Instructional Mask Autoencoder: A Scalable Learner for Hyperspectral Image Classification

Weili Kong, *Student Member, IEEE*, Baisen Liu, Xiaojun Bi, Jiaming Pei, *Student Member, IEEE*, and Zheng Chen

***Abstract*—Nowadays, an increasing number of hyperspectral images (HSIs) are becoming available. However, the utilization of unlabeled HSIs is extremely low due to high annotation costs. Thus, it is crucial to figure out how to use these unlabeled HSIs and enhance the classification performance. Fortunately, self-supervised training enables us to acquire latent features from unlabeled HSIs, thereby enhancing network performance via transfer learning. Whereas, most current networks for HSIs are inflexible, it is challenging for them to perform learning and accommodate multimodal HSIs. Therefore, we devise a scalable self-supervised network called instructional mask autoencoder, which can extract general patterns of HSIs by these unannotated data. It primarily consists of a spatial–spectral embedding block and a transformer-based masked autoencoder, which are utilized for projecting input samples into the same latent space and learning higher level semantic information, respectively. Moreover, we utilize a random token called ins_token to instruct the model learn components of global information, which are highly correlated with the target pixel in HSI samples. In the fine-tuning stage, we design a learnable aggregation mechanism to put all tokens into full play. The obtained results illustrate that our method exhibits robust generalization performance and accelerates convergence across diverse datasets. In cases of limited samples, we conducted experiments on three structurally distinct HSIs, all of which achieved competitive performance. Compared to state-of-the-art methods, our approach demonstrated respective improvements of 1.97%, 0.44%, and 3.35% on these three datasets.**

***Index Terms*—Mask autoencoder, multimodal hyperspectral image (HSI), self-supervised, transfer learning, unlabeled HSI.**

Manuscript received 6 October 2023; revised 13 November 2023; accepted 23 November 2023. Date of publication 28 November 2023; date of current version 14 December 2023. This work was supported by the Natural Science Foundation of Heilongjiang Province for Key projects, China, under Grant ZD2021F004. *(Corresponding author: Baisen Liu.)*

Weili Kong is with the School of Information and Communication Engineering, Harbin Engineering University, Harbin 150050, China (e-mail: kkweil@hrbeu.edu.cn).

Baisen Liu is with the School of Electronic and Information Engineering, Heilongjiang Institute of Technology, Harbin 150050, China, and also with the School of Information and Communication Engineering, Harbin Engineering University, Harbin 150050, China (e-mail: spedliu@126.com).

Xiaojun Bi is with the School of Information Engineering, Minzu University of China, Beijing 100081, China, and also with the Key Laboratory of Ethnic Language Intelligent Analysis and Security Governance of MOE, Minzu University of China, Beijing 100081, China (e-mail: bixiaojun@hrbeu.edu.cn).

Jiaming Pei is with the School of Computer Science, The University of Sydney, Sydney, NSW 2006, Australia (e-mail: jpei0906@uni.sydney.edu.au).

Zheng Chen is with the Key Laboratory of Ethnic Language Intelligent Analysis and Security Governance of MOE, Minzu University of China, Beijing 100081, China (e-mail: chenzheng@hrbeu.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2023.3337132

I. Introduction

IN RECENT years, hyperspectral remote sensing technology has made significant strides, which uses spectroscopy imagery technology to synchronously gather enormous spectral and spatial information of the observing targets at pixel level [1]. Thus, it enables us to conduct accurate classification for the observation targets [2], [3], [4]. Numerous fields, including ecological research [5], precision agriculture [6], mineral exploration [7], and medicine [8], are covered by the categorization tasks of a hyperspectral image (HSI) considering the advantage of a wealth of information contained in it. Unlike some other image classification missions, HSI classification is a dense task that assigns each of the pixels in the imagery into a specific category [9]. Hence, annotating HSIs is a quite expensive mission. Moreover, the sensors and payloads used for capturing HSIs are diverse, with a wide range of parameters. That means the gathered HSIs exhibit variations in wavelength range, spectral resolution, and spatial resolution. It is still necessary to train a model from scratch to analyze these HSIs due to their different structures. Therefore, how to leverage readily available unlabeled data to learn a shared feature extractor that can handle multimodal HSIs would be a meaningful work.

In the domain of remote sensing, numerous methodologies have been proposed for creating efficient HSI classifiers. In its early beginnings, researchers focus on the traditional machine learning approaches to train an HSI classifier. As graphics card computing power continues to advance and deep learning technology evolves, numerous ingenious networks have been devised for HSI classification, such as those based on convolutional neural networks (CNNs) [10], [11], recurrent neural networks (RNNs) [12], long short-term memory networks [13], [14], graph neural networks [15], [16], and graph convolutional networks [17], [18]. Compared to traditional machine learning methods, these approaches have all demonstrated remarkable performance. In particular, the CNN is often combined with other networks to enhance the extraction of spatial–spectral information. With the introduction of a CNN, the input structure of the model has changed from $\mathbb{R}^c$ to $\mathbb{R}^{s\times s\times c}$. The reason behind this transition is the prevalent local homogeneity observed in natural images, that is, pixels within the same area are likely to belong to the same land cover class, and their spectral and textural features are similar.

However, the potential of CNN-related methods has been constrained by the following limitations.

1) The CNN is constrained by the fixed size of its convolutional kernel, allowing it only to access short-term dependencies.
2) Local homogeneity introduced by convolutional operations may not be applicable to pixels located at the boundaries of land cover regions. Specifically, there may exist a variety of pixels belonging to distinct land cover classes within the same sample.
3) Owing to the sensitivity of convolutions to geometric textures in images, boundaries between land cover regions are also prone to extraction, introducing noise during classification [19].
4) When the sample size is fixed, the structure of the CNN becomes rigid, resulting in a singular input size and limited generalization performance [20].

Altering the sample size necessitates a corresponding modification in the CNN structure, rendering previously trained model parameters unusable. Therefore, although CNN-related methods have demonstrated strong performance, certain limitations persist, constraining the performance and generalization capability.

To surmount these inherent limitations of CNN-based methods, certain research endeavors opt to employ a transformer as a foundational structure in designing classification models [20], [21], [22], [23], [24], [25], [26], [27]. The core of the transformer is the self-attention (SA) mechanism. The transformer exploits long-term dependencies along data through SA. It captures the dependencies between all positions in sequence data by calculating similarities and performing weighted summation to integrate information from different positions. Besides, SA is parameter free and enables the model to process inputs of any length. Regrettably, the transformer model performs indiscriminate global SA calculations on input data. It lacks some a priori assumptions about the data, the so-called inductive bias, such as translation invariance and local homogeneity in the CNN. Therefore, this type of network often has a larger function domain and requires more data to train it effectively [28]. Although some approaches have used pretraining to alleviate this issue, they have only been performed on small datasets. When encountering new data, these models still require training from scratch. These methods still cannot effectively support multimodal HSIs, and their generalization performance in such scenarios has not been thoroughly validated.

In this article, for the purpose of effectively harnessing unannotated hyperspectral data and making the model compatible with multimodal inputs, such as data from varying spectral resolutions, spatial resolutions, and input sizes, a transformer-based self-supervised learner is specifically designed for the HSI, which shows strong generalization capabilities across multimodal HSI inputs. First, a spatial–spectral embedding block is designed to convert the multimodal HSIs to a shared token space. Afterward, we employ two self-supervised agent tasks, namely, masking reconstruction and model attention instruction, to train a unified shared encoder. During this process, each pixel can be analogously likened to words in the context of natural language processing (NLP), and the spatial relationships between these pixels are reminiscent of contextual information in NLP. Consequently, the network inherently acquires an understanding of spatial spectral information within HSIs as it undertakes the patch reconstruction task. In response to the inherent absence of inductive biases within transformer architectures, we propose a novel approach. This entails the incorporation of an ins_token at the input side of the encoder, initialized with random values. Leveraging a metric learning paradigm [29], we aim to align the output vector of this ins_token, postdecoding, as closely as possible with the embedding vector of the target pixel within a designated projection space. This strategic augmentation serves to direct the model's attention toward the specific target pixel. To accommodate variable input sizes, this study introduces adaptable conditional positional embedding [30]. Finally, instead of global average pooling [31], we introduce a mechanism to adaptively combine the tokens generated by the encoder to fully exploit the knowledge acquired by the network for downstream tasks. The resulting composite output is subsequently utilized as the ultimate classification vector, which is then fed into the classifier for supervised training. To facilitate the training of our model, we source a diverse collection of HSIs from the GaoFen-5 satellite. This dataset encompassed a broad spectrum of environmental scenarios, such as desert, forest, township, forest village, snowfield, village, city, and metropolis. Subsequently, we meticulously divided these unlabeled images into nonoverlapping patches, categorized into four distinct size parameters. When transferring pretrained model parameters to a new dataset, the process primarily involves the replacement of the input layer to accommodate varying spectral resolutions. Subsequently, supervised fine-tuning can be conducted with a limited number of samples. Under the same circumstances, compared to the similar methods, our technique delivered state-of-the-art performance under the same circumstances.

In summary, the primary contributions of this article are as follows.

1) We devise a self-supervised learner, which is capable of harnessing a substantial volume of unannotated HSIs. It significantly enhances data utilization efficiency and promotes downstream task performance, particularly in a sample-limited scenario.
2) Our proposed method exhibits robust generalization capabilities on multimodal HSI inputs while maintaining simplicity and ease of implementation.
3) We introduce a model attention instructor, denoted as the ins_token, a randomly initialized token that directs the model focus toward areas of human interest through metric learning.

II. Related Works

A. Deep-Learning-Based HSI Classification Methods

In the early stage of the study on HSI classification, most methods focus on exploring the discrepancy of original spectral signatures in HSIs to distinguish the pixels into different categories, including k-nearest neighbor [32], support vector machines (SVMs) [33], logistic regression [34], and so on. Several methods for dimension reduction and spectral information extraction have also been developed to handle the complex high-dimensional nonlinear distribution of HSIs, such as principal component analysis [35], [36], independent component analysis [37], and linear discriminant analysis [38]. However, the linear processing nature makes it challenging to process the complex spectrum properties of HSIs. In recent years, deep learning has emerged as a powerful tool in HSI classification. For instance, Ahmad et al. [39] and Mughees and Tao [40] gathered the feature sets by using an autoencoder (AE)-based method to extract HSI features. Zhong et al. [41] proposed a semisupervised deep belief network through regularizing pretraining and fine-tuning procedures by a diversity promoting prior over latent factors, thereby improving model classification performance. Nevertheless, owing to challenges in HSIs, such as spectral drift, spectral variability within identical materials, and material variability within identical spectra, methods that solely rely on spectral information still suffer from significant classification errors [42].

