A Dual-Branch Multiscale Transformer Network for Hyperspectral Image Classification

Cuiping Shi, *Member, IEEE*, Shuheng Yue, and Liguo Wang, *Member, IEEE*

***Abstract*— In recent years, convolutional neural networks (CNNs) have achieved great success in hyperspectral image (HSI) classification tasks. CNNs focus more on the local features of HSIs. The recently emerging Transformer network has shown great interest in the global features of HSIs. However, existing Transformer networks only consider single-scale feature extraction and do not combine the advantages of multiscale feature extraction and Transformer global feature extraction. To address this issue, this article proposes a dual-branch multiscale Transformer (DBMST) for HSI classification. First, a large-size spectral convolution kernel is utilized for the spectral dimension of the hyperspectral cube for downsampling feature extraction. Next, a channel shrink soft split module (CS3M) is proposed, which not only solves the problem of missing local information in large-scale tokens but also extracts shallow features and performs dimensionality reduction on channels. Then, considering the different dimensions of features extracted at different scales in two branches, a pooled activation fusion module (PAFM) is carefully designed. Finally, the proposed DBMST is evaluated on three commonly used HSI datasets. The experimental results show that DBMST achieves better classification performance compared to other advanced networks, demonstrating the effectiveness of the proposed method in HSI classification.**

***Index Terms*— Classification, feature extraction, hyperspectral images (HSIs), multiscale, Transformer.**

I. Introduction

With the rapid development of remote sensing technology and unmanned aerial vehicle (UAV) technology, the amount of information obtained from hyperspectral images (HSIs) is also becoming increasingly abundant. HSIs contain rich spatial features and continuous spectral information, with each pixel containing thousands of continuous spectral bands, providing generous spectral information. At present, HSIs have been widely applied in multiple fields, such as military surveys [1], vegetation analysis [2], biomedical imaging [3], and geological surveys [4]. In order to fully utilize the inherent potential of hyperspectral data, various data processing techniques have been explored, such as data compression [5], spectral unmixing [6], object detection [7], data reconstruction and recovery [8], and classification [9], [10], [11]. Among these technologies, classification as a mainstream application has attracted the attention of many researchers. In the past decade, a large number of feature extraction methods based on handcrafted and subspace learning were proposed for HSI classification, such as the k-nearest neighbor method [12], the support vector machine (SVM) [13], [14], [15], [16], and the Bayesian estimation method [17]. In addition, some methods for dimensionality reduction and spectral information extraction have also been proposed; typical methods include principal component analysis (PCA) [18], linear discriminant analysis (LDA) [19], [20], and local preserving projection (LPPS) [21]. However, the above methods did not fully utilize spatial features and ignored the correlation between adjacent pixels. In order to better learn the spatial features of images and effectively utilize the correlation between pixels, Sun et al. [10] fully utilized the spectral and spatial information of HSIs and proposed a multiscale spatial–spectral kernel method based on adjacent superpixels, which improved the classification performance. Duan et al. [22] proposed a sparse popular hypergraph method based on semisupervised geodesic to improve classification performance by combining hypergraph embedding feature extraction and sparse representation. The above methods are HSI classification methods based on handcrafted and subspace learning, which require researchers to have rich expert knowledge and require manual design of feature extractors, thus having significant limitations. In recent years, some HSI classification methods based on deep learning have attracted the attention of researchers.

With the rapid development of deep learning technology, image processing technology has also made significant progress in various fields, promoting innovation in remote sensing image processing technology. A large number of methods based on deep learning [23], [24], [25], [26], [27], [28], [29] were designed for HSI classification tasks. In addition to being applied in HSI classification, deep learning-based networks are also used in the multimodal classification of LiDAR data and HSIs, as well as cross-scene HSI classification methods [56], [57]. The mainstream backbone networks include autoencoders (AEs), convolutional neural networks (CNNs), recurrent neural networks (RNNs), generic

Manuscript received 26 August 2023; revised 10 December 2023; accepted 5 January 2024. Date of publication 8 January 2024; date of current version 22 January 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 42271409 and Grant 62071084, in part by the Heilongjiang Science Foundation Project of China under Grant LH2021D022, and in part by the Fundamental Research Funds in Heilongjiang Provincial Universities of China under Grant 145209149. *(Corresponding author: Cuiping Shi.)*

Cuiping Shi is with the Department of Communication Engineering, Qiqihar University, Qiqihar 161000, China, and also with the College of Information Engineering, Huzhou University, Huzhou 313000, China (e-mail: shicuiping@qqhru.edu.cn).

Shuheng Yue is with the Department of Communication Engineering, Qiqihar University, Qiqihar 161000, China (e-mail: 2021910320@qqhru.edu.cn).

Liguo Wang is with the College of Information and Communication Engineering, Dalian Nationalities University, Dalian 116000, China (e-mail: wangliguo@hrbeu.edu.cn).

Digital Object Identifier 10.1109/TGRS.2024.3351486

adversarial networks (GANs), capsule networks (CapsNets), graph convolutions networks (GCNs), and graph attention networks (GATs).

Graph neural networks have achieved remarkable performance in the field of HSI classification, as they can effectively extract data features for classification through adjacency matrices and graph nodes. Dong et al. [60] proposed WFCG for HSI classification. WFCG combines the feature extraction advantages of superpixel-based GAT with pixel-based CNN. The features extracted by GAT and CNN are fused with weighted features for classification. Li et al. [61] proposed a multilevel superpixel-guided sparse GAT (MSG GAT) for HSI classification. An SGAT method was proposed in MSG GAT to simplify the architecture of GAT and reduce the risk of overfitting while ensuring classification accuracy. To address the high spatial complexity of GNN, Liu et al. [62] proposed FDGC for HSI classification. A new dynamic GCN was designed, which can adaptively capture topology information and greatly reduce spatial complexity.

Thanks to the powerful image feature extraction capabilities of CNNs, they have become the most popular DL backbone network [30], [31], [32]. HSIs contain rich spatial and spectral features, and sufficient extraction of these features can effectively improve classification performance. In early research on HSIs' classification, many excellent CNN networks were proposed. For example, considering that 3-D convolution can extract spectral–spatial features, He et al. [33] proposed multiscale 3-D CNN. Usually, fixed-size convolutions are used for feature extraction in images, but they ignore the inherent spatial structure information of ground objects, resulting in the loss of spatial details. Therefore, Shang et al. [37] proposed an HSI classification method based on multiscale cross-branch response and second-order channel attention (MCRSCA). Zhu et al. [53] proposed a CNN DHCNet based on deformable convolution. The deformable convolution has a dynamic receptive field, which is not limited to fixed structures and avoids the neglect of spatial structure information. Due to the limited receptive field of convolution, in order to expand the range of receptive fields, it is usually necessary to add additional convolution layers to increase the network depth. As the depth of the network increases, network convergence is hindered and lower classification accuracy is generated. Therefore, Paoletti et al. [36] proposed a pyramid network of deep residuals (PyResNet), which gradually increases the network depth in the form of residuals and avoids the obstacles to convergence while obtaining a large receptive field. The 3-D convolutional paradigm has the advantage of spatial–spectral joint feature extraction, while the 2-D convolutional paradigm has the advantage of spatial feature extraction. The hybrid spectral CNN (Hybrid-SN) [35] was proposed, which greatly improves classification performance by combining the advantages of 2-D CNN and 3-D CNN in extracting spatial and spectral joint features. In order to highlight the query pixels and correctly extract the spatial information brought by the pixels around the query pixels, CAN was proposed by Liu et al. [58] to design a scaled dot product center attention module (SDPCA) for HSI classification. It can extract spectral–spatial information from the center pixel and pixels similar to the center pixel on the HSI patch for HSI classification. Although CNN-based methods have strong extraction capabilities for image features and spatial context information, they still have some limitations. The convolution of a limited receptive field makes it impossible for CNN-based methods to obtain global information. Even by deepening the network layers, the actual receptive field cannot achieve the global effect.

In the past two years, Transformer's success in natural language processing (NLP) has led to the development of computer vision [38], [39], [40], [41] and promoted the development of HSI classification. In the field of HSIs' classification, many excellent Transformer methods have been proposed. In [42], an HSI Transformer (HIT) was proposed to obtain subtle spectral–spatial differences. This network encodes spatial spectra along the height, width, and spectral dimensions through convolutional permutators and uses spectral adaptive 3-D convolutional mapping modules instead of linear mapping to obtain local spatial–spectral information. Hong et al. [43] rethought HSIs' classification from a sequence perspective and proposed SpectralTransformer (SF). It can generate grouped spectral embeddings by learning spectral information between adjacent bands of HSIs, thereby learning local spectral feature representations. Mei et al. [44] found that when the Transformer classifies HSIs with a large number of frequency bands, the features extracted by multihead self-attention may exhibit excessive dispersion. To address this issue, a group-aware hierarchical Transformer (GAHT) for HSI classification was proposed. It constructs the Transformer in a hierarchical manner and restricts multihead self-attention to a local spatial environment through a grouped pixel embedding module. Although these methods can learn spectral semantic information well, they ignore high-frequency information such as texture and edge. In order to better represent high-level semantic features and obtain spectral–spatial features, Sun et al. [45] proposed a spectral–spatial feature tokenization Transformer (SSFTT) that extracts shallow spectral and spatial features through a designed convolutional module and performs feature transformation through a Gaussian weighted feature marker. Similarly, Zhang et al. [46] proposed a convolutional transformer mixer (CTMixer) for hyperspectral classification, which is modeled using CNNs and Transformer frameworks to obtain global local hyperspectral features. Next, a group parallel residual block is constructed to extract local spectral–spatial features, achieving an effective combination of convolution and Transformer.

The multiscale features of images are particularly important in HSI classification. Multiscale features can not only avoid information loss and redundancy at a single scale and extract richer and more comprehensive feature information to improve classification performance but also improve generalization performance. Multiscale feature extraction has been widely applied in CNNs, and Zhong et al. [47], Zhang et al. [48], Wang et al. [49], Gao et al. [50], and Lu et al. [51] have demonstrated the importance of multiscale features in HSIs' classification. The Transformer network can obtain global dependencies and extract low-frequency information from images through the multihead self-attention module. The Transformer methods, such as SF and SSFTT, were

proposed for HSI classification based on single-scale feature extraction. They only consider the extraction of single-scale features and cannot extract multiscale features for HSI classification. In recent years, researchers have combined multiscale feature extraction with Transformers and proposed many multiscale Transformers, such as shunted Transformer [54] and MSNAT [59], which implements Transformer's multiscale feature extraction from the perspective of self-attention. In this article, in order to better combine multiscale feature extraction with Transformer, a dual-branch multiscale Transformer (DBMST) for HSI classification is proposed. Specifically, first, DBMST uses 3-D spectral convolution to extract spectral information of HSIs. Second, to achieve the division of tokens at different scales, a channel shrink soft split module (CS3M) is designed. Then, a token-to-token (T2T) feature extraction module is proposed to convert tokens into images and extract local information from the images. Next, based on the features extracted at different scales in different branches, this article proposes a pooled activation fusion module (PAFM) for fusion. Finally, the fused features predict the corresponding labels for each pixel through a linear layer.

The main contributions of this article are given as follows.

1) This article proposes a DBMST that fully utilizes the advantages of Transformer and multiscale features, with two branches performing feature extraction at different scales to obtain richer discriminative features. The proposed DBMST is a parallel DBMST for HSI classification. Experiments on three commonly used datasets have demonstrated that the DBMST can provide good HSI classification performance.
2) A CS3M is designed in DBMST, which is utilized for large-scale token partitioning to avoid information loss caused by conventional token partitioning methods for hyperspectral data. The input channel is shrunk to reduce the amount of parameters required for linear mapping.
3) A T2T module is proposed for replacing the feed-forward network (FFN) in Transformer. It can transform token into image, extract local information of image, obtain high-frequency information, and enhance global semantic information.
4) A PAFM module is designed for feature fusion of different dimensions. It can fuse features at different scales in different branches extracted by DBMST.

The remaining part of this article is arranged as follows. In Section II, the network structure of the proposed DBMST is introduced in detail. In Section III, the complexity analysis of DBMST, quantitative analysis of comparative experiments, and visual evaluation results are presented. The conclusions are provided in Section IV.

II. Methodology

The DBMST method proposed in this article mainly includes three modules, i.e., CS3M, the T2T local global feature extraction module, and PAFM for feature fusion of different dimensions.

The overall network framework is shown in Fig. 1. The input HSIs' data are $X \in \mathbb{R}^{H\times W\times L}$, where H, W, and L represent the length, width, and number of bands of HSIs, respectively, with the corresponding labels being $Y_i \in \{1, 2, 3, \ldots, \text{Class}\}$. In order to remove spectral redundancy, PCA is first performed to reduce the spectral dimension. Next, for the HSI classification method based on central pixel segmentation, in order to avoid the loss of edge information, edge filling is performed on X to obtain $X_{\text{in}} \in \mathbb{R}^{h\times w\times b}$ (where $h \times w$ refers to the spatial size of the processed image and b is the number of spectral bands after dimensionality reduction). Finally, the processed data are sent into DBMST to extract the features.

HSIs are 3-D data, and the spectrum has sequential properties. In this article, the 3-D + 2-D structure is utilized for shallow feature extraction of input data, which is more conducive to subsequent extraction of spectral features. First, the input data after PCA preprocessing are $X_{\text{in}} \in \mathbb{R}^{64\times 1\times h\times w\times 30}$. $X_{\text{in}} \in \mathbb{R}^{64\times 1\times h\times w\times 30}$ is input into a convolutional block containing a 3-D spectral convolution layer, batch normalization layer, and rectified linear unit (ReLU) to obtain the output feature $F_1 \in \mathbb{R}^{64\times 64\times h\times w\times 1}$. This process can be represented as

$$F_1 = \delta(f_{\text{BN}}(X_{\text{in}}\Theta W^{3D} + b^{3D})) \tag{1}$$

where δ represents the nonlinear activation function ReLU, f_{BN} represents the batch normalization, Θ represents the convolution operator, and W^{3D} and b^{3D} represent the weight and bias of 3-D convolution, respectively.

DMBST is a dual-branch multiscale Transformer. The input data obtained by the spectral processing layer have a size of $64 \times 64 \times h \times w$, where the first 64 represents the number of batches and the second 64 represents the number of channels. In the small-scale branch, the output of the spectral processing layer is divided by the 1×1 window to obtain the input with a size $64 \times 64 \times (h \times w)$ of the small-scale branch. The input size of the large-scale branch is $64 \times n \times c_1$, which is obtained by dividing the output of the spectral processing layer by the window with size $s \times s$.

