

# Investigation of Hierarchical Spectral Vision Transformer Architecture for Classification of Hyperspectral Imagery

Wei Liu, *Member, IEEE*, Saurabh Prasad, *Senior Member, IEEE*, Melba Crawford, *Life Fellow, IEEE*

**Abstract**—In the past three years, there has been significant interest in hyperspectral imagery (HSI) classification using vision Transformers for analysis of remotely sensed data. Previous research predominantly focused on the empirical integration of convolutional neural networks (CNNs) to augment the network’s capability to extract local feature information. Yet, the theoretical justification for vision Transformers out-performing CNN architectures in HSI classification remains a question. To address this issue, a unified hierarchical spectral vision Transformer architecture, specifically tailored for HSI classification, is investigated. In this streamlined yet effective vision Transformer architecture, multiple mixer modules are strategically integrated separately. These include the CNN-mixer, which executes convolution operations; the spatial self-attention (SSA)-mixer and channel self-attention (CSA)-mixer, both of which are adaptations of classical self-attention blocks; and hybrid models such as the SSA+CNN-mixer and CSA+CNN-mixer, which merge convolution with self-attention operations. This integration facilitates the development of a broad spectrum of vision Transformer-based models tailored for HSI classification. In terms of the training process, a comprehensive analysis is performed, contrasting classical CNN models and vision Transformer-based counterparts, with particular attention to disturbance robustness and the distribution of the largest eigenvalue of the Hessian. From the evaluations conducted on various mixer models rooted in the unified architecture, it is concluded that the unique strength of vision Transformers can be attributed to their overarching architecture, rather than being exclusively reliant on individual multi-head self-attention (MSA) components. Extensive experiments demonstrate that the derived vision Transformer models, based on the unified architecture, surpass the classical methods when applied to multiple hyperspectral benchmark datasets.

**Index Terms**—Hyperspectral imagery (HSI) classification, Unified vision Transformer architecture, Mixer, Disturbance robustness, Hessian eigenvalue.

## I. INTRODUCTION

**H**YPERSPECTRAL imagery (HSI) enables detailed material identification by representing the reflectance spectra of objects via hundreds of contiguous bands. HSI data are used

in diverse applications including environmental monitoring, precision agriculture, geology, urban mapping, and defense [1]–[4]. Owing to the rapid advancements in deep learning [5]–[10], CNN architectures have emerged as the predominant standard for HSI classification in recent years. In [11], a deep feature fusion CNN is utilized to categorize each pixel of HSI data. To bolster extraction of spectrally-based features, [12], [13] introduce 3D-CNNs for HSI classification. Additionally, an attention mechanism can be integrated into the CNN framework to facilitate band selection for HSI data, as demonstrated in [14]. The efficacy of CNN-based HSI classification faces two significant limitations: 1) CNNs often struggle to adequately capture long-range dependencies; 2) The adoption of small input image window patches serves as a compromise between the high dimensionality of HSI data and its corresponding lower spatial resolution. This restricts the design possibilities of the network, impacting its depth and width. In the past three years, the appeal of using vision Transformers for HSI classification has grown [15]–[17]. This is attributed to the understanding that the spectral dimension of HSI parallels sequence data, irrespective of whether analysis is conducted at the pixel or patch level. In [18], group-wise spectral embedding is employed for HSI classification. Similarly, [19] introduces a group-aware hierarchical vision Transformer to strengthen HSI classification. Furthermore, the LESSFormer design, as presented in [20], aims to increase the capture of local information using adaptive spectral-spatial tokens. However, some have suggested that this configuration compromises the inductive bias inherent in CNNs [16], [21]. To address this, some have integrated vision Transformer and CNN modules, either in parallel or sequentially, to harness the advantages of both [22], [23]. Owing to the scalability of vision Transformers, they typically have a higher number of parameters compared to traditional CNNs. Incorporating an additional CNN branch on top of the multi-head self-attention (MSA) typically leads to a further increase in the model’s parameter size. At the same time, it has been noted that the overarching structure of vision Transformers, rather than just the MSA mixer, is pivotal to delivering top-tier performance [24]. This notion is further emphasized in studies where MSAs are substituted for multi-layer perceptrons (MLPs), as highlighted in [25]–[28]. While vision Transformer-based network architectures presently have a pronounced edge in HSI classification-based metrics relative to CNNs, the associated exploration predominantly remains empirical. Thus, this field continues to struggle with pivotal questions: (1) Does MSA

© 2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Wei Liu is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: liu3044@purdue.edu).

