






# S<sup>2</sup>DCN: Spectral–Spatial Difference Convolution Network for Hyperspectral Image Classification

Zitong Zhang , Graduate Student Member, IEEE, Hanlin Feng , Graduate Student Member, IEEE, Chunlei Zhang , Qiaoyu Ma , Student Member, IEEE, and Yuntao Li 

**Abstract**—A novel spectral–spatial difference convolution network (S<sup>2</sup>DCN) is proposed for hyperspectral image (HSI) classification, which integrates the difference principle into the deep learning framework. S<sup>2</sup>DCN employs a learnable gradient encoding pattern to extract important detail features in spectral and spatial domains, alleviating the information loss caused by the oversmoothing effect in deep feature extraction. Specifically, the feature extraction modules in S<sup>2</sup>DCN are designed, namely spectral difference convolution (SeDC) module and spatial difference convolution (SaDC) module. The SeDC module performs 1-D difference convolution in the spectral domain to capture peak-valley information in sensitive narrow bands, enhance subtle spectral differences, and preserve fine-grained features. The SaDC module employs 2-D difference convolution in the spatial domain, integrating fine-structural features while preserving the deep abstract features extracted by vanilla convolutions. This further empowers the capability of the model to extract discriminative features. A series of experiments are performed on four publicly available HSI datasets to demonstrate the effectiveness of S<sup>2</sup>DCN method, which is compared with current state-of-the-art models. The experimental results show that the proposed S<sup>2</sup>DCN outperforms competitors and achieves optimal classification performance.

**Index Terms**—Deep learning, detail feature, difference convolution, hyperspectral image (HSI) classification.

## I. INTRODUCTION

**H**YPERSPECTRAL images (HSIs) have hundreds of contiguous and narrow spectral bands, which collect abundant spectral and spatial information for monitoring the surface of the Earth [1], [2]. Each pixel in HSI contains spectral information, representing reflectance or radiance intensity at specific

wavelengths, as well as spatial information, denoting its position within the image and its relational context with neighboring pixels [3]. HSI classification is a pivotal phase in analyzing HSI data, intending to extract ground object features from labeled pixels and assign corresponding labels to individual pixels. This technique finds extensive applications in diverse fields, including land cover analysis [4], military target detection [5], and agricultural monitoring [6].

Early research in HSI classification predominantly emphasized the utilization of spectral features, promoting the development of several pixelwise classification algorithms, such as k-nearest neighbors [7], support vector machine [8], and logistic regression [9]. However, these primary methods are constrained in adaptive feature extraction and often overlook vital spatial context information, resulting in unsatisfactory classification results. In recent years, more and more studies have emphasized the importance of the spatial relationships between adjacent pixels in HSI for object recognition, highlighting the critical role of spatial information in classifying HSI data [10]. Consequently, approaches that combine spectral and spatial features have shown substantial advantages in enhancing classification performance [11].

With the rapid advancement in computer technology, deep learning has achieved remarkable breakthroughs in computer vision fields, such as image processing and object detection [12]. In this context, deep learning has also been successfully applied to remote sensing data analysis [10]. Driven by data, deep learning employs an end-to-end learning approach, allowing it to extract and fuse spectral–spatial features adaptively, thus significantly improving HSI classification accuracy. Currently, various backbone networks that have proven effective in computer vision have been successfully applied to HSI classification, such as autoencoder [13], recurrent neural network [14], and convolutional neural network (CNN) [15]. In particular, CNN has become widely adopted in HSI classification due to its excellent ability to model local context. It achieves feature extraction from shallow to deep layers by stacking convolutional kernels [16]. Recently, the emerging network architectures based on self-attention mechanism, namely vision transformer (ViT) [17], [18] and MLP-mixer (Mixer) [19] with a pure multilayer perceptron (MLP) structure, have achieved remarkable results in vision tasks and have been successfully applied to HSI classification [20], [21]. Consequently, the current deep learning methods in HSI classification have formed three major backbone network families represented by CNN, ViT, and Mixer [22].

Manuscript received 20 October 2023; revised 12 December 2023; accepted 20 December 2023. Date of publication 3 January 2024; date of current version 16 January 2024. (Corresponding author: Hanlin Feng.)

Zitong Zhang is with the School of Earth Sciences and Resources, China University of Geosciences, Beijing 100083, China (e-mail: 3001200116@email.cugb.edu.cn).

Hanlin Feng is with the School of Computer and Information Technology, Northeast Petroleum University, Daqing 163318, China (e-mail: fhanlin@163.com).

Chunlei Zhang is with the Beijing Zhongdi Runde Petroleum Technology Company Ltd., Beijing 100083, China (e-mail: zcl\_3559@126.com).

Qiaoyu Ma is with the College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China (e-mail: xueyanshe\_mqy@163.com).

Yuntao Li is with the School of Mechanical and Electrical Engineering, China University of Mining and Technology- Beijing, Beijing 100083, China (e-mail: 11760014861@163.com).

Digital Object Identifier 10.1109/JSTARS.2023.3349175

Deep learning methods combine feature aggregation with layerwise mapping to extract spectral and spatial features from HSI data and learn discriminative features for classification. However, deep learning models, such as CNN, often suffer from oversmoothing due to the stacking of convolutional kernels and weighted averaging strategies. This limitation hampers the ability of the model to capture local detail patterns [23]. On the other hand, ViT and Mixer models, while enhancing the modeling of global features in images, come with a large number of parameters, resulting in redundant computations. These deep models generate an oversmoothing effect through layerwise feature extraction, leading to strong correlations among the interfeatures, weakening the modeling capacity for local detail structures, and losing valuable high-frequency information (i.e., sharply changing regions) in the spatial–spectral domain. Consequently, it is prone to overfitting and affects the classification performance [24].

Many studies have introduced attention mechanisms to improve the classification performance and generalization ability of the model, such as the channel attention mechanism, namely the squeeze-and-excitation block [25] and the convolutional block attention module that simultaneously incorporates both channel attention and spatial attention [26], [27]. Introducing attention mechanisms can make the model selectively focus on salient parts instead of treating each part equally [2]. However, the aforementioned attention-based models still rely on the initial weight that depends on the correlation between pixels in the image without explicitly modeling the spatial structural relationship. As a result, they have a weak ability to extract local patterns and preserve detail features, which may lead to the loss of subtle spectral feature changes and spatial morphological structural features. On the other hand, attention models based on transformers increase the parameter size and complexity of the model when calculating attention weights. The oversmoothing issue caused by high-dimensional computation results in strong correlations between extracted features, which may lead to the loss of detail features in the spectral and spatial domains, thereby reducing the discriminative ability of the model. Traditional feature engineering methods can be utilized to overcome these limitations and further extract local detail features. Local descriptor operators, such as local binary pattern (LBP) [28], histogram of oriented gradient (HOG) [29], and scale-invariant feature transform (SIFT) [30] are widely used to extract texture, edge, and other detail features from images. Among them, LBP is a powerful method for representing local detail information by calculating the difference in grayscale between neighboring and central pixels. However, LBP suffers from issues such as a single calculation method and fixed weight, which prevent it from achieving higher accuracy.

Given the aforementioned challenges, a spectral–spatial difference convolution network ( $S^2DCN$ ) is proposed for HSI classification. Inspired by the simple yet effective design in [31],  $S^2DCN$  incorporates the difference principle of LBP into a deep learning framework, augmenting additional spectral and spatial detail features. It combines deep learning with backpropagation for iterative weight optimization, enabling the learning of

quantifiable local encoding patterns. The  $S^2DCN$  model focuses on harnessing the spectral and spatial details present in HSI data with an attention-like weighted approach, which effectively merges the ability of deep abstract feature representation in vanilla convolution with the fine-grained feature perception in central difference convolution (CDiff), strengthening the ability of the model to extract and preserve detail features within shallow layers. Specifically, we design two modules: the spectral difference convolution (SeDC) module operating in the spectral domain and the spatial difference convolution (SaDC) module operating in the spatial domain. These two modules combine vanilla convolution with CDiff using different weight coefficients, with the goal of ensuring the learning of deep abstract features while incorporating the extraction of gradient-level detail features. The  $S^2DCN$  model can significantly enhance the modeling ability of local structures and reduce the oversmoothing effect while increasing intraclass compactness and interclass distinctiveness of spectral–spatial features. Therefore, this can improve classification performance. The main contributions of this article can be summarized as follows.

- 1) A novel  $S^2DCN$  is proposed to preserve gradient detail information and extract spectral–spatial features. The difference principle is introduced into the deep learning architecture for HSI classification tasks for the first time. By employing the fusion strategy of vanilla convolution and CDiff,  $S^2DCN$  simultaneously considers the extraction of high-level semantic and local detail features, thereby enhancing the discriminability of spectral–spatial features.
- 2)  $S^2DCN$  innovatively makes the local gradient information learnable by integrating difference information from LBP into the deep learning framework, which leverages the backpropagation algorithm to optimize the weights associated with detail features, strengthening its ability to extract local detail patterns in the spectral and spatial domains.
- 3) Two modules are developed according to the characteristics of HSI data. The SeDC module effectively captures subtle spectral changes and enhances the spectrum's local mutation frequency range (peaks and valleys) through 1-D difference convolution. The SaDC module aggregates difference information within the neighborhood through 2-D difference convolution, enhancing the representation of fine-grained structural features, such as morphology and texture, and reducing information loss in the propagation process.
- 4) Extensive experiments are performed on four benchmark HSI datasets. The experimental results indicate the outstanding classification performance of  $S^2DCN$ . In addition, the ablation experiments are designed to showcase the effectiveness of the proposed difference convolution modules.

The rest of this article is organized as follows. Section II introduces the related work of HSI classification and difference convolution. Section III details the principle of the proposed  $S^2DCN$ . The experimental results and analysis are illustrated in Section IV. Finally, Section V concludes this article.

## II. RELATED WORK

### A. HSI Classification via Deep Learning

For HSI classification tasks, feature extraction is pivotal in determining classifier performance. HSI data features are predominantly distributed in the spectral and spatial domains. Spectral features reflect variations in reflectance across different bands, furnishing insights into the spectral attributes of ground objects, such as their spectral shape and intensity. Spatial features encompass texture structures, morphological characteristics, interpixel spatial relationships, and spatial statistical information, providing insights into the diverse spatial distributions. Given the limited resolution of HSI data acquisition, spectral mixing phenomena can introduce complexity to the relationship between spectra and ground objects. In such scenarios, relying solely on spectral information for HSI classification often fails to achieve the desired outcomes. Therefore, integrating spectral features with spatial features allows for a comprehensive and effective representation of interclass differences, facilitating accurate ground object identification.

Deep learning stands at the forefront of HSI classification, with distinctive encoding patterns and a talent for learning intrinsic data characteristics, and robust feature extraction ability. Recently, classic backbone networks from computer vision have been adeptly utilized for HSI classification tasks. CNN-based networks excel at modeling local information in the spectral and spatial domains and representing intricate nonlinear features [32], [33]. MLP-based models boast superior flexibility and universality, making them particularly apt for managing the inherent dense features of HSI data [34], [35]. Models rooted in the ViT paradigm demonstrate a distinct edge in addressing long-term dependencies in HSI data, capturing global features effectively [36], [37].

However, current deep learning methods still have limitations in extracting and preserving valuable detail features. During the feature extraction stage, these models typically perform convolution, linear mapping, or pooling operations on adjacent pixel information, which leads to the abstraction of detail features during the layerwise propagation, resulting in the over-smoothing effect and reducing the classification performance. To better extract and preserve local detail features, researchers have proposed various methods. Introducing attention mechanisms is a commonly used approach, such as self-attention, channel attention, and spatial attention. Zhang et al. optimized the feature extraction process based on CNN by constructing self-attention modules in spectral and spatial domains and score-weighted fusion [38]. This adaptive approach effectively combines local features with long-term dependencies related to the target pixel for better capturing detailed texture features. Paoletti et al. proposed an automatic attention-based CNN based on channel attention, which automatically designs and optimizes convolutional neural structures for HSI processing. It utilizes convolutional architecture to extract discriminative spectral–spatial features [39]. Guo et al. introduced a spectral–spatial attention mechanism that focuses on the redundancy and differences between frequency bands through a proposed feature grouping strategy to obtain more useful information [40]. In addition, researchers have

enhanced the perception of local textures by adjusting the shape and size of convolutional kernels [41]. However, when attention mechanisms are applied to HSI data, they may overly focus on highly correlated regions, suppressing spatial structure modeling and resulting in biased and selective feature learning, leading to the loss of critical information about ground objects. Furthermore, attention mechanisms based on transformers increase network complexity and computational requirements, making it challenging to effectively handle features at different scales. Therefore, improving the preservation of fine-grained features in deep learning models remains a promising research direction.

