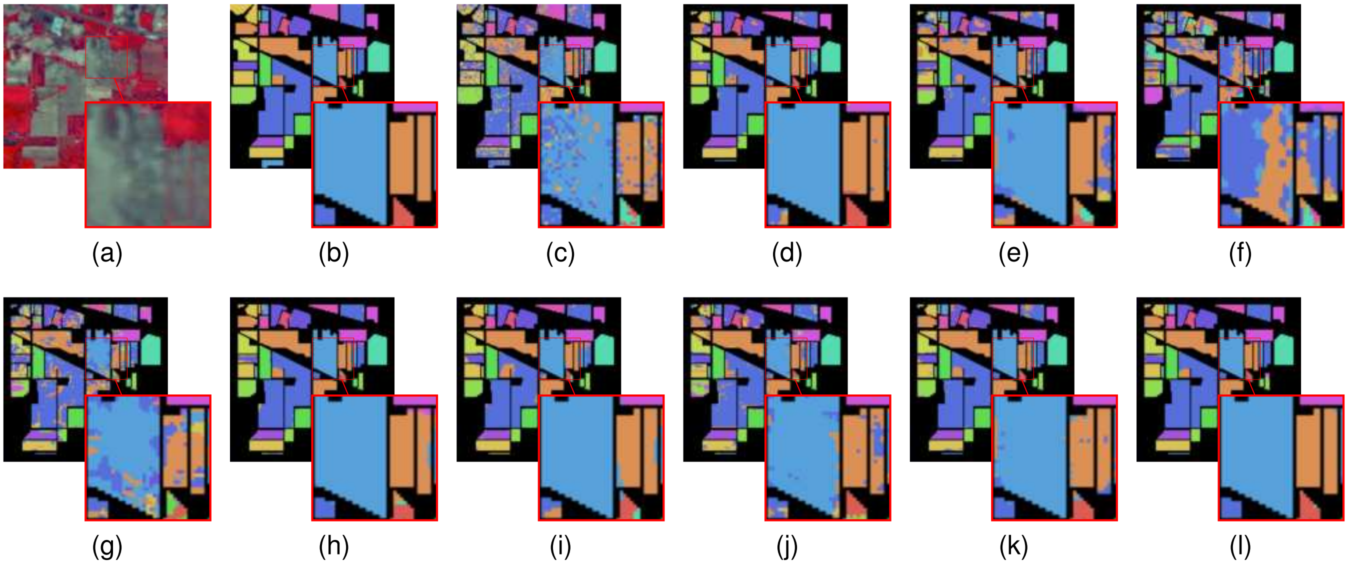


Fig. 6. Visualization of the classification maps based on the Indian Pines dataset. (a) Original image, (b) ground truth, (c) Mou, (d) 2D-CNN, (e) 3D-CNN, (f) HybridSN, (g) ViT, (h) SSFTT, (i) morphFormer, (j) HiT, (k) SS-TMNet, and (l) DCTN(ours).