Saurabh Prasad is with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77204-4005 USA (e-mail: saurabh.prasad@ieee.org).

Melba Crawford is with the Lyles School of Civil Engineering and School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: mcrawford@purdue.edu).

serve as the *critical component* in vision Transformers that enhances HSI classification? (2) What fundamental differences exist relative to the training process for vision Transformer-based models and CNNs in analysis of hyperspectral datasets?

To address these questions, this paper proposes a unified hierarchical spectral vision Transformer architecture designed to integrate discriminative features for HSI classification. Notably, the simple yet effective unified architecture can be seamlessly integrated with any type of mixer block to construct a novel vision Transformer model. In this paper, various mixer modules, including the CNN-mixer, spatial self-attention (SSA)-mixer, channel self-attention (CSA)-mixer, SSA+CNN-mixer, and CSA+CNN-mixer, are independently integrated into the unified architecture, resulting in multiple vision Transformer models. A comprehensive analysis is conducted on these derived vision Transformer models and classical models, considering both the macroscopic aspect of the disturbance robustness and the microscopic aspect of the distribution of the maximum eigenvalue of the Hessian after the training process. In this paper, the term 'Hessian' specifically refers to the Hessian of the loss function relative to the parameters of the network. A key goal of this study is to explore the influence of different mixers on training vision Transformer-based models. A comprehensive comparison is conducted to explore and highlight fundamental differences between the vision Transformer and CNN models. To the best of our knowledge, this is the first paper to thoroughly investigate the key factors behind the superior performance of vision Transformers in HSI classification. Other contributions include: a) Conducting a rigorous evaluation of the training process for both CNNs and vision Transformers; b) Demonstrating that the unified architecture, rather than the MSA modules contribute to the superior performance observed with vision Transformers in HSI classification.

The remainder of this paper is structured as follows: Related work is summarized in Section II. The proposed method is detailed in Section III. The experimental setup and results are presented in Section IV. Section V includes the conclusion.

## II. RELATED WORK

**CNN-based HSI classification:** As highlighted in the first review of deep learning-based HSI classification [1], traditional machine learning techniques often fall short in addressing the unique challenges inherent in HSI classification, and particularly the significant spatial variability of spectral signatures. Over the past decade, the application of CNN models has advanced significantly, both in terms of enhanced performance and efficiency in HSI classification. Compared to traditional machine learning techniques, CNN-based methods excel in their ability to capture localized and discriminative spatial information, all while exhibiting resilience to translations and other variations. In [29], a streamlined, end-to-end CNN structure utilizing  $1 \times 1$  convolutional layers is adopted for HSI classification. [30] introduces a dual-channel CNN, crafted to jointly exploit spectral-spatial features from HSI. [31] develops a spectral-spatial latent reconstruction framework that concurrently reconstructs spectral and spatial features, while also

performing pixel-wise classification with high accuracy. [32] formulates a novel enhanced multiscale feature fusion network to extract sufficiently multiscale features from the parallel multipath architecture of three stages for HSI classification. Additionally, [33] implements a novel online spectral information compensation network for HSI classification. However, conventional 1D and 2D-CNNs often fall short in concurrently leveraging both spatial and spectral discriminative information. Recognizing this gap, researchers pioneered 3D-CNN architectures. For instance, [34] investigates an enhanced 3D deep CNN encompassing five layers. Furthermore, [35] proposes a distinctive recurrent 3D-CNN, designed to refine the 3D-CNN model by progressively diminishing the patch size. [36] formulates a streamlined 3D-CNN model with minimal parameters, resulting in a notable reduction in duration to convergence, while boosting accuracy. However, it should be noted that 3D-CNN models may encounter challenges such as overfitting and substantial computational demands. Aiming to alleviate such issues, [37] suggests a synergistic methodology that intertwines 2D-CNN and 3D-CNN. In this approach, the 2D-CNN is employed to extract spatial features, while the 3D-CNN, using small kernels, focuses on inter-band correlations. Complementing this, [38] proposes a 2D-3D CNN that incorporates a multi-branch feature fusion architecture. Some researchers specifically design CNN variants to efficiently extract feature representations [39], [40]. Notably, [40] proposes a novel geometry-aware convolutional foundation model that excels in learning unique geometry- and category-aware features and is informed by vehicle kinematics information to significantly enhance inclusive object detection and extend the perception range. Additionally, HSI shows category imbalance and complex spatial-spectral distributions, limiting adaptation performance. To address these issues, [41] proposes a class-aligned and class-balancing generative domain adaptation method for HSI classification. Similarly, [42] presents a novel framework with multigranularity generators and discriminators that uses adversarial and contrastive learning to continuously improve discriminator classification performance with diverse generated samples.