### B. Difference Convolution

In specific tasks such as object detection, edge detection, and texture analysis in computer vision, detail information such as object boundaries, texture details, and line structures are crucial. These details are often encoded in the shallow explicit features of an image, and classical local descriptors have been effective methods for extracting detail features in the early stages. For instance, LBP [29], HOG [42], SIFT [43], and mathematical morphology methods [44] can effectively represent local features. LBP is an effective method for extracting explicit gradient features from images. It characterizes the edge context's abrupt changes and detail features through central difference operations and can depict local gradient information, such as texture. HOG represents detail texture and shape information by calculating local gradient directions in the image, while SIFT extracts local features by detecting key points and calculating their descriptors. Mathematical morphology methods use morphological operations, such as erosion and dilation to process and represent image details. In comparison, LBP focuses on describing local texture information and is more sensitive to slight texture and detail variations in the image, thus performing well in tasks that require emphasis on image details. However, traditional local descriptors heavily rely on handcrafted features, resulting in weak generalization of the methods and limited by inherent encoding patterns and shallow representation capabilities, leading to lower accuracy. Therefore, researchers are increasingly devoting more effort to utilizing deep learning to extract features and improve classification accuracy.

In computer vision, where deep learning dominates, the conventional approach directly aggregates local intensity-level information (usually measured in grayscale) using vanilla convolutions. Such an approach primarily relies on the feature values of current position and the surrounding neighborhood, which makes it susceptible to external disturbances and challenging to represent fine-grained features. To address this issue, Yu et al. first proposed the CDiff [45], which extracted fine-grained information, such as edges and textures, by performing CDiff on images. Experimental results demonstrated that CDiff surpassed other enhanced convolution operators, such as LBConv [46] and GaborConv [47], in feature extraction capabilities. In later research, noting the redundancy of CDiff and the challenges in network optimization due to the aggregation of gradients in various directions, Yu et al. introduced the cross-CDiff [48]. This method decouples CDiff into

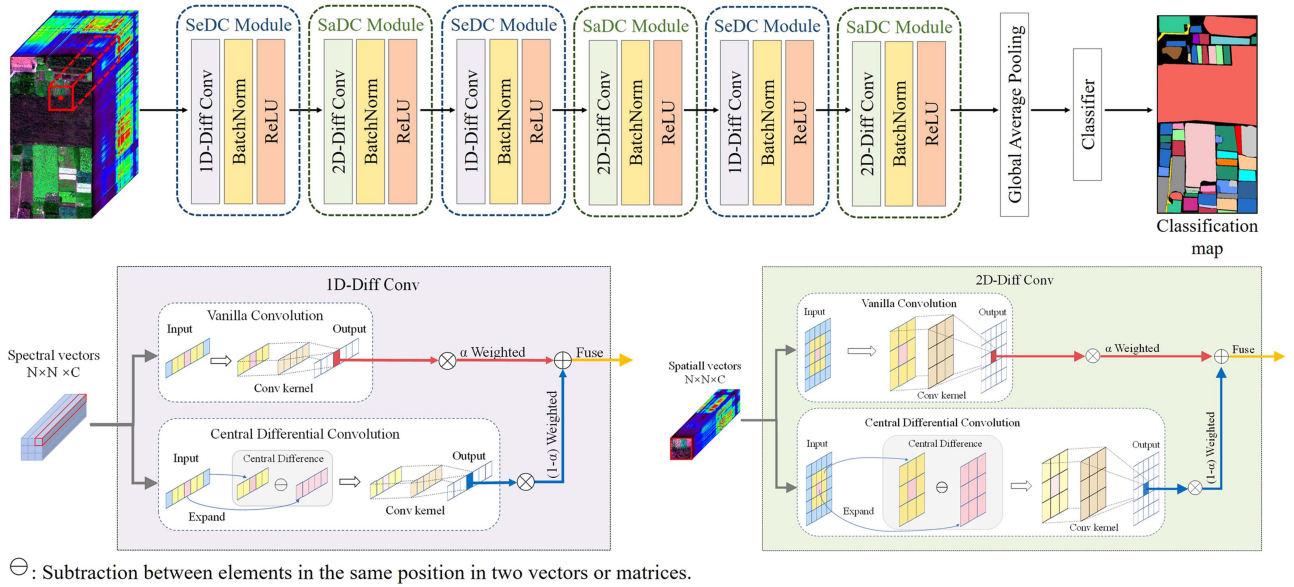


Fig. 1. Diagram of the proposed  $S^2DCN$  for HSI classification.

two symmetrically crossed suboperators, horizontal–vertical and diagonal, substantially reducing the parameters while maintaining comparable performance to the CDiff.

CDiff enhances details related to gradients by introducing the idea of difference operation, which calculates the disparity between local pixels in an image. The extracted spatial difference features are effective in capturing finer grained intrinsic features. Meanwhile, CDiff successfully integrates the interpretable LBP operator into the deep learning framework to improve performance and achieve the learnability of gradient encoding patterns. In addition, the extended versions of CDiff have demonstrated excellent performance in various tasks, such as edge detection [49], video gesture recognition [50], face recognition [51], and object detection [52]. Research results indicate that integrating difference information into deep learning can augment the ability of the model to extract detail image features.

In HSI data, adjacent spatial pixels share certain structural relationships. For instance, buildings typically possess polygonal shapes, while vegetation demonstrates fractal-like appearances. Given the dependency on these spatial structures, incorporating detail features, such as edges, textures, and shapes, can further improve the accuracy and robustness of models [10]. Therefore, we propose the  $S^2DCN$  to enhance classification performance.

### III. PROPOSED METHOD

#### A. Overall Architecture of $S^2DCN$

In this section, an  $S^2DCN$  method is developed to enhance the performance of HSI classification, which emphasizes the extraction of detail features from both spectral and spatial domains. Fig. 1 displays the overall classification workflow,  $S^2DCN$  comprises two feature extraction modules, namely the SeDC module and SaDC module. To elaborate, the one-dimensional difference convolution (1-D-Diff Conv)

layer in the SeDC module helps capture subtle changes in spectral bands, improving the discriminability of spectral features. On the other hand, the two-dimensional difference convolution (2-D-Diff Conv) layer in the SaDC module can fully retain explicit features that exist in the shallow layers of the network. These features are crucial in describing the fine spatial structure of ground objects, such as textures, edges, and shapes, thereby improving the ability of the model to express the discriminative features of ground objects. While both SeDC and SaDC modules employ the difference convolution layer, it is worth noting that this layer effectively fuses features extracted by vanilla convolution and CDiff via adjusting the weight coefficient  $\alpha$ . This fusion approach helps to supplement new detail features in the high-level abstract features extracted by vanilla convolutions, enabling a more precise representation of invariant features within ground objects. After feature extraction is carried out alternately through spectral domain and spatial domain, the extracted features are transported into the global average pooling (GAP) layer, where the feature maps are converted into fixed-length vectors. Finally, the feature vector is fed into the classifier to obtain the final HSI classification result.

#### B. Difference Convolution in SeDC Module

The SeDC module primarily comprises a difference convolution layer, a batch normalization (BN) layer, and a ReLU activation function layer. Among these components, the 1-D-Diff Conv operation within the difference convolution layer is the pivotal step for feature extraction. 1-D-Diff Conv operation excels at capturing subtle variations in spectral bands, enabling the expression of more distinctive spectral characteristics across various types of ground objects, which effectively mitigates the oversmoothing effect of vanilla convolution. The following section outlines the fundamental principle underlying 1-D-Diff Conv in the spectral domain.

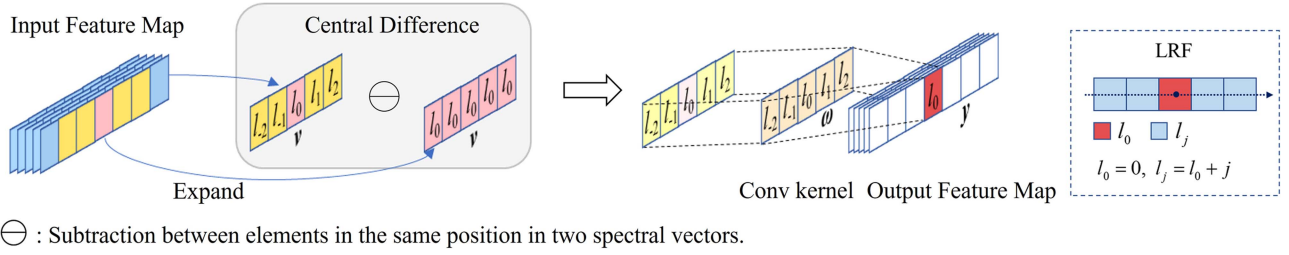


Fig. 2. CDiff in SeDC module. The dashed box on the right displays an example diagram of the LRF position index. Features obtained in the spectral domain after 1-D CDiff considerably preserve subtle variations in the spectral curve, capturing local detail characteristics.

The original HSI data cube is defined as  $\mathcal{I} \in \mathbb{R}^{H \times W \times C}$ , where  $H$  and  $W$  refer to the height and width of HSI data, respectively,  $C$  is the number of spectral bands. We assume that  $x_{h_0, w_0, c_0} \in \mathbb{R}$  represents the pixel located in the  $c_0$ th band at the spatial position  $(h_0, w_0)$ , where  $h_0$  and  $w_0$  represent the height and width index of the image, respectively,  $c_0$  represents the spectral band index, and  $h_0 = 1, 2, \dots, H$ ,  $w_0 = 1, 2, \dots, W$ ,  $c_0 = 1, 2, \dots, C$ . Then, the spectrum vector at this spatial position can be expressed as  $\mathbf{x}_{h_0, w_0, :} \in \mathbb{R}^{1 \times C}$ , which denotes the input data of the SeDC module as a 1-D sequence data of length  $C$ .

The vanilla 1-D convolution consists of two main steps as follows.

- 1) First, it conducts local neighborhood sampling on the pixels within the input data.
- 2) Second, the pixels in the receptive field are aggregated by updating the weight vector  $\omega$  through backpropagation iterations.

For the spectral vectors, the vanilla 1-D convolution can be expressed as follows:

$$\begin{cases} \mathbf{y}(l_0) = \sum_{j \in \mathcal{U}_{se}} \omega(l_j) \cdot \mathbf{v}(l_j) \\ \mathbf{v} = \mathbf{x}_{h_0, w_0, c_0:c_p} \in \mathbb{R}^{1 \times (c_p - c_0)} \end{cases} \quad (1)$$

where  $\mathbf{y}(l_0)$  represents the output feature vector,  $\mathbf{v}$  is the local sampling of the input spectral vector  $\mathbf{x}_{h_0, w_0, :} \in \mathbb{R}^{1 \times C}$  from  $c_0$ th to  $c_p$ th bands, i.e., the local receptive field (LRF). Taking the convolution kernel of  $1 \times 5$  as an example,  $l_0$  represents the center position of the LRF,  $j \in \mathcal{U}_{se} = \{-2, -1, 0, 1, 2\}$  enumerates all offsets within the local receptive field relative to the center position.