A. Channel Shrink Soft Split Module

In the past few years, Transformer has been applied to HSI classification. Assuming that the data input to the network is $X_{\text{in}} \in \mathbb{R}^{h\times w\times b}$. When using a Transformer for modeling, if the scale of the token is large, there will be a problem of spatial information loss. Furthermore, most Transformers are single scale, which results in the inability of multiscale features to be extracted by Transformers. Therefore, many multiscale Transformers have been proposed by researchers in recent years, such as the shunted Transformer [54] and CVT [55]. Shunted Transformer implements multiscale feature extraction for Transformer from the perspective of self-attention. CVT achieves multiscale feature extraction from the perspective of input token scale. Its multiscale token is obtained by mapping the input image by introducing 2-D convolutions with size Patchsize × Patchsize. Unlike them, in this article, the multiscale feature extraction is achieved through parallel branches. In order to achieve multiscale Transformer to extract richer image features and avoid the loss of spatial information in dividing large-scale tokens, a CS3M is proposed, which

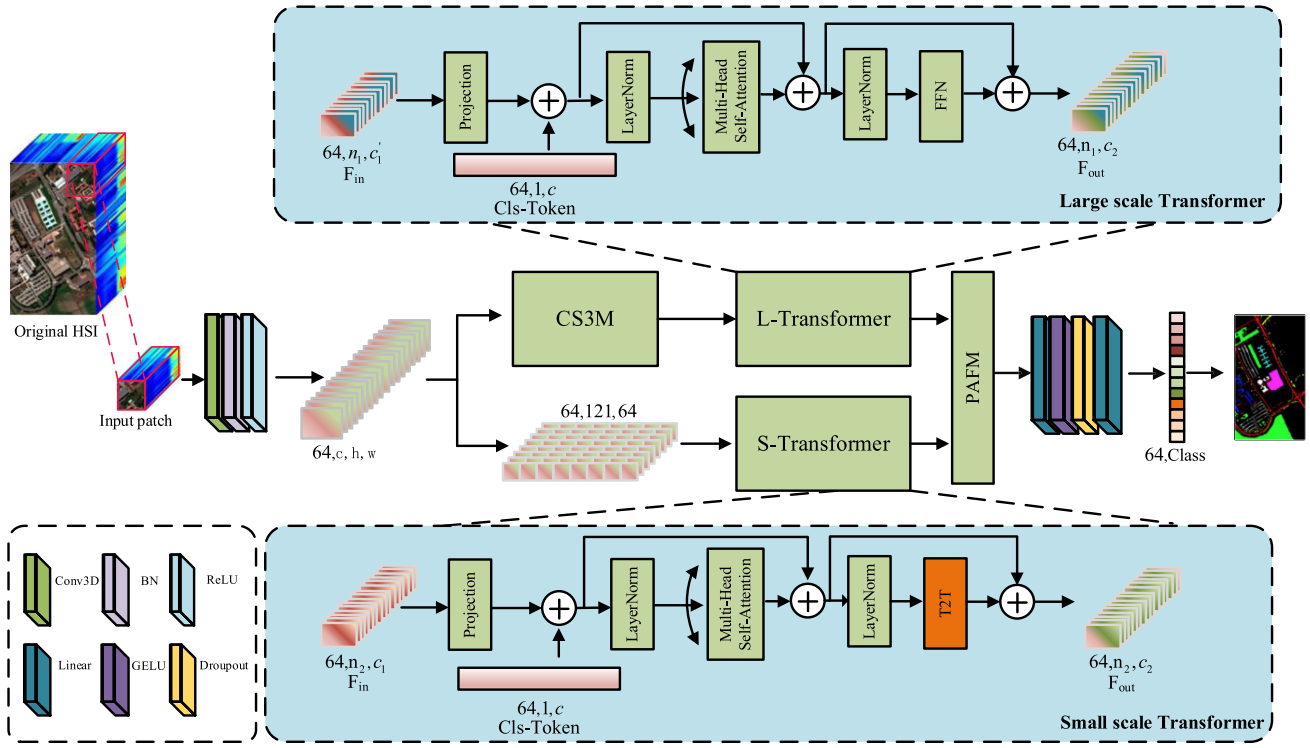


Fig. 1. Overall structure diagram of the proposed DBMST.

utilizes window sliding algorithm to partition and obtain large-scale input tokens. Different from CVT, we do not partition large-scale input tokens through fixed size convolution operations. The structure of CS3M is shown in Fig. 2.

In this section, a detailed introduction of the proposed CS3M module is provided, which is composed of some channel shrink modules and soft splitting modules. A channel shrink block is composed of a convolutional layer, a batch normalization layer, and a nonlinear activation layer. Its main purpose is to shrink the channels, remove redundant channel information, and further extract local information such as the texture and edges of the image. Its process can be represented as

$$X_1 = \delta(f_{\mathrm{BN}}(X'_{\mathrm{in}} \Theta W^{2D} + b^{2D})) \tag{2}$$

where δ represents the nonlinear activation function ReLU, f_{BN} represents the batch normalization, Θ represents the convolution operator, W^{2D} and b^{2D} represent the weight and offset of 2-D convolution, respectively, and $F_1 \in \mathbb{R}^{h\times w\times c_1}$ is the output feature of the channel shrink block.

Usually, the spatial size $h \times w$ of F_1 is small. If the traditional token partitioning method is used to obtain the token set $T_a \in \{t_1, t_2, \ldots, t_s\}$, some spatial information will be lost, as shown in the green border area in Fig. 3(a). The token partitioning method using soft split (SS) can avoid the loss of some spatial information, which is shown in Fig. 3(b). The resulting token set is $T_b \in \{t_1, t_2, \ldots, t_n\}$, where $n > s$. Compared with traditional token partitioning methods, the SS partitioning method can retain more information. The process is represented as

$$F_T = \mathrm{SS}(F_2) \tag{3}$$

$$\mathrm{SS} = \mathrm{Flatten}\left(\sum_{j=0}^{w}\sum_{i=0}^{h} \mathrm{Slice}\left(x_{(j,i)}^{(j+p,i+p)}\right)\right) \tag{4}$$

where $\mathrm{Slice}(x_{(j,i)}^{(j+p,i+p)})$ represents a token partition of $p \times p$ size from position (j, i) to position $(j + p, i + p)$ of the input HSIs cube. Flatten$(\cdot)$ represents the flattening function. The output of SS$(\cdot)$ is $F_T \in \mathbb{R}^{64\times n\times c_1}$. The calculation process of n and c_1 can be represented as

$$n = \mathrm{patchsize\_}w \times \mathrm{patchsize\_}h \tag{5}$$

$$c_1 = (h - \mathrm{patchsize\_}h + 1) \times (w - \mathrm{patchsize\_}w + 1) \times c' \tag{6}$$

where $h \times w$ refers to the spatial size of the HSIs cube, patchsize_h and patchsize_w represent the length and width of the token sampling window, respectively, and c represents the number of channels after passing through CS3M.

It is worth noting that the large-scale tokens obtained by CS3M will be dimensionally reduced through linear layers. The calculation for linear layer training parameters is

$$p = C_{\mathrm{in}} \times C_{\mathrm{out}} \tag{7}$$

where p represents the calculated parameter quantity, C_{in} represents the number of input channels, and C_{out} represents the number of output channels.

The number of channels for the input feature $F_1 \in \mathbb{R}^{64\times h\times w\times c}$ is c, which is shrunk by CS3M to c_1, resulting

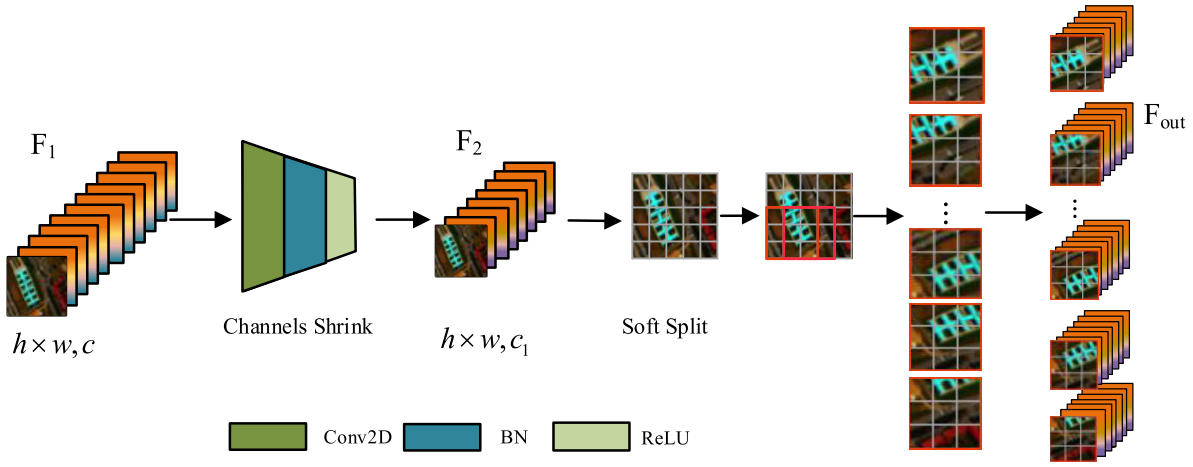


Fig. 2. Structure diagram of CS3M.

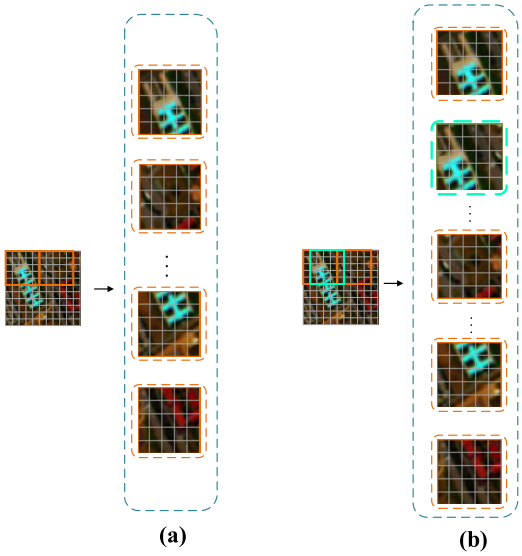


Fig. 3. (a) Traditional token partitioning method. (b) SS.

in the output feature $F_T \in \mathbb{R}^{64\times n\times c_1}$, where $c > c_1$. According to formulas (7) and (6), compared with the parameter quantity without CS3M, the parameter quantity required for linear mapping with CS3M is reduced to c_1/c times. Therefore, channel shrink not only has the advantages of shrinking the number of channels, reducing channel redundancy, and extracting local features of the image but also reduces the training parameters of the network.

The size of input and output in CS3M proposed in this article is independent of the dataset. The input and output sizes in CS3M are the same in the Indian Pines, Pavia, Salinas, and Houston 2013 datasets. Overall, for the proposed CS3M, the size of the input data is $64 \times 64 \times h \times w$, where the first 64 represents the number of batches, the second 64 represents the number of channels, and $h \times w$ represents the spatial size of the input data. First, the input data are fed into the channel shrink module. The feature size after channel shrink is $64 \times c \times h \times w$, where C represents the number of channels after shrink. Then, the features after channel contraction are input into the module SS. Finally, the output feature size of CS3M is $64 \times n \times c_1$.

B. L-Transformer

L-Transformer is a traditional Transformer, which is shown in Fig. 4. Traditional ViT mainly consists of multihead self-attention, FFN, and layer normalization (LN). First, multihead self-attention maps the input into three vectors: query, key, and value. Next, perform point multiplication on the query vector and key vector to obtain a correlation matrix. Then, softmax activation is performed on the obtained correlation matrix. In order to alleviate the gradient disappearance caused by the softmax function, the correlation matrix is scaled before using softMax. The multihead self-attention can be represented as

$$\text{Attention}(Q, K, V) = \text{SoftMax}\left(\frac{QK^T}{\sqrt{D_k}}\right)V \tag{8}$$

$$\text{MultiHeadAttn}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_H)W^o \tag{9}$$

$$\text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{10}$$

where Q, K, and V represent the query vector, the key vector, and the value vector, respectively, SoftMax$(\cdot)$ represents the SoftMax activation function, $\sqrt{D_K}$ represents the contraction coefficient, Concat represents the cascade function, W^o represents the weight of linear mapping, and W_i^j represents the ith attention information of the jth vector.

The forward propagation network FFN is composed of two layers and a multilayer perceptron, which can mine the nonlinear relationship of features and enhance the representation ability of features.

First, the feature with size $64 \times n \times c_1$ obtained through the CS3M module is used as input for the L-Transformer. Finally, the size of the features extracted through multihead self-attention and FFN is $64 \times n \times c_1$.

C. Token-to-Token Feature Extraction Module

In this section, a T2T module is constructed. The FFN of the traditional L-Transformer is replaced with the T2T

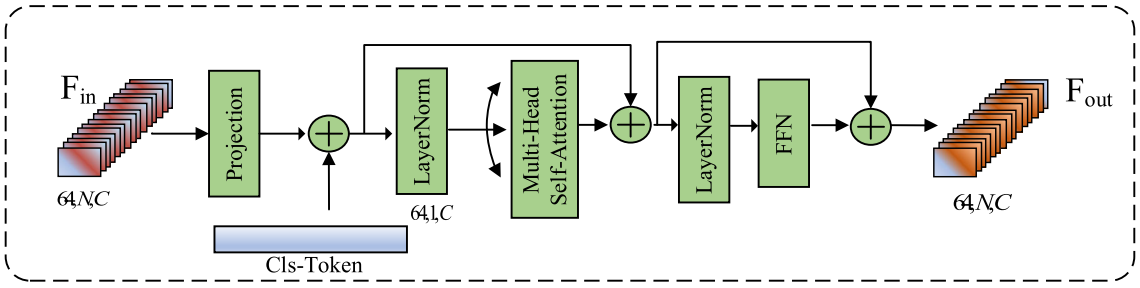


Fig. 4. L-Transformer structure diagram.

module, and named S-Transformer, which is a Transformer for small-scale token feature extraction. The T2T structure of the S-Transformer is shown in Fig. 5. Compared to FFN, T2T extracts features from feature matrices, while FFN extracts features from feature vectors. The feature vector only has two positional relationships; the positional relationships of the feature matrix are more abundant, which are more closely related to the spatial structure of the image. Through convolutional blocks, more abundant local features can be extracted.

In this section, a detailed introduction to the T2T module is provided. T2T includes token to image (T2I), image feature extraction, and image to token (I2T). The input of T2T is a small-scale token divided by a 1×1 window. The size of the input feature is $64 \times N \times C$. First, the input feature $F_{\text{in}} \in \mathbb{R}^{64\times N\times C}$ goes through LN, multihead self-attention, and residual connection to obtain the intermediate feature $F_m \in \mathbb{R}^{64\times N\times C}$. In the T2I stage, the intermediate feature $F_m \in \mathbb{R}^{64\times N\times C}$ is converted from a feature vector to a feature matrix. Compared with the feature vector, the feature matrix has more spatial location information and can extract more local information. It is worth noting that in order to preserve category information, before converting the feature vector to the feature matrix, we separate Cls-Token from $F_m \in \mathbb{R}^{64\times N\times C}$ to obtain $F_C \in \mathbb{R}^{64\times 1\times C}$ and $F'_m \in \mathbb{R}^{64\times(N-1)\times C}$. Then, $F'_m \in \mathbb{R}^{64\times(N-1)\times C}$ obtains feature $F_I \in \mathbb{R}^{64\times C\times h\times w}$ through T2I. The process of T2I is represented as

$$F_I = \text{Reshape}\big(\text{Separate}\big(\big[F'_m, F_C\big]\big)\big) \tag{11}$$

where F'_m represents the feature token, F_C represents the classification token, Separate$(\cdot)$ represents the separation operation, used to separate the feature token from the classification token, and F_I is the feature image after the Reshape$(\cdot)$ conversion operation.