Recently, attention modules have gained widespread popularity in the field of deep learning, owing to their plug-and-play capability and their effectiveness in enhancing neural network performance [43]–[45]. In [43], a hierarchical network for efficient and accurate outdoor LiDAR point cloud registration is proposed by introducing an attention-based neighbor encoding module to gather neighborhood information. In pioneering work in instance-level HSI classification, [44] proposes a novel spectral-spatial feature pyramid network, which integrates multi-scale spectral and spatial information for instance segmentation in HSI. In [45], a ghost attention mechanism is proposed to significantly reduce both the parameters and FLOPs of the vision Transformer while achieving similar or better accuracy. The introduction of attention modules offers an alternative approach to boost HSI classification accuracy. These modules, by selectively emphasizing the most discriminative regions of an input small window patch or feature map, guide the network to focus on pivotal areas. Through the allocation of differential weights to various

pixels, the attention mechanism captures essential details, ignoring extraneous information. This refinement contributes to the network’s more accurate predictions. In [46], a pixel classification CNN is complemented with a superpixel-based graph attention network. The work in [47] melds a spectral-spatial attention network with ResNet for HSI classification. Recognizing the potential of harnessing long-range semantic information, [48] introduces an adaptive projection attention technique. Concurrently, several studies corroborate that the integration of attention modules significantly improves HSI classification accuracies, as evidenced by [49]–[51].

**Transformer-based HSI classification:** Over the last three years, the vision Transformer has excelled in the realm of HSI classification, showcasing its distinctive advantage in handling data sequences. The work in [18] introduces SpectralFormer, which integrates group-wise spectral embedding and cross-layer adaptive fusion modules. Specifically, the group-wise spectral embedding is adept at capturing feature embeddings from adjacent spectral bands. This combination facilitates the capture of detailed local spectral representations and promotes the transmission of memory-like components from superficial to deeper layers. Meanwhile, [4] presents a multiscale and cross-level attention learning network designed to holistically harness both global and local multiscale features of pixels for enhanced classification. In [19], a technique is introduced that employs grouped pixel embedding to better represent local representations. [52] proposes the spectral-spatial feature tokenization Transformer (SSFTT) approach, crafted to efficiently encapsulate HSI’s low, mid, and high-level semantic features. Aiming to optimize classification and reduce computational overhead, [53] devises a neighborhood-centric representation of multi-scale HSI features. In [54], a novel local vision Transformer, complemented by a spatial partition restore network, is introduced for HSI classification. [20] details LESSFormer, a design for HSI classification that converts HSI into adaptable spectral-spatial tokens. These tokens are then enriched to capture both local and extensive data nuances. Addressing the vision Transformer’s predominant focus on global data, [55] integrates it with a CNN, aiming to extract local features and thereby enhance classification. [23] develops a hybrid Transformer, merging multi-granularity tokens with spatial-spectral attention to model spatial-spectral information. Additionally, [56] implements a dual-branch architecture, combining the CNN and vision Transformer to seamlessly fuse spectral and spatial features. [57] proposes a novel hybrid deep learning network that systematically combines hierarchical CNNs and Transformers for feature extraction and fusion. This approach effectively learns spatial-spectral features in HSIs and elevation information in LiDAR, significantly enhancing the accuracy of the joint classification. Similarly, [58] introduces a novel layered architecture that integrates Transformer with CNN, utilizing a feature dimensionality reduction module and a Transformer-style CNN module to extract shallow features and enforce texture constraints, while employing the original Transformer encoder to extract deep features. Inspired by the observation that high-frequency information captures local details and low-frequency information provides global smooth variations, [59] develops a frequency domain feature