To alleviate this issue, CNNs and their variants are being used to explore joint spatial–spectral features in HSIs. For example, Xu et al. [43] designed a multiple-spectral-resolution 3-D CNN, which combined the 3-D convolution layer and residual connection to better adapt to the 3-D cubic form of hyperspectral data and make efficient use of spectral information in different bands. Li et al. [44] combined depthwise separable convolution and the 3-D CNN; this work successfully accelerated the training speed and achieved good classification performance. Although CNNs and their variants have shown promise in achieving accurate results, their inherent network architecture and focus on local spatial information may not effectively capture useful spectral sequentiality information. As a result, these models may have limitations in achieving higher accuracy in HSI classification tasks.

B. Pretraining-Based HSI Classification Methods

In the realm of NLP, pretrained large-scale models have exhibited remarkable performance, showcasing robust generalization and transfer capabilities, even when exposed to a limited amount of downstream task-specific annotations [45], [46]. Prominent examples include BERT [47] and the GPT series [48], [49], [50]. Building upon the foundation laid by vision transformers (ViTs) [51], researchers have devised pretraining models tailored for the visual domain, such as Google's BEiT [52] and the MAE [53] model developed by the team led by He et al. These methods employ self-supervised learning techniques for model pretraining and have consistently achieved state-of-the-art performance in downstream tasks. Scholars, drawing inspiration from the vision self-supervised framework like BEiT [52] and the MAE [53], have devised pretrained models tailored for hyperspectral imagery. These models have demonstrated commendable performance in classification tasks, exemplified by the masked autoencoder spectral–spatial transformer (MAEST) designed by Ibanez et al. [54], spectral–spatial masked transformer (SS-MTr) proposed by Huang et al. [55], and masked spatial–spectral model proposed by Scheibenreif et al. [56]. However, when employing these models on different datasets, apart from fine-tuning the new data, retraining the new dataset is often necessary. Moreover, it is noteworthy that these models have primarily leveraged a limited subset of hyperspectral data available in the public domain, such as Indian Pines (IP), PaviaU (PU), and Salinas (SA) datasets. They have not fully harnessed the extensive reservoir of unlabeled hyperspectral data that are accessible and have still maintained certain constraints on network inputs.

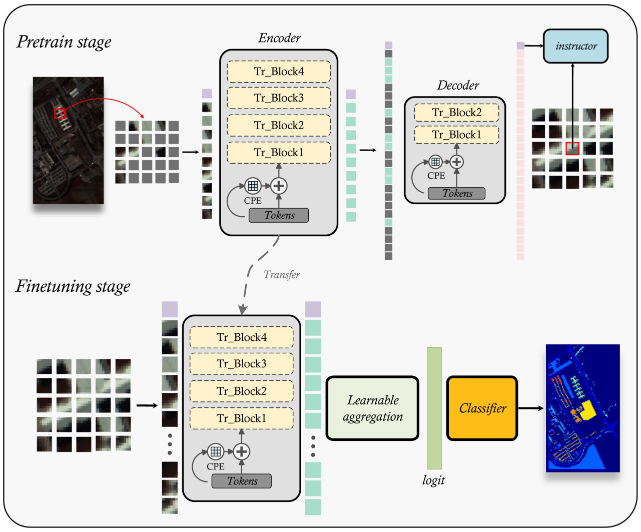


Fig. 1. Overall architecture of IMAE. During pretraining, we first perform spatial–spectral embedding on the provided HSI samples, converting them into a shared token space. Then, 50% of these tokens are masked out, allowing visible tokens to enter an AE for the reconstruction task. Simultaneously, a randomly initialized ins_token is introduced to guide model attention. After pretraining, we discard the decoder and the encoder is applied to unspoiled HSI samples. Finally, a learnable aggregation is applied to the outputs of encoder for the classification task.

III. Proposed Methodology

A. Overview of Instructional Mask Autoencoder (IMAE)

In this section, we will introduce the proposed method in detail. The complete workflow of IMAE is as follows: To begin with, we start by performing spatial–spectral embedding on the provided HSI sample, transforming it into a sequence of tokens. Subsequently, these token sequences undergo random masking, allowing the visible tokens to be fed into an AE for reconstruction. At the same time, a randomly initialized ins_token is fed into the AE to conduct model attention instruction. The output of ins_token at the decoder end, along with the spectral vectors of target pixels, is projected into a metric space; then, their distance within this metric space is minimized. The overall architecture of the IMAE is illustrated in Fig. 1.

B. Spatial–Spectral Embedding

Spatial–spectral embedding is mainly composed of two basic components: spectral embedding and position embedding. In the spectral embedding strategy, a 1×1 2-D CNN layer is employed as the input layer to unify all HSI data into the same dimension. Then, a 1×1 3-D CNN layer is utilized to extensively explore spectral information. Finally, we project the

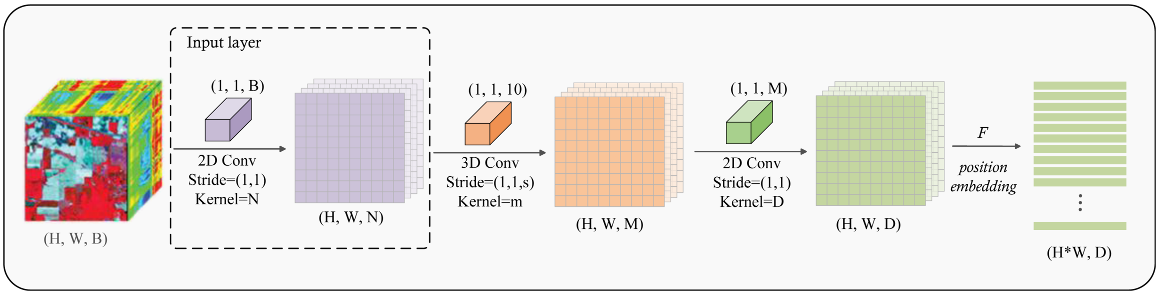


Fig. 2. Overall architecture of the spatial–spectral embedding module.

features into a shared space through another 1 × 1 2-D CNN layer. If the spectral bands of the downstream task data differ from those of the pretraining data, we can modify the parameters of the input layer to continue leveraging the expertise acquired by the model. Fig. 2 illustrates the overview of spatial-spectral embedding module.

Specifically, given a training sample $\mathbf{X}$, $\mathbf{X} \in \mathbb{R}^{h \times w \times b}$, where h and w represent the height and width of the input patch, respectively, b represents the number of bands. In the 2-D convolution operation, the lth convolution kernel $\mathbf{W}_{2d}^{(l)} \in \mathbb{R}^{1 \times 1 \times C_{2d}}$ and the feature map of $\mathbf{W}_{2d}^{(l)}$ is $\mathbf{Z}_{2d}^{(l)}$. For illustration, considering the input layer, we calculate $\mathbf{Z}_{2d}^{(l)}$ as follows:

$$\mathbf{Z}_{2d} = \text{Conv2D}(\mathbf{X})$$

$$\mathbf{Z}_{2d\,(i,j)}^{(l)} = \sum_{n=1}^{b} \mathbf{X}_{(i,j,n)} \times \mathbf{W}_{2d\,(i,j,n)}^{(l)} \tag{1}$$

In the 3-D convolution operation, the lth convolution kernel $\mathbf{W}_{3d}^{(l)} \in \mathbb{R}^{1 \times 1 \times C_{3d}}$ and the feature map of $\mathbf{W}_{3d}^{(l)}$ is $\mathbf{Z}_{3d}^{(l)}$. As a demonstration, supposing that the input data $\mathbf{Z}_{2d} \in \mathbb{R}^{h \times w \times m}$, we calculate $\mathbf{Z}_{3d}^{(l)}$ as follows:

$$\mathbf{Z}_{3d} = \text{Conv3D}(\mathbf{Z}_{2d})$$

$$\mathbf{Z}_{3d\,(i,j,k)}^{(l)} = \sum_{n=1}^{C_{3d}} \mathbf{Z}_{2d(i,j,(k-1)*s+n)} \times \mathbf{W}_{3d\,(i,j,n)}^{(l)} \tag{2}$$

where s represents the stride of the 3-D convolution kernel on the third dimension of the input; the third dimension c of $\mathbf{Z}_{3d}^{(l)}$ can be computed as

$$c = \left\lceil \frac{m - C_{3d}}{s} \right\rceil + 1. \tag{3}$$

We construct the spectral embedding (SE) module using two 2-D convolution layers and a 3-D convolution layer, and its expression is

$$\mathbf{Z} = SE(\mathbf{X}) = \text{Conv2D}(\text{Conv3D}(\text{Conv2D}(\mathbf{X}))). \tag{4}$$

In addition, position embedding plays a crucial role in the transformer-based model. Through the SA mechanism, the transformer-based model can learn the relationships between tokens and pay attention to essential facts, but it is unable to learn the position information of each token, thus necessitating the input of extra token position information to the model. The common position embedding methods are predefined; the length of position token sequence is fixed even if the position tokens are learnable, which will make the model unable to handle sequences exceeding the predefined length. The sequence length growth in an HSI patch is a square term of its size, so using the length fixed embedding method will prevent the model from generalizing to larger patch inputs. Furthermore, the predefined methods just add a particular encoding to each token in accordance with the sequence, disregarding the relationship between the pixels in the patch and the neighborhood in which they are located.

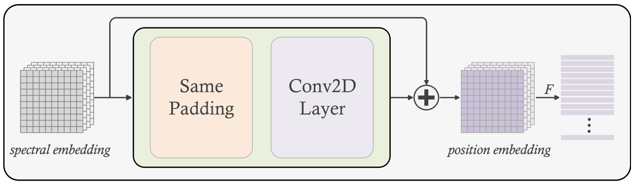


Fig. 3. Structure of CPE. Note that F is a flatten function that flattens the 2-D position embedding from $\mathbb{R}^{h \times w \times c}$ to $\mathbb{R}^{hw \times c}$.

Conditional position embedding (CPE) is a flexible parameter-free approach that can solve this defect. It hinges on the input token and its neighborhood to dynamically produce the position embedding token associated with the input token. Moreover, CPE is translation invariant, which allows it to efficiently leverage the local homogeneity of natural images. CPE can be easily implemented by the 2-D convolution layer and same padding layers. Fig. 3 illustrates the structure of CPE. After spectral embedding (SE) and position embedding (PE), the input of the transformer is

$$\mathbf{X}_{\text{embedded}} = SE(\mathbf{X}) + PE(SE(\mathbf{X})) \tag{5}$$

where $\mathbf{X}_{\text{embedded}} \in \mathbb{R}^{hw \times c}$; c represents the embedding dimension.