In order to reduce overfitting and avoid gradient disappearance, the residual connection is added to the feature extraction stage. In the feature extraction stage, first, $F_I \in \mathbb{R}^{64\times C\times h\times w}$ extracts local information through 2-D convolutional blocks to obtain $F'_I \in \mathbb{R}^{64\times C\times h\times w}$. Then, global feature enhancement is performed on the obtained $F'_I \in \mathbb{R}^{64\times C\times h\times w}$ through mean pooling, batch normalization, and residual connection layers to obtain $F_e \in \mathbb{R}^{64\times C\times h\times w}$. The process of the feature extraction can be represented as

$$F'_I = f_{\text{BN}}(\delta_G(F_I \Phi W^{2D} + b^{2D})) \tag{12}$$

where δ_G represents the Gaussian error linear unit (GELU), f_{BN} represents batch normalization, and Φ, W^{2D}, and b^{2D} are convolution operators, weights, and biases, respectively,

$$F''_I = f_{\text{BN}}\big(f_{\text{Avgpool}}\big(f_{\text{BN}}\big(f_{\text{Avgpool}}\big(F'_I\big) + F'_I\big)\big) + F'_I\big) \tag{13}$$

$$F'''_I = \delta_G\big(f_{\text{BN}}\big(F''_I \Theta W^{2D} + b^{2D}\big)\big) \tag{14}$$

$$F_e = \delta_G\big(\big[f_{\text{Avgpool}}\big(F'''_I\big)\Theta W^{2D} + b^{2D}\big]\big) \tag{15}$$

where f_{Avgpool} and f_{BN} represent mean pooling and batch normalization, respectively, while δ_G is a GELU.

Usually, features in a Transformer are transmitted in the form of vectors. Therefore, we need to convert $F_e \in \mathbb{R}^{64\times C\times h\times w}$ from the structure of the matrix to a vector token, that is, the feature matrix $F_e \in \mathbb{R}^{64\times C\times h\times w}$ is transformed into the feature vector $F'_m \in \mathbb{R}^{64\times(N-1)\times C}$. Then, cascade the results with Cls-Token $F_C \in \mathbb{R}^{64\times 1\times C}$. Finally, a layer of MLP is used to obtain the nonlinear relationship of features, enhance feature representation, and obtain output feature $F_s \in \mathbb{R}^{64\times N\times C}$. The process of I2T can be represented as

$$F''_m = \text{Concatenate}\big(\big[\text{Flatten}\big(F'_m\big), F_C\big]\big) \tag{16}$$

$$F_s = \delta_G\big(f_{\text{LN}}\big(F''_m * W + b\big)\big) \tag{17}$$

where Flatten$(\cdot)$ represents the flattening function used to convert a 2-D image into a 1-D vector, Concatenate$(\cdot)$ is the cascading function, and W and b are the weights and biases of the linear mapping.

D. Pooled Activation Fusion Module

DBMST includes two branches, namely, the L-Transformer branch and the S-Transformer branch. It is worth noting that their inputs are tokens of two different scales, which means that the output feature dimensions obtained by the two Transformer modules are different, making it difficult for features to be directly fused. In this article, a PAFM is proposed. It adopts the adaptive mean pooling to downsample two output features to the same dimension for weighted fusion, which is shown in Fig. 6.

Specifically, in order to better fuse the two image features, we first perform adaptive mean pooling on the two input features $F_s \in \mathbb{R}^{64\times n_1\times c_2}$ and $F_l \in \mathbb{R}^{64\times n_2\times c_2}$, downsampling the sequence of n_1 and n_2 to the same length n and obtain $F'_s \in \mathbb{R}^{64\times n\times c_2}$ and $F'_l \in \mathbb{R}^{64\times n\times c_2}$. Next, extend the dimensions of features $F'_s \in \mathbb{R}^{64\times n\times c_2}$ and $F'_l \in \mathbb{R}^{64\times n\times c_2}$ after adaptive pooling to obtain features $F''_s \in \mathbb{R}^{64\times 1\times n\times c_2}$ and $F''_s \in \mathbb{R}^{64\times 1\times n\times c_2}$. Then, stack along the extended dimensions to obtain the feature $F_z \in \mathbb{R}^{64\times 2\times n\times c_2}$. Finally, softmax activation is performed on the stacked features along the

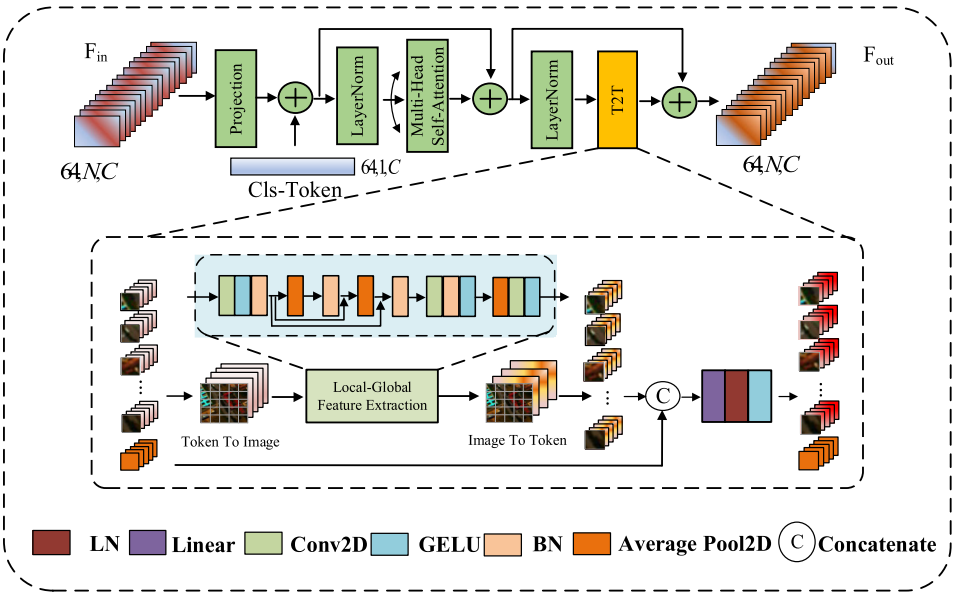


Fig. 5. Structure diagram of T2T.

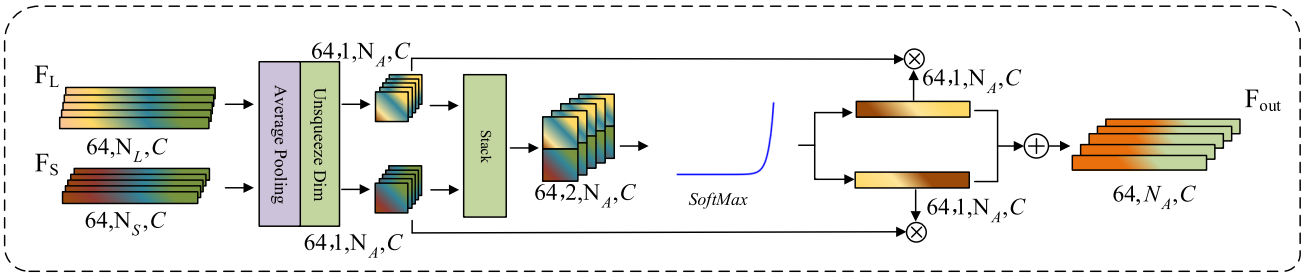


Fig. 6. Structure of PAFM.

stacking dimension to obtain the weight matrices of the two features, and the two matrices are weighted and fused with the corresponding features after adaptive pooling. The PAFM process can be represented as

$$F_z = \text{Stack}\left[\zeta((\phi_{z=\min\{n_1,n_2\}}(F_s))), \zeta((\phi_{z=\min\{n_1,n_2\}}(F_L)))\right] \tag{18}$$

$$F = \frac{e^{F_z}}{\sum_{i=1}^{n} e^{F_i}} \otimes (\phi(F_s) + \phi(F_L)) \tag{19}$$

where $\phi_{z=\min\{n_1,n_2\}}(\cdot)$ represents the adaptive mean pooling function, the pooled dimension is the minimum of n_1 and n_2, $\zeta(\cdot)$ represents the dimension extension function, Stack$(\cdot)$ is a tensor stacking operation, and $\otimes$ represents feature weighting.

E. Implementation Details

In this section, the implementation details of the proposed network DBMST are provided. Taking the Indian Pines dataset as an example, the size of the Indian Pines dataset is $145 \times 145 \times 200$. First, the output size of the data after PCA preprocessing and 3-D cube partitioning is $11 \times 11 \times 30$, obtaining the input $X \in \mathbb{R}^{64\times1\times11\times11\times30}$ of the network. Then, input X into Conv3D with 64 convolutional kernels and a size of $1 \times 1 \times 30$ for spectral feature extraction to obtain $F_x \in \mathbb{R}^{64\times64\times11\times11}$. Next, $F_x \in \mathbb{R}^{64\times64\times11\times11}$ is input into the small-scale branch and the large-scale branch, respectively. In the small-scale branch, the features are first divided into 1×1 small-scale token to obtain $F_{1\times1} \in \mathbb{R}^{64\times n_1\times c_1}$. Then, input the obtained small-scale token into the S-Transformer. The T2T in S-Transformer converts token to Image, extracts and enhances features through convolution and pooling, and then converts image to token. Finally, small-scale branching is used to obtain feature $F_s \in \mathbb{R}^{64\times n_1\times c_2}$. In large-scale branches, first, CS3M divided feature $F_x \in \mathbb{R}^{64\times64\times11\times11}$ into large-scale tokens with a scale of 5×5 and shrunk the channels to obtain $F_{5\times5} \in \mathbb{R}^{64\times n_2\times c_1'}$. Then, the large-scale token $F_{5\times5} \in \mathbb{R}^{64\times n_2\times c_1'}$ is input into the L-Transformer, which is worth noting as a traditional Transformer network used for global information modeling. Finally, feature $F_l \in \mathbb{R}^{64\times n_2\times c_2}$ is obtained by large-scale branching. Two branches of tokens with different scales are extracted into features $F_s \in \mathbb{R}^{64\times n_1\times c_2}$ and $F_l \in \mathbb{R}^{64\times n_2\times c_2}$, and then, the two features are fused through the PAFM module to obtain the fused features. Finally, the predicted category of samples is output through a linear layer. The process of the proposed DBMST is described in Algorithm 1.

III. Experimental Results and Analysis

A. Dataset Description

To verify the generalization ability of the proposed DBMST, three common datasets were used for some experiments, including the Indian Pines dataset, the Pavia dataset, and the

Algorithm 1 Implementation Process of DBMST

Input: HSI image data $X \in \mathbb{R}^{H\times W\times L}$ with label $Y \in \mathbb{R}^{H\times W}$, PCA parameter $b = 30$, and spatial size $s = 11$. Set the Adam optimizer and learning rate r=0.0005, select batch size B=64 and training iterations T=200

Output: The classification accuracy and visual classification map of each category.

1. Firstly, perform PCA processing on HSIs data, then perform edge filling, slice and extract cubes according to the spatial size s, and obtain the processed data $x \in \mathbb{R}^{64\times 1\times 11\times 11\times 30}$.
2. **for** T=1 to 200 do
3. Select x to execute Conv3D and obtain $F_x \in \mathbb{R}^{64\times 64\times 11\times 11}$
4. Divide $F_x \in \mathbb{R}^{64\times 64\times 11\times 11}$ into tokens on a scale of 1×1 in small-scale branches to obtain $F_{1\times 1} \in \mathbb{R}^{64\times n_1\times c_1}$.Among them, the first 64 represents the number of batches, and the second 64 represents the number of channels.
5. Input feature $F_{1\times 1} \in \mathbb{R}^{64\times n_1\times c_1}$ into S-Transformer
6. Transform between Token and Image using the T2T module in S-Transformer to extract local features and enhance global features
7. S-Transformer output feature $F_s \in \mathbb{R}^{64\times n_1\times c_2}$
8. In the large-scale branch, $F_x \in \mathbb{R}^{64\times 64\times 11\times 11}$ is divided into large-scale Token $F_{5\times 5} \in \mathbb{R}^{64\times n_2\times c_1'}$ by module CS3M according to a scale of 5×5. Among them, the first 64 represents the number of batches, and the second 64 represents the number of channels
9. Large scale Token $F_{5\times 5} \in \mathbb{R}^{64\times n_2\times c_1'}$ is input into L-Transformer to obtain output feature $F_l \in \mathbb{R}^{64\times n_2\times c_2}$
10. $F_s \in \mathbb{R}^{64\times n_1\times c_2}$ and $F_l \in \mathbb{R}^{64\times n_2\times c_2}$ are fused through PAFM
11. The fused features are extracted through linear layers
12. Output classification labels
13. **end for**
14. Save the parameters of the optimal model, obtain the classification accuracy of labeled samples, and visualize the classification map of feature categories.

TABLE I
CATEGORY NAMES AND NUMBER OF DATA SAMPLES DIVIDED IN THE INDIAN PINES DATASET

Indian Pines			
Class	Class name	Training	Test
1	Alfalfa	4	42
2	Corn-notill	142	1286
3	Corn-mintill	82	748
4	Corn	23	214
5	Grass-pasture	48	435
6	Grass-trees	72	658
7	Grass-pasture-mowed	3	25
8	Hay-windrowed	47	431
9	Oats	3	17
10	Soybean-notill	97	875
11	Soybean-mintill	245	2210
12	Soybean-clean	59	534
13	Wheat	20	185
14	Woods	126	1139
15	Bldg-Grass-Tree-Drivers	38	348
16	Stone-Steel-Towers	9	84
/	Total	1018	9231

TABLE II
CATEGORY NAMES AND NUMBER OF DATA SAMPLES DIVIDED IN THE PAVIA DATASET

Pavia			
Class	Class name	Training	Test
1	Asphalt	66	6565
2	Meadows	186	18463
3	Gravel	20	2079
4	Trees	30	3034
5	Painted metal sheets	13	1332
6	Bare Soil	50	4979
7	Bitumen	13	1317
8	Self-blocking bricks	36	3646
9	Shadows	9	938
/	Total	423	42353

Salinas dataset. The category names and data divisions of all datasets are shown in Tables I–IV, respectively.

1) Indian Pines Dataset: The HSIs' data captured by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor in 1992 included 145 × 145 pixels and 224 spectral bands, leaving 200 wavebands after removing the water absorption and low signal-to-noise ratio bands.