extraction vision Transformer network for HSI classification. The work in [60] puts forward three essential elements for efficient HSI classification through the integration of vision Transformer and CNN networks: extensive exploration of available features, effective reuse of representative features, and differentiated fusion of multi-domain features. Utilizing masked autoencoders’ self-supervised training paradigm [61], some researchers adopt a masked image modeling strategy for remote sensing image classification [62]–[64]. [62] develops a novel 3D generative pretrained Transformer architecture based on masked autoencoders for remote sensing applications. [63] introduces LFSMIM, a self-supervised network for HSI classification that employs low-pass filtering to construct the target domain within the masked image modeling framework. [64] proposes an unsupervised band selection framework that captures nonlinear relationships between bands and leverages spatial information in HSI. From these studies, it is evident that while vision Transformers have advanced HSI classification accuracy compared to CNN models, the majority of research has concentrated on empirical modifications to the self-attention modules, such as integrating CNN modules or altering feature embeddings. Specifically, the current analysis of model enhancements relies heavily on metric outcomes and prediction maps, without a thorough exploration of the variations throughout the training phase. This overlooks a critical element that contributes to the superior performance of vision Transformer architecture in HSI classification.

To bridge these gaps, a unified architecture for HSI classification built upon the vision Transformer is proposed in this paper. Based on the unified architecture, the attributes of the vision Transformer equipped with multiple mixers are investigated from a model training perspective, and the influence of different mixer modules on model performance is explored.

### III. PROPOSED METHOD

#### A. Overall architecture construction.

The HSI classification model based on the vision Transformer primarily consists of two main modules, as depicted in Fig. 1: 1) a unified architecture and 2) mixer block options. Subsequent sections provide detailed descriptions of these modules. Additionally, rather than relying exclusively on prediction accuracy and empirical analysis of various models, this paper evaluates disturbance robustness and the distribution of the largest eigenvalue of the Hessian. This evaluation provides insights into the model training process from both macro and micro perspectives.

The original HSI data are denoted as  $\mathbf{I} \in \mathbb{R}^{h \times w \times c}$ , where  $h$  and  $w$  represent the spatial height and width, respectively, and  $c$  signifies the number of spectral bands. The HSI data  $\mathbf{I}$  are divided into patches using a patch window size of  $s \times s$ , with each patch represented as  $\mathbf{P} \in \mathbb{R}^{s \times s \times c}$ . The label assigned to the center point of a patch determines its true label. The proposed vision Transformer model for HSI classification is designed to categorize the center point of each patch cube. The hierarchical vision Transformer architecture for HSI classification is depicted in Figure 1, referred