C. IMAE for HSI Spectral–Spatial Feature Extraction

In this section, we focus on how to extract general features in HSIs through the IMAE. Concretely, we perform self-supervised training for the IMAE through constructing two proxy tasks: 1) constructing a pixel-level masked AE to reconstruct the random masked input and 2) designing an instructor token to direct the model to concentrate on the region we are interested in.

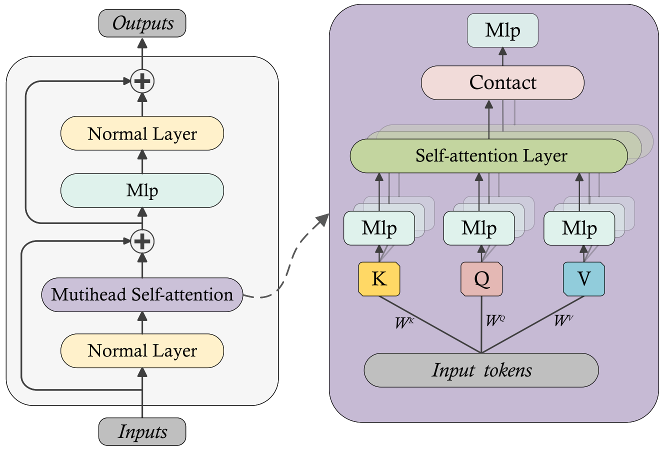


Fig. 4. Structure of a transformer block.

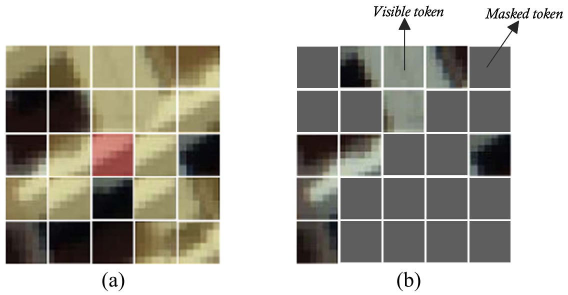


Fig. 5. 5 × 5 HSI patch. (a) Illustration of local homogeneity. The red pixel is the target pixel; the yellow area is made of the similar material as the target pixel. (b) Illustration of the visible token and the mask token.

The transformer is a flexible module known for its strong generalization capabilities, making it particularly well suited for transfer learning. Through the SA mechanism, it can capture long-term dependencies within input data and flexibly process inputs of different lengths. When dealing with HSIs with different spatial resolutions, the sizes of HSI samples would have a great impact on the final classification performance. To ensure that the IMAE possesses robust generalization capacity, we utilize the transformer as the fundamental module for constructing the network. A transformer encoder or decoder includes several blocks; each block is composed of a multihead self-attention layer (MSA), a multilayer perceptron (MLP), layer normalization (LN), and residual connection. The structure of the transformer block is shown in Fig. 4. The output token $\mathbf{Z}^{(l)}$ of the lth block can be computed as

$$\widehat{\mathbf{Z}}^{(l)} = \mathrm{MSA}(\mathrm{LN}(\mathbf{Z}^{(l-1)})) + \mathbf{Z}^{(l-1)}$$
$$\mathbf{Z}^{(l)} = \mathrm{LN}(\mathrm{MLP}(\widehat{\mathbf{Z}}^{(l)})) + \widehat{\mathbf{Z}}^{(l)}. \tag{6}$$

The attention mechanism can be achieved through three learnable matrices, namely, $\mathbf{W}^K$, $\mathbf{W}^Q$, and $\mathbf{W}^V$. These matrices allow the input tokens $\mathbf{X} = \{x_1, x_2, \ldots, x_n | x \in \mathbb{R}^d\}, \mathbf{X} \in \mathbb{R}^{n\times d}$ to be mapped into an assembly of query, key, and value vectors, respectively. They can be generated by matrix operation as follows:

$$\mathbf{Q} = \mathbf{X}\mathbf{W}^{Q^\top} = \{\mathbf{W}^Q x_1, \mathbf{W}^Q x_2, \ldots, \mathbf{W}^Q x_n\} \tag{7}$$
$$\mathbf{K} = \mathbf{X}\mathbf{W}^{K^\top} = \{\mathbf{W}^K x_1, \mathbf{W}^K x_2, \ldots, \mathbf{W}^K x_n\} \tag{8}$$
$$\mathbf{V} = \mathbf{X}\mathbf{W}^{V^\top} = \{\mathbf{W}^V x_1, \mathbf{W}^V x_2, \ldots, \mathbf{W}^V x_n\} \tag{9}$$

where $\mathbf{K}$, $\mathbf{Q}$, and $\mathbf{V}(\mathbf{K}, \mathbf{Q}, \mathbf{V} \in \mathbb{R}^{n\times m})$ represent the matrices that combined by the query, key, and value vectors, respectively. d represents the dimension of input tokens and m represents the dimension of tokens after mapping. Afterward, we use scaled dot product to compute the attention map by $\mathbf{K}$ and $\mathbf{Q}$ and generate the output tokens by $\mathbf{V}$ and the attention map, as follows:

$$\mathrm{Attr}(\mathbf{K}, \mathbf{Q}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \tag{10}$$

where $\mathrm{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}})$ represents the attention map and d_k represents the dimension of key tokens.

The multihead attention mechanism involves performing various attention operations on the tokens independently, followed by a weighted linear combination of the output through a learnable matrix $\mathbf{W}^O$. To be more specific, suppose that there are p heads$(\mathbf{H}_1, \mathbf{H}_1, \ldots, \mathbf{H}_p)$; the output of MSA can be computed as follows:

$$\mathbf{H}_i = \mathrm{Attr}(\mathbf{X}\mathbf{W}_i^{K^\top}, \mathbf{X}\mathbf{W}_i^{Q^\top}, \mathbf{X}\mathbf{W}_i^{V^\top}) \tag{11}$$
$$\mathbf{H} = \left[\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_p\right]\mathbf{W}^O \tag{12}$$

where $\mathbf{H}_i \in \mathbb{R}^{n\times m}$, $\left[\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_p\right] \in \mathbb{R}^{n\times pm}$, and $\mathbf{W}^O \in \mathbb{R}^{pm\times m}$.

In HSI analysis, the surrounding neighborhood of a target pixel is often used as input to expand information and enhance model performance. This approach may introduce semantic redundancy. In [53], it can help the model holistically understand beyond low-level image statistics through masking a high portion of random patches of natural images. Inspired by this work, we randomly mask the input data to destroy the semantic redundancy. After that, it conducts representation learning and reconstructs the original unmasked input via an AE. In this way, the model can implicitly learn the context and texture features in HSI samples. Fig. 5 illustrates the local homogeneity in HSI and our mask strategy.

Regrettably, since the transformer model performs indiscriminate global SA calculations on input tokens, lacks inductive bias, has a broad function domain, and disperses local attention, training the transformer network requires a large amount of data. In the context of HSI analysis, our primary focus lies in understanding the relationships between target pixels and their neighboring contexts. This constitutes a significant prior knowledge. Hence, we aimed to design a mechanism that can learn this prior knowledge during the network's pretraining phase. We introduced a random initialized token similar to the "*cls_token*" found in the ViT, which we refer to as the "*ins_token*" to represent global features of the input. Furthermore, we devised a proxy task for self-supervised training. To elaborate, we project the output of the *ins_token* and the spectral vector of the target pixel into a specific metric space and minimize the distance

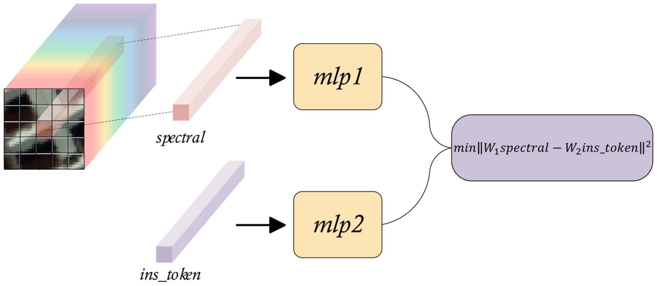


Fig. 6. Working mechanism of the instructor.

between them. This operation enables us to guide the model in learning components in the global information that are highly correlated with the target pixel, allowing the model to naturally focus on regions of interest and enhanced the local attention to target pixel. Theoretically, this instructional term can be considered as a form of regularization constraint for the AE, which serves to constrict the model's functional domain, subsequently diminishing the quantity of data necessary for fitting. Moreover, it works as a helpful manual for the aggregate of encoder tokens in the downstream task. Its working mechanism is shown in Fig. 6.

Specifically, after spatial–spectral embedding $\mathbf{X}_{\text{embedded}}$, we randomly mask and flatten $\mathbf{X}_{\text{embedded}}$ and then contact the ins_token to it as the input of encoder $\mathbf{X}_{\text{masked}} = \{ins\_token, x_1, x_2, \ldots, x_n \quad | ins\_token, x_i \in \mathbb{R}^c\}$. Let $\mathbf{Z}$ represent the latent features of $\mathbf{X}_{\text{masked}}$

$$\mathbf{Z} = \text{encoder}(\mathbf{X}_{\text{masked}}) = \{ins\_token, z_1, z_2, \ldots, z_n\}. \quad (13)$$

Afterward, we move the visible token to its original position and then fill the masked token with a random token, called $fill(\cdot)$

$$\mathbf{Z}_{\text{filled}} = fill(\mathbf{Z}). \quad (14)$$

Finally, we use $\mathbf{Z}_{\text{filled}}$ as the input of the decoder to reconstruct the original HSI patch $\mathbf{X}'$ as well as conduct instruction

$$\mathbf{X}' = \text{decoder}(\mathbf{Z}_{\text{filled}}) = \{ins\_token, x'_1, x'_2, \ldots, x'_{hw}\} \quad (15)$$

$$\min ||x_c - ins\_token||^2 \quad (16)$$

where x_c represents the center pixel. The loss function of the pretraining stage is

$$\begin{aligned} l &= l_r + \alpha l_{\text{ins}} \\ &= \frac{1}{hw} \sum_{i=1}^{hw} ||x_i - x'_i||^2 + \alpha ||x_c - ins\_token||^2. \end{aligned} \quad (17)$$

D. Learnable Aggregation

In the downstream task, in order to make full use of the information learned by the network, we propose a learnable aggregation to combine the tokens from the encoder and then feed its outputs to the classifier as the final logit for supervised training. Specifically, we use the uncovered patch $\mathbf{X} \in \mathbb{R}^{h \times w \times b}$ as model input in forward propagation. Let the output of encoder $\mathbf{Z} = \{ins\_token, z_1, z_2, \ldots, z_{hw} | ins\_token, z_i \in \mathbb{R}^d\}$. The final $logit$ can be computed as follows:

$$Z = [z_1, z_2, \ldots, z_{hw}]^T, \quad Z \in \mathbb{R}^{hw \times d} \quad (18)$$

$$Z' = [f(z_1), f(z_2), \ldots, f(z_{hw})]^T, \quad Z' \in \mathbb{R}^{hw \times d} \quad (19)$$

$$logit = clf(Z^T Z' g(ins\_token) + ins\_token)\} \quad (20)$$

where f and g represent MLP mapping, b represents the spectral bands of input, and d represents the embedding dimension of encoder. Finally, we employ the cross-entropy loss function to train the classifier, as follows:

$$\underset{\theta}{\text{minimize}}\, E(y, logits) = -\sum_{i=1}^{n} y_i \log(logit_i) \quad (21)$$

where θ represents the parameters of the model, y represents the ground truth of training data, and n represents the amount of training data.