As shown in (1), vanilla convolution extracts local features through weighted summation. However, as the number of network layers gradually increases, local detail information progressively aggregated, leading to an oversmoothing effect. To enhance the representation ability of fine-grained spectral features, we introduce the concept of the difference operation into vanilla convolution. Fig. 2 illustrates a schematic diagram of the 1-D CDiff, which applies CDiff operation to the LRF after taking the neighborhood so that it aggregates central gradient information to represent feature details. The CDiff in SeDC module can be written as follows:

$$\mathbf{y}(l_0) = \sum_{j \in \mathcal{U}_{se}} \omega(l_j) \cdot (\mathbf{v}(l_j) - \mathbf{v}(l_0)). \quad (2)$$

The CDiff operation can effectively capture the significant mutation information in the spectral signal while fully retaining key features in the subtle and sensitive bands. However, the extraction of deep abstract features is also crucial for accurately classifying HSI data with rich nonlinear characteristics. Therefore, the combination of vanilla convolution and CDiff becomes necessary to provide more robust feature modeling capabilities. The generalized form of the combined 1-D-Diff Conv operator is as follows:

$$\begin{aligned} \mathbf{y}(l_0) = & \alpha \cdot \underbrace{\sum_{j \in \mathcal{U}_{se}} \omega(l_j) \cdot (\mathbf{v}(l_j) - \mathbf{v}(l_0))}_{\text{central difference convolution}} \\ & + (1 - \alpha) \cdot \underbrace{\sum_{j \in \mathcal{U}_{se}} \omega(l_j) \cdot \mathbf{v}(l_j)}_{\text{vanilla convolution}} \end{aligned} \quad (3)$$

where  $\alpha \in [0, 1]$  represents the weight coefficient used to balance the contribution of high-level abstract features and gradient-level detail information. A larger value of  $\alpha$  places greater importance on central gradient information. When  $\alpha = 0$ , the difference convolution operator degenerates into a vanilla convolution operator, which solely aggregates deep abstract features. When  $\alpha = 1$ , it becomes the CDiff operator, focusing solely on central gradient information. It is worth noting that  $\omega(l_j)$  is shared between vanilla convolution and CDiff.

### C. Difference Convolution in SaDC Module

The SaDC module mainly consist of a 2-D-Diff Conv layer, a BN layer, and a ReLU activation function layer. In this section, the emphasis is on elucidating the operational principles of the difference convolution layer within spatial domain.

We define  $\mathbf{P}_{c_0}(h_0, w_0) \in \mathbb{R}^{N \times N}$  represents an image patch in the  $c_0$ th band, consisting of the set of pixels within the neighborhood of its center point  $(h_0, w_0)$ , with a spatial size of  $N \times N$  as the input to SaDC module. A zero-padding strategy is employed for pixels outside the image boundaries when sampling the neighborhoods. For the vanilla convolution operation of 2-D images, the output feature map  $\mathbf{F}$  can be expressed as

$$\begin{cases} \mathbf{F}(d_{0,0}) = \sum_{(j,k) \in \mathcal{U}_{sa}} \mathbf{W}(d_{j,k}) \cdot \mathbf{Q}(d_{j,k}) \\ \mathbf{Q} = \mathbf{P}_{c_0}(h_0, w_0) \in \mathbb{R}^{N \times N} \end{cases} \quad (4)$$

where  $\mathbf{W}$  represents the convolution kernel,  $\mathbf{Q}$  represents the input patch. Taking the convolution kernel of  $3 \times 3$  as an

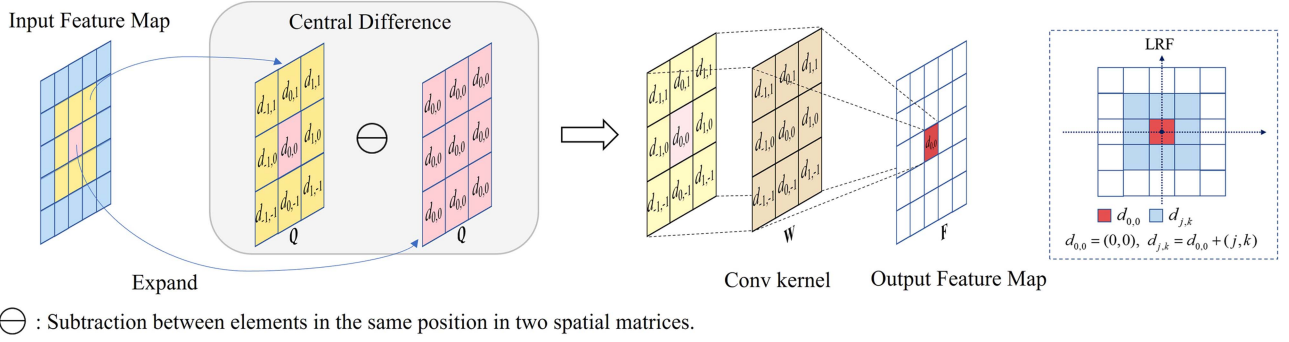


Fig. 3. CDiff in SaDC module. The dashed box on the right displays an example diagram of the LRF position index. Features obtained in the spatial domain after 2-D CDiff can maintain higher feature resolution and preserve fine-structured information that may have been attenuated due to oversmoothing effect.

example,  $d_{0,0}$  represents the central position of the receptive field,  $(j, k) \in \mathbf{U}_{sa} = \{(-1, -1), (-1, 0), \dots, (1, 0), (1, 1)\}$  enumerates the coordinate offsets of all positions in the receptive field with respect to the central position, while  $d_{j,k}$  enumerates all positions in the receptive field. Fig. 3 shows a schematic diagram of the 2-D CDiff operation. The equation is expressed as follows:

$$\mathbf{F}(d_{0,0}) = \sum_{(j,k) \in \mathbf{U}_{sa}} \mathbf{W}(d_{j,k}) \cdot (\mathbf{Q}(d_{j,k}) - \mathbf{Q}(d_{0,0})). \quad (5)$$

The 2-D CDiff operation aims to extract detail gradient-level information from the image while preserving the typical textures, morphologies, and edge features of various ground objects to the maximum extent possible. In image data, the deep intensity-level information (i.e., grayscale information) extracted by vanilla convolution and the fine-structured information obtained through CDiff are vital factors determining classification performance. In the proposed SaDC module, the difference convolution operator combines the features extracted by vanilla convolution and CDiff to enhance the expressive capability of the model, and this can be written as follows:

$$\mathbf{F}(d_{0,0}) = \underbrace{\alpha \cdot \sum_{(j,k) \in \mathbf{U}_{sa}} \mathbf{W}(d_{j,k}) \cdot (\mathbf{Q}(d_{j,k}) - \mathbf{Q}(d_{0,0}))}_{\text{central difference convolution}} + (1 - \alpha) \cdot \underbrace{\sum_{(j,k) \in \mathbf{U}_{sa}} \mathbf{W}(d_{j,k}) \cdot \mathbf{Q}(d_{j,k})}_{\text{vanilla convolution}} \quad (6)$$

where  $\alpha \in [0, 1]$  represents the weight coefficient employed to balance the contribution of high-level abstract features and shallow fine-structured features in the spatial domain.

#### D. Classifier

After feature extraction through the SeDC and SaDC modules from the original HSI data  $\mathcal{Z}$ , the network fuses the obtained spectral and spatial features and feeds them into the GAP layer. Let  $\mathcal{G} \in \mathbb{R}^{H_1 \times W_1 \times C_1}$  represent the input features of the GAP layer, where  $H_1$ ,  $W_1$ , and  $C_1$  correspond to the

height, width and number of channels of the input, respectively. For a pixel  $g_{h_1, w_1, c_1} \in \mathbb{R}$  at position  $(h_1, w_1, c_1)$  in  $\mathcal{G}$ , where  $h_1 = 1, 2, \dots, H_1$ ,  $w_1 = 1, 2, \dots, W_1$ ,  $c_1 = 1, 2, \dots, C_1$ . The GAP operation can be written as follows:

$$\mathbf{g} = \text{GAP}(\mathcal{G}), \quad g_c = \frac{1}{H_1 W_1} \sum_{h_1, w_1} g_{h_1, w_1, c_1} \quad (7)$$

where,  $\mathbf{g} = [g_1, g_2, \dots, g_c, \dots, g_{C_1}] \in \mathbb{R}^{1 \times C_1}$  denotes the feature vector.

Finally, classification phase is performed through linear mapping, and the predicted value  $\hat{\mathbf{y}} = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_T] \in \mathbb{R}^T$  is calculated as follows:

$$\hat{y}_t = \sum_{m=1}^{C_1} \theta_{m,t} g_c + b_t, \quad t = 1, 2, \dots, T \quad (8)$$

where  $T$  represents the class of ground objects,  $\theta_{m,t} \in \mathbb{R}$  is the weight, and  $b_t \in \mathbb{R}$  is the bias.

Let  $\mathbf{Y} = [Y_1, Y_2, \dots, Y_T] \in \mathbb{R}^T$  represent a one-hot encoded vector with  $T$  classes of ground objects, where  $Y_t \in \{0, 1\}$ ,  $t = 1, 2, \dots, T$ . The proposed S<sup>2</sup>DCN model employs the softmax function from (9) to calculate the likelihood probabilities of the output units, and outputs the class with the highest probability as the classification result. The model is trained by minimizing the cross-entropy loss function ( $L_{CE}$ ), as shown in (10)

$$\text{softmax}(\hat{y}_t) = \frac{\exp(\hat{y}_t)}{\sum_{i=1}^T \exp(\hat{y}_i)} \quad (9)$$

$$L_{CE} = - \sum_{t=1}^T Y_t \log(\text{softmax}(\hat{y}_t)). \quad (10)$$

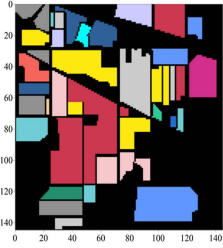
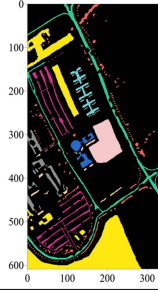
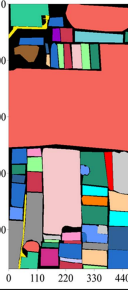
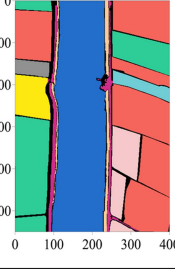
## IV. EXPERIMENTS

### A. HSI Datasets

Four publicly available HSI datasets were considered to verify the proposed S<sup>2</sup>DCN, including the Indian Pines (IP) [53], University of Pavia (UP), WHU-Hi-HongHu (HH), and WHU-Hi-LongKou (LK) [54]. The datasets details are shown in Table I.

1) *Indian Pines*: IP dataset was collected by the airborne visible infrared imaging spectrometer sensor at the IP test site

TABLE I  
GROUNDTRUTH CLASSES FOR THE FOUR HSI DATASETS

Indian Pines (IP)			University of Pavia (UP)			WHU-Hi-HongHu (HH)			WHU-Hi-LongKou (LK)						
															
No.	Color	Category	Samples	Color	Category	Samples	Color	Category	Samples	Color	Category	Samples			
1		Alfalfa	46		Asphalt	6631		Red roof	14041		Corn	34511			
2		Corn-notill	1428		Meadows	18649		Road	3512		Cotton	8374			
3		Corn-min	830		Gravel	2099		Bare soil	21821		Sesame	3031			
4		Corn	237		Trees	3064		Cotton	163285		Broad-leaf soybean	63212			
5		Grass/Pasture	483		Painted metal sheets	1345		Cotton firewood	6218		Narrow-leaf soybean	4151			
6		Grass/Trees	730		Bare soil	5029		Rape	44557		Rice	11854			
7		Grass/pasture-mowed	28		Bitumen	1330		Chinese cabbage	24103		Water	67056			
8		Hay-windrowed	478		Self-Blocking Bricks	3682		Pakchoi	4054		Roads and houses	7124			
9		Oats	20		Shadows	947		Cabbage	10819		Mixed weed	5229			
10		Soybean-notill	972					Tuber mustard	12394						
11		Soybean-min	2455					Brassica parachinensis	11015						
12		Soybean-clean	593					Brassica chinensis	8954						
13		Wheat	205					Small Brassica chinensis	22507						
14		Woods	1265					Lactuca sativa	7356						
15		Buidings-Grass-Tree-Drives	386					Celtuce	1002						
16		Stone-steel towers	93					Film covered lettuce	7262						
17								Romaine lettuce	3010						
18								Carrot	3217						
19								White radish	8712						
20								Garlic sprout	3486						
21								Broad bean	1328						
22								Tree	4040						
Total			10249	Total			42776	Total			386693	Total			204542

in the northwest of Indiana, USA, recording 20 m per pixel and consisting of  $145 \times 145$  pixels. The dataset provides 224 bands, which are usually reduced to 200 by removing water absorption and noisy bands, with a sensitivity in the wavelength range of 400–250 nm. The ground truth comprises 16 classes representing different crops and vegetation types, with a total of 10 249 samples.