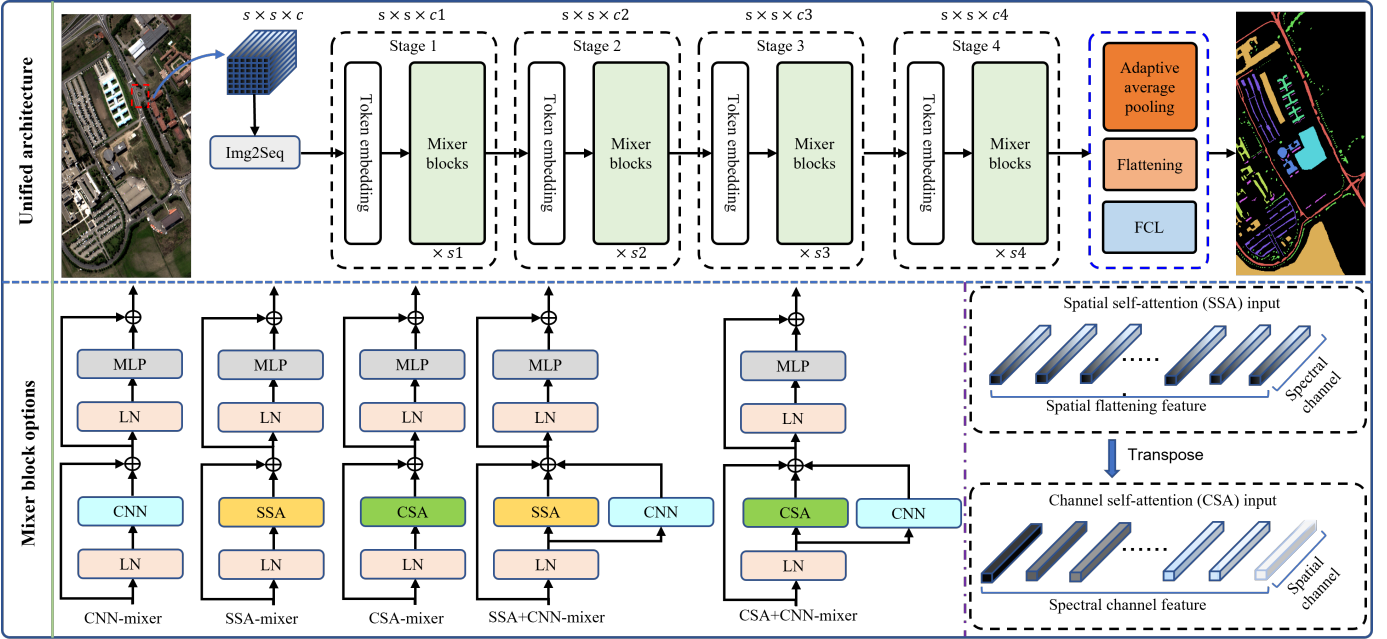


Fig. 1: Overall framework for HSI classification. The model consists of a unified architecture and mixer block options. The unified architecture is based on a novel hierarchical spectral vision Transformer, specifically tailored for HSI classification. Mixer block options include five common mixer blocks. When different mixers are individually chosen by the mixer blocks, it results in the creation of five unique Transformer models. The visualization in the bottom right corner demonstrates how the SSA-mixer and CSA-mixer can be easily converted on sequence inputs using the transpose operator. *Img2Seq*: transfer the image to sequence. LN: linear normalization. MLP: multilayer perceptron. CNN: convolutional neural network. SSA: spatial self-attention. CSA: channel self-attention. FCL: fully connected layer.

to as the *unified architecture*. The network comprises four stages: *Stage 1*, *Stage 2*, *Stage 3*, and *Stage 4*. In each stage, feature information is extracted through the iterative stacking of token embedding and the mixer module. The number of layers in each stage is represented by  $s_1$ ,  $s_2$ ,  $s_3$ , and  $s_4$  respectively. Given the unique characteristics of the input patch window, discriminative features are accumulated across various layers, emphasizing the information in the spectral and spatial dimensions. The respective channel numbers for each stage are denoted as  $c_1$ ,  $c_2$ ,  $c_3$ , and  $c_4$ . Similar with the Swin Transformer’s linear embedding technique [65], this paper utilizes  $nn.Conv2d(\cdot)$  for processing raw-valued features. To guarantee compatibility in data shape with the mixer blocks modules,  $Seq2Img(\cdot)$  and  $Img2Seq(\cdot)$  are judiciously placed before and after the  $nn.Conv2d(\cdot)$  operation, respectively. Importantly,  $Seq2Img(\cdot)$  serves as the inverse operation to  $Img2Seq(\cdot)$ . The token embedding strategy employed here is designed to project the spectral dimension to an arbitrary dimension, without impacting its spatial dimension. Employing the token embedding module, channel/pixel feature information is consolidated to produce a hierarchical representation that prioritizes the spectral/spatial dimension. Once processed by the token embedding module, the feature signals are relayed to the mixer blocks module for further discriminative feature extraction. After the feature extraction through four stages, the latent representation will further undergo processing by the adaptive average pooling, flattening, and a fully connected layer to output the predicted

values for the center position of each patch. Notably, the Swin Transformer achieves hierarchical representation by reducing resolution and simultaneously expanding the number of channels. In contrast, the HSI datasets from Houston 2013, Botswana, and Pavia University consist of high-dimensional input channels, with 144, 145, and 103 channels respectively, which always include redundant information. To effectively extract latent representations without substantially enlarging the parameter size of the vision Transformer architecture, the feature dimensions for the initial three layers are reduced, while the spatial dimension size remains constant. Therefore, the hierarchical paradigm for the proposed unified architecture is achieved by leveraging the spectral dimension.

To promote the model’s capacity to generalize across classification tasks, the label smoothing cross-entropy is selected as the loss. It is computed as follows:

$$\mathcal{L}(y, \hat{y}) = - \sum_{c=1}^C \left( (1 - \alpha) \cdot y_c + \frac{\alpha}{C} \right) \log(\hat{y}_c) \quad (1)$$

where the  $y$  is the ground truth label for the one-hot vector.  $\hat{y}$  is the predicted probability distribution.  $C$  is the class number.  $\alpha$  is set at 0.1 to control the extent of smoothing.  $y_c$  is the value of the  $c$ -th element in the true label vector, while  $\hat{y}_c$  is the value of the  $c$ -th element in the probability distribution vector predicted by the model.