IV. Experiment Results and Analysis

A. Description of Datasets

In the pretraining stage, we selected HSIs from a variety of scenes, including desert, forest, township, forest village, snowfield, village, city, and metropolis, and divided them into four patches of varying sizes 9, 15, 29, and 33, respectively. These HSIs were gathered by GaoFen-5 satellite, which contain 330 spectral bands in the wavelength range from 0.4 × 10^{-6} to 2.5 × 10^{-6} m. The spectral resolution of VNIR and SWIR is 10 and 20 nm, respectively. The size of each hyperspectral image is 2008 × 2083 and the spatial resolution of these data is 30 m per pixel. Thirty-three water absorption bands are removed in the process of data preprocessing. After pretraining, the performance of the proposed method is evaluated on three hyperspectral datasets, including IP, PU, and SA. These three datasets possess different spectral bands and spatial resolutions and are widely utilized for HSI classification tasks. Conducting experiments on these three datasets allows us to effectively evaluate the generalization performance of our model as well as its classification performance on downstream tasks.

1) *Indian Pines:* The IP dataset contains 145×145 pixels, which is gathered by the AVIRIS sensor in Northwestern Indiana, where AVIRIS stands for airborne visible infrared imaging spectrometer. The original IP dataset contains 220 spectral channels in the wavelength range from 0.4 × 10^{-6} to 2.5 × 10^{-6} m with a spatial resolution of 20 m. In this article, 20 bands corrupted by water absorption effects are discarded. It contains 16 classes and 10 249 labeled pixels in total.
2) *PaviaU:* The PU dataset contains 610×340 pixels collected by the ROSIS sensor at the University of Pavia, where ROSIS stands for reflective optics system imaging spectrometer. This image scene contains 103 spectral bands in the wavelength range from 0.43 × 10^{-6} to 0.86 × 10^{-6} m with a spatial resolution of 1.3 m. The dataset was provided by Prof. Paolo Gamba from the Telecommunications and Remote Sensing Laboratory,

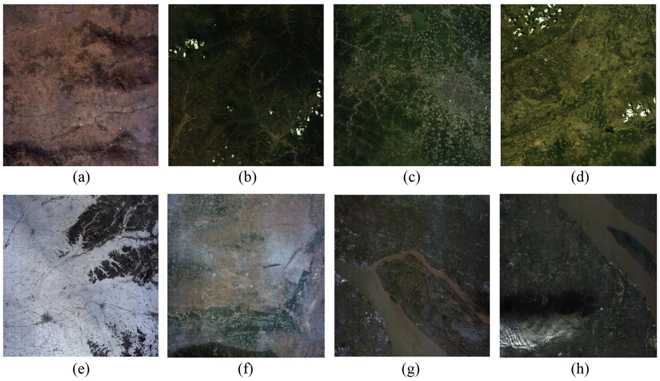


Fig. 7. False-color images of GaoFen-5 pretraining dataset. (a) Desert. (b) Forest. (c) Township. (d) Forest village. (e) Snowfield. (f) Village. (g) City. (h) Metropolis.

TABLE I
LAND COVER CLASS ILLUSTRATION AND NUMBER OF TRAINING AND TESTING SAMPLES FOR THE INDIAN PINES DATASET

No.	Class	Training	Testing	Total
1	Alfalfa	20	26	46
2	Corn-notill	20	1408	1428
3	Corn-mintill	20	810	830
4	Corn	20	217	237
5	Grass-pasture	20	463	483
6	Grass-trees	20	710	730
7	Grass-pasture-mowed	14	14	28
8	Hay-windrowed	20	450	478
9	Oats	10	10	20
10	Soybean-notill	20	952	972
11	Soybean-mintill	20	2435	2455
12	Soybean-clean	20	573	593
13	Wheat	20	185	205
14	Woods	20	1245	1265
15	Buildings-Grass-Trees	20	366	386
16	Stone-Steel-Towers	20	73	93
	Total	304	9945	10249

University of Pavia. It contains nine classes and 42 776 labeled pixels in total.

3) *Salinas:* The SA dataset contains 512×217 pixels also collected by the AVIRIS sensor over Salinas Valley, California. These data contain 224 spectral bands range from 0.4×10^{-6} to 2.5×10^{-6} m with a spatial resolution of 3.7 m. It contains 16 classes and 50 929 labeled pixels in total. In this article, 20 water absorption bands (108–112, 154–167, and 224) are removed during data preprocessing.

The false-color images of the GaoFen-5 dataset are shown in Fig. 7. The false-color images and ground truth of three widely used datasets are illustrated in Fig. 8. In the sample-limited scenario, the details of training and testing samples split on three widely used datasets are shown in Tables I–III. The descriptions of all the datasets are summarized in Table IV.

B. Training Details and Experimental Settings

In the pretraining phase, the HSIs in the GaoFen-5 dataset are sliced into samples with four divergent sizes 9, 15, 29, and 33. Samples of the same size are uncovered. (For instance, suppose that the size of the HSI is 100 × 100; we divide it into patches with two different sizes 10 and 20. Consequently, the number of samples with size 10 is 100, and the number of samples with size 20 is 25.) To compensate for the discrepancy in the number of samples of different sizes, we resample samples of larger size to align the number of samples of different sizes, hence eliminating the model's bias with regard to the input sample size. After aligning, the number of total samples is about 300 000.

TABLE II
LAND COVER CLASS ILLUSTRATION AND NUMBER OF TRAINING AND TESTING SAMPLES FOR THE SALINAS DATASET

No.	Class	Training	Testing	Total
1	Broccoli green weeds 1	20	1989	2009
2	Broccoli green weeds 2	20	3726	3726
3	Fallow	20	1956	1976
4	Fallow rough plow	20	1374	1394
5	Fallow smooth	20	2658	2678
6	Stubble	20	3939	3959
7	Celery	20	3559	3579
8	Grapes untrained	20	11251	11271
9	Soil vineyard develop	20	6183	6203
10	Corn senesced green weeds	20	3258	3278
11	Lettuce romaine 4 week	20	1048	1068
12	Lettuce romaine 5 week	20	1907	1927
13	Lettuce romaine 6 week	20	896	916
14	Lettuce romaine 7 week	20	1050	1070
15	Vineyard untrained	20	7248	7268
16	Vineyard vertical trellis	20	1787	1807
	Total	320	50609	50929

TABLE III
LAND COVER CLASS ILLUSTRATION AND NUMBER OF TRAINING AND TESTING SAMPLES FOR THE PAVIAU DATASET

No.	Class	Training	Testing	Total
1	Asphalt	20	6611	6631
2	Meadows	20	18629	18649
3	Gravel	20	2079	2099
4	Trees	20	3044	3064
5	Mental sheets	20	1325	1345
6	Bare soil	20	5009	5029
7	Bitumen	20	1310	1330
8	Bricks	20	3662	3682
9	Shadow	20	927	947
	Total	180	42596	42776

The mini-batch training strategy was employed during the training process. Besides, we designed a custom data loader; when sampling from the dataset, each step in each epoch has a separate size, so as to guarantee that the model will not be biased by the sizes of samples, as illustrated in Fig. 9.

During the fine-tuning stage, 20 samples per class were randomly selected as the training data. In case a certain class has fewer than 40 samples, 50% of them are assigned as training data. Details of the data assignments can be found in Tables I and II. Given that the number of spectral bands in the downstream task's data differs from that of the pretrained network, we have to substitute the input layer of the trained IMAE encoder with an alternative input layer that can adapt to the new hyperspectral data. Otherwise, the network is unable to execute matrix operations due to dimension mismatch. Subsequently, as aforementioned we aggregate the output tokens of the encoder and submit the output feature vectors to a randomly initialized classifier for supervised classification training.

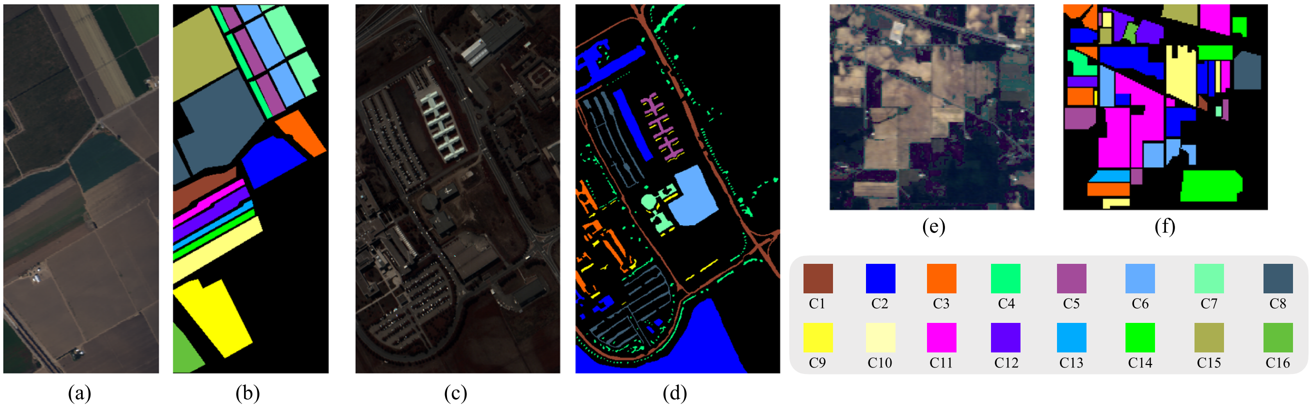


Fig. 8. False-color images and ground truth of three widely used dataset. (a) False-color images of Salinas. (b) Ground truth of Salinas. (c) False-color images of PaviaU. (d) Ground truth of PaviaU. (e) False-color images of Indian Pines. (f) Ground truth of Indian Pines.