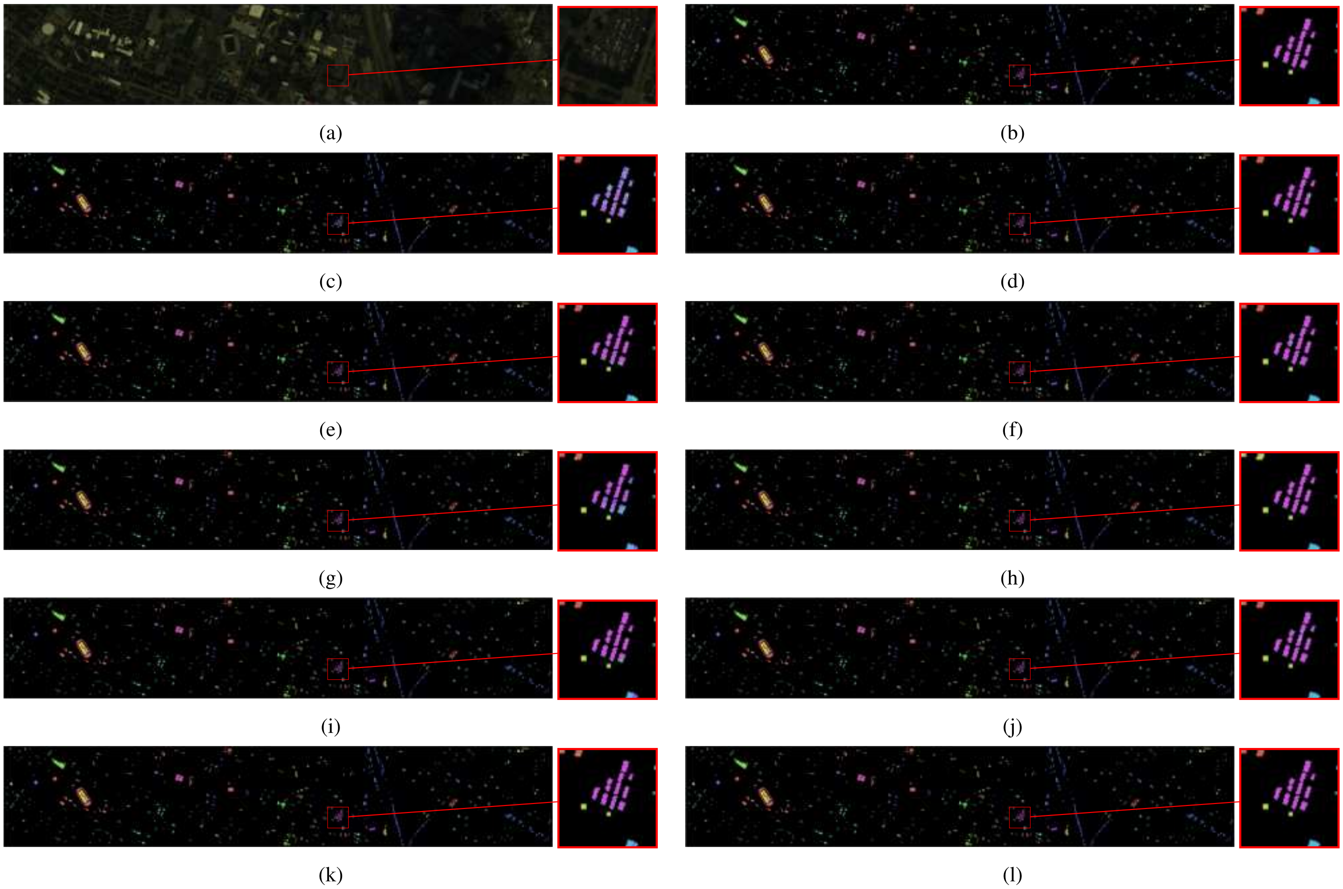
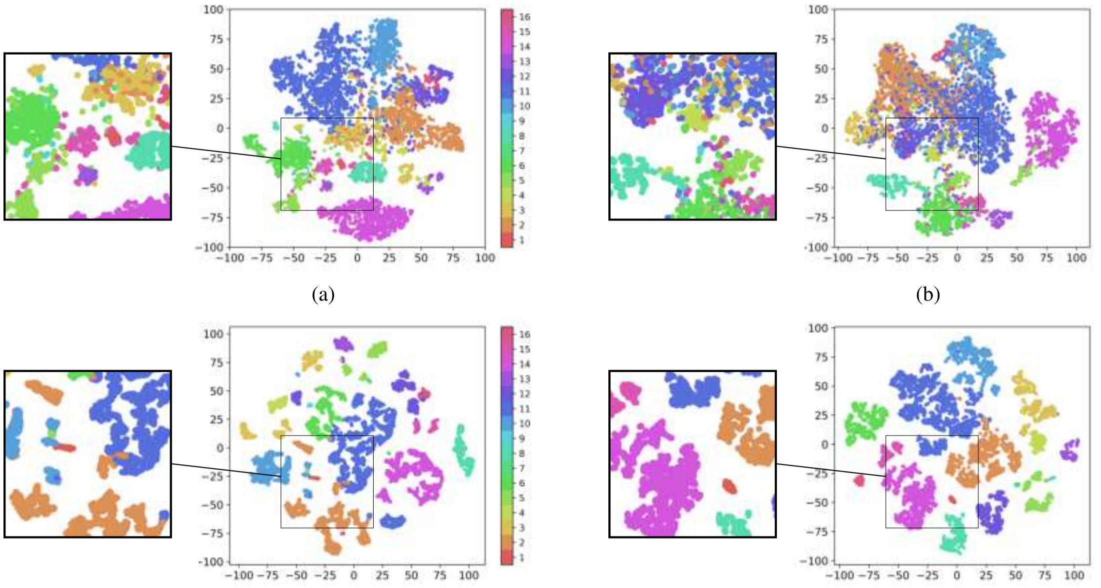