2) *University of Pavia*: UP dataset was captured by the reflective optics spectrographic imaging system (ROSIS) sensor near the UP in Italy, with a spatial resolution of 1.3 m. This dataset comprises 115 spectral bands, ranging from 430 to 860 nm, and has a total size of  $610 \times 340$  pixels. It contains 103 valid bands after eliminating the 12 bands influenced by atmospheric absorption and noise. There are nine land-cover classes in UP, with a total of 42 776 samples.

3) *WHU-Hi-HongHu*: HH dataset was acquired in Honghu City, Hubei Province, China, using a 17-mm focal length Headwall Nanohyperspec imaging sensor equipped on a DJI Matrice 600 Pro UAV platform. There are 22 classes in HH dataset, totaling 386 693 samples, showcasing a diverse agricultural scene with numerous crop classes. It also features various cultivars of the same crop type, such as Chinese cabbage and cabbage, as well as Brassica chinensis and small Brassica chinensis. The imagery dimensions measure  $940 \times 475$  pixels, encompassing

270 bands spanning from 400 to 1000 nm, and the spatial resolution is about 0.043 m.

4) *WHU-Hi-LongKou*: LK dataset was obtained using an 8-mm focal length Headwall Nanohyperspec image sensor in Longkou Town, Hubei province, China. This scene is a simple agricultural scene with nine classes, totaling 204 542 samples. The imagery size is  $550 \times 400$  pixels, with a spatial resolution of about 0.463 m, and there are 270 bands from 400 to 1000 nm.

## B. Experimental Settings

1) *Implementation Details*: The proposed S<sup>2</sup>DCN was implemented on the PyTorch platform using a workstation with Inter Core i7-11700 K CPU, 128 G RAM, and an NVIDIA GeForce RTX3090 24 GB GPU. The AdamW optimizer was adopted with a batch size of 128. The number of training epochs was set to 300, and the learning rate was 0.0005. More details of the proposed S<sup>2</sup>DCN with base parameter settings are listed in Table II.

2) *Evaluation Metrics and Comparison Models*: Different metrics have been considered for the the evaluation of results, namely, overall accuracy (OA), average accuracy (AA), kappa coefficient ( $\kappa \times 100$ ), and the number of model parameters

TABLE II  
DETAILS OF THE PROPOSED S<sup>2</sup>DCN

Module	Layer setting	Input size	Kernel size	Output size
SeDC_1	1-D-Diff Conv	15 × 15 × C	7, strid = 1	15 × 15 × 24
	BatchNorm	15 × 15 × 24	–	15 × 15 × 24
	ReLU	15 × 15 × 24	–	15 × 15 × 24
SaDC_1	2-D-Diff Conv	15 × 15 × 24	3 × 3, strid = [1,1]	15 × 15 × 24
	BatchNorm	15 × 15 × 24	–	15 × 15 × 24
	ReLU	15 × 15 × 24	–	15 × 15 × 24
SeDC_2	1-D-Diff Conv	15 × 15 × 24	7, strid = 1	15 × 15 × 24
	BatchNorm	15 × 15 × 24	–	15 × 15 × 24
	ReLU	15 × 15 × 24	–	15 × 15 × 24
SaDC_2	2-D-Diff Conv	15 × 15 × 24	3 × 3, strid = [1,1]	15 × 15 × 24
	BatchNorm	15 × 15 × 24	–	15 × 15 × 24
	ReLU	15 × 15 × 24	–	15 × 15 × 24
SeDC_3	1-D-Diff Conv	15 × 15 × 24	7, strid = 1	15 × 15 × 24
	BatchNorm	15 × 15 × 24	–	15 × 15 × 24
	ReLU	15 × 15 × 24	–	15 × 15 × 24
SaDC_3	2-D-Diff Conv	15 × 15 × 24	3 × 3, strid = [1,1]	15 × 15 × 128
	BatchNorm	15 × 15 × 128	–	15 × 15 × 128
	ReLU	15 × 15 × 128	–	15 × 15 × 128
GAP	Global Average Pooling	15 × 15 × 128	15 × 15, strid = [15,15]	128
Classifier	Linear projection	128	–	T
	softmax	T	–	T

\* C and T are the numbers of input channels and land-cover classes, respectively.

TABLE III  
CLASSIFICATION RESULTS OF COMPARISON METHODS ON IP DATASET AT 3% TRAINING RATIO

No.	ResNet	ViT	Mixer	NesT	Cycle	PiDiNet	S <sup>2</sup> DCN
1	98.15±2.15	<b>100.0±0.00</b>	98.24±1.44	<b>100.0±0.00</b>	<b>100.0±0.00</b>	<b>100.0±0.00</b>	<b>100.0±0.00</b>
2	95.86±0.37	91.09±0.40	90.27±0.44	91.81±0.36	91.38±0.31	<b>95.92±1.13</b>	95.43±0.34
3	79.78±1.25	81.50±0.65	92.93±0.60	84.86±0.52	89.01±0.36	86.34±1.75	<b>95.58±0.40</b>
4	83.43±1.86	92.38±1.52	87.47±0.94	95.07±0.78	88.82±1.33	<b>96.36±1.66</b>	95.62±0.81
5	90.69±0.92	88.25±0.85	85.77±1.00	92.22±0.96	88.91±0.91	91.51±2.05	<b>94.15±0.71</b>
6	98.34±0.42	97.56±0.26	96.24±0.35	96.28±0.50	94.47±0.47	95.94±1.16	<b>98.64±0.34</b>
7	<b>100.0±0.00</b>	<b>100.0±0.00</b>	<b>100.0±0.00</b>	<b>100.0±0.00</b>	<b>100.0±0.00</b>	<b>100.0±0.00</b>	<b>100.0±0.00</b>
8	99.97±0.08	<b>100.0±0.00</b>	98.41±0.43	<b>100.0±0.00</b>	<b>100.0±0.00</b>	99.50±0.54	<b>100.0±0.00</b>
9	99.38±1.98	<b>100.0±0.00</b>	<b>100.0±0.00</b>	<b>100.0±0.00</b>	<b>100.0±0.00</b>	<b>100.0±0.00</b>	<b>100.0±0.00</b>
10	93.16±0.76	89.04±0.46	94.57±0.45	87.81±0.76	91.07±0.65	89.15±2.43	<b>97.48±0.24</b>
11	95.39±0.28	91.92±0.15	95.08±0.20	92.32±0.28	94.52±0.31	95.58±0.65	<b>98.75±0.08</b>
12	69.62±1.22	72.31±0.80	77.23±0.83	71.33±0.47	81.84±0.57	<b>84.76±3.16</b>	83.53±0.58
13	89.88±1.59	87.76±1.39	95.18±0.69	97.80±0.49	<b>100.0±0.00</b>	93.98±3.16	95.45±0.72
14	97.42±0.36	97.92±0.28	92.39±0.26	96.96±0.08	94.82±0.12	97.40±0.38	<b>99.60±0.08</b>
15	92.48±0.71	82.55±0.98	85.93±0.89	82.28±1.05	82.98±1.26	<b>94.47±1.61</b>	88.96±1.16
16	99.18±0.96	96.58±0.66	<b>100.0±0.00</b>	94.30±1.19	98.57±0.04	89.24±5.92	96.58±0.66
OA (%)	92.53±0.13	90.59±0.06	92.18±0.07	91.09±0.10	92.23±0.11	93.80±0.36	<b>96.45±0.09</b>
AA (%)	92.67±0.24	91.80±0.13	93.11±0.10	92.69±0.10	93.52±0.07	94.38±0.40	<b>96.24±0.10</b>
$\kappa \times 100$	91.48±0.15	89.26±0.07	91.10±0.08	89.84±0.12	91.14±0.13	92.94±0.40	<b>95.95±0.10</b>
Params (M)	11.29	0.68	26.21	40.34	2.74	4.06	<b>0.33</b>

The best results are shown in bold.

(Params). To alleviate the randomness in the experimental results, each dataset was subjected to ten repetitions of the experiments, utilizing the mean and standard deviation of the first three evaluation metrics for quantitative analysis. The six comparison methods based on deep learning are ResNet [55], ViT [17], Mixer [19], nested transformer (NesT) [56], CycleMLP (Cycle) [57], and pixel difference network (PiDiNet) [58].

### C. Comparison With State-of-the-Art Methods

In comparative experiments, the input patch size is set to 15×15, and 3% of samples are randomly selected from each class for training. Tables III–VI display the evaluation metrics

and class-specific accuracies of the proposed S<sup>2</sup>DCN and the comparative methods on four datasets. The results highlighted in bold in the tables represent the best classification performance. It is evident that S<sup>2</sup>DCN outperforms other SOTA deep learning methods in terms of OA, AA,  $\kappa \times 100$ , and Params. Figs. 4–7 depict the classification maps of all the methods for visualization.

Table III shows the classification results of all experimental methods on IP dataset. S<sup>2</sup>DCN has the best classification performance in 10 out of 16 classes and achieves the highest OA of 96.45%, demonstrating significant advantages. The PiDiNet exhibits suboptimal performance with an OA of 93.80%, which utilizes the pixel difference convolution (PDC) operation to



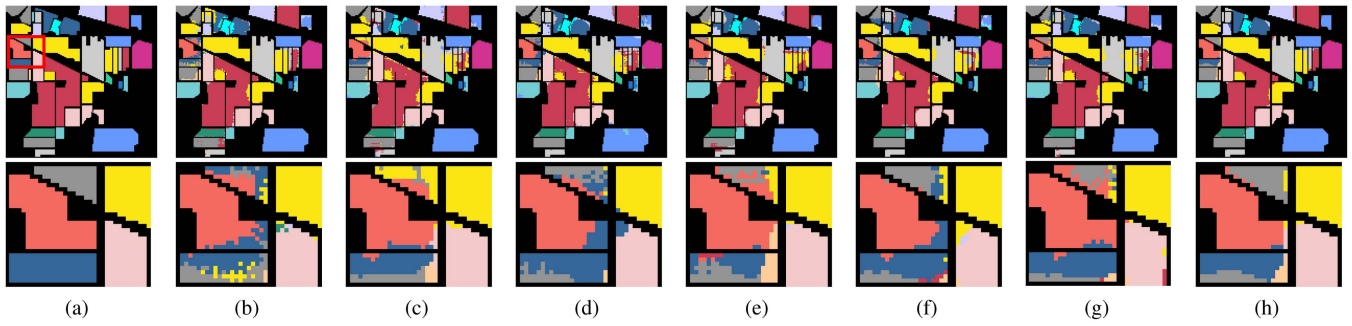


Fig. 4. Classification maps obtained by the different methods for the IP dataset at 3% training ratio. The overall classification accuracies are given in the parentheses. (a) Ground truth. (b) ResNet (92.73%). (c) ViT (90.63%). (d) Mixer (92.23%). (e) NesT (91.19%). (f) Cycle (92.28%). (g) PiDiNet (93.88%). (h) S<sup>2</sup>DCN (96.52%).