TABLE IV
DETAILS OF ALL DATASETS

Dataset	Sensor	Bands	Spatial Resolution	Spectral Resolution	Size	Annotated Samples	Classes	Acquisition Year
GaoFen-5	AHSI	330	30m	10nm&20nm	2008*2083	-	-	2019
Indian Pines	AVIRIS	200	20m	30nm	145*145	10249	16	1992
PaviaU	ROSIS	103	1.3m	30nm	610*340	42776	9	2001
Salinas	AVIRIS	204	3.7m	30nm	512*217	50929	16	1998

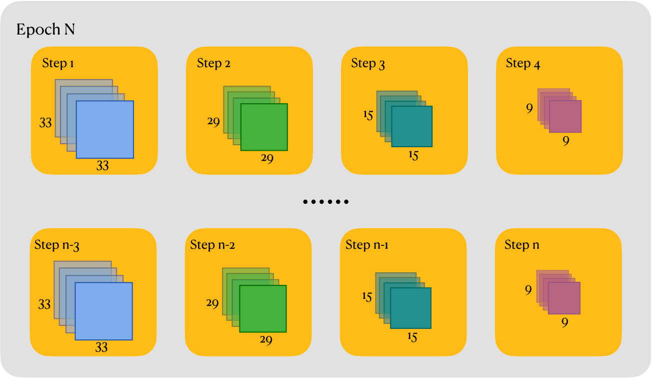


Fig. 9. Sample strategy of custom data loader. In this diagram, the yellow blocks depict the steps within an epoch. The blue, green, turquoise, and pink squares represent the HSI samples of varying sizes. Each sample encompasses a set of pixels centered around the target pixel.

The implementation of our method is very sample, which is completed entirely on the PyTorch platform. In the pretraining stage, a server with two A40 computing cards and 256-GB memory was employed as the hardware platform; the mask ratio, embedding dimension, depth, and heads of encoder were set to 0.5, 256, 4, and 8, respectively; the number of parameters of the decoder was half of it. AdamW was utilized as an optimizer, and the learning rate was set to 8×10^{-4}. In the downstream task, we use a terminal with an RTX3090 graphics card and 56-GB memory as the computing platform; the learning rate of the encoder and the classifier was set to 10^{-5} and 10^{-3}, respectively.

In order to quantify the classification performance of our method, the overall accuracy (OA), average accuracy (AA), and kappa coefficient (Kappa) were employed as evaluation measures. OA is the ratio of the number of correctly labeled hyperspectral pixels to the total number of hyperspectral pixels in test samples. AA is the mean of accuracy in different land cover categories. Kappa measures the consistency between classification results and ground truth. The larger values of OA, AA, and Kappa represent the better classification results.

C. Classification Results

To verify the advancement of the our method, we compared the classification results with SVM [57], RNN [58], 3-D CNN [59], ViT, HIT [22], MAEST [54], SSTN [23], and SS-MTr [55]. Among these comparative methods, SVM is a classic machine learning method. RNN and 3-D CNN are mainstream deep learning methods. ViT, HIT, and SSTN are transformer-based methods, in particular, ViT is the first transformer-based model used for image processing. HIT and SSTN have implemented some improvements on its basis to make it more suitable for HSI classification tasks. Similar to our method, MAEST and SS-MTr are pretraining methods with backbone network as MAE. The training data assignments for all compared methods as the same as those for the IMAE; the size of the input samples for CNN-based and transformer-based methods was set to 15×15. Tables V–VII record the classification results of different methods on IP, PU, and SA datasets, including accuracy for each class and OA, AA, and Kappa for all classes. The best results are highlighted in bold. Figs. 10–12 illustrate the classification maps of different methods.

D. Discussion

Based on the empirical evidence derived from our experiments, it becomes apparent that traditional machine learning

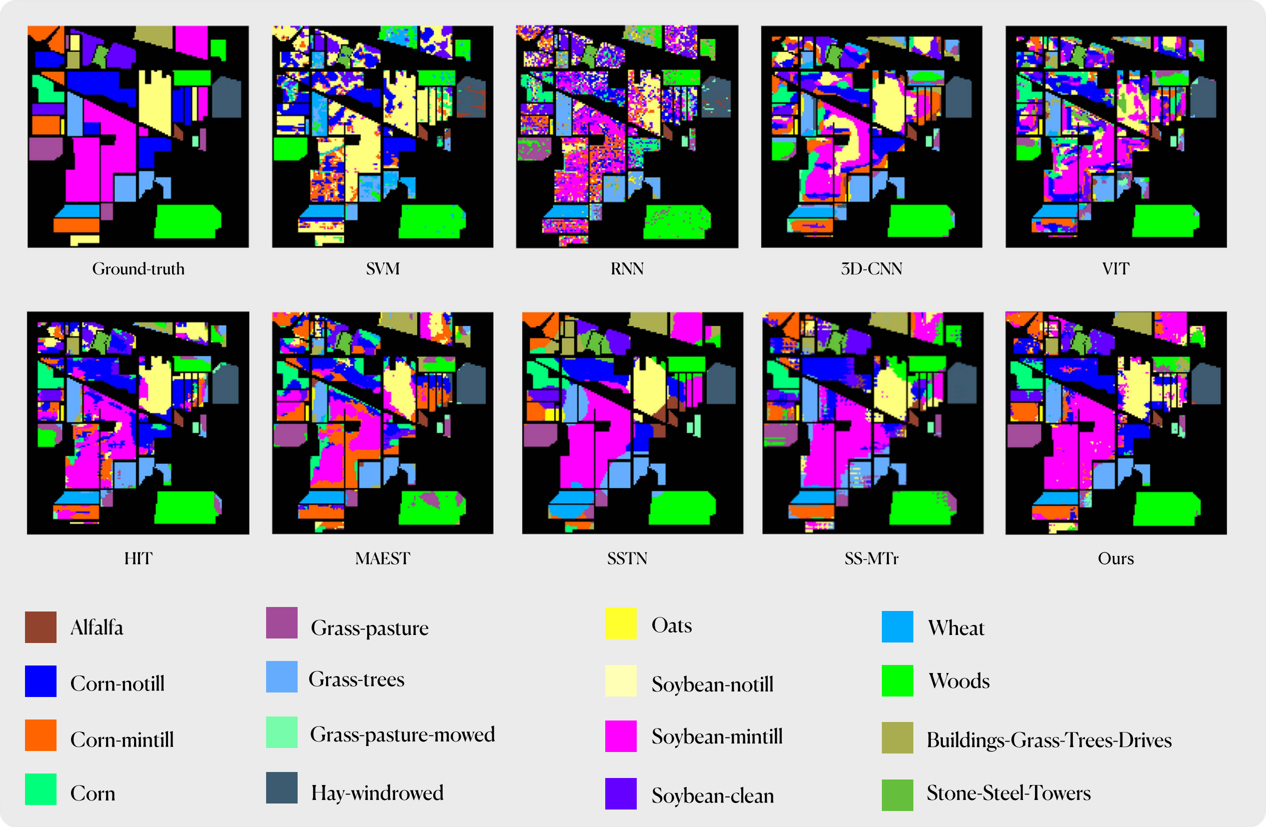


Fig. 10. Classification maps using different methods on the Indian Pines dataset.

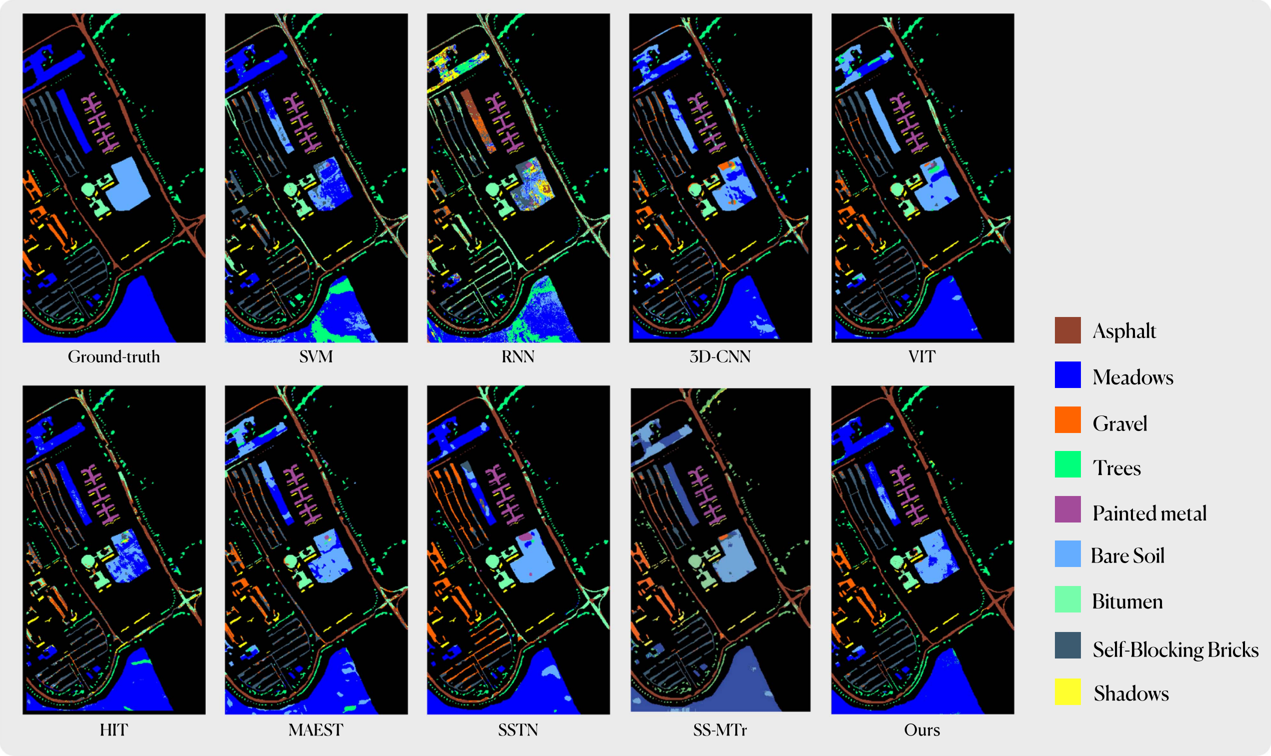


Fig. 11. Classification maps using different methods on the PaviaU dataset.