2) Pavia Dataset: The dataset is obtained by the catoptrics spectral image system (ROSIS-3), which contains 115 spectral bands with wavelengths ranging from 0.43 to 0.86 μm The image space size is 610 × 340, including nine types of ground cover, with 103 bands remaining after removing the water absorption band and low signal-to-noise ratio band.

3) Salinas Dataset: Over the Salinas Valley, the HSI data captured by the AVIRIS sensor have an image space size of 512 × 217, containing 224 spectral bands. Remove the noise bands 108–112, 154–167, and 224, and there are still 200 spectral bands left. Salinas has a spatial resolution of 3.7 m and includes 16 types of ground cover.

4) Houston2013 Dataset: The Houston 2013 dataset contains 15 types of land cover, collected by the HSI analysis team and NCALM on the University of Houston campus and nearby urban areas. A total of 144 spectral bands are included, and the image space size is 349 × 1905.

B. Experimental Setup

1) Evaluation Indicators: For HSI classification, there are three commonly used performance evaluation indicators, namely, overall classification accuracy (OA), average accuracy (AA), and kappa coefficient (kappa). Assume that the confusion matrix $H = (a_{i,j})_{n\times n}$, where n is the number of categories and $a_{i,j}$ is the number of categories j, is divided into i. The calculation of OA is

$$\mathrm{OA} = \frac{\sum_{i=1}^{n} a_{i,j}}{M} \times 100\% \tag{20}$$

where M represents the total number of samples and OA represents the percentage of accurately classified samples to the total number of samples.

TABLE III
CATEGORY NAMES AND NUMBER OF DATA SAMPLES DIVIDED FOR THE SALINAS DATASET

Salinas			
Class	Class name	Training	Test
1	Brocoil-green-weeds_1	20	1989
2	Brocoil-green-weeds_2	37	3689
3	Fallow	19	1957
4	Fallow-rough-plow	13	1381
5	Fallow-smooth	26	2652
6	Stubble	39	3920
7	Celery	35	3544
8	Grapes-untrained	112	11159
9	Soil-vinyard-develop	62	6141
10	Corn-senesced-green-weeds	32	3246
11	Lettuce-romaine-4wk	10	1056
12	Lettuce-romaine-5wk	19	1908
13	Lettuce-romaine-6wk	9	907
14	Lettuce-romaine-7wk	10	1060
15	Vinyard-untrained	72	7198
16	Vinyard-vertical-trellis	18	1789
/	Total	533	53596

TABLE IV
CATEGORY NAMES AND NUMBER OF DATA SAMPLES DIVIDED FOR THE HOUSTON2013 DATASET

Houston2013			
Class	Class name	Training	Test
1	Healthy Grass	63	1188
2	Stressed Grass	63	1191
3	Synthetic Grass	35	662
4	Trees	62	1182
5	Soil	62	1180
6	Water	16	309
7	Residential	63	1205
8	Commerical	62	1182
9	Road	63	1189
10	Highway	61	1166
11	Railway	62	1173
12	Parking Lot 1	62	1171
13	Parking Lot 2	23	446
14	Tennis Court	21	407
15	Running Track	33	627
/	Total	751	14278

The average classification accuracy represents the average classification accuracy of each category, and the calculation of AA is

$$\mathrm{AA} = \frac{1}{n}\sum_{i=1}^{n}\frac{a_{i,j}}{\sum_{j=1}^{n} a_{i,j}}. \tag{21}$$

The calculation of the kappa matrix is

$$\mathrm{k}appa = \frac{\sum_{i=1}^{n} a_{i,i} - \frac{\sum_{i=1}^{n}(a_{i,\_}a_{\_,j})}{M}}{M - \frac{\sum_{i=1}^{n}(a_{i,\_}a_{\_,j})}{M}} \tag{22}$$

where $a_{i,\_}$ and $a_{\_,j}$ denote all the column elements of row i and all the row elements of column j, respectively.

2) Comparison Methods: In order to evaluate the effectiveness of the proposed method, some advanced HSI classification methods are chosen for comparison. The classification methods based on CNNs include LS2CM-Res [34], FADCNN [60], HybridSN [35], PyResNet [36], and MCRSCA [37]. Transformer-based classification methods include VIT [52], SSTN [47], SSFTT [45], SpectralFormer [43], HIT [42], GAHT [44], and CTMixer [46]. HybridSN is a hybrid CNN network that combines 2DCNN and 3DCNN. PyResNet is a residual classification network composed of three pyramid bottleneck residual blocks and convolutional layers. MCRSCA is an HSI classification method based on MCRSCA. Unlike the above methods, LS2CM-Res is a lightweight classification method that replaces the convolutional layer with a lightweight spectral space convolutional module (LS2CM). VIT is a classic visual Transformer classification network. SSTN is a spectral space Transformer, which determines the hierarchical operation selection and block level order of the network through the factorization architecture search (FAS) framework. SpectralFormer rethinks the HSIs' classification problem from the perspective of spectral sequence attributes and constructs a Transformer-based classification network. Unlike the Transformer-based methods mentioned above, HIT and CTMixer are hybrid classification methods that combine CNN and Transformer. GAHT proposed a grouped pixel embedding module and constructed a Transformer classification network in a hierarchical manner.

3) Implementation Details: The method proposed in this article is implemented on the Python platform and uses a desktop PC with an Intel[1] Core[2] i9-9900K CPU, NVIDIA GeForce RTX 3090TiGPU, and 128-GB random access memory. The Adam optimizer is used, and the batch size, initial learning rate, and training rounds are set to 64, 5e-3, and 200, respectively.

For a fair comparison, all experiments in this article were conducted in the same experimental environment, and all experimental results were taken as the average of 20 experiments.

C. Model Analysis

1) Some Ablation Experiments:

a) Some ablation experiments for PAFM module: Compared to single-scale Transformers, multiscale Transformers contain more abundant classification features. Due to the different dimensions of feature extraction by branches at different scales, additive fusion cannot be performed. To solve the problem of branch feature fusion at different scales, we propose a PAFM module that can assign weights to the features of different scale branches and perform additive fusion. In order to verify the effectiveness of PAFM, this article conducted some ablation experiments on four datasets: Indian Pines, Salinas, Pavia, and Houston2013. The results are shown in Table V. From the table, it can be seen that on the Indian Pines dataset, the average classification accuracy with PAFM

[1]Registered trademark.
[2]Trademarked.

is 1.33% higher than that without PAFM. For the Salinas dataset, without PAFM, the average classification accuracy would decrease by 0.44%. For the Pavia dataset, compared to the absence of PAFM, the classification with PAFM improved by 1.44%. For the Houston2013 dataset, compared to the absence of PAFM, the classification with PAFM improved by 0.35%.In summary, the ablation experiments of the PAFM module on four datasets have demonstrated the effectiveness of PAFM.

b) Some ablation experiments of the proposed DBMST: The DBMST proposed in this article is a multiscale Transformer network, which mainly includes PAFM, T2T, and CS3M. CS3M is used to implement large-scale token division, shrink channel information, and avoid redundancy. T2T replaces the FFN of the S-Transformer and is used to extract contextual information. PAFM is used to solve the problem of the fusion of features extracted from different scale branches. To verify the validity of the above components, we conducted an ablation study on three commonly used datasets. The results of the ablation experiments are shown in Table VI. In the first case, the network has only a single-scale token as input, and the final classification accuracy is the worst on all four datasets. In the second case, the network is a multiscale token as input, and the classification effect is substantially improved on all four datasets, with the largest improvement of 3.56% on the Pavia dataset. In the third case, the network contains multiscale strategy, PAFM and T2T modules, and the classification accuracy is improved slightly in the four datasets. In the fourth case, the network contains the multiscale strategy, PAFM and CS3M modules, and the average classification accuracy OA is improved by 0.65%, 0.4%, 0.49%, and 0.10% on the Indian Pines, Salinas, Pavia, and Houston2013 datasets, respectively. In the last case, the network achieves optimal classification on all three datasets when the network contains all the components. The results of the ablation experiments fully demonstrate the effectiveness of the above components.

2) Parameter Sensitivity Analysis: For deep learning networks, parameter settings have an impact on the performance of the network. Among them, the learning rate and batch size directly determine the effectiveness of weight updates. Analyzed from the perspective of model optimization, the learning rate and the batch size are important parameters that affect the convergence of the model. The learning rate determines the convergence of the model, and the batch size affects the generalization performance of the model. To explore the optimal combination of learning rate and batch size suitable for DBMST, we conducted experiments on four datasets with different combinations of learning rates and batch sizes, where the chosen learning rate combination is $\{1e-4, 5e-4, 1e-3, 5e-3\}$ and the batch size combination is {128, 64, 32, 16}. The experimental results are shown in Fig. 7.

For the Indian Pines dataset, a trend of increasing and then decreasing OA values can be observed as the batch size increases. The maximum value is taken at the batch size of 64, as shown in Fig. 7(a). In the Salinas dataset, it shows a phenomenon that the OA value becomes larger and then smaller as the learning rate becomes larger. The optimal OA result can be taken at a learning rate of $5e-4$, as shown in Fig. 7(b). In the Pavia dataset, the optimal combination of batch size and learning rate is 64 and $5e-4$, as shown in Fig. 7(c). In the Houston 2013 data, it can be observed that when the learning rate is fixed, the OA value first increases and then decreases with the increase in the number of batches. The optimal result is achieved when the number of batches is 64, as shown in Fig. 7(d). In summary, we choose $5e-4$ and 64 as the learning rate and batch size of DBMST.

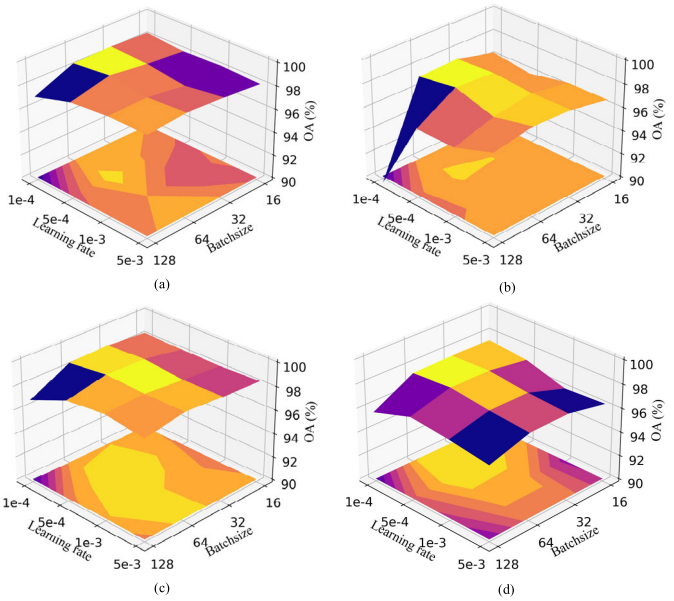


Fig. 7. Experimental results for different combinations of learning rates and batch sizes on four datasets. (a) Indian Pines dataset. (b) Salinas dataset. (c) Pavia dataset. (d) Houston 2013 dataset.

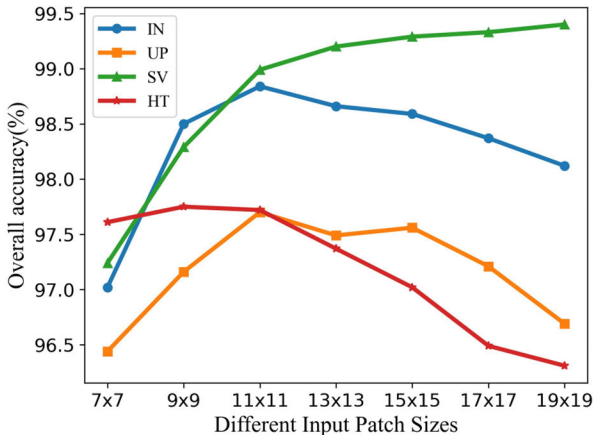


Fig. 8. Impact of different input space sizes on classification accuracy.

3) Different Input Space Sizes: For the HSI classification task, the input to the network is usually a 3-D cube obtained by slicing. Therefore, there is also a large impact of different input space sizes on the classification performance. In order to explore the optimal input space size of the network on different datasets, some experiments were carried out on these datasets. The input space sizes chosen for the experiments are 7×7, 9×9, 11×11, 13×13, 15×15, 17×17, and 19×19, respectively. The experimental results are shown in Fig. 8.

For the Indian Pines dataset, the OA values show a tendency to increase and then decrease as the input network

TABLE V
IMPACT OF THE PROPOSED PAFM MODULE ON OA ON DIFFERENT DATASETS (%)

With or Without PAFM	Indian Pines	Salinas	Pavia	Houston
With PAFM	98.84	98.97	97.70	97.72
Without PAFM	97.51	98.53	96.26	97.37

TABLE VI
IMPACT OF DIFFERENT MODULES ON NETWORK OA VALUES (%)

case	Signal Scaled	Multi-Scaled	PAFM	T2T	CS3M	Indian Pines	Salinas	Pavia	Houston
1	√	–	-	-	-	94.77	97.66	93.38	97.38
2	-	√	√	-	-	97.90	98.24	96.94	97.44
3	-	√	√	√	-	98.07	98.30	97.08	97.58
4	-	√	√	-	√	98.55	98.74	97.43	97.54
5	-	√	√	√	√	98.84	98.97	97.70	97.72

TABLE VII
CLASSIFICATION RESULTS ON THE INDIAN PINES DATASET (THE BEST CLASSIFICATION RESULTS ARE BOLDED)

Methods	CNNs					Transformer							
	LS2CM-Res	PyResNet	Hybrid-SN	MCRSCA	FADCNN	VIT	SF	SSTN	HIT	GAHT	CTMixer	SSFTT	Proposed
OA(100%)	97.73	91.87	95.83	92.43	96.60	93.10	81.52	98.04	90.46	83.82	98.07	98.07	**98.84**
AA(100%)	96.46	92.22	95.19	89.11	83.73	94.49	79.26	92.02	91.96	87.02	97.64	97.13	**98.24**
k×100	97.41	90.72	95.24	91.35	96.13	92.12	78.82	97.76	89.09	81.46	97.81	97.79	**98.68**
1	94.35	96.02	99.16	76.76	53.44	95.45	62.24	80.00	97.36	90.44	98.73	**99.43**	99.18
2	98.01	87.69	91.86	91.70	97.49	91.03	77.66	98.61	85.37	79.73	98.52	97.93	**98.79**
3	97.41	91.87	94.00	89.83	96.96	91.84	76.88	97.09	87.71	78.75	**99.54**	96.74	98.82
4	98.46	94.76	95.70	79.26	88.02	94.87	67.68	**99.64**	91.84	87.50	99.37	98.26	99.28
5	97.02	89.75	99.11	94.77	86.45	94.62	87.65	97.32	95.35	89.91	97.66	**99.13**	98.27
6	98.29	96.05	98.49	99.16	98.20	98.43	90.11	98.37	93.57	86.00	98.99	98.9	**99.29**
7	85.70	89.85	92.67	80.45	23.89	92.58	79.95	88.44	91.46	91.42	93.83	90.51	**95.37**
8	99.75	98.28	98.66	99.79	98.44	97.87	89.41	96.40	97.48	92.24	**99.95**	99.39	99.51
9	97.06	86.85	85.06	69.38	30.00	**100.0**	66.95	30.00	94.26	95.28	96.64	93.04	95.30
10	95.60	93.68	93.56	88.10	96.99	91.73	78.30	97.67	89.59	81.14	94.99	97.32	**98.02**
11	98.58	92.52	97.15	93.45	98.39	91.23	80.53	98.31	89.83	81.87	98.78	98.15	**99.36**
12	95.50	87.62	95.13	79.83	94.84	89.31	71.34	96.90	85.78	73.67	96.15	97.40	**97.56**
13	99.45	94.94	99.45	99.15	97.24	99.09	91.24	99.88	96.35	93.26	99.34	99.06	**99.64**
14	**99.29**	97.20	98.11	97.82	98.93	96.02	91.07	98.79	94.88	92.56	99.22	**99.32**	99.14
15	95.16	93.63	95.67	89.45	86.28	92.98	76.82	97.41	91.18	82.13	95.60	97.44	**98.82**
16	93.70	84.79	89.26	96.89	94.09	94.82	91.17	**97.47**	89.48	96.43	95.00	92.04	95.57

size increases. The peak of OA is reached when the input space size is 11 × 11. For the Pavia dataset, the input space becomes larger and two local extreme points appear for OA values, which are input space sizes of 11 × 11 and 15 × 15. From the overall point of view, the OA value shows a trend of increasing and then decreasing, with the maximum value at the extreme point 11 × 11. For the Salinas dataset, the OA values increase as the input space becomes larger. For the Houston 2013 dataset, the OA values show a tendency to increase and then decrease as the input network size increases. The peak of OA is reached when the input space size is 11 × 11. From Fig. 7, we can see that the growth rate of OA values before 11 × 11 is fast and the growth of OA values after 11 × 11 becomes slow. In summary, we choose the input of 11 × 11 space size as the input of the network for all four datasets.