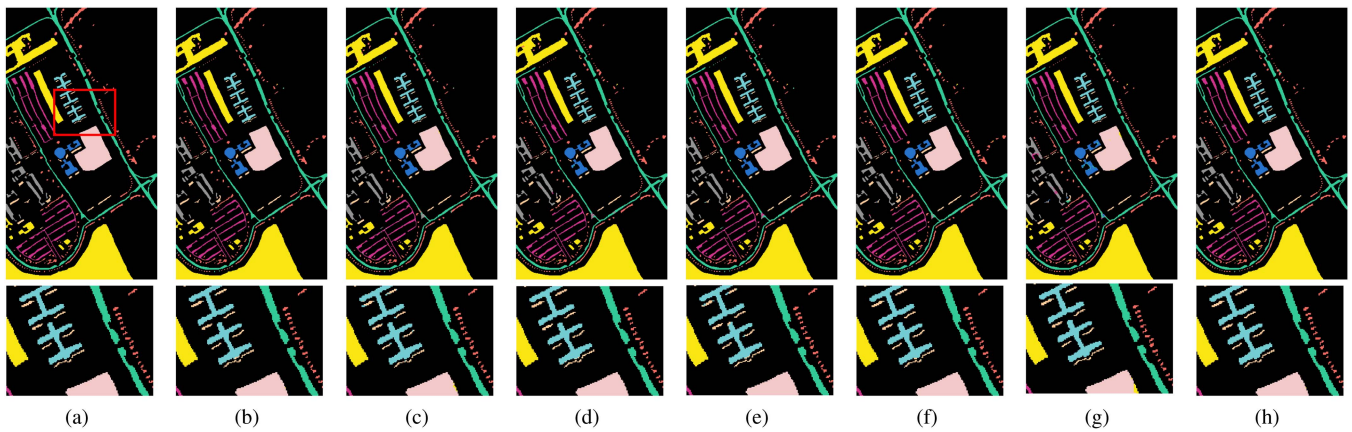


Fig. 5. Classification maps obtained by the different methods for the UP dataset at 3% training ratio. The overall classification accuracies are given in the parentheses. (a) Ground truth. (b) ResNet (99.70%). (c) ViT (98.91%). (d) Mixer (99.40%). (e) NesT (99.20%). (f) Cycle (99.43%). (g) PiDiNet (98.64%). (h) S<sup>2</sup>DCN (99.72%).

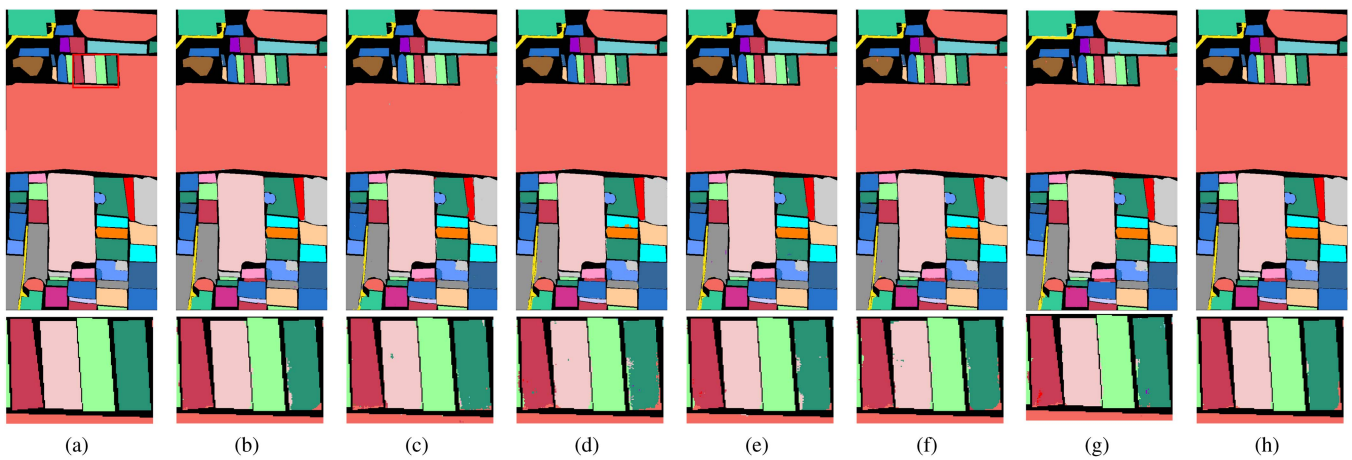


Fig. 6. Classification maps obtained by the different methods for the HH dataset at 3% training ratio. The overall classification accuracies are given in the parentheses. (a) Ground truth. (b) ResNet (99.48%). (c) ViT (99.32%). (d) Mixer (99.15%). (e) NesT (99.40%). (f) Cycle (99.10%). (g) PiDiNet (99.30%). (h) S<sup>2</sup>DCN (99.65%).

TABLE IV  
CLASSIFICATION RESULTS OF COMPARISON METHODS ON UP DATASET AT 3% TRAINING RATIO

No.	ResNet	ViT	Mixer	NesT	Cycle	PiDiNet	S <sup>2</sup> DCN
1	<b>99.96±0.12</b>	98.53±0.34	98.70±0.36	99.09±0.23	99.24±0.29	98.18±0.31	99.88±0.21
2	<b>99.98±0.03</b>	99.81±0.06	99.86±0.05	99.95±0.04	99.93±0.06	99.93±0.02	<b>99.98±0.03</b>
3	99.63±0.24	98.51±1.89	97.71±0.71	99.40±0.60	98.64±0.36	93.61±0.82	<b>99.70±0.26</b>
4	97.80±0.62	96.11±0.46	<b>98.42±0.26</b>	94.80±1.22	96.86±0.95	97.44±0.47	98.25±0.64
5	99.54±0.29	99.58±0.69	99.64±0.34	99.01±0.62	99.65±0.23	<b>100.0±0.00</b>	99.97±0.09
6	<b>99.99±0.02</b>	99.74±0.13	<b>99.99±0.02</b>	<b>99.99±0.02</b>	99.95±0.15	99.00±0.22	<b>99.99±0.02</b>
7	<b>99.99±0.04</b>	99.83±0.36	99.25±0.58	99.96±0.13	99.80±0.63	94.90±0.90	99.96±0.13
8	<b>99.39±0.19</b>	97.67±0.45	<b>99.39±0.39</b>	98.99±0.39	99.59±0.19	98.63±0.43	99.14±0.27
9	97.54±0.64	91.20±2.55	<b>98.43±0.42</b>	93.40±0.92	95.66±0.90	92.35±1.27	97.79±0.73
OA (%)	99.69±0.05	98.90±0.08	99.39±0.04	99.18±0.06	99.41±0.05	98.63±0.11	<b>99.71±0.04</b>
AA (%)	99.31±0.13	97.89±0.21	99.04±0.10	98.29±0.12	98.81±0.14	97.12±0.22	<b>99.41±0.12</b>
$\kappa \times 100$	99.59±0.07	98.54±0.10	99.19±0.05	98.91±0.08	99.22±0.07	98.18±0.14	<b>99.61±0.05</b>
Params (M)	11.23	0.65	25.76	39.89	2.71	3.75	<b>0.18</b>

The best results are shown in bold.

TABLE V  
CLASSIFICATION RESULTS OF COMPARISON METHODS ON HH DATASET AT 3% TRAINING RATIO

No.	ResNet	ViT	Mixer	NesT	Cycle	PiDiNet	S <sup>2</sup> DCN
1	99.11±0.19	98.96±0.14	98.90±0.16	99.04±0.18	98.85±0.18	98.96±0.10	<b>99.50±0.12</b>
2	97.25±2.62	92.04±1.07	94.34±3.82	95.39±2.07	93.19±1.12	96.31±0.30	<b>97.84±0.76</b>
3	98.34±0.48	<b>99.17±0.15</b>	97.33±0.74	98.60±0.34	98.67±0.18	99.06±0.07	99.01±0.26
4	99.92±0.01	99.83±0.04	99.91±0.04	99.94±0.01	99.88±0.03	99.92±0.01	<b>99.99±0.02</b>
5	98.61±0.27	99.11±0.89	98.00±0.21	98.83±0.18	98.24±0.34	97.83±0.32	<b>99.57±0.18</b>
6	99.87±0.14	99.80±0.06	99.79±0.11	99.85±0.12	99.79±0.07	99.85±0.03	<b>99.92±0.09</b>
7	99.23±0.09	98.96±0.08	99.17±0.12	<b>99.40±0.13</b>	98.63±0.14	98.98±0.11	99.30±0.16
8	98.77±0.28	98.24±0.47	98.67±0.23	99.01±0.36	96.43±0.76	96.40±0.41	<b>99.32±0.28</b>
9	99.76±0.10	99.23±0.19	99.67±0.14	99.81±0.06	99.78±0.31	99.78±0.07	<b>99.91±0.04</b>
10	99.22±0.11	99.20±0.24	98.01±0.53	<b>99.43±0.27</b>	98.47±0.18	98.42±0.21	98.76±0.38
11	98.98±0.16	98.83±0.13	98.32±0.19	98.74±0.23	98.16±0.26	98.97±0.14	<b>99.42±0.17</b>
12	99.12±0.36	98.90±0.18	98.46±0.11	98.83±0.18	97.77±0.13	97.84±0.30	<b>99.10±0.17</b>
13	98.67±0.32	97.96±0.44	98.14±0.29	98.39±0.0	98.44±0.38	98.08±0.14	<b>99.07±0.12</b>
14	<b>99.49±0.25</b>	99.41±0.49	98.47±0.29	98.61±0.31	97.88±0.27	99.18±0.14	99.10±0.21
15	96.63±0.77	97.98±1.04	94.87±1.20	97.82±0.78	92.10±2.26	99.07±0.84	<b>98.07±0.64</b>
16	<b>99.84±0.13</b>	99.17±0.13	97.28±0.74	99.64±0.11	98.40±0.36	99.70±0.06	99.62±0.14
17	98.86±0.60	99.02±0.41	98.29±1.47	<b>99.10±0.46</b>	96.38±0.63	99.29±0.18	98.97±0.34
18	98.10±0.51	94.69±0.78	97.54±0.38	97.96±0.88	95.19±0.94	96.96±0.55	<b>98.87±0.55</b>
19	98.51±0.28	98.87±0.15	98.32±0.23	99.07±0.18	97.60±0.44	97.67±0.24	<b>99.48±0.16</b>
20	<b>99.49±0.54</b>	98.83±0.36	98.97±0.39	98.62±0.36	98.89±0.49	97.91±0.63	99.43±0.32
21	98.52±0.65	98.33±0.71	95.18±1.82	<b>99.10±0.66</b>	95.45±1.47	96.22±0.74	97.78±0.99
22	99.30±0.22	99.61±0.24	99.37±0.33	99.66±0.15	98.99±0.22	99.04±0.30	<b>99.81±0.21</b>
OA (%)	99.46±0.02	99.30±0.03	99.13±0.03	99.45±0.02	99.09±0.03	99.29±0.02	<b>99.63±0.01</b>
AA (%)	98.89±0.10	98.46±0.11	98.05±0.16	98.86±0.12	97.60±0.18	98.43±0.10	<b>99.17±0.05</b>
$\kappa \times 100$	99.32±0.03	99.11±0.03	98.91±0.04	99.30±0.03	98.86±0.03	99.10±0.03	<b>99.53±0.02</b>
Params (M)	11.33	0.70	26.54	40.66	2.76	4.28	<b>0.43</b>

The best results are shown in bold.

capture gradient information. In the process of PDC operation, the original pixels in the local feature map patch covered by the convolution kernels are replaced by pixel differences. Although the addition of rich gradient information makes the performance of PiDiNet better, our method combines vanilla convolution with difference convolution to extract more comprehensive features, so it is obviously superior to it. The worst performing method is ViT, which requires a larger amount of training data. It is prone to overfitting and generalization ability decline in the case of limited training samples. The classification maps for comparative methods on IP dataset are shown in Fig. 4. The classification results of S<sup>2</sup>DCN show more explicit boundaries than other competitors, with fewer occurrences of the salt-and-pepper phenomenon, thus conforming to the actual distribution of land covers. In the enlarged part displayed in the red box [see Fig. 4(a)], the classification performance of classes no.3 (*corn-min*), no.4 (*corn*), and no.12 (*soybean-clean*) is unsatisfactory among comparative methods, particularly at the

boundaries where severe misclassification occurs. In contrast, the classification map generated by S<sup>2</sup>DCN demonstrates superior intraclass spatial consistency. For the class with the largest number of samples, the no.11 (*soybean-min*) class, other models exhibit more pixels misclassified as the class no.2 (*corn-notill*) at the boundaries, showing a salt-and-pepper phenomenon. While S<sup>2</sup>DCN has the least misclassified pixels, indicating that the difference convolution operation effectively preserves edge details.