TABLE V
CLASSIFICATION RESULTS OF DIFFERENT METHODS USING 20 TRAINING SAMPLES PER CLASS ON THE INDIAN PINES DATASET

Classes	SVM	RNN	3D-CNN	VIT	HIT	MAEST	SSTN	SS-MTr	Ours
1	97.83%	84.78%	97.83%	91.3%	95.65%	**100%**	**100%**	96.15%	95.65%
2	47.76%	30.74%	35.64%	35.85%	66.32%	41.81%	57.29%	72.51%	**75.35%**
3	10.12%	31.45%	36.39%	19.88%	35.06%	66.27%	49.76%	**78.40%**	74.22%
4	17.72%	63.71%	44.73%	64.98%	81.86%	76.79%	93.25%	97.24%	**98.31%**
5	0%	62.32%	13.25%	25.47%	39.75%	**87.37%**	80.12%	77.32%	81.16%
6	39.86%	85.34%	79.45%	63.97%	88.63%	84.11%	88.9%	**99.15%**	92.47%
7	0%	92.86%	**100%**	**100%**	**100%**	96.43%	**100%**	**100%**	96.43%
8	80.75	88.28%	89.96%	85.98%	85.77%	98.95%	99.79%	**100%**	98.54%
9	0%	85%	**100%**	**100%**	**100%**	**100%**	80%	100%	**100%**
10	74.07%	44.96%	59.88%	35.49%	64.71%	57.10%	66.36%	**87.5%**	76.13%
11	1.87%	38.7%	34.50%	38.04%	55.11%	43.14%	88.47%	69.86%	**86.03%**
12	22.09%	52.45%	41.15%	32.04%	51.77%	31.53%	70.49%	**72.95%**	72.68%
13	99.51%	98.05%	77.56%	98.54%	96.59%	99.51%	98.54%	**100%**	99.51%
14	**94.31%**	88.30%	74.23%	79.76%	83.72%	72.72%	92.17%	91.97%	85.77%
15	3.37%	48.19%	23.83%	31.87%	43.01%	81.34%	**100%**	95.08%	95.60%
16	90.32%	96.77%	**100%**	93.55%	98.92%	97.84%	**100%**	**100%**	**100%**
OA	38.26%	54.36%	49.18%	46.95%	64.17%	61.09%	79.39%	81.82%	**83.79%**
AA	42.47%	68.24%	63.02%	62.30%	74.18%	77.18%	85.32%	**89.88%**	89.24%
Kappa	32%	49.21%	44.17%	41.37%	59.98%	56.76%	76.8%	79.45%	**81.60%**

TABLE VI
CLASSIFICATION RESULTS OF DIFFERENT METHODS USING 20 TRAINING SAMPLES PER CLASS ON THE PAVIAU DATASET

Classes	SVM	RNN	3D-CNN	VIT	HIT	MAEST	SSTN	SS-MTr	Ours
1	61.02%	51.95%	63.17%	67.26%	53.55%	70.68%	71.62%	90.15%	**95.34%**
2	70.57%	47.36%	66.89%	61.67%	82.34%	76.14%	**98.04%**	74.66%	92.92%
3	16.15%	29.25%	55.60%	58.89%	36.11%	72.89%	87.28%	**90.52%**	90.33%
4	**96.70%**	90.05%	81.46%	85.93%	92.20%	89.46%	58.02%	85.48%	89.85%
5	99.11%	99.11%	85.50%	**100%**	99.85%	99.78%	**100%**	99.40%	99.26%
6	33.27%	34.86%	64.94%	80.71%	46.25%	80.97%	**92.42%**	91.08%	80.49%
7	95.26%	98.50%	78.65%	77.82%	89.10%	87.44%	**100%**	99.01%	89.47%
8	81.72%	48.18%	70.56%	**89.33%**	62.49%	79.44%	87.05%	85.94%	87.62%
9	72.63%	**99.05%**	92.19%	92.93%	99.26%	99.89%	39.5%	91.80%	93.24%
OA	67.18%	53.20%	68.40%	71.16%	71.50%	78.56%	87.78%	84.95%	**91.13%**
AA	72.63%	66.48%	73.22%	79.39%	73.46%	84.08%	81.55%	89.78%	**90.95%**
Kappa	58.10%	44.52%	60.55%	64.66%	63.22%	72.7%	83.78%	80.91%	**88.28%**

and deep learning algorithms struggle to perform effectively in scenarios marked by a paucity of available samples.

It is indicated that such methods may not be adequate for small-sample training. For the SVM, the high-dimensional feature space of HSIs can cause the curse of dimensionality and increase computational complexity, leading to suboptimal classification results if inappropriate feature subset or extraction methods are chosen. Moreover, noise and spectral mixing in HSIs can result in irregular sample distribution and affect SVM performance. In addition, a large number of support vectors may lead to overfitting problems. For the RNN, it is a sequence model primarily used to handle data with temporal sequence features. Although HSIs are also sequential data, their temporal sequence features are not prominent and the data dimensionality is high, with complex interrelationships between different dimensions. Therefore, using RNN to process the spatiotemporal information of HSIs may face significant challenges. In addition, in HSI classification tasks, data preprocessing is typically required, such as dimensionality reduction, denoising, and normalization. These operations may result in the loss of temporal sequence features in the data, thereby affecting the performance of RNN in HSI classification tasks. For the 3-D CNN, it is a complex model that has a large number of parameters to learn, necessitating a substantial amount of data for fitting. Therefore, in the sample-limited scenario, training the 3-D CNN becomes challenging and prone to overfitting. In addition, during the preprocessing stage, operations like dimensionality reduction, denoising, and normalization may cause the loss of spatiotemporal information in the data. Since the 3-D CNN relies on this information to improve classification performance, the preprocessing steps can potentially impact the effectiveness of the 3-D CNN. ViT, as the first model designed for visual tasks based on transformers, performs poorly in limited-sample high spectral classification tasks for several possible reasons. Firstly, it requires more data for training due to the lack of inductive bias and the complex architecture with numerous parameters, requiring more data for training. Insufficient training samples can result in incomplete learning and failure to converge. Second, ViT is more suitable for handling image data with clear spatial structures, as it relies on SA mechanisms. However, in the case of high spectral images, the spatial correlation between pixels is relatively weak. ViT primarily focuses on global correlations within input sequences, potentially failing to fully leverage the local and spatial features of high spectral images. Third, high spectral images often possess a large number of spectral bands, resulting in high-dimensional feature spaces. ViT is sensitive to the length of input

TABLE VII
CLASSIFICATION RESULTS OF DIFFERENT METHODS USING 20 TRAINING SAMPLES PER CLASS ON THE SALINAS DATASET

Classes	**SVM**	**RNN**	**3D-CNN**	**VIT**	**HIT**	**MAEST**	**SSTN**	**SS-MTr**	**Ours**
1	99.65%	97.31%	54.95%	74.51%	79.24%	90.69%	**100%**	98.59%	99.15%
2	48.66%	96.56%	96.46%	64.04%	97.80%	43.26%	**99.97%**	95.84%	97.58%
3	40.79%	87.35%	92.86%	85.98%	96.96%	65.64%	**99.9%**	97.96%	92.76%
4	98.78%	98.78%	90.32%	95.98%	96.56%	93.97%	99.14%	97.89%	**99.57%**
5	97.98%	98.77%	88.87%	80.77%	95.33%	81.07%	89.32%	**100%**	94.88%
6	96.84%	99.49%	95.33%	95.38%	95.38%	97.42%	**99.97%**	99.92%	99.72%
7	98.60%	99.69%	91.95%	95.39%	93.66%	95.95%	**99.78%**	99.75%	98.97%
8	62.31%	33.86%	83.44%	65.84%	72.62%	61.76%	68.32%	70.3%	**85.18%**
9	95.81%	99.85%	97.79%	93.36%	96.5%	98.61%	97.94%	**99.98%**	97.10%
10	1.98%	70.44%	81.94%	88.87%	86.15%	45.85%	98.54%	**98.68%**	80.29%
11	66.10%	91.39%	69.94%	87.83%	93.07%	96.16%	**100%**	**100%**	98.69%
12	88.69%	98.65%	88.22%	93.15%	92.79%	**100%**	98.86%	98.32%	98.24%
13	99.02%	99.02%	93.23%	93.45%	94.21%	98.14%	100%	**100%**	98.91%
14	88.22%	91.50%	88.13%	91.21%	94.02%	99.44%	99.44%	**100%**	**100%**
15	65.63%	87.08%	22.43%	72.34%	64.72%	66.39%	**96.87%**	30.81%	85.66%
16	41.84%	97.51%	57.44%	60.71%	65.58%	90.81%	99.89%	**100%**	89.71%
OA	71.70%	81.25%	78.16%	80.03%	84.45%	76.61%	91.76%	83.84%	**92.20%**
AA	74.43%	90.45%	80.83%	83.68%	88.41%	82.82%	**96.62%**	93%	94.78%
Kappa	68.53%	79.37%	75.69%	82.86%	82.86%	74.11%	90.88%	81.96%	**91.34%**

sequences, and when there are too many bands, the input sequence length becomes long, leading to increased computational complexity and potential difficulties in model training. Finally, high spectral images commonly suffer from issues such as noise and spectral mixing. ViT lacks robustness against noise and may be overly sensitive to outliers when facing high levels of noise or severe spectral mixing, leading to decreased classification performance. However, the well-designed transformer-based networks have achieved impressive performance, such as HIT and SSTN. Their innovation lies in the integration of CNNs and transformers to overcome the limitation of transformers in capturing only global features while neglecting local features. Specifically, they employ CNNs for local feature extraction and utilize transformers to capture long-term dependencies between these local features. Yet, the introduction of CNNs has impeded the generalization performance of these approaches, making it difficult to accommodate the samples of varying sizes. The pretraining-based networks have also obtained competitive performance, like MAEST and SS-MTr. The difference between them is that the MAEST applies masking in the spectral dimension, allowing it to capture hidden information within the spectrum and effectively suppress noise. This approach focuses on learning spectral features for improved performance. On the other hand, SS-MTr applies masking in the spatial dimension and combines it with a convolutional network to learn spatial–spectral features. Despite that, limitations imposed by their model architectures and training strategies hinder their ability to exploit extensive pools of unlabeled data for pretraining, leaving room for further enhancement.

Our methodology incorporates an ins_token, enhancing the model's ability to capture features relevant to both the global context and specific target pixels. Furthermore, we leverage extensive pretraining on a large-scale dataset of HSIs, facilitating the comprehensive learning of generic features embedded in the data. Consequently, in comparable conditions, our approach surpasses existing methods, achieving state-of-the-art performance in the field. Specifically, on the IP dataset, we attained an OA of 83.79%, an AA of 89.24%, and a kappa coefficient of 81.60%. Similarly, on the SA dataset, our model achieved an OA of 92.2%, an AA of 94.78%, and a Kappa of 91.34%. On the PU dataset, our performance metrics were recorded at 91.13% for OA, 90.95% for AA, and 88.28% for Kappa. Across these three datasets, our model outperforms traditional machine learning and deep learning methods by a substantial margin. In comparison to the enhanced ViT model, our approach, including the best performing model SSTN within it, exhibits notable improvements across various performance indicators. Furthermore, in comparison to similar pretraining methods, our model surpasses MAEST in terms of AA, OA, and Kappa on all datasets. Compared to SS-MTr, the AA score of IMAE is on par with it except for the Indian Pines dataset. In all other datasets, our model consistently outperforms SS-MTr across various performance metrics.