4) Different Scales' Token: The proposed DBMST is a multiscale Transformer, and different scales of tokens also have an impact on the classification effect of the network. To explore the optimal scale token, some experiments were conducted on four datasets. The token scale size chosen for the experiment is $\{3 \times 3, 5 \times 5, 7 \times 7, 9 \times 9\}$. The results are shown in Fig. 9. For the Indian Pines dataset, the OA value first increases and then decreases as the adopted token scale becomes larger. The OA reaches its maximum value when the token scale is 5 × 5. For the Pavia dataset, it is obvious that the best classification accuracy is obtained when the token scale is at 5 × 5. For the Salinas dataset, the OA decreases as the scale of token increases, but the OA at the scales of 3 × 3 and 5 × 5 does not have a large difference. For the Houston2013 dataset, the OA value first increases and then decreases as the adopted token scale becomes larger. The OA

TABLE VIII
CLASSIFICATION RESULTS ON THE PAVIA DATASET (THE BEST CLASSIFICATION RESULTS ARE BOLDED)

Methods	CNNs					Transformer							
	LS2CM-Res	PyResNet	Hybrid-SN	MCRSCA	FADCNN	VIT	SF	SSTN	HIT	GAHT	CTMixer	SSFTT	Proposed
OA(100%)	97.08	88.82	94.46	94.91	93.05	90.21	81.16	92.62	86.62	86.68	96.82	96.57	**97.69**
AA(%)	96.10	89.05	92.57	92.43	89.35	88.52	77.5	89.22	82.22	84.6	95.78	95.66	**96.82**
k×100	96.12	84.98	92.63	93.23	90.72	86.89	74.33	90.11	82.07	82.09	95.79	95.44	**96.94**
1	96.17	83.75	93.64	94.67	84.79	87.50	82.84	84.29	81.98	85.61	97.10	96.59	**97.35**
2	98.70	93.39	97.91	98.90	97.87	93.47	83.20	97.67	92.50	90.28	99.20	98.43	**99.21**
3	90.99	76.76	82.36	79.93	75.69	73.05	47.41	75.39	56.78	61.50	84.33	92.29	**92.76**
4	97.35	95.49	96.95	95.78	95.09	94.98	85.35	95.54	96.76	96.74	96.90	**97.68**	97.47
5	98.18	95.58	97.11	**99.67**	97.81	98.13	95.74	97.16	96.94	98.31	98.45	98.57	98.41
6	98.99	93.66	97.47	90.24	98.54	89.31	77.38	99.37	85.47	83.33	98.56	97.50	**99.03**
7	94.48	90.61	90.84	81.86	84.64	84.62	57.58	78.34	55.89	70.37	95.81	97.08	**98.03**
8	92.04	77.01	81.26	91.01	81.26	82.04	69.12	75.47	77.98	77.54	**95.36**	87.05	91.74
9	97.97	95.19	95.58	**99.80**	88.45	93.61	98.98	**99.80**	95.75	97.77	96.39	95.70	97.43

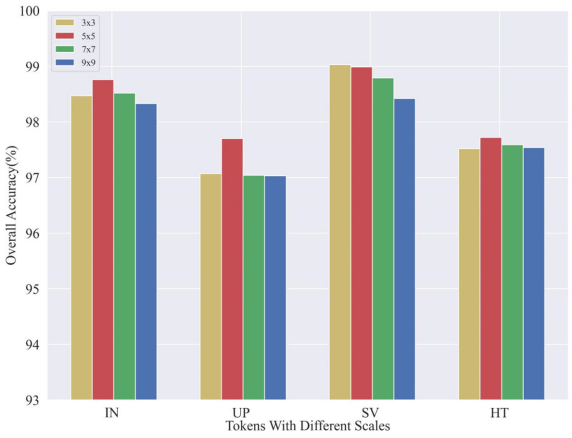


Fig. 9. Impact of the token with different scales on classification accuracy.

reaches its maximum value when the token scale is 5×5. In summary, in the four datasets, we choose the token of 5×5 as the input of the large-scale branch Transformer.

D. Analysis of Results

1) Quantitative Analysis: Tables VII–X give the classification results of OA, AA, kappa, and each category of the Indian Pines, Pavia, Salinas, and Houston2013 datasets, respectively. A rough observation shows that both Transformer and CNN-based methods achieve better classification results. Compared with other methods, the proposed method in this article has the highest OA on the three datasets. Specifically, among the CNN-based methods, 3-D CNN and PyResNet are less effective in classification. The main reason is that the network structure is simple and fewer features with discriminative properties are extracted, so the classification performance of the model is poor. Hybrid-SN combines 2DCNN and 3DCNN for extracting spatial features and spatial–spectral features, respectively, and, finally, achieves better classification results. LS2CM-Res designed an LS2CM instead of standard convolution and obtained better classification results. CTMixer constructs a two-branch network to extract global–local spectral features by combining convolution and Transformer. Similar to CTMixer, SSFTT is also a Transformer-based classification method. SSFTT designed a Gaussian feature weight module to extract advanced semantic features. Among the Transformer-based methods, SSFTT, SSTN, and CTMixer achieved better classification results on all four datasets, and the classification accuracy on the Indian Pines dataset was higher than most of the CNN-based methods.

Finally, the analysis shows that the classification method proposed in this article, which combines multiscale features with Transformer, finally obtains the best classification results among CNN- and Transformer-based methods. Compared with the best CNNs method for classification in CNNs, the OA values of the proposed method in this article are higher, 1.11%, 0.61%, and 2.05%, respectively, on the four datasets. Compared with the Transformer-based classification method, the OA values are higher than the best classification method on the three datasets, 0.75%, 0.87%, and 3.30%, respectively. On the Indian Pines dataset, the worst classification result is obtained by SF with an OA value of 81.52%. The reason is that SF only considers the spectral information of HSIs for classification. Although the OA of SF is not satisfactory, on the Pavia dataset, SF achieved suboptimal classification results for the ninth category, only lower than the optimal classification accuracy of SSTN and MCRSCA. The classification performance of GAHT is poor on the Indian Pines, Pavia, and Salinas datasets, but the classification accuracy on the Houston dataset is higher than the three classification methods of MCRSCA, SF, and HIT. In the Indian Pines dataset, the proposed method in this article achieves the best results of all compared methods in nine categories, including Corn-notill and Grass-trees. In the Pavia dataset, DBMST achieved the highest classification accuracy in six categories compared to other methods. In the Salinas dataset, our proposed method, compared to other comparison methods, achieves the highest classification accuracy in 11 categories. In categories 5, 13, and 16 of the Salinas dataset, not only is the classification accuracy the highest among all methods, but the OA value is also very close to 100%. In the Houston 2013 dataset, compared with other methods, the six subcategories proposed in this article achieved the best classification accuracy. This also fully proved that the multiscale Transformer proposed in

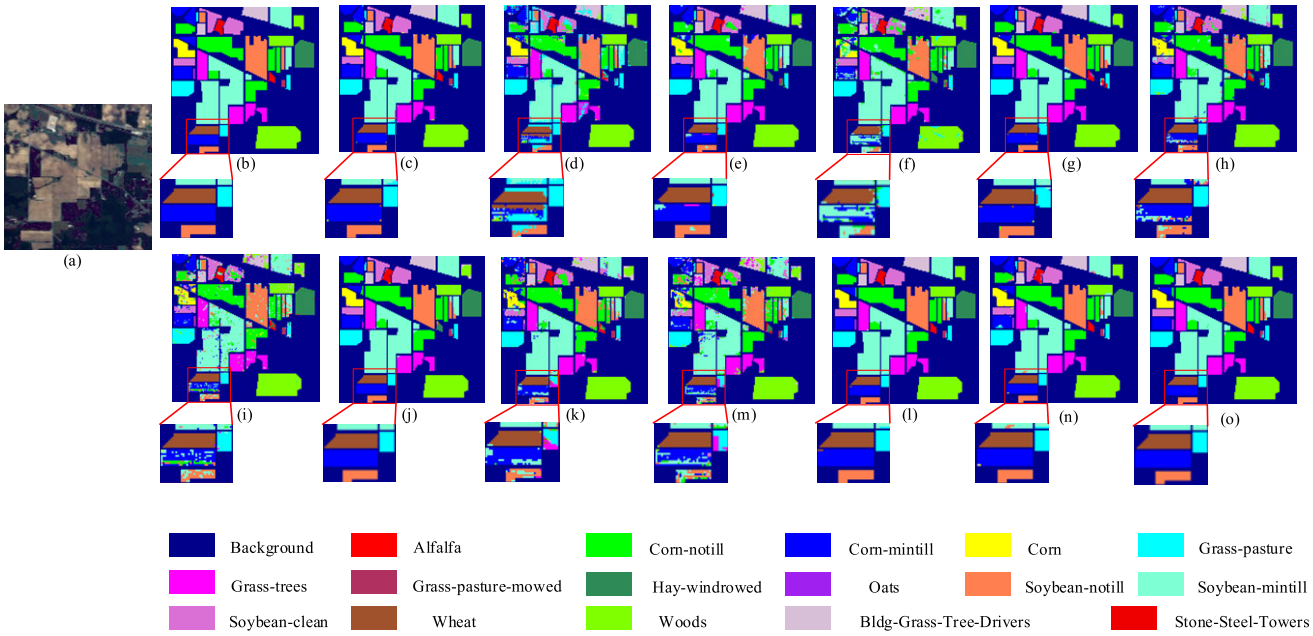


Fig. 10. Classification maps of each method on the Indian Pines dataset, with OA values in brackets. (a) Pseudocolor map. (b) Ground-truth map. (c)–(n) LS2CM-Res (97.73%), PyResNet (91.87%), HybridSN (95.83%), MCRSCA (92.43%), FADCNN(96.60), VIT (93.10%), SF (81.52%), SSTN (98.04%), HIT (90.46%), GAHT (83.82%), CTMixer (98.07%), and SSFTT (98.07), respectively. (o) DBMST (98.84%).

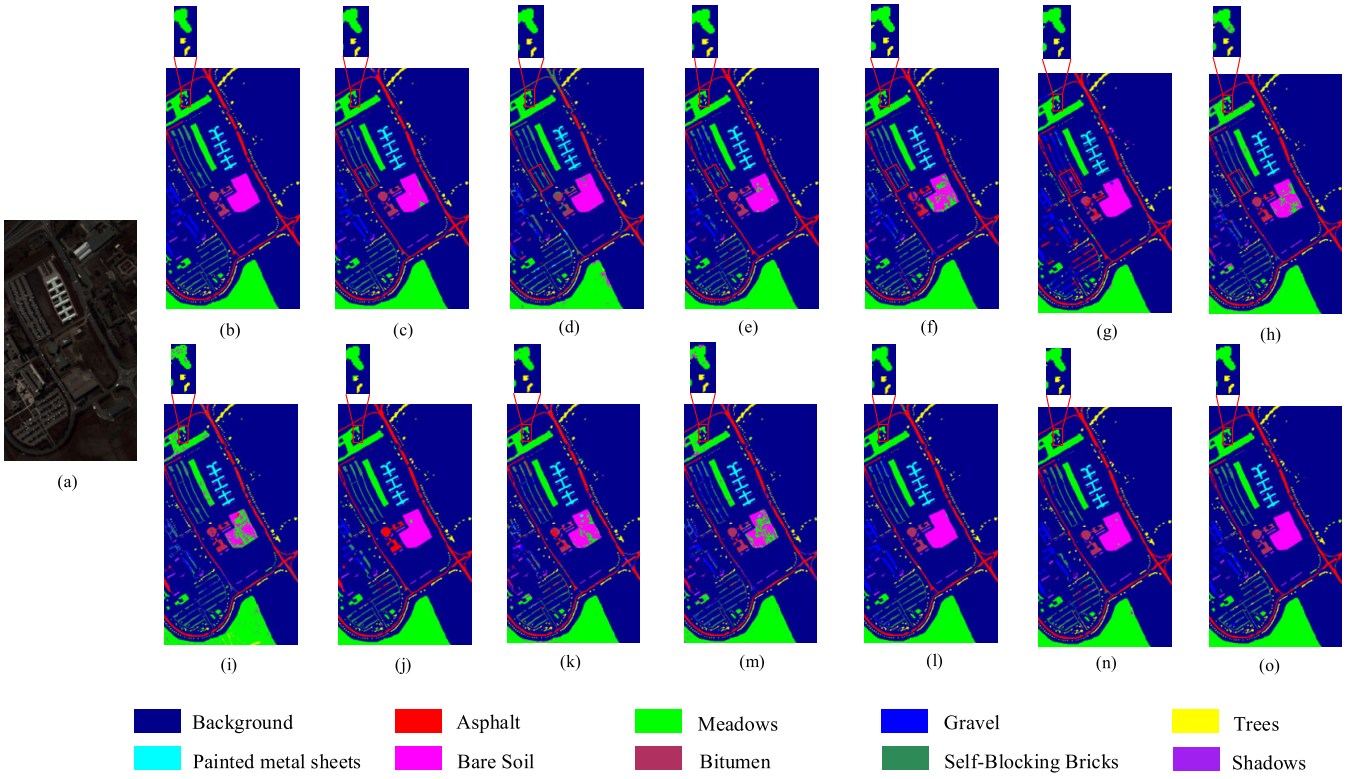


Fig. 11. Classification maps of each method on the Pavia dataset, with OA values in brackets. (a) Pseudocolor map. (b) Ground-truth map. (c)–(n) LS2CM-Res (97.47%), PyResNet (88.82%), HybridSN (94.62%), MCRSCA (94.91%), FADCNN (93.05), VIT (90.21%), SF (81.16%), SSTN (92.62%), HIT (86.62%), GAHT (86.68%), CTMixer (96.82%), and SSFTT (96.57), respectively. (o) DBMST (97.69%).

this article, by combining multiscale features and Transformer, can extract more discriminative features.