Fig. 7. Visualization of the classification maps based on the Houston2013 dataset. (a) Original image, (b) ground truth, (c) Mou, (d) 2D-CNN, (e) 3D-CNN, (f) HybridSN, (g) ViT, (h) SSFTT, (i) morphFormer, (j) HiT, (k) SS-TMNet, and (l) DCTN(ours).

information modeling and cannot effectively capture long-range spectral dependencies, resulting in poor results.

Fig. 5 shows that the classification maps produced by the 2D-CNN method and the morphFormer method are relatively similar to the DCTN method. However, they still manifest slight noise, lacking the stability observed in our proposed approach. The underlying reason for this dissimilarity is that the 2D-CNN method concentrates solely on extracting local spatial features, neglecting the modeling of long-range spectral information and global spatial-spectral characteristics, thereby engendering noise. Furthermore, the morphFormer method falls short of fully leveraging the benefits of both convolution



(a)

(b)

(c)

(d)

Fig. 8. t-SNE visualization results of hidden features obtained by various methods on Indian Pines dataset. (a) HybridSN. (b) ViT. (c) morphFormer. (d) DCTN.

and Transformer, leading to suboptimal feature modeling for local and global fusion.

The visualization results of HiT and SS-TMNet methods on the three datasets have also exhibited promising results. These methods incorporate convolution into the Transformer architecture, extracting features across three dimensions: height, width, and spectral. Nevertheless, they still generate a certain degree of noise. The plausible cause for this outcome stems from their disregard for dimensional interactions and inadequate integration and utilization of the extracted information. In conclusion, our proposed DCTN method accurately identifies nearly all categories, yields minimal noise, and surpasses all other comparative methods. This not only underscores the effectiveness of our proposed approach but also showcases its robust generalization capabilities.

As shown in Fig. 8, we conducted a t-SNE visualization comparative analysis of hidden features using various methods on the Indian Pines dataset. The figure shows that our proposed DCTN method is able to reduce interclass misclassification and demonstrate a stronger clustering tendency. As indicated in the enlarged portion of the figure, the DCTN method produces more densely stacked features for different categories, tighter intraclass clustering, and more distinct interclass separation. This indicates that the DCTN method effectively captures global feature dependencies and local feature relationships among different materials in HSIs. One possible reason is that the DCTN method employs EISA, facilitating the model in better capturing global dependencies and local feature representations through the interactive extraction and fusion of features across different dimensions.

3) Complexity Analysis: We assessed various methods' model complexity on the Indian Pines dataset, including calculations of floating-point operations (FLOPs), parameter numbers (Param), training time, and testing time, as shown in Table IV. It is noteworthy that our proposed DCTN method demonstrates commendable performance. However, it is essential to acknowledge that there is room for improvement in speed and efficiency, which constitutes a limitation of the DCTN method. In a broader context, compared to RNN-based and CNN-based methods, it becomes apparent that most Transformer-based methods require a longer time to acquire intricate feature representations. For instance, the training durations for 2D-CNN and SS-TMNet amount to 31.29 and 333.93 s, respectively. This difference can be primarily attributed to the substantial attention computational workload imposed by Transformer-based methods to pursue robust feature representations.

TABLE IV

Complexity Analysis Results Between DCTN and the Comparative Methods

Methods	FLOPs (GB)	Param (MB)	Training time (s)	Testing time (s)
Mou	0.01	0.26	25.07	2.68
2D-CNN	0.03	2.76	31.29	2.04
3D-CNN	0.13	1.54	80.05	6.54
HybridSN	0.53	4.32	79.27	7.78
ViT	0.13	2.61	83.01	5.31
SSFTT	0.02	0.15	18.92	1.18
morphFormer	0.07	0.21	74.13	9.19
HiT	1.17	51.23	159.8	11.68
SS-TMNet	2.67	83.33	333.93	31.12
DCTN	1.48	45.32	253.16	20.69

Regarding computational complexity, our DCTN method exhibits FLOPs and Param numbers within the moderate range among Transformer-based methods. Specifically, the Param numbers for HiT and SS-TMNet are 51.23 and 83.33 MB, respectively, whereas DCTN's Param is 45.32 MB. It is essential to highlight that while DCTN may not be the most efficient method compared to others, it consistently achieves state-of-the-art accuracy across all datasets, as indicated in Tables I–III.