The ground objects in the UP dataset display a narrow strip-like spatial distribution pattern, such as classes no.1 (*asphalt*), no.5 (*painted metal sheets*), and no.8 (*self-blocking bricks*). There are noise pixels that can be introduced and result in misclassification when extracting morphological features from these classes. Table IV shows the classification results of experimental methods on the UP dataset. All metrics of S<sup>2</sup>DCN are the best among all contrasting methods, which shows that it performs well in classifying ground objects with apparent spatial morphological features. The robustness of S<sup>2</sup>DCN is

TABLE VI  
CLASSIFICATION RESULTS OF COMPARISON METHODS ON LK DATASET AT 3% TRAINING RATIO

No.	ResNet	ViT	Mixer	NesT	Cycle	PiDiNet	S <sup>2</sup> DCN
1	99.90±0.02	99.80±0.03	99.96±0.02	<b>99.98±0.06</b>	99.96±0.03	99.84±0.03	99.96±0.02
2	99.72±0.04	99.47±0.19	99.73±0.08	99.88±0.06	99.90±0.27	99.51±0.15	<b>99.91±0.06</b>
3	<b>99.95±0.04</b>	99.32±1.15	98.36±0.63	99.05±0.35	98.43±0.57	98.51±0.31	99.33±0.26
4	99.79±0.03	99.53±0.15	99.45±0.10	99.66±0.08	99.50±0.15	99.60±0.05	<b>99.85±0.04</b>
5	<b>99.87±0.05</b>	99.64±0.43	97.71±1.39	99.60±0.92	99.29±1.65	98.94±0.29	99.46±0.34
6	99.63±0.06	99.78±0.10	99.62±0.07	99.62±0.09	99.94±0.06	99.42±0.15	<b>99.97±0.02</b>
7	99.94±0.01	99.74±0.09	99.94±0.01	99.95±0.01	<b>99.97±0.03</b>	99.97±0.01	99.96±0.02
8	97.16±0.16	97.29±0.42	97.09±0.21	97.48±0.65	97.84±0.41	97.39±0.25	<b>98.07±0.31</b>
9	95.82±0.13	96.73±0.93	97.20±0.36	94.97±1.04	97.26±0.39	<b>98.03±0.26</b>	97.72±0.37
OA (%)	99.66±0.02	99.51±0.04	99.53±0.02	99.61±0.02	99.64±0.01	99.60±0.02	<b>99.78±0.02</b>
AA (%)	99.09±0.04	99.03±0.29	98.78±0.11	98.91±0.08	99.12±0.13	99.02±0.06	<b>99.36±0.06</b>
$\kappa \times 100$	99.55±0.02	99.35±0.05	99.38±0.03	99.48±0.02	99.52±0.02	99.48±0.02	<b>99.71±0.02</b>
Params (M)	11.33	0.70	26.53	40.66	2.76	4.27	<b>0.43</b>

The best results are shown in bold.

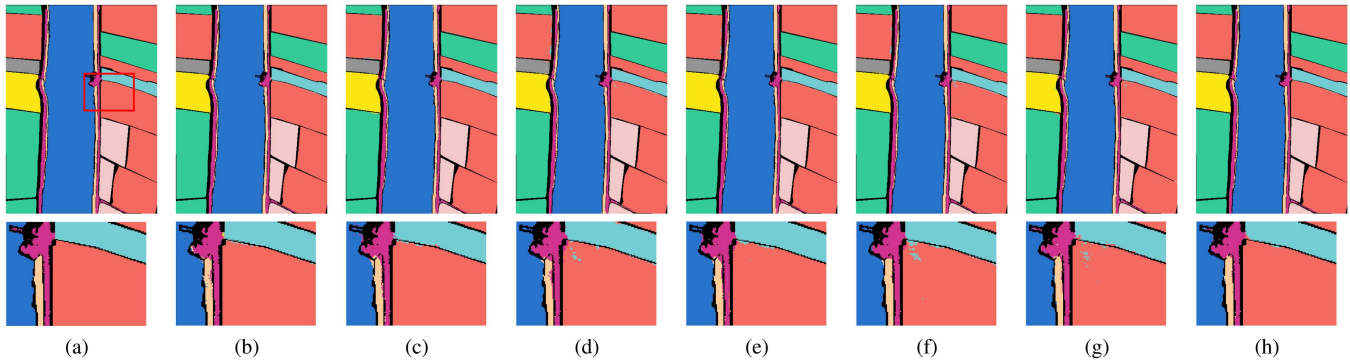


Fig. 7. Classification maps obtained by the different methods for the LK dataset at 3% training ratio. The overall classification accuracies are given in the parentheses. (a) Ground truth. (b) ResNet (99.68%). (c) ViT (99.52%). (d) Mixer (99.54%). (e) NesT (99.62%). (f) Cycle (99.65%). (g) PiDiNet (99.61%). (h) S<sup>2</sup>DCN (99.79%).

also excellent in terms of standard deviation. Fig. 5 shows the visualization results of all methods on the UP dataset. In the enlarged region displayed in the red box [see Fig. 5(a)], classes no.5 and no.6 are easily misclassified as each other. While in Fig. 5(h), S<sup>2</sup>DCN provides the highest consistency with the actual distribution, exhibiting the fewest misclassified pixels.

The HH and LK datasets are typical agricultural scenes acquired from different agricultural areas with various crop types in Hubei Province, China. These two datasets possess high spatial resolution and finer class division. Table V provides the obtained results for the HH dataset. Following the same pattern of classification enhancement this trend. The proposed S<sup>2</sup>DCN reaches the best OA with 99.63%, also providing the highest values for AA and  $\kappa \times 100$ . The standard deviation indicates the stability of the model. Fig. 6 shows that S<sup>2</sup>DCN is excel at classifying elongated block areas. In the enlarged area highlighted in the red box [see Fig. 6(a)], the classification maps of the comparative methods often exhibit many misclassified pixels existing along the boundaries. In contrast, the classification map of S<sup>2</sup>DCN shows the fewest misclassified pixels, indicating that the proposed S<sup>2</sup>DCN can preserve spatial texture and edge information, significantly improving the classification accuracy.

Table VI shows the obtained classification results on the LK dataset. The classification maps are presented in Fig. 7. As shown in the enlarged parts, S<sup>2</sup>DCN demonstrates advantages in classes no.8 (*roads and houses*) and no.9 (*mixed weed*), which

are narrow-shaped ground objects, as well as at the boundary between classes no.4 (*broad-leaf soybean*) and no.5 (*narrow-leaf soybean*). The classification map of S<sup>2</sup>DCN achieves impressive visual effects in maintaining the correctness of the boundary area and spatial consistency. It can be concluded that the proposed S<sup>2</sup>DCN excels in extracting distinctive features of ground objects with various spatial distribution patterns, demonstrating its strong robustness.

To validate the effectiveness of feature extraction capability, the t-distributed stochastic neighbor embedding (t-SNE) approach [59] is employed to visualize the features extracted from the final layer of the different networks on the IP and LK datasets, as illustrated in Figs. 8 and 9. Each dot represents one pixel, and pixels with the same label share the same color. The tight clustering of intraclass samples and the wide separation of interclass samples indicate the superior feature extraction and representation capabilities of the proposed S<sup>2</sup>DCN. It can be observed that samples of the same categories gather together, and intraclass variance is minimized in Figs. 8(f) and 9(f), indicating the powerful feature extraction capabilities of the proposed S<sup>2</sup>DCN.

#### D. Performance Under Different Training Ratio

To verify the generalization capability of S<sup>2</sup>DCN, different numbers of training samples are randomly selected at training

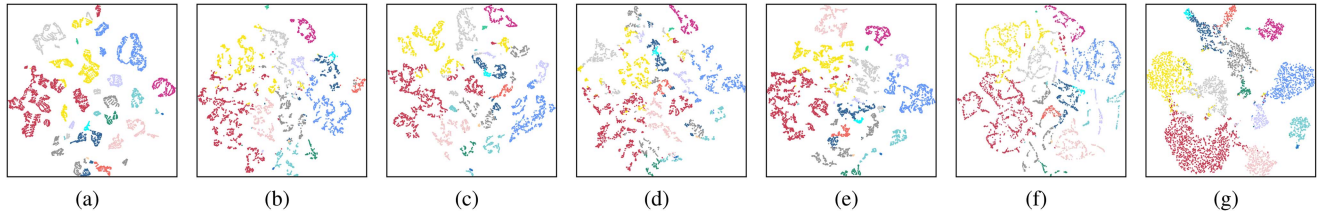


Fig. 8. Feature distribution visualization by t-SNE for IP dataset. (a) ResNet. (b) ViT. (c) Mixer. (d) NesT. (e) Cycle. (f) PiDiNet. (g) S<sup>2</sup>DCN.

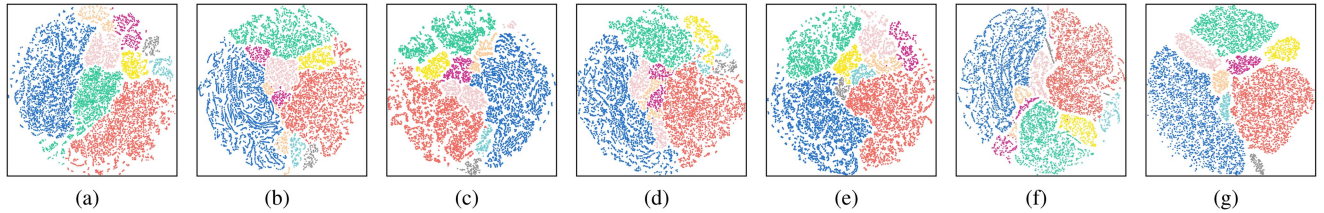


Fig. 9. Feature distribution visualization by t-SNE for LK dataset. (a) ResNet. (b) ViT. (c) Mixer. (d) NesT. (e) Cycle. (f) PiDiNet. (g) S<sup>2</sup>DCN.

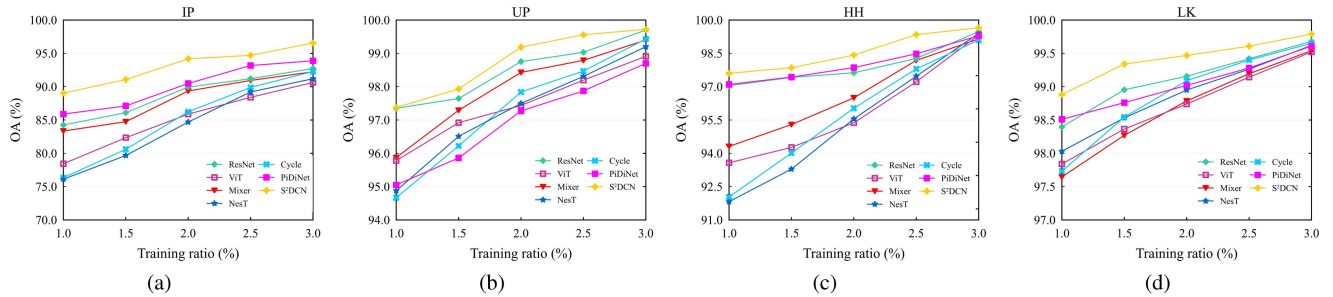


Fig. 10. Classification results with  $15 \times 15$  patch size and different training ratio. (a) IP. (b) UP. (c) HH. (d) LK.

TABLE VII

CLASSIFICATION RESULTS OF COMPARISON METHODS ON UP, HH, AND LK DATASETS IN THE FEW-SHOT SCENARIO. (THE BEST RESULTS ARE SHOWN IN BOLD)

Dataset		ResNet	ViT	Mixer	NesT	Cycle	PiDiNet	S <sup>2</sup> DCN
UP	OA (%)	86.87	85.38	86.51	87.14	88.03	81.86	<b>90.95</b>
	AA (%)	93.07	89.01	91.20	90.77	92.49	85.37	<b>93.87</b>
	$\kappa \times 100$	83.24	81.23	82.72	83.47	84.61	76.81	<b>88.30</b>
HH	OA (%)	87.87	88.66	89.47	89.50	88.65	88.34	<b>91.02</b>
	AA (%)	84.50	86.97	89.19	88.22	86.46	86.32	<b>91.47</b>
	$\kappa \times 100$	84.78	85.83	86.89	86.87	85.78	85.48	<b>88.77</b>
LK	OA (%)	96.52	96.61	96.97	97.05	96.63	96.63	<b>97.67</b>
	AA (%)	96.14	94.25	95.28	95.15	94.53	91.97	<b>96.32</b>
	$\kappa \times 100$	95.46	95.57	96.04	96.15	95.59	92.39	<b>96.95</b>

The best results are shown in bold.

ratios of 1%, 1.5%, 2%, 2.5%, and 3%. The OA results in Fig. 10 show that the proposed S<sup>2</sup>DCN compares favorably with other methods and allow two important observations to be highlighted: 1) the proposed method achieves the optimal OA values over four datasets for all experiment configurations, even with small training ratios; 2) S<sup>2</sup>DCN exhibits the least sensitivity to the training ratio, even at a training ratio of 1%, it can still learn valuable detail features and deliver satisfactory results. In contrast, the comparative methods are sensitive to the training ratio, and there is a noticeable inflection point where the network's classification performance improves significantly as

the number of training samples increases. The above-mentioned phenomenon indicates that the classification performance and generalization capability of S<sup>2</sup>DCN is superior with a small number of samples.