2) Visual Assessment: Figs. 10–13 show the maps of classification results for all methods on the Indian Pines, Pavia, Salinas, and Houston2013 datasets, respectively. The results show that the classification map of DBMST proposed in this article is closest to the real feature results. Due to the good local feature extraction ability of CNNs, it is not difficult to find that some CNN-based classification result maps are smoother, such as LS2CM-Res and HybridSN. Due to the LS2CM model design in LS2CM-Res and the network design of HybridSN combined with 2DCNN and 3DCNN, both of

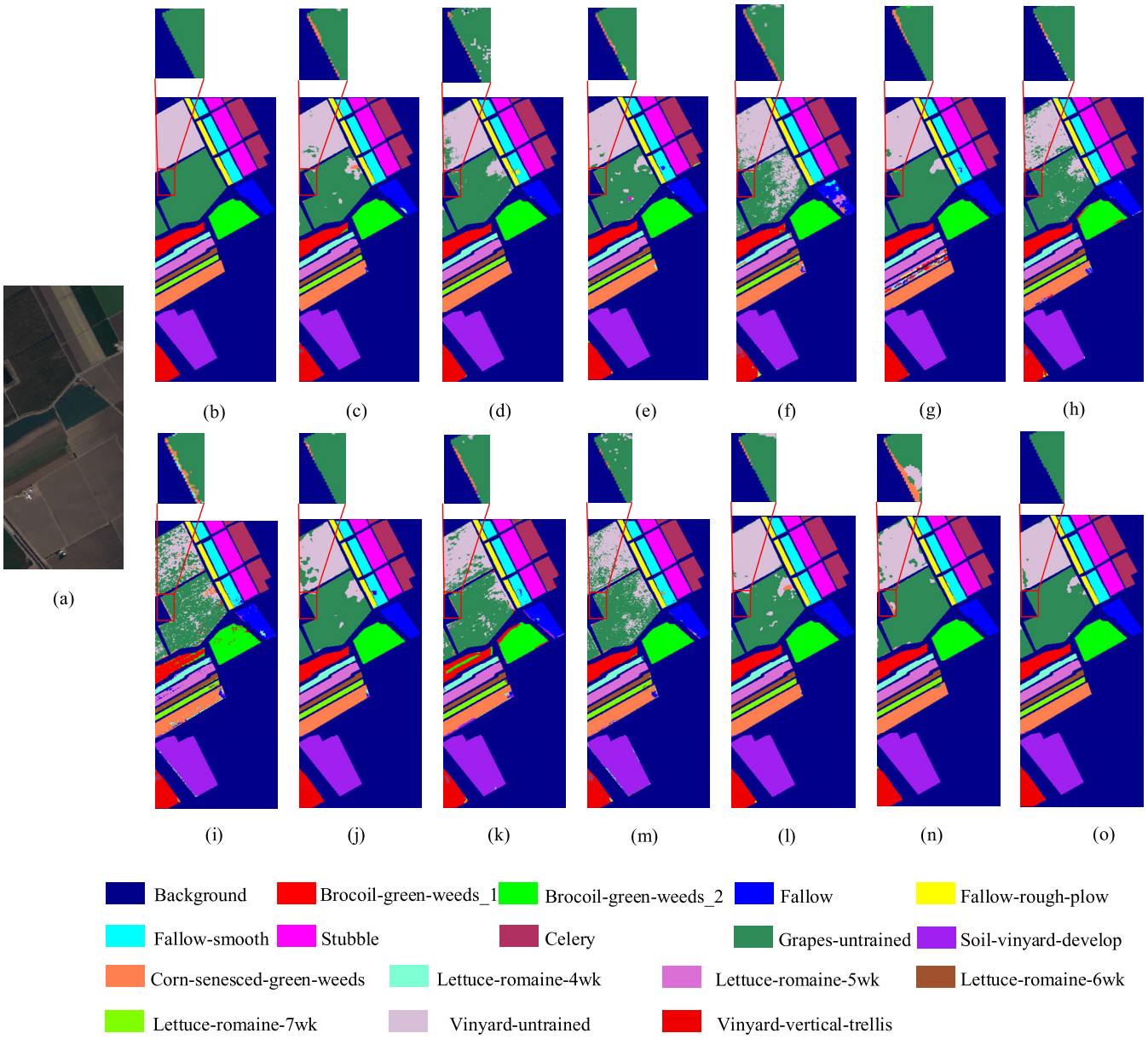


Fig. 12. Classification maps of each method on the Salinas dataset. (a) Pseudocolor map. (b) Ground-truth map. (c)–(n) LS2CM-Res (96.92%), PyResNet (96.18%), HybridSN (97.94%), MCRSCA (91.80%), FADCNN(97.86), VIT (93.30), SF (88.01%), SSTN (95.67%), HIT (93.51%), GAHT (91.50%), CTMixer (95.67%), and SSFTT (98.25), respectively. (o) for DBMST (98.97%).

them are able to extract rich spectral–spatial information. Transformer is able to acquire global dependencies and extract low-frequency information of images. The importance of multiscale features is ignored in current Transformer-based HSIs' classification methods, and only feature extraction at a single scale is considered, for example, the adjacent parts of Grass-pasture and Soybean-notill in the Indians Pines dataset are easily confused, leading to misclassification. Since the input of the Transformer is a vector, the contextual information of the image is not available, so it is unfriendly to classify some small targets, such as Self-Blocking Bricks and Bitumen in the Pavia dataset. The DBMST proposed in this article, from a multiscale perspective, designs the Transformer framework, converts the input vector of the Transformer into an image in the feature extraction process, and extracts the contextual information of the image, and it can be found through visual analysis that the effectiveness of the method proposed in this article can be verified on all four datasets.

Figs. 14–17 show the visualization of the distribution of t-distributed stochastic neighbor embedding (t-SNE) data for the four methods with the best classification results on different datasets. It can be seen that in the Indian Pines dataset, compared with the other three methods, the proposed method in this article has the best clustering results with large interclass distances, small intraclass distances, and low confusion between classes. On the Pavia dataset, DBMST has a smaller intraclass distance and better classification in the class represented by the orange particles compared to the other three methods. In the Salinas dataset, our proposed method still achieves optimal clustering results compared to other

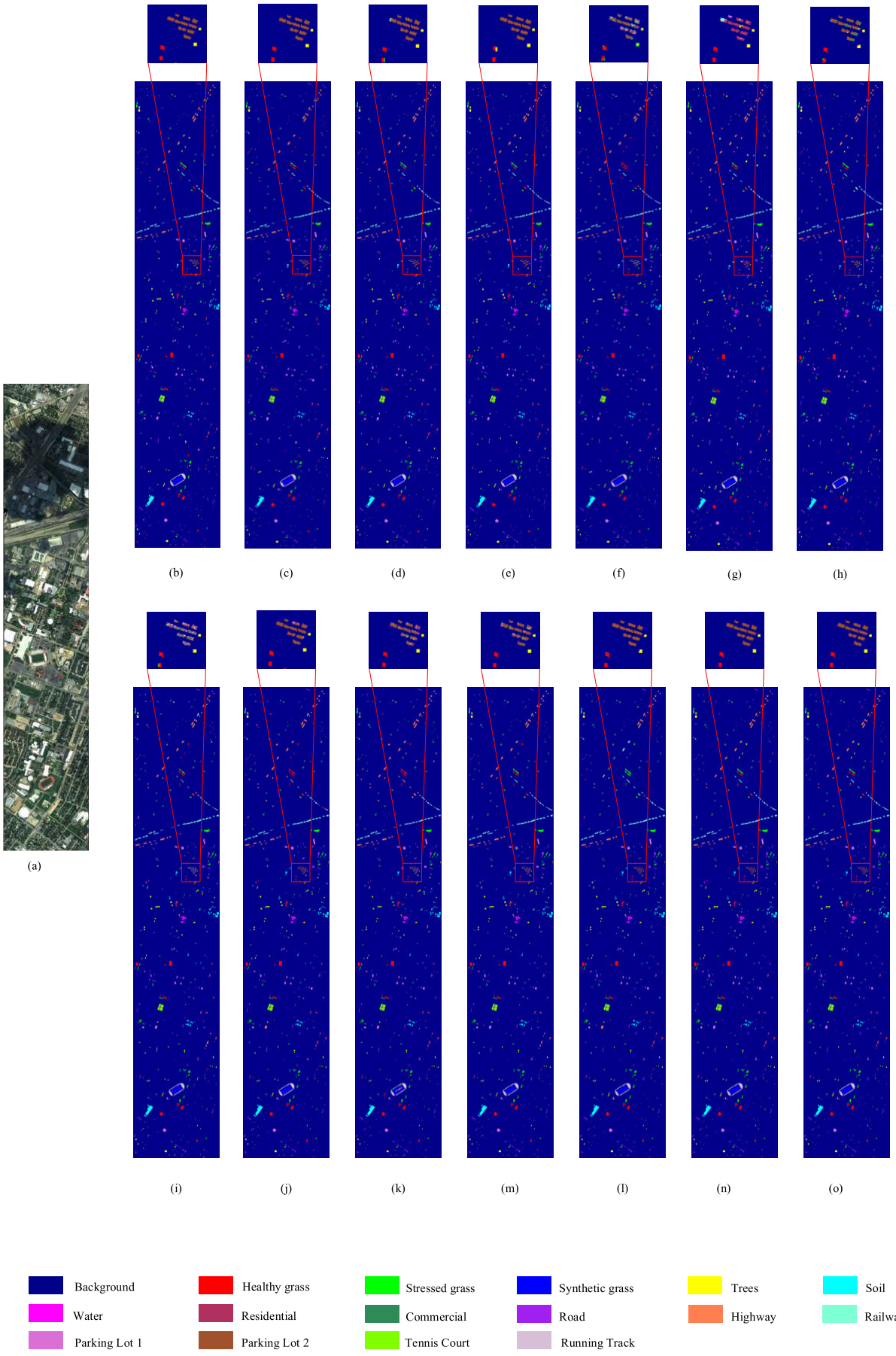


Fig. 13. Classification maps of each method on the Houston dataset. (a) Pseudocolor map. (b) Ground-truth map. (c)–(n) LS2CM-Res (95.66%), PyResNet (94.93%), HybridSN (95.94%), MCRSCA (86.06%), FADCNN (95.64), VIT (94.88), SF (84.83%), SSTN (92.17%), HIT (89.96%), GAHT (89.92%), CTMixer (95.40%), and SSFTT (97.43), respectively. (o) DBMST (97.72%).

TABLE IX
CLASSIFICATION RESULTS ON THE SALINAS DATASET (OPTIMAL CLASSIFICATION RESULTS ARE BOLDED)

Methods	CNNs					Transformer							
	LS2CM-Res	PyResNet	Hybrid-SN	MCRSCA	FADCNN	VIT	SF	SSTN	HIT	GAHT	CTMixer	SSFTT	Proposed
OA(100%)	96.92	96.18	97.94	91.80	97.86	93.30	88.01	95.67	93.51	91.5	95.67	98.25	**98.97**
AA(100%)	97.92	97.35	98.20	93.41	97.37	95.86	92.18	97.52	95.01	94.46	97.96	98.66	**99.18**
k×100	96.57	95.75	97.71	90.86	97.62	92.54	86.63	95.17	92.78	90.55	95.18	98.05	**98.72**
1	**100.00**	99.14	99.24	92.98	98.03	97.47	96.59	**100.00**	96.33	99.09	**100.00**	**100.00**	99.96
2	99.76	99.81	99.56	95.86	98.59	98.54	93.20	99.66	97.07	99.30	98.43	99.36	**99.83**
3	97.22	99.86	99.94	90.24	99.31	96.16	90.55	97.02	94.62	93.82	99.73	**100.00**	99.98
4	96.05	95.12	97.40	**98.91**	97.47	97.85	96.74	97.54	96.14	97.46	97.08	98.03	97.61
5	99.45	98.82	97.60	95.98	98.87	97.14	96.31	98.45	97.98	98.20	99.08	99.17	**99.61**
6	99.95	99.70	99.68	99.91	99.55	98.84	99.29	99.84	98.57	99.44	99.75	**99.97**	99.93
7	99.40	99.71	99.56	99.78	99.58	99.03	96.53	99.65	97.65	96.97	**100.00**	99.72	99.77
8	96.88	95.88	96.93	87.38	97.55	86.66	76.09	91.49	88.80	84.61	95.48	96.60	**98.40**
9	99.52	99.81	99.81	99.27	99.82	99.07	98.21	99.29	97.99	98.79	99.76	99.78	**99.79**
10	96.90	98.67	98.38	91.80	97.04	95.23	86.11	97.67	95.19	91.14	97.40	98.80	**99.42**
11	96.54	98.16	96.14	76.70	97.77	93.24	85.90	96.25	89.75	90.60	97.76	96.04	**98.26**
12	99.26	97.83	99.56	98.98	97.76	98.65	95.56	98.43	97.70	98.39	**99.90**	99.23	99.53
13	98.83	94.58	96.91	97.39	93.20	98.18	94.29	99.16	95.45	96.02	97.57	99.26	**99.58**
14	98.88	96.10	96.40	97.45	89.16	98.52	92.57	99.16	95.27	96.10	**99.02**	97.20	98.37
15	88.35	85.61	94.43	77.06	94.78	81.45	66.38	86.79	84.55	75.39	86.66	95.63	**96.12**
16	99.80	98.72	99.61	94.82	99.51	97.78	97.25	99.95	97.26	96.14	99.75	99.81	**99.93**

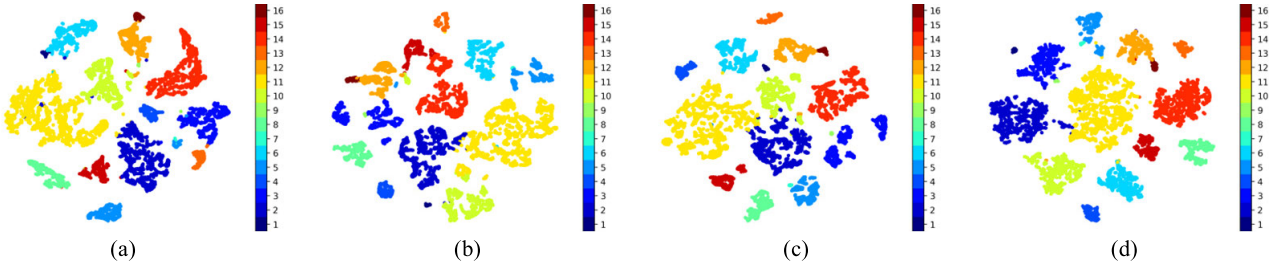


Fig. 14. Visualization of t-SNE data analysis on the Indian Pines dataset. (a)–(d) LS2CM-Res, SSTN, CTMixer, and DBMST, respectively.