D. Ablation Studies

To validate the effectiveness of the proposed modules, we conducted ablation experiments on Houston2013 dataset for the SFPM module, the EISA module with interactive

TABLE V
ABLATION STUDY RESULTS OF THE PROPOSED COMPONENTS ON THE HOUSTON2013 DATASETS

Nets	Components				Metric(%)	
#	MDCP	EISA(w/int.)	EISA	CNN branch	OA	Kappa
ViT	×	×	×	×	91.28	90.58
ViT-SFPM	✓	×	×	×	97.51 (**↑6.23%**)	97.31 (**↑6.73%**)
DCTN-EISA(w/Int.)	✓	✓	×	×	98.09 (**↑6.81%**)	97.94 (**↑7.36%**)
DCTN-EISA	✓	×	✓	×	98.24 (**↑6.96%**)	98.10 (**↑7.52%**)
DCTN(w/Ext.)	✓	×	✓	✓	**98.31** (**↑7.03%**)	**98.17** (**↑7.59%**)

self-attention, the EISA module without interactive self-attention, and the CNN branch. Table V shows the OA and Kappa results. The "EISA(w/int.)" designation signifies the EISA module without the interactive mechanism, eliminating fusion among different dimensional branches. The ViT model was employed as the benchmark model, and ablation experiments were performed on each module. The experimental results were obtained by averaging ten iterations. The bold data in parentheses represent the performance changes relative to the benchmark model ViT.

"ViT-SFPM" refers to applying our proposed SFPM module for feature encoding before ViT. Analysis of the table reveals a significant enhancement in classification accuracy compared to the original ViT method. Specifically, the OA and Kappa metrics increased by 6.23% and 6.73%, respectively. These results serve as clear evidence of the effectiveness of the proposed SFPM module. The SFPM module can combines local spatial information extracted by the R2D branch with fused spatial-spectral information from the A3D branch, resulting in a powerful representation of shallow features.

The term "DCTN-EISA(w/int.)" represents the DCTN method without the interactive mechanism of the EISA module and the CNN branch. In this configuration, the EISA module calculates attention independently for the height, width, and spectral branches without engaging in dimensionwise interaction. The results in the table demonstrate that DCTN-EISA(w/int.) achieves better classification accuracy compared to ViT-SFPM. The OA and Kappa metrics increased by 0.58% and 0.63%, respectively. These findings indicate that our designed convolutional Transformer architecture is more effective than ViT in deep feature extraction. These results provide partial confirmation of the effectiveness of the proposed EISA module.

"DCTN-EISA" denotes the DCTN method with the CNN branch removed. Table V shows that this approach attains enhanced classification accuracy compared to DCTN-EISA(w/int.) across all three datasets. The OA and Kappa metrics increased by 0.15% and 0.16%, respectively. These improvements affirm the effectiveness of our proposed self-attention-based interactive mechanism. Moreover, these results offer substantial evidence for the effectiveness of the proposed EISA module, which not only extracts features from diverse dimensions but also captures fused interactive features.

"DCTN(w/Ext.)" represents the original, unmodified DCTN method. The table demonstrates that this method achieves the best classification results across all three datasets, surpassing DCTN-EISA comprehensively. The OA and Kappa metrics are 0.07% higher than those of DCTN-EISA, underscoring the efficacy of our designed CNN branch. The CNN branch effectively captures local spatial information and integrates it into the network to enhance the network's capability for spatial feature extraction. In conclusion, each component within our design exhibits promising results and plays a crucial role in the feature extraction process.