The generalization performance of the model is the core metric of our method. In this section, we first test the reconstruction ability of the pretrained model. Afterward, we to assess the generalizability of IMAE from the perspective of training and inference of downstream tasks. Finally, we analyze the influence of pretrained weights on model convergence speed.

As abovementioned, we random mask 50% HSI samples and then reconstruct it to the original samples through a transformer-based AE. PSNR and SSIM are employed to evaluate the reconstruction performance. The average value of PSNR and SSIM on test set are 50 dB and 0.99, respectively, which means the latent knowledge of HSI was fully learnt by our model, and the overfitting did not take place. Fig. 13 shows the original samples, masked samples, and reconstructed samples.

In the pretrained stage, the IMAE was trained by HSI samples with four different sizes, namely, 33, 29, 15, and 9. To examine the generation capacity of our pretrained model on the sample size, we random selected 10% samples of per class in three widely used datasets with three different sizes, which are distinct from it in the pretraining dataset. The classification results are shown in Fig. 14. It is evident that despite the fact that the size

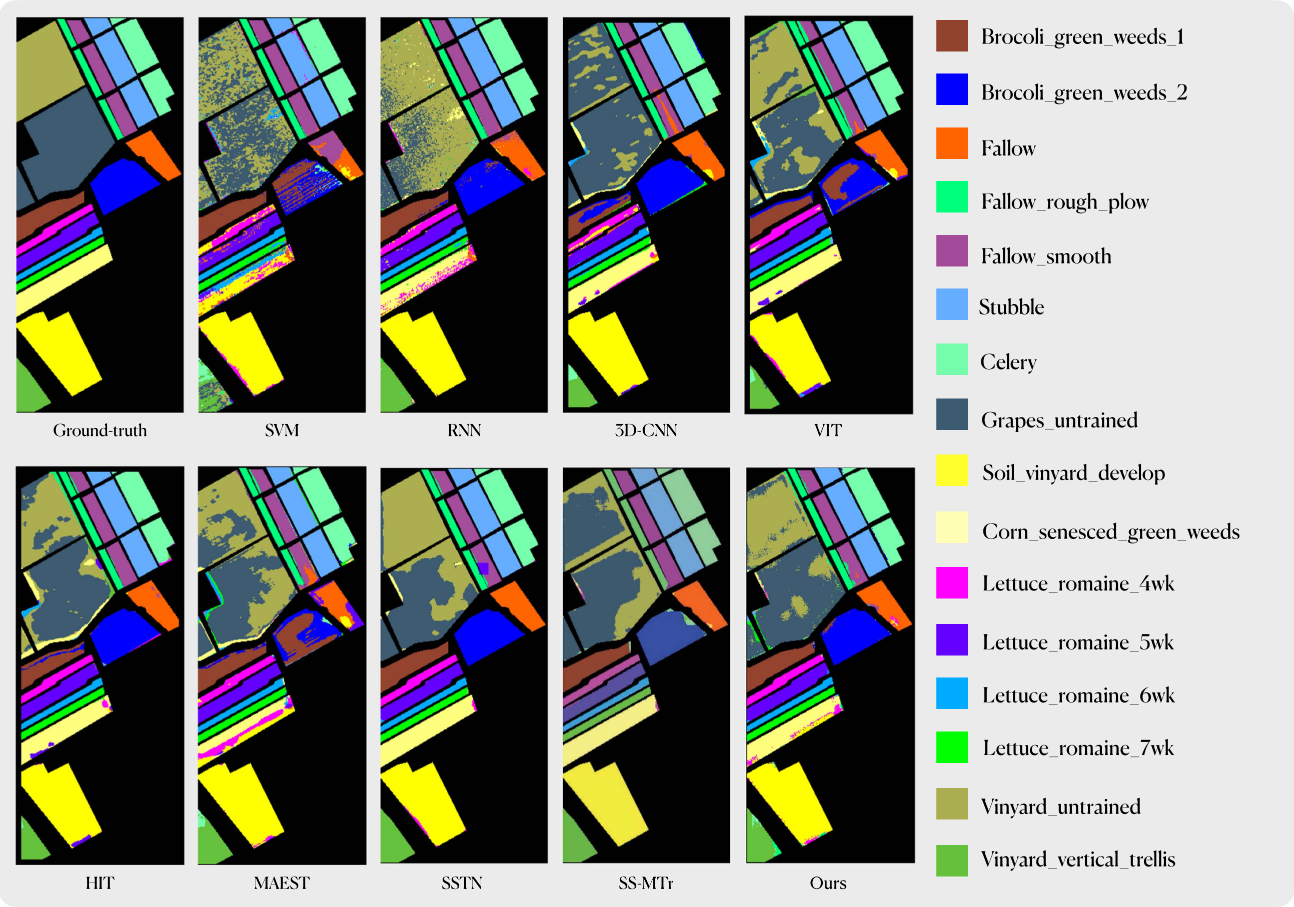


Fig. 12. Classification maps using different methods on the Salinas dataset.

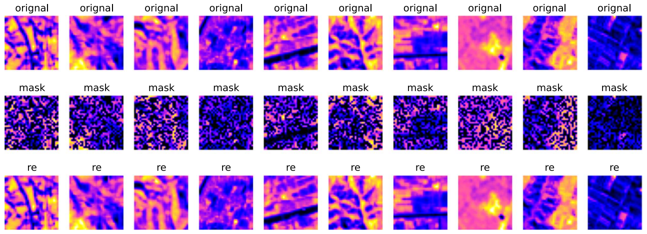


Fig. 13. Reconstruction examples obtained by the IMAE with 50% masking ratio.

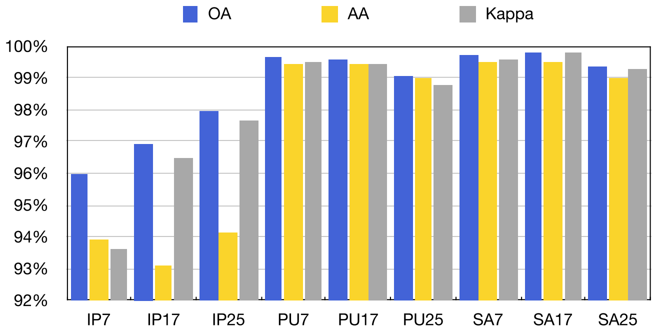


Fig. 14. Classification results of different training sample sizes different from the pretraining dataset.

and spectral resolution of the training data in the downstream tasks are not consistent with those in the pretraining dataset, our method still achieves excellent classification results on these data.

In the inference stage, we only fine-tune on the training set with a sample size of 15. Then, we evaluate the classification accuracy of our inferences using samples whose sizes differ from those in the training set. The experiment result is illustrated in Fig. 15.

Obviously, the common feature of the three curves in Fig. 15 is that when the input sample size is small, the inference accuracy is also small. As the input sample size increases, the inference accuracy also increases sharply until the inference sample size is equal to the training sample size. The inference accuracy gradually declines as the inference sample is larger than the training sample. We postulate that the reason for this phenomenon is that when the input sample size is small, the model is unable to learn enough contextual information, leading to low inference accuracy; when the input sample size is large, due to the presence of ins_token, the model prefers to focus on areas close to the center pixel, allowing the model to suppress invalid information brought on by the increase in input sample size, thereby lessening the impact on inference accuracy.

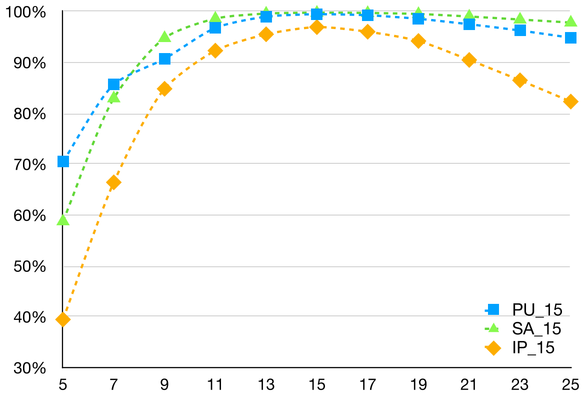
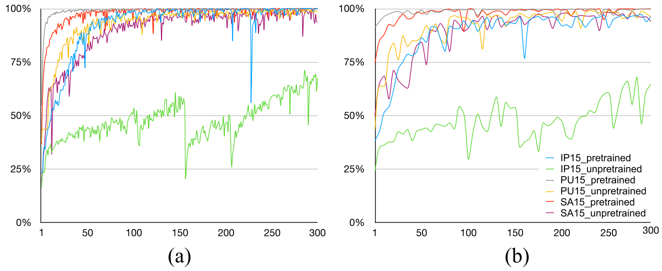


Fig. 15. Inference performance on different input sample sizes where the model was fine-tuned on training samples with a fixed size of 15.



(a)

(b)

Fig. 16. Accuracy curves in the training process with 10% training data each class. The accuracy curves during (a) training and (b) testing.

In the classification task, as shown in Fig. 16, our method can greatly improve the performance and speed up the convergence rate especially when the training data are relatively small. By observing the curves in the figure, we find that training with randomly initialized weights converges slowly on PU and SA datasets, and it not converges on the IP dataset. When pretrained weights were utilized in the training process, it notably expedited convergence on SA and PU datasets, achieving a substantial level of convergence on the IP dataset. The resulting accuracy was comparable with that of some state-of-the-art methods. To achieve this result, all we did was simply replace the input layer of the pretrained IMAE.

V. Conclusion

In this article, we devised a pretraining model tailored for the HSI based on the principles of self-supervised learning. This approach leverages copious amounts of unlabeled hyperspectral data as training material. Through a masking and reconstruction mechanism, it captures intrinsic spectral spatial characteristics prevalent within HSIs. In addition, it employs metric learning to guide the model's focus toward points of interest. Our method exhibits robust generalization capabilities, which we have rigorously tested in both training and inference phases. Remarkably, using a consistent set of pretraining weights, our model demonstrates outstanding generalization performance across multimodal inputs with varying spectral resolutions, spatial resolutions, and input sample sizes. For fine-tuning the IMAE on new datasets, a simple adjustment of the input layer to accommodate different spectral resolutions suffices. This adaptation significantly expedites model convergence and enhances performance in downstream tasks, particularly in scenarios characterized by limited samples. When compared to classical and state-of-the-art methods under identical conditions, our model attains state-of-the-art performance. The approach we have introduced opens up new possibilities for the application of large pretrained models in the domain of hyperspectral imagery. However, owing to the diversity in hyperspectral sensor parameters, many HSIs come with varying numbers of spectral channels. Constructing different input channels and training them can be a highly resource-intensive task. Our future research endeavors will focus on exploring methods to unify the channel numbers of HSIs with different spectral resolutions. This approach allows for the seamless integration of HSIs generated by various sensors without necessitating the replacement of the input layer and promoting generalization, efficiency, and cost-effectiveness.