### E. Performance in the Few-Shot Scenario

To further verify the performance of the proposed S<sup>2</sup>DCN, we randomly select ten samples per class as training sets for few-shot classification on UP, HH, and LK datasets. The experimental results in Table VII demonstrate that the proposed

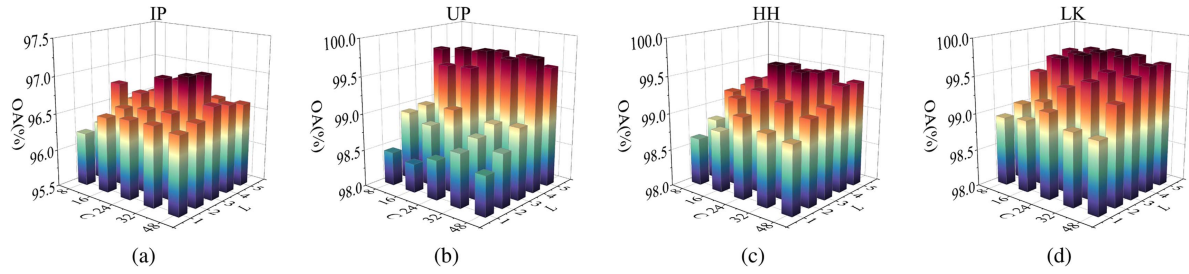


Fig. 11. Performance of the S<sup>2</sup>DCN with different  $C$  and  $L$  on the four HSI datasets. (a) IP. (b) UP. (c) HH. (d) LK.

S<sup>2</sup>DCN outperforms other methods in the few-shot classification, as it can still learn valuable detailed information even with a limited number of training samples. However, the prediction accuracy of the compared methods is generally unsatisfactory. This phenomenon indicates that S<sup>2</sup>DCN can achieve better classification performance even in the few-shot scenario.

### F. Hyperparameter Sensitivity Analysis

The hyperparameter settings in deep learning models are crucial for achieving excellent classification performance. We select the hyperparameters that significantly impact the model training process and classification performance to determine the suitable settings for S<sup>2</sup>DCN.

1) *Depth and Width of S<sup>2</sup>DCN*: In terms of network architecture, the selected hyperparameters are the number of difference convolution modules ( $L$ ) and the number of channels in a difference convolution module ( $C$ ), which denote the depth and width of S<sup>2</sup>DCN. Notably, the difference convolution modules here refer to the SeDC and SaDC modules collectively. The network depth affects the expressive power of difference features, while the network width influences the effectiveness of spectral–spatial feature extraction.

Specifically, the hyperparameters of the network architecture are validated at the 3% training ratio on four datasets, as shown in Fig. 11. Increasing  $L$  can significantly improve OA, suggesting that increasing network depth enhances feature representation and learning capabilities, which is sufficient for extracting non-linear abstract features. The performance of the model gradually improves until it reaches the peak, after which it tends to decline. This decline may be attributed to degradation problems, such as gradient instability caused by too deep a network. When  $C$  increases, the classification performance of the model generally exhibits a trend of initially increasing and then decreasing. There is often a tradeoff between classification performance and model complexity. Considering all factors, the hyperparameter  $L$  is set to 3 and  $C$  is set to 24 for subsequent experiments.

2) *Weight Coefficients in the SeDC and SaDC Modules*: The weight coefficient  $\alpha$  is set to 0.6 in the SaDC module when verifying the impact of  $\alpha$  in the SeDC module. According to (3), when  $\alpha$  is set to 0, the difference convolution degenerates into the vanilla convolution; when  $\alpha$  is set to 1, it becomes the CDiff. We investigate the influence of the contribution levels from CDiff and vanilla convolution on classification performance to determine the optimal values of  $\alpha$  in four datasets.

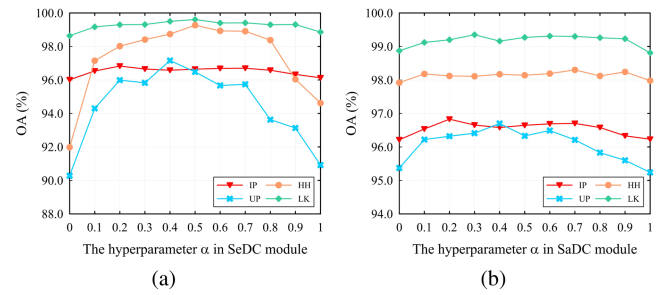


Fig. 12. Influence of the  $\alpha$  on classification performance (a) SeDC module. (b) SaDC module.

Fig. 12(a) shows the influence of the weight coefficient  $\alpha$  in the SeDC module. It can be observed that the  $\alpha$  applicable to the four datasets are different. Utilizing difference convolution can capture and enhance subtle differences in spectral curves. The inclusion of suitable difference information can effectively enhance spectral details and provide fine-grained features about the objects, thereby significantly improving classification accuracy. The impact of  $\alpha$  variations on classification performance is not particularly significant for the IP dataset with relatively low spectral resolution and the LK dataset with high spectral resolution but most block-like regions ground object distributed. When  $\alpha = 0$ , only the vanilla convolution is employed to extract spectral features. As the variation of  $\alpha$  value increases, adding gradient information extracted by difference convolution enhances the classification performance. When the difference information is increased to a certain extent, the classification performance will gradually decline until the difference information is completely used.

In the IP dataset, the highest accuracy achieved is 96.79% when  $\alpha$  is set to 0.7, resulting in a 0.63% improvement compared to only utilizing vanilla convolution for extracting spectral features. The UP dataset consists of urban scenes with significant spectral differences among ground objects. When  $\alpha$  is set to 0.4, the highest accuracy achieved is 97.16%, representing a 2.86% improvement compared to vanilla convolution. In the HH dataset, the employment of the SeDC module results in the most significant improvement in OA values, reaching the highest accuracy of 99.27% when  $\alpha$  is set to 0.5, which is a 9.29% improvement compared to vanilla convolution. In the LK dataset, the highest accuracy achieved is 99.61% when  $\alpha$  is set to 0.5, which represents a 0.97% improvement compared to vanilla convolution.

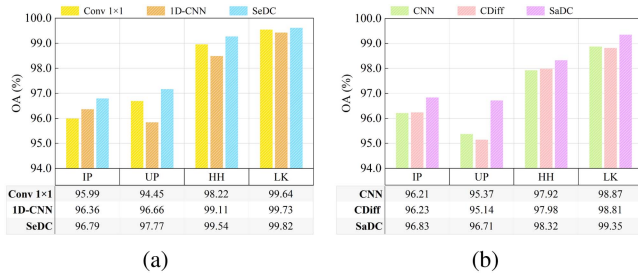


Fig. 13. Effectiveness of difference convolution modules. (a) SeDC module. (b) SaDC module.

When verifying the impact of the weight coefficient  $\alpha$  on the classification performance in the SaDC module, the  $\alpha$  value in the SeDC module is set to 0.6. As observed from Fig. 12(b), the difference information, which contains essential details in spatial features, improves the classification accuracy on the four datasets compared to vanilla convolution. Among them, the most significant effects are observed in the IP and UP datasets, with an improvement of up to 0.81% in the IP dataset and up to 1.33% in the UP dataset. The spatial resolution of IP dataset is the lowest among the four datasets, resulting in a larger surface area covered by a single pixel and extraction of larger-scale features. For the UP dataset, the spatial distribution of land covers often appears as narrow and elongated strips, making it susceptible to noise interference during spatial feature extraction. Therefore, using gradient-level information in difference convolution can improve the accuracy of identifying edge regions. The optimal weight coefficient  $\alpha$  values in the SaDC module for the four datasets are 0.2 in the IP dataset, 0.4 in the UP dataset, 0.7 in the HH dataset, and 0.3 in the LK dataset, respectively.

### G. Ablation Experiment

The effectiveness of the proposed SeDC and SaDC modules are validated by comparing with popular convolution modules, as shown in Fig. 13. The SeDC module is compared with conventional 1D-CNN [60] and pointwise convolution (Conv  $1 \times 1$ ) [61]. The ablation experimental results in spectral domain are presented in Fig. 13(a).

The valuable insights can be derived as follows: the performance of employing only vanilla convolution ( $\alpha = 0$ ) and CDiff ( $\alpha = 1$ ) is unsatisfactory. This indicates that relying solely on intensity- or gradient-level information leads to a one-sided feature extraction, incapable of integrating high-level abstract and shallow-level detail features from the spectral curves. The accuracies of the proposed SeDC module are significantly higher than that of using two other encoding patterns. Furthermore, the SeDC module achieves the best classification results over all four datasets, confirming the effectiveness and generalization capability.

Fig. 13(b) shows the accuracies of the proposed SaDC module compared to vanilla convolution (CNN) and CDiff. It is worth mentioning that combining deep abstract features and shallow fine-structured features can improve classification performance, as using either of them alone would lead to information loss. The ablation experimental results show the importance of difference

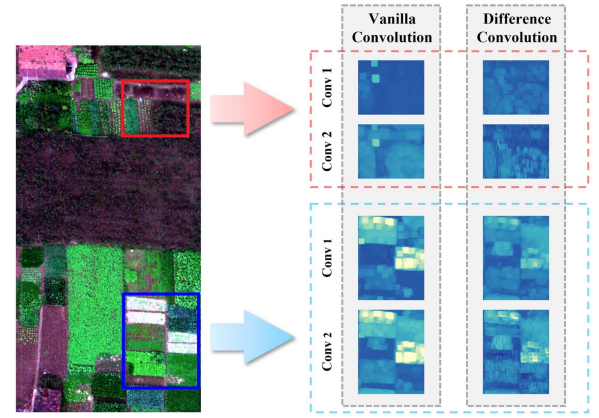


Fig. 14. Comparison of feature responses between difference convolution and vanilla convolution in the spatial domain. The RGB color image synthesized from the HH dataset is shown (band numbers are R: 113, G: 68, B: 23). Vanilla convolution primarily focuses on representing deep semantic features but lacks in capturing fine-grained intrinsic features. Difference convolution can effectively compensate for the oversmoothing effect of vanilla convolution, reflecting shallow-level details, such as edges, textures, and morphology that are closer to reality.

information in spatial feature extraction and representation, thus validating the effectiveness of the proposed SaDC module.

To further explore the effect of difference information, Fig. 14 visually compares difference convolution operations in the SaDC module with vanilla convolution. The difference convolution is advantageous in preserving shallow-level features, such as texture and morphological edges. Moreover, the difference convolution exhibits superior capability in preserving the resolution of features, effectively retaining important fine-structural features, and alleviating the oversmoothing issue observed in vanilla convolution. The comparison reveals that the difference convolution in the SaDC module outperforms in reserving detail features, making it suitable for HSI data with rich and dense spatial features.

## V. CONCLUSION

An effective end-to-end framework called  $S^2$ DCN is proposed for HSI classification tasks.  $S^2$ DCN integrates difference information into the deep learning architecture to enhance the ability of extracting and retaining detail features. To better learn the spectral and spatial information in the HSI, the SeDC and SaDC modules are designed. The SeDC can strengthen subtle spectral differences and extract more discriminative spectral features. The SaDC module can effectively improve the detail expression of important spatial information, including texture, edges, and morphology. Extensive experiments are conducted on four publicly HSI datasets to validate the proposed  $S^2$ DCN. The experimental results demonstrate that  $S^2$ DCN outperforms competitors quantitatively and visually. Moreover, the ablation studies are carried out to further confirm the effectiveness of difference convolution and the importance of detail features in achieving high accuracy HSI classification. In the future, we will explore a 3-D difference convolution architecture with adaptive weight coefficient to realize dynamic joint extraction of spectral-spatial features.