TABLE VI
OA(%) RESULTS FOR DCTN METHOD ACROSS VARIED PATCH SIZES

Patch Size	Datasets		
#	Pavia University	Indian Pines	Houston2013
5 × 5	**96.57 ± 0.14**	84.74 ± 1.10	96.69 ± 0.31
7 × 7	95.91 ± 0.07	84.41 ± 2.48	96.75 ± 0.17
9 × 9	95.35 ± 0.05	**92.85 ± 0.41**	97.75 ± 0.41
11 × 11	94.31 ± 0.06	92.02 ± 0.41	97.97 ± 0.20
13 × 13	93.32 ± 0.05	91.08 ± 0.21	98.23 ± 0.22
15 × 15	92.26 ± 0.06	89.42 ± 0.25	**98.31 ± 0.16**
17 × 17	91.17 ± 0.04	87.34 ± 0.23	97.87 ± 0.19
19 × 19	90.09 ± 0.04	85.24 ± 0.23	97.29 ± 0.16

E. Patch Size Studies

To evaluate the impact of varying patch sizes on the classification results of our proposed DCTN model, we conducted experiments on three distinct datasets, and the OA metric are presented in Table VI. Specifically, we investigated a range of patch sizes, from 5 × 5 to 19 × 19 with two-unit intervals. We performed ten independent trials and recorded the standard ± deviations for the sake of result stability and reliability.

Upon examination of the table, it becomes evident that our model's optimal patch size differs across datasets. This disparity may arise from distinctions in dataset characteristics, noise levels, and land cover distributions. The optimal patch size in the Pavia University dataset was 5 × 5, resulting in an OA of 96.57 ± 0.14. Moreover, for the Indian Pines dataset, we observed that a patch size of 9 × 9 yielded the highest OA, achieving 92.85 ± 0.41. In the case of the Houston2013 dataset, the optimal patch size was identified as 15 × 15, yielding an OA of 98.31 ± 0.16. These findings underscore the importance of selecting an appropriate patch size tailored to specific contexts to maximize classification performance.

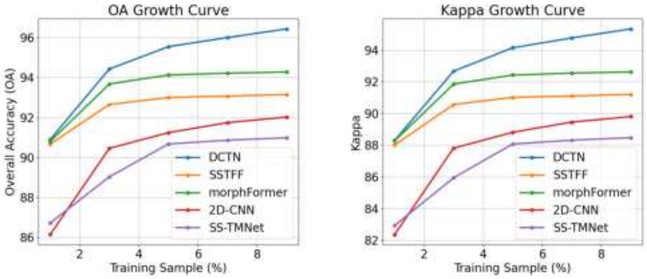
F. Assessing Results With Varied Training Sample Ratios

In order to assess the scability of our proposed method, we conduct the experiments with different number of training samples on the Indian Pines dataset, the training samples ranging from 10% to 50% with 10% intervals. The classification results of metric OA are presented in Table VII.

TABLE VII

CLASSIFICATION RESULTS(OA) BY THE DCTN METHOD AND COMPARISON METHODS WITH DIFFERENT TRAINING SAMPLES ON THE INDIAN PINES DATASET

Training Sample	**Methods**									
#	**Mou**	**2D-CNN**	**3D-CNN**	**HybridSN**	**ViT**	**SSFTT**	**morphFormer**	**HiT**	**SS-TMNet**	**DCTN**
10%	75.27±0.77	85.98±0.42	74.20±2.41	67.26±13.98	66.21±0.89	91.11±0.40	91.98±0.57	82.13±2.65	84.67±1.25	**92.85±0.41**
20%	79.24±0.62	89.69±0.19	82.25±2.26	68.08±12.81	77.58±0.56	92.55±0.25	94.40±0.03	88.52±0.55	87.62±0.97	**95.37±0.29**
30%	82.18±0.42	90.51±0.22	86.88±1.33	88.95±1.37	82.77±0.80	92.92±0.23	94.56±0.12	89.73±0.41	89.48±0.41	**95.81±0.11**
40%	83.05±0.52	90.96±0.16	89.10±0.56	88.39±2.31	85.63±0.53	93.02±0.22	94.66±0.12	90.39±0.32	90.26±0.46	**96.01±0.19**
50%	84.16±0.41	91.04±0.25	89.63±0.36	89.86±1.10	87.72±0.56	93.16±0.30	94.77±0.12	90.65±0.30	90.48±0.47	**96.10±0.17**



(a) (b)

Fig. 9. Classification results achieved by the proposed DCTN and the comparative methods with varying training samples on the Pavia University dataset. (a) OA results. (b) Kappa results.

From the table, it is evident that as the number of training samples increases, the classification accuracy of the DCTN method gradually improves and consistently surpasses the performance of other comparative methods. This substantiates the stability and remarkable generalization ability of our proposed approach. Compared to the HybridSN method, when the quantity of training samples increases from 30% to 40%, the classification accuracy declines from 88.95% to 88.39%, highlighting the method's poor stability. One possible factor contributing to this instability is the ineffective integration of 3-D convolutional and 2-D convolutional operations and the disregard for extracting long-range spectral information.