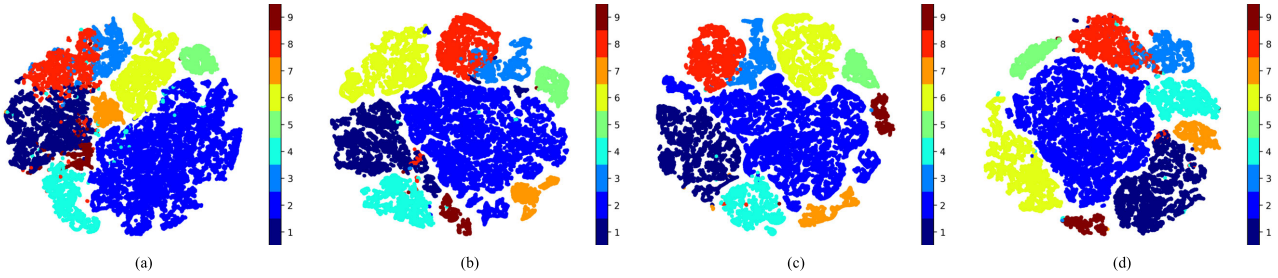


Fig. 15. Visualization maps of t-SNE data analysis on the Pavia dataset. (a)–(d) LS2CM-Res, SSTN, CTMixer, and DBMST, respectively.

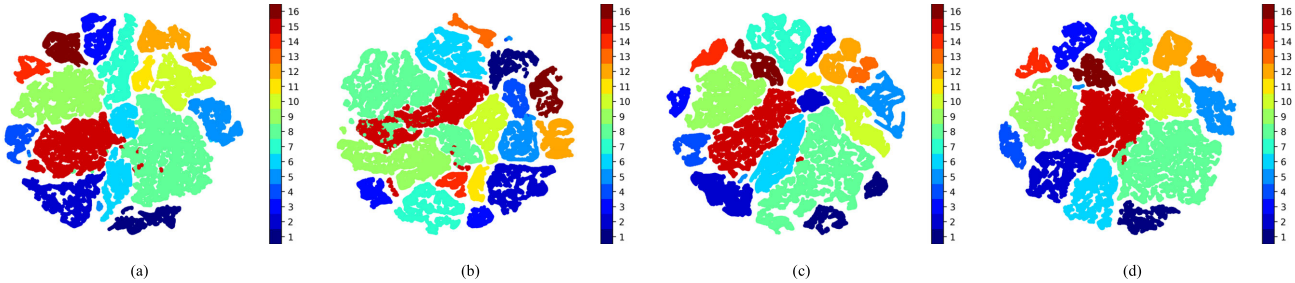
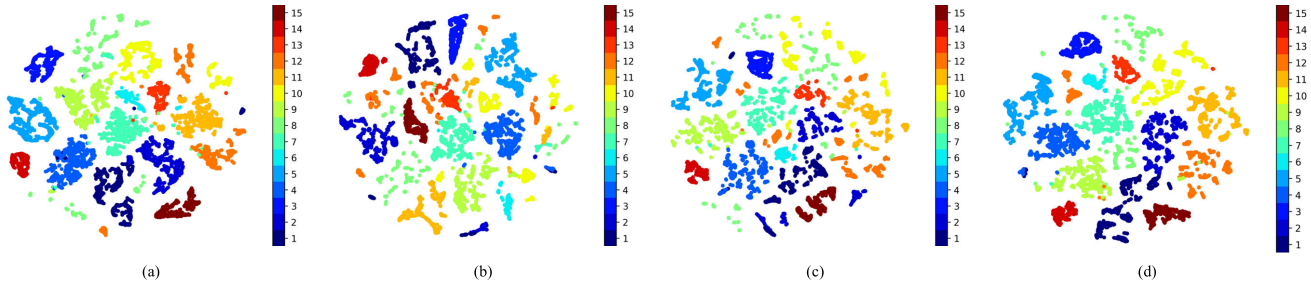


Fig. 16. Visualization of t-SNE data analysis on the Salinas dataset. (a)–(d) LS2CM-Res, SSTN, CTMixer, and DBMST, respectively.

methods. Through observation, it can be found that in the Houston dataset, the intraclass distance of CTMixer is larger than that of the method proposed in this article, while the interclass distance is smaller. The method proposed in this



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

Fig. 17. Visualization of t-SNE data analysis on the Houston dataset. (a)–(d) LS2CM-Res, SSTN, CTMixer, and DBMST, respectively.

TABLE X

CLASSIFICATION RESULTS ON THE HOUSTON DATASET (OPTIMAL CLASSIFICATION RESULTS ARE BOLDED)

Methods	CNNs					Transformer							
	LS2CM-Res	PyResNet	Hybrid-SN	MCRSCA	FADCNN	VIT	SF	SSTN	HIT	GAHT	CTMixer	SSFTT	Proposed
OA(100%)	95.66	94.93	95.54	86.06	95.64	94.88	84.83	92.17	89.96	89.92	95.4	97.43	**97.72**
AA(100%)	96.04	95.34	95.94	87.19	95.12	95.51	86.85	93.63	90.05	90.62	96.27	97.58	**97.77**
k×100	95.31	94.52	95.18	84.92	95.28	94.46	83.58	91.54	89.14	89.1	95.03	97.22	**97.53**
1	92.54	96.52	97.46	90.28	91.45	96.95	88.97	89.00	94.68	95.00	92.07	**98.17**	97.53
2	98.20	97.11	98.10	87.36	98.35	99.02	96.65	94.18	94.76	96.40	95.07	98.72	**99.13**
3	99.82	98.6	99.28	90.63	99.21	99.6	98.53	98.46	96.17	98.31	99.12	99.23	**99.94**
4	94.81	95.79	93.43	**97.65**	95.99	96.68	97.33	98.67	93.64	96.56	96.89	97.58	97.55
5	99.64	98.45	98.48	95.24	97.93	97.25	94.94	99.25	95.39	97.34	99.30	**99.80**	99.56
6	97.73	96.91	98.46	91.74	98.82	98.22	97.98	97.89	86.55	90.33	**99.32**	98.41	98.09
7	94.91	95.64	93.82	87.05	93.11	89.54	78.55	91.44	89.31	83.52	94.55	97.49	**97.70**
8	98.40	92.14	95.29	86.67	86.78	91.27	77.55	95.24	87.46	87.43	98.09	98.00	**99.33**
9	93.66	94.23	94.86	81.49	94.89	89.01	73.68	93.51	83.44	85.35	94.62	95.60	**96.17**
10	91.48	88.56	95.05	77.37	**95.55**	92.41	77.98	76.75	85.86	82.18	89.84	95.26	95.15
11	96.41	96.23	95.36	82.12	96.5	95.52	77.12	93.69	88.09	83.92	97.03	**98.55**	98.04
12	94.54	93.98	92.05	77.25	95.72	94.39	72.09	87.36	83.92	84.86	95.4	95.17	**96.61**
13	92.92	94.98	92.41	83.31	86.05	94.3	79.96	92.91	83.58	91.46	95.83	**96.12**	94.6
14	97.36	96.67	98.36	84.91	98.86	99.31	94.45	98.48	93.39	90.33	**99.39**	98.95	98.35
15	98.18	94.24	96.78	94.78	97.63	**99.20**	96.98	97.55	94.49	96.25	97.53	96.60	98.79

TABLE XI

TRAINING TIME (S) AND TESTING TIME (S) OF EACH METHOD ON THE FOUR DATASETS

Methods		Indian Pines		Pavia		Salinas		Houston	
		training	test	training	test	training	test	training	test
CNNs	LS2CM-Res	0.30	0.81	0.12	2.96	0.18	7.27	0.17	0.90
	PyResNet	2.31	7.00	0.97	36.30	1.22	45.51	1.73	11.50
	Hybrid-SN	0.34	0.95	0.14	4.91	0.18	6.09	0.25	1.55
	MCRSCA	0.52	1.63	0.21	7.53	0.30	10.39	0.38	2.39
	FADCNN	0.34	0.64	0.14	1.80	0.18	2.71	0.08	0.61
Transformer	VIT	0.30	1.00	0.16	3.71	0.16	6.29	0.09	0.60
	SF	0.42	1.37	0.18	5.81	0.23	9.00	0.28	1.90
	SSTN	0.39	1.18	0.16	5.34	0.23	7.71	0.28	1.72
	HIT	22.29	170.99	7.08	652.74	12.60	1111.63	12.62	210.19
	GAHT	0.38	1.12	0.17	5.38	0.22	7.17	0.28	1.70
	CTMixer	0.41	1.37	0.15	5.10	0.23	8.29	0.27	1.74
	SSFTT	0.32	0.47	0.12	2.37	0.19	6.11	0.12	0.75
	Proposed	0.30	0.95	0.12	4.85	0.17	6.07	0.23	1.60

article has a larger interclass distance compared to LS2CM-Res and SSTN, making it easier for different categories to be correctly classified. In summary, the visual results from the clustering perspective on the three datasets validate the effectiveness of our proposed method.

3) Analysis of Model complexity: In order to better illustrate the impact of CS3M on model parameters in different datasets, this article presents the parameter impact of CS3M on four different datasets using radar charts. As shown in Fig. 18, it can be observed that with CS3M, the network parameters

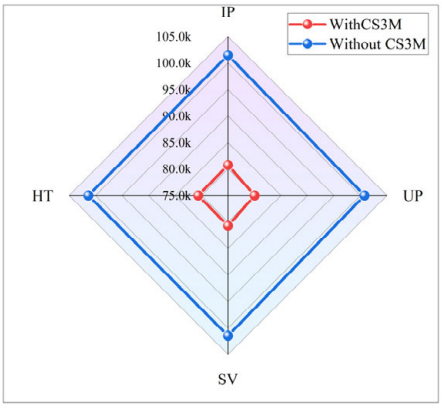


Fig. 18. Impact of CS3M on model parameters on different datasets.

are fewer than that without CS3M. According to the distance between the red lines with CS3M and the blue lines without CS3M in this graph, it can be analyzed that CS3M has a significant impact on the parameter quantity of the network model, further demonstrating the effectiveness of the proposed method in this article.

In order to evaluate the network complexity of the proposed methods, this article analyzes the complexity of the models in terms of training and testing time. Table XI shows the training time and the testing time of each method on the four datasets. As shown in Table XI, the training time of DBMST is the optimal result in the Indian Pines dataset, and the testing time, although not optimal, is also the suboptimal result. In the Pavia dataset, DBMST has the shortest training time, and the test time exceeds 80% of the comparison methods, just below LS2CM-Res and VIT. The reason is that LS2CM-Res is a classification method designed with the goal of being lightweight, so it has a shorter training time and testing time. In the Salinas dataset, the training time of the proposed DBMST is optimal. In the Houston 2013 dataset, the testing and training times of the method proposed in this article are only lower than the LS2CM Res, FADCNN, and SSFTT methods and are superior to the remaining nine methods Therefore, considering three different datasets together, the network complexity of the proposed DBMST is the best among all methods.

IV. Conclusion

In this article, a DBMST network is proposed for HSI classification from a multiscale perspective. First, DBMST proposes the CS3M for HSIs to achieve large-scale token partitioning and avoid the loss of spatially adjacent information. Then, considering that the input of the Transformer is a vector after flattening, the T2T local–global feature extraction module is designed for small-scale branches to transform tokens into images and extract the contextual information of images. Finally, the PAFM for feature fusion in different scale branches is proposed for different dimensionality of extracted features. In order to verify the effectiveness of the proposed method, a large number of quantitative experiments and visual analyses were conducted on three datasets, and the experimental results proved the effectiveness of the DBMST method proposed in this article. In future work, we will continue to explore the extraction of multiscale features in the Transformer and try to combine Transformer with other networks to improve the Transformer structure.

References

[1] J.-P. Ardouin, J. Levesque, and T. A. Rea, "A demonstration of hyperspectral image exploitation for military applications," in *Proc. 10th Int. Conf. Inf. Fusion*, Jul. 2007, pp. 1–8.

[2] C. M. Gevaert, J. Suomalainen, J. Tang, and L. Kooistra, "Generation of spectral–temporal response surfaces by combining multispectral satellite and hyperspectral UAV imagery for precision agriculture applications," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 3140–3146, Jun. 2015.

[3] S. S. M. Noor, K. Michael, S. Marshall, J. Ren, J. Tschannerl, and F. J. Kao, "The properties of the cornea based on hyperspectral imaging: Optical biomedical engineering perspective," in *Proc. Int. Conf. Syst., Signals Image Process. (IWSSIP)*, Bratislava, Slovakia, May 2016, pp. 1–4.

[4] J. Wang, L. Zhang, Q. Tong, and X. Sun, "The spectral crust project—Research on new mineral exploration technology," in *Proc. 4th Workshop Hyperspectral Image Signal Process., Evol. Remote Sens. (WHISPERS)*, Shanghai, China, Jun. 2012, pp. 1–4.

[5] J. Mielikainen and P. Toivanen, "Lossless compression of hyperspectral images using a quantized index to lookup tables," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 3, pp. 474–478, Jul. 2008.

[6] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "An augmented linear mixing model to address spectral variability for hyperspectral unmixing," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1923–1938, Apr. 2019.

[7] S. Yang and Z. Shi, "Hyperspectral image target detection improvement based on total variation," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2249–2258, May 2016.

[8] L. Yan, X. Wang, M. Zhao, M. Kaloorazi, J. Chen, and S. Rahardja, "Reconstruction of hyperspectral data from RGB images with prior category information," *IEEE Trans. Comput. Imag.*, vol. 6, pp. 1070–1081, 2020.

[9] L. Sun et al., "Low rank component induced spatial–spectral kernel method for hyperspectral image classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 10, pp. 3829–3842, Oct. 2020.

[10] L. Sun, C. Ma, Y. Chen, H. J. Shim, Z. Wu, and B. Jeon, "Adjacent superpixel-based multiscale spatial–spectral kernel for hyperspectral classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 6, pp. 1905–1919, Jun. 2019.

[11] L. Sun, Z. Wu, J. Liu, L. Xiao, and Z. Wei, "Supervised spectral–spatial hyperspectral image classification with weighted Markov random fields," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1490–1503, Mar. 2015.

[12] C. Cariou and K. Chehdi, "A new k-nearest neighbor density-based clustering method and its application to hyperspectral images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 6161–6164.

[13] 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.

[14] Q. Ye, P. Huang, Z. Zhang, Y. Zheng, L. Fu, and W. Yang, "Multiview learning with robust double-sided twin SVM," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 12745–12758, Dec. 2022.