References

[1] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, Jun. 2013.

[2] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019.

[3] P. Ghamisi et al., "Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 37–78, Dec. 2017.

[4] A. Plaza et al., "Recent advances in techniques for hyperspectral image processing," *Remote Sens. Environ.*, vol. 113, pp. S110–S122, 2009.

[5] R. Behling, M. Bochow, S. Foerster, S. Roessner, and H. Kaufmann, "Automated GIS-based derivation of urban ecological indicators using hyperspectral remote sensing and height information," *Ecol. Indicators*, vol. 48, pp. 218–234, 2015.

[6] 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.

[7] E. Bedini, "The use of hyperspectral remote sensing for mineral exploration: A review," *J. Hyperspectral Remote Sens.*, vol. 7, no. 4, pp. 189–211, 2017.

[8] G. Lu and B. Fei, "Medical hyperspectral imaging: A review," *J. Biomed. Opt.*, vol. 19, no. 1, 2014, Art. no. 010901.

[9] P. Ghamisi et al., "New frontiers in spectral-spatial hyperspectral image classification: The latest advances based on mathematical morphology, Markov random fields, segmentation, sparse representation, and deep learning," *IEEE Geosci. Remote Sens. Mag.*, vol. 6, no. 3, pp. 10–43, Sep. 2018.

[10] 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.

[11] W. Wang, Y. Chen, X. He, and Z. Li, "Soft augmentation-based siamese CNN for hyperspectral image classification with limited training samples," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2021, Art. no. 5508505.

[12] R. Hang, Q. Liu, D. Hong, and P. Ghamisi, "Cascaded recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5384–5394, Aug. 2019.

[13] Q. Liu, F. Zhou, R. Hang, and X. Yuan, "Bidirectional-convolutional LSTM based spectral-spatial feature learning for hyperspectral image classification," *Remote Sens.*, vol. 9, no. 12, 2017, Art. no. 1330.

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

[15] H. Hu, M. Yao, F. He, and F. Zhang, “Graph neural network via edge convolution for hyperspectral image classification,” *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2021, Art. no. 5508905.

[16] 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.

[17] S. Wan, C. Gong, P. Zhong, S. Pan, G. Li, and J. Yang, “Hyperspectral image classification with context-aware dynamic graph convolutional network,” *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 597–612, Jan. 2021.

[18] B. Liu, K. Gao, A. Yu, W. Guo, R. Wang, and X. Zuo, “Semisupervised graph convolutional network for hyperspectral image classification,” *J. Appl. Remote Sens.*, vol. 14, no. 2, 2020, Art. no. 026516.

[19] B. Liu, W. Kong, and Y. Wang, “Deep convolutional asymmetric autoencoder-based spatial-spectral clustering network for hyperspectral image,” *Wireless Commun. Mobile Comput.*, vol. 2022, 2022, Art. no. 2027981.

[20] J. He, L. Zhao, H. Yang, M. Zhang, and W. Li, “HSI-BERT: Hyperspectral image classification using the bidirectional encoder representation from transformers,” *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 165–178, Jan. 2020.

[21] Y. Qing, W. Liu, L. Feng, and W. Gao, “Improved transformer net for hyperspectral image classification,” *Remote Sens.*, vol. 13, no. 11, 2021, Art. no. 2216.

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

[23] 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, 2021, Art. no. 5514715.

[24] H. Yu, Z. Xu, K. Zheng, D. Hong, H. Yang, and M. Song, “MSTNet: A multilevel spectral–spatial transformer network for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5532513.

[25] L. Yang et al., “FusionNet: A convolution–transformer fusion network for hyperspectral image classification,” *Remote Sens.*, vol. 14, no. 16, 2022, Art. no. 4066.

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

[27] C. Zhao et al., “Hyperspectral image classification with multi-attention transformer and adaptive superpixel segmentation-based active learning,” *IEEE Trans. Image Process.*, vol. 32, pp. 3606–3621, 2023.

[28] A. Vaswani et al., “Attention is all you need,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[29] M. Kaya and H. Ş. Bilge, “Deep metric learning: A survey,” *Symmetry*, vol. 11, no. 9, 2019, Art. no. 1066.

[30] X. Chu, Z. Tian, B. Zhang, X. Wang, and C. Shen, “Conditional positional encodings for vision transformers,” in *Proc. 11th Int. Conf. Learn. Representations*, 2022.

[31] M. Lin, Q. Chen, and S. Yan, “Network in network,” 2013, *arXiv:1312.4400*.

[32] L. Samaniego, A. Bárdossy, and K. Schulz, “Supervised classification of remotely sensed imagery using a modified k-NN technique,” *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 7, pp. 2112–2125, Jul. 2008.

[33] Y. Bazi and F. Melgani, “Toward an optimal SVM classification system for hyperspectral remote sensing images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 11, pp. 3374–3385, Nov. 2006.

[34] J. Li, J. M. Bioucas-Dias, and A. Plaza, “Semisupervised hyperspectral image classification using soft sparse multinomial logistic regression,” *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 2, pp. 318–322, Mar. 2013.

[35] 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.

[36] S. Prasad and L. M. Bruce, “Limitations of principal components analysis for hyperspectral target recognition,” *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 4, pp. 625–629, Oct. 2008.

[37] A. Villa, J. A. Benediktsson, J. Chanussot, and C. Jutten, “Hyperspectral image classification with independent component discriminant analysis,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 12, pp. 4865–4876, Dec. 2011.

[38] T. V. Bandos, L. Bruzzone, and G. Camps-Valls, “Classification of hyperspectral images with regularized linear discriminant analysis,” *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 3, pp. 862–873, Mar. 2009.

[39] M. Ahmad, A. M. Khan, M. Mazzara, and S. Distefano, “Multi-layer extreme learning machine-based autoencoder for hyperspectral image classification,” in *Proc. Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2019, pp. 75–82.

[40] A. Mughees and L. Tao, “Efficient deep auto-encoder learning for the classification of hyperspectral images,” in *Proc. IEEE Int. Conf. Virtual Reality Vis.*, 2016, pp. 44–51.

[41] P. Zhong, Z. Gong, S. Li, and C.-B. Schönlieb, “Learning to diversify deep belief networks for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3516–3530, Jun. 2017.

[42] D. Hong et al., “Interpretable hyperspectral artificial intelligence: When nonconvex modeling meets hyperspectral remote sensing,” *IEEE Geosci. Remote Sens. Mag.*, vol. 9, no. 2, pp. 52–87, Jun. 2021.

[43] H. Xu, W. Yao, L. Cheng, and B. Li, “Multiple spectral resolution 3D convolutional neural network for hyperspectral image classification,” *Remote Sens.*, vol. 13, no. 7, 2021, Art. no. 1248.

[44] W. Li, H. Chen, Q. Liu, H. Liu, Y. Wang, and G. Gui, “Attention mechanism and depthwise separable convolution aided 3DCNN for hyperspectral remote sensing image classification,” *Remote Sens.*, vol. 14, no. 9, 2022, Art. no. 2215.

[45] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, “Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing,” *ACM Comput. Surv.*, vol. 55, no. 9, pp. 1–35, 2023.

[46] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, “Transformers in vision: A survey,” *ACM Comput. Surv.*, vol. 54, no. 10, pp. 1–41, 2022.

[47] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, *arXiv:1810.04805*.

[48] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving language understanding by generative pre-training,” 2018.

[49] A. Radford et al., “Language models are unsupervised multitask learners,” *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[50] T. Brown et al., “Language models are few-shot learners,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 1877–1901.

[51] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” 2020, *arXiv:2010.11929*.

[52] H. Bao, L. Dong, S. Piao, and F. Wei, “BEiT: BERT pre-training of image transformers,” 2021, *arXiv:2106.08254*.

[53] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16000–16009.

[54] D. Ibanez, R. Fernandez-Beltran, F. Pla, and N. Yokoya, “Masked autoencoding spectral–spatial transformer for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5542614.

[55] L. Huang, Y. Chen, and X. He, “Spectral-spatial masked transformer with supervised and contrastive learning for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5508718.

[56] L. Scheibenreif, M. Mommert, and D. Borth, “Masked vision transformers for hyperspectral image classification,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 2165–2175.

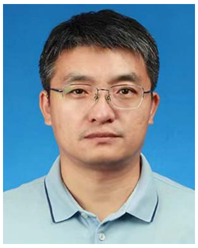
[57] M. Pal and G. M. Foody, “Feature selection for classification of hyperspectral data by SVM,” *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 5, pp. 2297–2307, May 2010.

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

[59] M. Ahmad et al., “A disjoint samples-based 3D-CNN with active transfer learning for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5539616.

Weili Kong (Student Member, IEEE) received the B.S. degree in electronic information engineering from the Heilongjiang Institute of Technology, Harbin, China, in 2021. He is currently working toward the Ph.D. degree in information and communication engineering with Harbin Engineering University, Harbin.

His research interests include remote sensing image analysis, multimodal image analysis, and deep learning.



Baisen Liu received the Ph.D. degree in information and communication engineering from the Information and Communication School, Harbin Engineering University, Harbin, China, in 2011.

He joined the School of Signal and Information Processing, Heilongjiang Institute of Technology, Harbin, as the Director. His research interests include multimodal image analysis, remote sensing image classification, and remote sensing image clustering.

Xiaojun Bi received the Ph.D. degree in information and communication engineering from the College of Information and Communication Engineering, Harbin Engineering University, Harbin, China, in 2006.

She is currently a Professor with Minzu University of China, Beijing, China, where she is also the Director of the Key Laboratory of Ethnic Language Intelligent Analysis and Security Governance of MOE. Before this, she was a Professor with Harbin Engineering University. Her research interests include evolutionary computation, multiobjective optimization, image/video analysis, deep learning, and natural language processing.

Jiaming Pei (Student Member, IEEE) received the B.S. degree in information system and information management from Taizhou University, Taizhou, China, in 2021. He is currently working toward the master's degree with the University of Sydney, Sydney, NSW, Australia.

From 2021 to 2022, he visited the Southwestern University of Finance and Economics, Chengdu, China. He has authored or coauthored and worked on some papers in the refereed journals, such as *Neural Computing and Applications*, *Information Processing and Management*, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, and IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING. His research interests include the application of data mining and computer science.

Zheng Chen received the Ph.D. degree in information and communication engineering from the College of Information and Communication Engineering, Harbin Engineering University, Harbin, China, in 2023.

He is currently a Lecturer with the Minzu University of China, Beijing, China, where he is also the leading Member of the Key Laboratory of Ethnic Language Intelligent Analysis and Security Governance of MOE, Minzu University of China. His research interests include image/video analysis, lightweight neural networks, machine translation, and natural language processing.