To further validate our proposed method's stability and generalization ability, we conducted training sample parameter experiments on the Pavia University dataset and visualized the results in Fig. 9. The figure demonstrates that even with smaller training samples ranging from 1% to 9%, as the number of training samples increases, our proposed DCTN method consistently outperforms the compared methods in terms of OA and Kappa results. This finding suggests that our method exhibits strong classification capabilities even with limited sample sizes, providing additional evidence of its stability and exceptional generalization ability.

V. CONCLUSION

In this work, we propose a novel dual-branch convolutional Transformer method for HSI classification tasks called DCTN, which incorporates interactive self-attention. We effectively integrate convolution operations into the Transformer architecture and introduce a new SFPM to learn shallow feature representations. Different from directly inputting the raw pixel features in ViT model, our proposed SFPM module utilizes 2-D grouped convolutional branches extract spatial features in groups and employs 3-D convolutional branches to extract spatial-spectral features, ultimately generating a feature mapping sensitive to spatial-spectral information. Additionally, we design a new interactive self-attention module called EISA to enable interactive feature encoding across height, width, and spectral dimensions, effectively capturing globally and locally fused spatial-spectral information. We also introduce a new CNN branch to extract and integrate local spatial information into the network efficiently. Furthermore, extensive experiments conducted on three datasets validate our proposed method's effectiveness and generalization capability. The experimental results demonstrate that the DCTN method outperforms other state-of-the-art and classical methods.

In our future work, we will explore using self-supervised and transfer learning techniques to facilitate efficient learning in situations with limited samples. We will also investigate solutions to address the overlap issue during training. These efforts are aimed at further enhancing the performance of our model.

REFERENCES

[1] M. Ahmad et al., "Hyperspectral image classification—Traditional to deep models: A survey for future prospects," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 968–999, 2022.

[2] C. Wang et al., "A review of deep learning used in the hyperspectral image analysis for agriculture," *Artif. Intell. Rev.*, vol. 54, no. 7, pp. 5205–5253, Oct. 2021.

[3] B. Zhang, D. Wu, L. Zhang, Q. Jiao, and Q. Li, "Application of hyperspectral remote sensing for environment monitoring in mining areas," *Environ. Earth Sci.*, vol. 65, no. 3, pp. 649–658, Feb. 2012.

[4] F. D. van der Meer et al., "Multi-and hyperspectral geologic remote sensing: A review," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 14, no. 1, pp. 112–128, Feb. 2012.

[5] G. Vivone, "Multispectral and hyperspectral image fusion in remote sensing: A survey," *Inf. Fusion*, vol. 89, pp. 405–417, Jan. 2023.

[6] X. Yang, W. Cao, Y. Lu, and Y. Zhou, "Self-supervised learning with prediction of image scale and spectral order for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5545715.

[7] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, Jan. 2015, Art. no. 258619.

[8] W. Zhao and S. Du, "Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Aug. 2016.

[9] W. Song, S. Li, L. Fang, and T. Lu, "Hyperspectral image classification with deep feature fusion network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3173–3184, Jun. 2018.

[10] M. He, B. Li, and H. Chen, "Multi-scale 3D deep convolutional neural network for hyperspectral image classification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3904–3908.

[11] J. Zheng, Y. Feng, C. Bai, and J. Zhang, "Hyperspectral image classification using mixed convolutions and covariance pooling," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 522–534, Jan. 2021.

[12] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.

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

[14] 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, Jul. 2017.

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

[16] Y. Ding et al., "Multi-feature fusion: Graph neural network and CNN combining for hyperspectral image classification," *Neurocomputing*, vol. 501, pp. 246–257, Aug. 2022.

[17] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[18] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. 9th Int. Conf. Learn. Represent.*, 2021, pp. 1–12.

[19] S. Mei, C. Song, M. Ma, and F. Xu, "Hyperspectral image classification using group-aware hierarchical transformer," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5539014.

[20] E. Xie, N. Chen, J. Peng, W. Sun, Q. Du, and X. You, "Semantic and spatial–spectral feature fusion transformer network for the classification of hyperspectral image," *CAAI Trans. Intell. Technol.*, vol. 8, no. 4, pp. 1308–1322, 2023.