## REFERENCES

- [1] L. Deng et al., “M2H-Net: A reconstruction method for hyperspectral remotely sensed imagery,” *ISPRS J. Photogrammetry Remote Sens.*, vol. 173, pp. 323–348, 2021.
- [2] J. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza, and J. Li, “Visual attention-driven hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 8065–8080, Oct. 2019.
- [3] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, “Deep learning for hyperspectral image classification: An overview,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019.
- [4] C. Kwan et al., “Deep learning for land cover classification using only a few bands,” *Remote Sens.*, vol. 12, no. 12, 2020, Art. no. 2000.
- [5] K. C. Tiwari, M. K. Arora, and D. Singh, “An assessment of independent component analysis for detection of military targets from hyperspectral images,” *Int. J. Appl. Earth Observation Geoinf.*, vol. 13, no. 5, pp. 730–740, 2011.
- [6] B. Lu, P. D. Dao, J. Liu, Y. He, and J. Shang, “Recent advances of hyperspectral imaging technology and applications in agriculture,” *Remote Sens.*, vol. 12, no. 16, 2020, Art. no. 2659.
- [7] L. Ma, M. M. Crawford, and J. Tian, “Local manifold learning-based  $k$ -nearest-neighbor for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4099–4109, Nov. 2010.
- [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, Aug. 2004.
- [9] M. Khodadadzadeh, J. Li, A. Plaza, and J. M. Bioucas-Dias, “A subspace-based multinomial logistic regression for hyperspectral image classification,” *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 12, pp. 2105–2109, Dec. 2014.
- [10] N. Audebert, B. Le Saux, and S. Lefèvre, “Deep learning for classification of hyperspectral data: A comparative review,” *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 2, pp. 159–173, Jun. 2019.
- [11] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, “Hyperspectral remote sensing data analysis and future challenges,” *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, Jun. 2013.
- [12] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, “Deep learning for visual understanding: A review,” *Neurocomputing*, vol. 187, pp. 27–48, 2016.
- [13] 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.
- [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] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, “Deep feature extraction and classification of hyperspectral images based on convolutional neural networks,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.
- [16] S. Jia, S. Jiang, Z. Lin, N. Li, M. Xu, and S. Yu, “A survey: Deep learning for hyperspectral image classification with few labeled samples,” *Neurocomputing*, vol. 448, pp. 179–204, 2021.
- [17] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” in *Proc. Int. Conf. Learn. Representations*, 2021.
- [18] X. Cao, H. Lin, S. Guo, T. Xiong, and L. Jiao, “Transformer-based masked autoencoder with contrastive loss for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–12, 2023.
- [19] I. O. Tolstikhin et al., “MLP-mixer: An all-MLP architecture for vision,” *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 24261–24272, 2021.
- [20] P. Lv, W. Wu, Y. Zhong, F. Du, and L. Zhang, “SCViT: A spatial-channel feature preserving vision transformer for remote sensing image scene classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–12, 2022.
- [21] X. He and Y. Chen, “Modifications of the multi-layer perceptron for hyperspectral image classification,” *Remote Sens.*, vol. 13, no. 17, 2021, Art. no. 3547.
- [22] G. Wei, Z. Zhang, C. Lan, Y. Lu, and Z. Chen, “Active token mixer,” in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2023, pp. 2759–2767.
- [23] C. Chen, X. Chen, and H. Cheng, “On the over-smoothing problem of CNN based disparity estimation,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8996–9004.
- [24] H. Zhou, X. Zhang, C. Zhang, and Q. Ma, “Vision transformer with contrastive learning for hyperspectral image classification,” *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.
- [25] P. Valsalan et al., “Hyperspectral image classification model using squeeze and excitation network with deep learning,” *Comput. Intell. Neurosci.*, vol. 2022, 2022, Art. no. 9430779.
- [26] G. Jiang, Y. Sun, and B. Liu, “A fully convolutional network with channel and spatial attention for hyperspectral image classification,” *Remote Sens. Lett.*, vol. 12, no. 12, pp. 1238–1249, 2021.
- [27] J. Liu et al., “An investigation of a multidimensional CNN combined with an attention mechanism model to resolve small-sample problems in hyperspectral image classification,” *Remote Sens.*, vol. 14, no. 3, 2022, Art. no. 785.
- [28] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.
- [29] J.-L. Xu and A. A. Gowen, “Spatial-spectral analysis method using texture features combined with PCA for information extraction in hyperspectral images,” *J. Chemometrics*, vol. 34, no. 2, 2020, Art. no. e3132.
- [30] Y. Li, H. Tang, W. Xie, and W. Luo, “Multidimensional local binary pattern for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–13, 2022.
- [31] Z. Yu, J. Wan, Y. Qin, X. Li, S. Z. Li, and G. Zhao, “NAS-FAS: Static-dynamic central difference network search for face anti-spoofing,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 9, pp. 3005–3023, Sep. 2021.
- [32] S. Zhang, M. Xu, J. Zhou, and S. Jia, “Unsupervised spatial-spectral CNN-based feature learning for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–17, 2022.
- [33] W. Yao, C. Lian, and L. Bruzzone, “ClusterCNN: Clustering-based feature learning for hyperspectral image classification,” *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 11, pp. 1991–1995, Nov. 2021.
- [34] N. Gong, C. Zhang, H. Zhou, K. Zhang, Z. Wu, and X. Zhang, “Classification of hyperspectral images via improved cycle-MLP,” *IET Comput. Vis.*, vol. 16, no. 5, pp. 468–478, 2022.
- [35] X. Tan and Z. Xue, “Spectral-spatial multi-layer perceptron network for hyperspectral image land cover classification,” *Eur. J. Remote Sens.*, vol. 55, no. 1, pp. 409–419, 2022.
- [36] Z. He, K. Xia, P. Ghamisi, Y. Hu, S. Fan, and B. Zu, “HyperViTGAN: Semisupervised generative adversarial network with transformer for hyperspectral image classification,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 6053–6068, 2022.
- [37] Z. Zhang, Q. Ma, H. Zhou, and N. Gong, “Nested transformers for hyperspectral image classification,” *J. Sensors*, vol. 2022, 2022, Art. no. 6785966.
- [38] X. Zhang et al., “Spectral–spatial self-attention networks for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022.
- [39] M. E. Paoletti, S. Moreno-Álvarez, Y. Xue, J. M. Haut, and A. Plaza, “AAAtt-CNN: Automatic attention-based convolutional neural networks for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–18, 2023.
- [40] W. Guo, H. Ye, and F. Cao, “Feature-grouped network with spectral–spatial connected attention for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–13, 2022.
- [41] C. Mu, Z. Guo, and Y. Liu, “A multi-scale and multi-level spectral-spatial feature fusion network for hyperspectral image classification,” *Remote Sens.*, vol. 12, no. 1, 2020, Art. no. 125.
- [42] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 886–893.
- [43] T. Lindeberg, “Scale invariant feature transform,” *Scholarpedia*, vol. 7, no. 5, p. 10491, 2012, doi: [10.4249/scholarpedia.10491](https://doi.org/10.4249/scholarpedia.10491).
- [44] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, “Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles,” *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 11, pp. 3804–3814, Nov. 2008.
- [45] Z. Yu et al., “Searching central difference convolutional networks for face anti-spoofing,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Los Alamitos, CA, USA, 2020, pp. 5294–5304, doi: [10.1109/CVPR42600.2020.00534](https://doi.org/10.1109/CVPR42600.2020.00534).
- [46] F. Juefei-Xu, V. Boddeti, and M. Savvides, “Local binary convolutional neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Los Alamitos, CA, USA, 2017, pp. 4284–4293, doi: [10.1109/CVPR.2017.456](https://doi.org/10.1109/CVPR.2017.456).
- [47] S. Luan, C. Chen, B. Zhang, J. Han, and J. Liu, “Gabor convolutional networks,” *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4357–4366, Sep. 2018.

- [48] Z. Yu, Y. Qin, H. Zhao, X. Li, and G. Zhao, "Dual-cross central difference network for face anti-spoofing," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, 2021, pp. 1281–1287, doi: [10.24963/jcai.2021/177](https://doi.org/10.24963/jcai.2021/177).
- [49] Z. Su et al., "Pixel difference networks for efficient edge detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 5097–5107, doi: [10.1109/ICCV48922.2021.00507](https://doi.org/10.1109/ICCV48922.2021.00507).
- [50] Z. Yu et al., "Searching multi-rate and multi-modal temporal enhanced networks for gesture recognition," *IEEE Trans. Image Process.*, vol. 30, pp. 5626–5640, 2021.
- [51] W. Liu, Z. Su, and L. Liu, "Beyond vanilla convolution: Random pixel difference convolution for face perception," *IEEE Access*, vol. 9, pp. 139248–139259, 2021.
- [52] X. Wu, D. Ma, X. Qu, X. Jiang, and D. Zeng, "Depth dynamic center difference convolutions for monocular 3D object detection," *Neurocomputing*, vol. 520, pp. 73–81, 2023.
- [53] 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.
- [54] Y. Zhong, X. Hu, C. Luo, X. Wang, J. Zhao, and L. Zhang, "WHU-Hi: UAV-borne hyperspectral with high spatial resolution (H2) benchmark datasets and classifier for precise crop identification based on deep convolutional neural network with CRF," *Remote Sens. Environ.*, vol. 250, 2020, Art. no. 112012.
- [55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Los Alamitos, CA, USA, 2016, pp. 770–778, doi: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- [56] Z. Zhang, H. Zhang, L. Zhao, T. Chen, and T. Pfister, "Nested hierarchical transformer: Towards accurate, data-efficient and interpretable visual understanding," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 3, 2022, pp. 3417–3425.
- [57] S. Chen, E. Xie, C. Ge, R. Chen, D. Liang, and P. Luo, "CycleMLP: A MLP-like architecture for dense prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023.
- [58] Z. Su et al., "Pixel difference networks for efficient edge detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 5117–5127.
- [59] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.
- [60] H. Zhang, Y. Li, Y. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network," *Remote Sens. Lett.*, vol. 8, no. 5, pp. 438–447, 2017.
- [61] M. Lin, Q. Chen, and S. Yan, "Network in network," in *Proc. Int. Conf. Learn. Representations*, 2014.



**Zitong Zhang** (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in mathematics from the China University of Geosciences, Beijing, China, in 2017 and 2020, respectively. She is currently working toward the Ph.D. degree in geophysical prospecting and information technology with the School of Earth Sciences and Resources, China University of Geosciences.

Her research interests include deep learning and hyperspectral image classification.



**Hanlin Feng** (Graduate Student Member, IEEE) received the B.Eng. degree in Internet of Things engineering from North China Electric Power University, Beijing, China, in 2021. He is currently working toward the M.Eng. degree in computer science and technology with Northeast Petroleum University, Daqing, China.

His research interests include deep learning, image processing, pattern recognition, and computer vision.



**Chunlei Zhang** received the M.Eng. degree in oil and gas, coal field geology from the Taiyuan University of Technology, Taiyuan, China, in 1997, and the Ph.D. degree in mineral resource prospecting and exploration from the China University of Petroleum, Beijing, China, in 2000. He majored in geostatistics, reservoir characterization, and reservoir engineering.

He is currently a Senior Engineer. From 2002 to 2004, he did postdoctoral research with the China University of Petroleum. From 2004 to present, he is with energy industry. His research interests include geostatistics, pattern recognition, deep learning, and computer vision.



**Qiaoyu Ma** (Student Member, IEEE) received the B.Eng. degree in computer science and technology from Shanghai University, Shanghai, China, in 2018, and the M.Eng. degree in computer technology from the China University of Geosciences, Beijing, China, in 2021. He is currently working toward the Ph.D. degree in computer science and technology with China Agricultural University, Beijing, China.

His research interests include deep learning, computer vision, and pattern recognition.



**Yuntao Li** received the B.Eng. degree in automation from Zhengzhou University, Zhengzhou, China, in 2018. He is currently working toward the M.Eng. degree in mechanical engineering with the China University of Mining and Technology-Beijing, Beijing, China.

His research interests include machine learning, pattern recognition, mechanical engineering, and computer vision.