[15] Q. Ye et al., "L1-norm distance minimization-based fast robust twin support vector k-plane clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4494–4503, Sep. 2018.

[16] Y.-N. Chen, T. Thaipisutikul, C.-C. Han, T.-J. Liu, and K.-C. Fan, "Feature line embedding based on support vector machine for hyperspectral image classification," *Remote Sens.*, vol. 13, no. 1, p. 130, Jan. 2021.

[17] Y. E. SahIn, S. Arisoy, and K. Kayabol, "Anomaly detection with Bayesian Gauss background model in hyperspectral images," in *Proc. 26th Signal Process. Commun. Appl. Conf. (SIU)*, May 2018, pp. 1–4.

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

[19] Q. Ye, J. Yang, F. Liu, C. Zhao, N. Ye, and T. Yin, "L1-norm distance linear discriminant analysis based on an effective iterative algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 1, pp. 114–129, Jan. 2018.

[20] L. Fu et al., "Learning robust discriminant subspace based on joint $L_{2,p}$- and $L_{2,s}$-norm distance metrics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 1, pp. 130–144, Jan. 2022.

[21] X. He and P. Niyogi, "Locality preserving projections," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 16. 2004, pp. 1–8.

[22] Y. Duan, H. Huang, and T. Wang, "Semisupervised feature extraction of hyperspectral image using nonlinear geodesic sparse hypergraphs," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022, Art. no. 5515115.

[23] F. Luo, T. Zhou, J. Liu, T. Guo, X. Gong, and J. Ren, "Multiscale diff-changed feature fusion network for hyperspectral image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–13, 2023, Art. no. 5502713.

[24] Y. Duan, F. Luo, M. Fu, Y. Niu, and X. Gong, "Classification via structure-preserved hypergraph convolution network for hyperspectral image," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–13, 2023, Art. no. 5507113.

[25] H. Zhou, F. Luo, H. Zhuang, Z. Weng, X. Gong, and Z. Lin, "Attention multihop graph and multiscale convolutional fusion network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–14, 2023, Art. no. 5508614.

[26] T. Guo, R. Wang, F. Luo, X. Gong, L. Zhang, and X. Gao, "Dual-view spectral and global spatial feature fusion network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–13, 2023, Art. no. 5512913.

[27] M. E. Paoletti et al., "Capsule networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2145–2160, Apr. 2019, doi: 10.1109/TGRS.2018.2871782.

[28] J. Wang, S. Guo, R. Huang, L. Li, X. Zhang, and L. Jiao, "Dual-channel capsule generation adversarial network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2022, Art. no. 5501016.

[29] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5966–5978, Jul. 2021.

[30] 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, May 2017.

[31] Y. Xu, L. Zhang, B. Du, and F. Zhang, "Spectral–spatial unified networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 5893–5909, Oct. 2018.

[32] H. Zhai, H. Zhang, L. Zhang, and P. Li, "Total variation regularized collaborative representation clustering with a locally adaptive dictionary for hyperspectral remote sensing imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Fort Worth, TX, USA, Jul. 2017, pp. 3755–3758.

[33] 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)*, Beijing, China, Sep. 2017, pp. 3904–3908.

[34] Z. Meng, L. Jiao, M. Liang, and F. Zhao, "A lightweight spectral–spatial convolution module for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[35] 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. 2020.

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

[37] R. Shang, H. Chang, W. Zhang, J. Feng, Y. Li, and L. Jiao, "Hyperspectral image classification based on multiscale cross-branch response and second-order channel attention," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2022, Art. no. 5532016.

[38] 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)*, Montreal, QC, Canada, Oct. 2021, pp. 11916–11925.

[39] B. Graham et al., "LeViT: A vision transformer in ConvNet's clothing for faster inference," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 12239–12249.

[40] D. Zhou et al., "DeepViT: Towards deeper vision transformer," 2021, *arXiv:2103.11886*.

[41] L. Yuan et al., "Tokens-to-token ViT: Training vision transformers from scratch on ImageNet," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 538–547.

[42] X. Yang, W. Cao, Y. Lu, and Y. Zhou, "Hyperspectral image transformer classification networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022, Art. no. 5528715.

[43] D. Hong et al., "SpectralFormer: Rethinking hyperspectral image classification with transformers," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022, Art. no. 5518615.

[44] S. Mei, C. Song, M. Ma, and F. Xu, "Hyperspectral image classification using group-aware hierarchical transformer," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022, Art. no. 5539014.

[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, pp. 1–14, 2022, Art. no. 5522214.

[46] 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, pp. 1–5, 2022.

[47] Z. Zhong, Y. Li, L. Ma, J. Li, and W.-S. Zheng, "Spectral-spatial transformer network for hyperspectral image classification: A factorized architecture search framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022, Art. no. 5514715.

[48] C. Zhang, G. Li, and S. Du, "Multi-scale dense networks for hyperspectral remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 9201–9222, Nov. 2019.

[49] D. Wang, B. Du, L. Zhang, and Y. Xu, "Adaptive spectral–spatial multiscale contextual feature extraction for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 2461–2477, Mar. 2021.

[50] H. Gao, Y. Yang, C. Li, L. Gao, and B. Zhang, "Multiscale residual network with mixed depthwise convolution for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 4, pp. 3396–3408, Apr. 2021.

[51] Z. Lu, B. Xu, L. Sun, T. Zhan, and S. Tang, "3-D channel and spatial attention based multiscale spatial–spectral residual network for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 4311–4324, 2020.

[52] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[53] J. Zhu, L. Fang, and P. Ghamisi, "Deformable convolutional neural networks for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 8, pp. 1254–1258, Aug. 2018.

[54] S. Ren, D. Zhou, S. He, J. Feng, and X. Wang, "Shunted self-attention via multi-scale token aggregation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 10843–10852.

[55] H. Wu et al., "CvT: Introducing convolutions to vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 22–31.

[56] M. Zhang, W. Li, Y. Zhang, R. Tao, and Q. Du, "Hyperspectral and LiDAR data classification based on structural optimization transmission," *IEEE Trans. Cybern.*, vol. 53, no. 5, pp. 3153–3164, May 2023.

[57] Y. Zhang, W. Li, W. Sun, R. Tao, and Q. Du, "Single-source domain expansion network for cross-scene hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 32, pp. 1498–1512, 2023.

[58] H. Liu, W. Li, X.-G. Xia, M. Zhang, C.-Z. Gao, and R. Tao, "Central attention network for hyperspectral imagery classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 8989–9003, Nov. 2023.

[59] X. Qiao, S. K. Roy, and W. Huang, "Multiscale neighborhood attention transformer with optimized spatial pattern for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–15, 2023, Art. no. 5523815.

[60] Y. Dong, Q. Liu, B. Du, and L. Zhang, "Weighted feature fusion of convolutional neural network and graph attention network for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 31, pp. 1559–1572, 2022.

[61] W. Li, Q. Liu, S. Fan, H. Bai, and M. Xin, "Multistage superpixel-guided hyperspectral image classification with sparse graph attention networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–18, 2023, Art. no. 5519718.

[62] Q. Liu, Y. Dong, Y. Zhang, and H. Luo, "A fast dynamic graph convolutional network and CNN parallel network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022, Art. no. 5530215.

Cuiping Shi (Member, IEEE) received the M.S. degree from Yangzhou University, Yangzhou, China, in 2007, and the Ph.D. degree from the Harbin Institute of Technology (HIT), Harbin, China, in 2016.

From 2017 to 2020, she held post-doctoral research at the College of Information and Communications Engineering, Harbin Engineering University, Harbin. She is currently a Professor with the Department of Communication Engineering, Qiqihar University, Qiqihar, China. She is also work with the College of Information Engineering, Huzhou University, Huzhou, China. She has published two academic books about remote sensing image processing and more than 90 papers in journals and conference proceedings. Her main research interests include remote sensing image processing, pattern recognition, and machine learning.

Dr. Shi's doctoral dissertation won the Nomination Award of Excellent Doctoral Dissertation of the Harbin University of Technology (HIT) in 2016.

Shuheng Yue received the bachelor's degree from Shandong Jiaotong University, Jinan, China, in 2021. He is currently pursuing the master's degree with Qiqihar University, Qiqihar, China.

His research interests include hyperspectral image processing and machine learning.

Liguo Wang (Member, IEEE) received the M.S. and Ph.D. degrees in signal and information processing from the Harbin Institute of Technology, Harbin, China, in 2002 and 2005, respectively.

From 2006 to 2008, he held a post-doctoral research position at the College of Information and Communications Engineering, Harbin Engineering University, Harbin, where he is currently a Professor. Since 2020, he has been with the College of Information and Communication Engineering, Dalian Nationalities University, Dalian, China. He has published two books about hyperspectral image processing and more than 130 papers in journals and conference proceedings. His main research interests include remote sensing image processing and machine learning.

宁波大学图书馆 NBULIB

文献检索报告
SCI 收录

报告编号：**202436000Z193120(S)**

数据库：科学引文索引 (Science Citation Index Expanded) 查证方式：文献被收录及所在期刊JCR期刊影响因子、中国科学院文献情报中心期刊分区情况 时间范围：1900 年 - 2024 年	委托人：石翠萍 委托单位：湖州师范学院 信息工程学院	检索人员：钱 绍 检索日期：2024年5月16日

检索结果：被 SCI-E 收录文献 3 篇

#	作者	标题	来源出版物	JCR影响因子	中科院分区	文献类型	入藏号
1	Shi, CP; Chen, JX; Wang, LG	Hyperspectral image classification based on a novel Lush multi-layer feature fusion bias network	***EXPERT SYSTEMS WITH APPLICATIONS*** 2024, 247: 123155.	• 8.5 (2022);	• 小类(升级版) (2023) 运筹学与管理科学 [2区]; • 小类(升级版) (2023) 计算机：人工智能 [2区]; • 小类(升级版) (2023) 工程：电子与电气 [2区]; • 大类(升级版) (2023) 计算机科学 [1区] Top 期刊;	J Article	WOS:001170528000001
2	Shi, CP; Yue, SH; Wang, LG	A Dual-Branch Multiscale Transformer Network for Hyperspectral Image Classification	***IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*** 2024, 62: 5504520.	• 8.2 (2022);	• 小类(升级版) (2023) 遥感 [2区]; • 小类(升级版) (2023) 成像科学与照相技术 [2区]; • 小类(升级版) (2023) 工程：电子与电气 [2区]; • 小类(升级版) (2023) 地球化学与地球物理 [1区]; • 大类(升级版) (2023) 地球科学 [1区];	J Article	WOS:001173248900023
3	Shi, CP; Liu, ZQ; Qu, JG; Deng, YX	The Expansion Methods of Inception and Its Application	***SYMMETRY-BASEL*** 2024, 16 (4): 494.	• 2.7 (2022);	• 小类(升级版) (2023) 综合性期刊 [3区]; • 大类(升级版) (2023) 综合性期刊 [3区];	J Article	WOS:001210657600001
						合计	3

备注	中科院期刊分区数据：大类分区(升级版), 小类分区(升级版) 影响因子/期刊分区的年份选择：最新年份

收录文献附录

第 1 条，共 3 条：
标题：Hyperspectral image classification based on a novel Lush multi-layer feature fusion bias network
作者：Shi, CP (Shi, Cuiping); Chen, JX (Chen, Jiaxiang); Wang, LG (Wang, Liguo)
来源出版物：EXPERT SYSTEMS WITH APPLICATIONS 卷：247 文献号：123155 出版年：AUG 1 2024
Web of Science 核心合集中的 "被引频次"：0
被引频次合计：0
入藏号：WOS:001170528000001
文献类型：Article 出版物类型：J

作者地址：[Shi, Cuiping] Huzhou Univ, Coll Informat Engn, Huzhou 313000, Peoples R China.; [Shi, Cuiping; Chen, Jiaxiang] Qiqihar Univ, Coll Commun & Elect Engn, Qiqihar 161000, Peoples R China.; [Wang, Liguo] Dalian Nationalities Univ, Coll Informat & Commun Engn, Dalian 116000, Peoples R China.
所属机构：Huzhou University; Qiqihar University; Dalian Minzu University
通讯作者地址：Shi, CP (corresponding author), Huzhou Univ, Coll Informat Engn, Huzhou 313000, Peoples R China.

第 2 条，共 3 条：
标题：A Dual-Branch Multiscale Transformer Network for Hyperspectral Image Classification
作者：Shi, CP (Shi, Cuiping); Yue, SH (Yue, Shuheng); Wang, LG (Wang, Liguo)
来源出版物：IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING 卷：62 文献号：5504520 出版年：2024
Web of Science 核心合集中的 "被引频次"：0
被引频次合计：0
入藏号：WOS:001173248900023
文献类型：Article 出版物类型：J
作者地址：[Shi, Cuiping] Qiqihar Univ, Dept Commun Engn, Qiqihar 161000, Peoples R China.; [Shi, Cuiping] Huzhou Univ, Coll Informat Engn, Huzhou 313000, Peoples R China.; [Yue, Shuheng] Qiqihar Univ, Dept Commun Engn, Qiqihar 161000, Peoples R China.; [Wang, Liguo] Dalian Nationalities Univ, Coll Informat & Commun Engn, Dalian 116000, Peoples R China.
所属机构：Qiqihar University; Huzhou University; Qiqihar University; Dalian Minzu University
通讯作者地址：Shi, CP (corresponding author), Qiqihar Univ, Dept Commun Engn, Qiqihar 161000, Peoples R China.; Shi, CP (corresponding author), Huzhou Univ, Coll Informat Engn, Huzhou 313000, Peoples R China.

第 3 条，共 3 条：
标题：The Expansion Methods of Inception and Its Application
作者：Shi, CP (Shi, Cuiping); Liu, ZQ (Liu, Zhenquan); Qu, JG (Qu, Jiageng); Deng, YX (Deng, Yuxin)
来源出版物：SYMMETRY-BASEL 卷：16 期：4 文献号：494 出版年：APR 2024
Web of Science 核心合集中的 "被引频次"：0
被引频次合计：0
入藏号：WOS:001210657600001
文献类型：Article 出版物类型：J
作者地址：[Shi, Cuiping] Huzhou Univ, Coll Informat Engn, Huzhou 313000, Peoples R China.; [Shi, Cuiping; Liu, Zhenquan; Qu, Jiageng; Deng, Yuxin] Qiqihar Univ, Coll Commun & Elect Engn, Qiqihar 161000, Peoples R China.; [Liu, Zhenquan] Northwest Normal Univ, Coll Phys & Elect Engn, Lanzhou 730070, Peoples R China.
所属机构：Huzhou University; Qiqihar University; Northwest Normal University - China
通讯作者地址：Shi, CP (corresponding author), Huzhou Univ, Coll Informat Engn, Huzhou 313000, Peoples R China.; Shi, CP (corresponding author), Qiqihar Univ, Coll Commun & Elect Engn, Qiqihar 161000, Peoples R China.

检索人员：[illegible] 绍
教育部科技查新工作站（Z19）
2024年5月16日