[21] J. He, L. Zhao, H. Yang, M. Zhang, and W. Li, "HSI-BERT: Hyperspectral image classification using the bidirectional encoder representation from transformers," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 165–178, Sep. 2019.

[22] X. Zhou, W. Zhou, X. Fu, Y. Hu, and J. Liu, "MDvT: Introducing mobile three-dimensional convolution to a vision transformer for hyperspectral image classification," *Int. J. Digit. Earth*, vol. 16, no. 1, pp. 1469–1490, 2023.

[23] J. Zhang, Z. Meng, F. Zhao, H. Liu, and Z. Chang, "Convolution transformer mixer for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 6014205.

[24] Y. Fang, Q. Ye, L. Sun, Y. Zheng, and Z. Wu, "Multi-attention joint convolution feature representation with lightweight transformer for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5513814.

[25] H. Yu et al., "Global spatial and local spectral similarity-based manifold learning group sparse representation for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3043–3056, May 2020.

[26] Y. Su, L. Gao, M. Jiang, A. Plaza, X. Sun, and B. Zhang, "NSCKL: Normalized spectral clustering with kernel-based learning for semisupervised hyperspectral image classification," *IEEE Trans. Cybern.*, vol. 53, no. 10, pp. 6649–6662, Oct. 2023.

[27] S. Mei, X. Li, X. Liu, H. Cai, and Q. Du, "Hyperspectral image classification using attention-based bidirectional long short-term memory network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5509612.

[28] Y. Su, M. Jiang, L. Gao, X. Sun, X. You, and P. Li, "Graph-cut-based collaborative node embeddings for hyperspectral images classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 6010905.

[29] Y. Huang et al., "Two-branch attention adversarial domain adaptation network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5540813.

[30] N. He et al., "Feature extraction with multiscale covariance maps for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 755–769, Feb. 2019.

[31] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2018.

[32] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, "HybridSN: Exploring 3-D–2-D CNN feature hierarchy for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 277–281, Feb. 2019.

[33] B. Heo, S. Yun, D. Han, S. Chun, J. Choe, and S. J. Oh, "Rethinking spatial dimensions of vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 11936–11945.

[34] B. Graham et al., "LeViT: A vision transformer in ConvNet's clothing for faster inference," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12259–12269.

[35] Q. Hou, Z. Jiang, L. Yuan, M. Cheng, S. Yan, and J. Feng, "Vision permutator: A permutable MLP-like architecture for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 1328–1334, Jan. 2023.

[36] X. Yang, W. Cao, Y. Lu, and Y. Zhou, "QTN: Quaternion transformer network for hyperspectral image classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 12, pp. 7370–7384, Dec. 2023.

[37] K. Wu, J. Fan, P. Ye, and M. Zhu, "Hyperspectral image classification using spectral–spatial token enhanced transformer with hash-based positional embedding," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5507016.

[38] B. Zhang, Y. Chen, Y. Rong, S. Xiong, and X. Lu, "MATNet: A combining multi-attention and transformer network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5506015.

[39] D. Hong et al., "SpectralFormer: Rethinking hyperspectral image classification with transformers," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5518615.

[40] X. Chen, S.-I. Kamata, and W. Zhou, "Hyperspectral image classification based on multi-stage vision transformer with stacked samples," in *Proc. IEEE Region 10 Conf. (TENCON)*, Dec. 2021, pp. 441–446.

[41] Y. Qing, W. Liu, L. Feng, and W. Gao, "Improved transformer net for hyperspectral image classification," *Remote Sens.*, vol. 13, no. 11, p. 2216, Jun. 2021.

[42] X. He, Y. Chen, and Q. Li, "Two-branch pure transformer for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 6015005.

[43] X. He, Y. Chen, and Z. Lin, "Spatial–spectral transformer for hyperspectral image classification," *Remote Sens.*, vol. 13, no. 3, p. 498, Jan. 2021.

[44] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.

[45] 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, 2022, Art. no. 5522214.

[46] E. Ouyang, B. Li, W. Hu, G. Zhang, L. Zhao, and J. Wu, "When multigranularity meets spatial–spectral attention: A hybrid transformer for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 4401118.

[47] W. Qi, C. Huang, Y. Wang, X. Zhang, W. Sun, and L. Zhang, "Global–local 3-D convolutional transformer network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5510820.

[48] F. Zhao, S. Li, J. Zhang, and H. Liu, "Convolution transformer fusion splicing network for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, 2023, Art. no. 5501005.

[49] H. Yang, H. Yu, K. Zheng, J. Hu, T. Tao, and Q. Zhang, "Hyperspectral image classification based on interactive transformer and CNN with multilevel feature fusion network," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, 2023, Art. no. 5507905.

[50] X. Yang, W. Cao, Y. Lu, and Y. Zhou, "Hyperspectral image transformer classification networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5528715.

[51] X. Huang, Y. Zhou, X. Yang, X. Zhu, and K. Wang, "SS-TMNet: Spatial–spectral transformer network with multi-scale convolution for hyperspectral image classification," *Remote Sens.*, vol. 15, no. 5, p. 1206, 2023.

[52] A. Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.

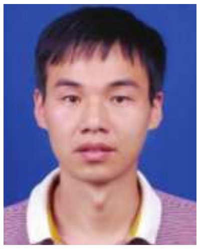
[53] V. Sharma, A. Diba, T. Tuytelaars, and L. Van Gool, "Hyperspectral CNN for image classification & band selection, with application to face recognition," KU Leuven, ESAT, Leuven, Belgium, Tech. Rep. KUL/ESAT/PSI/1604, 2016.

[54] X. Yang, Y. Ye, X. Li, R. Y. K. Lau, X. Zhang, and X. Huang, "Hyperspectral image classification with deep learning models," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5408–5423, Sep. 2018.

[55] S. K. Roy, A. Deria, C. Shah, J. M. Haut, Q. Du, and A. Plaza, "Spectral–spatial morphological attention transformer for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5503615.

Yunfei Zhou received the B.S. degree from Zaozhuang University, Zaozhuang, China, in 2021. He is currently pursuing the M.S. degree in computer technology with East China Jiaotong University, Nanchang, China.

He is currently a Researcher with the Institute for Data Science and Deep Learning, East China Jiaotong University. His research interests include deep learning and hyperspectral image classification.



Xiaohui Huang (Member, IEEE) received the B.Eng. and master's degrees from Jiangxi Normal University, Nanchang, China, in 2005 and 2008, respectively, and the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 2014.

He is currently an Associate Professor of Computer Science with East China Jiaotong University, Nanchang. His research interests are in the areas of machine learning, deep learning, and clustering algorithm.

Xiaofei Yang received the B.S. degree from Suihua University, Suihua, China, in 2011, and the M.S. and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 2014 and 2019, respectively.

He was a Post-Doctoral Researcher with the Department of Computer and Information Science, University of Macau, Macau, China, from 2020 to 2023. He is currently with the School of Electronic and Communication Engineering, Guangzhou University, Guangzhou, China. His research interests are in the areas of semisupervised learning, deep learning, remote sensing, transfer learning, and graph mining.

Jiangtao Peng (Senior Member, IEEE) received the B.S. and M.S. degrees from Hubei University, Wuhan, China, in 2005 and 2008, respectively, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2011.

He is currently a Professor at the Faculty of Mathematics and Statistics, Hubei University. His research interests include machine learning and hyperspectral image processing.

Yifang Ban (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from Nanjing University, Nanjing, China, in 1984 and 1987, respectively, and the Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada, in 1996.

Before joining KTH Royal Institute of Technology (KTH), Stockholm, Sweden, in 2004, she was a tenured Associate Professor at York University, Toronto, ON, Canada. She is currently the Chair Professor and the Director of the Geoinformatics Division, KTH Royal Institute of Technology, Stockholm, and an Associate Director of Digital Futures, Stockholm. Her research interests include Earth observation big data analytics, machine learning/deep learning and their applications in mapping urban and land cover, monitoring urbanization, wildfires, and other environmental changes, and assessing environmental impact. She has published extensively on these topics.

Dr. Ban is the Co-Chair of the ICA Commission on Sensor-Driven Mapping and the Co-Lead of the Group on Earth Observations (GEO) initiative "Global Urban Observation and Information" (from 2012 to 2022). She has been an Associate Editor and the Guest Editor for major remote sensing journals and an Invited Expert for EU and national grant application evaluations. Since 2016, she has been an Invited Expert of the UN Habitat Technical Committee on Human Settlements Indicators for UN Sustainable Development Goals (SDGs).