### B. Mixer block options

HSI has tens to hundreds of spectral bands. In HSI, each pixel is characterized by a spectrum comprising reflectance values across these bands. This provides a rich representation of the scene or object, allowing for in-depth analysis and identification of materials or features through their unique spectral signatures. As a result, beyond the patch flattening, each pixel in HSI can also be interpreted as a sequence of data. This characteristic makes it possible to devise a variety of mixer blocks tailored to their specific attributes, including the CNN-mixer, SSA-mixer, CSA-mixer, SSA+CNN-mixer, and CSA+CNN-mixer.

**CNN-mixer:** Similar to the vision Transformer block from the Swin Transformer, the CNN-mixer module embeds an MLP, but opts for a CNN in place of the MSA mechanism. Notably, it sets itself apart by integrating an inductive bias, which fosters local feature connections. As highlighted in [66], the CNN-mixer module possesses the capability to model locality, which is governed by the kernel size, as well as scale-invariance. To avoid the influence of the attention mechanism on model performance, this paper employs a simple two-layer convolutional module. The module's representation is as follows:

$$CN(\mathbf{X}) = Conv_{3 \times 3}(SiLU(BN(Conv_{3 \times 3}(\mathbf{X})))) \quad (2)$$

where the  $Conv_{3 \times 3}$  operation amplifies the channel count fourfold using  $3 \times 3$  filters. This is followed by the application of the  $BN$  batch normalization. The module further integrates the  $SiLU$  activation function, which precedes the  $Conv_{3 \times 3}$  operation to refine the features. In this paper, unless stated otherwise,  $\mathbf{X}$  represents each patch input of  $\mathbf{P} \in \mathbb{R}^{s \times s \times c}$ .

The CNN-mixer module, incorporating the CNN block, is computed as follows:

$$\begin{aligned} \hat{\mathbf{X}} &= Seq2Img(\mathbf{X}) \\ \mathbf{Y} &= \hat{\mathbf{X}} + CN(\hat{\mathbf{X}}) \\ \hat{\mathbf{Y}} &= Img2Seq(\mathbf{Y}) \\ \mathbf{Z} &= \hat{\mathbf{Y}} + MLP(LN(\hat{\mathbf{Y}})) \end{aligned} \quad (3)$$

where  $CN(\cdot)$  is the CNN block.  $MLP(\cdot)$  is the multilayer perceptron operation. The function  $Seq2Img(\cdot)$  denotes a basic reshaping operation that transforms a one-dimensional sequence into a feature map.  $Img2Seq(\cdot)$  signifies the inverse operation of  $Seq2Img(\cdot)$ . Utilizing these reshaping techniques ensures the smooth integration of the CNN-mixer module within the vision Transformer framework.

**SSA-mixer:** To maximize the benefits of the numerous spectral bands in HSI, the SSA-mixer regards each pixel within a patch window as a sequence. Consequently, the length of the input sequence corresponds to the spectral bands' feature dimension, while the number of sequences is defined by the window size,  $s \times s$ . This sequential feature information is then input to the MSA module to further distill discriminative features.

$$\begin{aligned} \mathbf{Y} &= \mathbf{X} + MSA(LN(\mathbf{X})) \\ \mathbf{Z} &= \mathbf{Y} + MLP(LN(\mathbf{Y})) \end{aligned} \quad (4)$$

where  $LN(\cdot)$  is linear normalization.  $MSA(\cdot)$  is the computation of MSA. It can be described as follows:

$$Attention(Q, K, V) = SoftMax(QK^T / \sqrt{d})V \quad (5)$$

where  $Q, K, V$  are the vectors of *query*, *key* and *value*. These vectors are produced by projecting the input token embeddings through three distinct linear projection layers.  $d$  is the token embedding dimension.

**CSA-mixer:** In a similar manner, the sequential information derived from token embedding is converted into three-dimensional feature data using the  $Seq2Img(\cdot)$  function. These data are subsequently transposed and processed via the  $Img2Seq(\cdot)$  function, facilitating its transformation into sequences for each channel. This sequential feature data is then channeled through the MSA module to further refine and extract key features. The process can be detailed as follows:

$$\begin{aligned} \hat{\mathbf{X}} &= Img2Seq(Transpose(Seq2Img(\mathbf{X}))) \\ \mathbf{Y} &= \hat{\mathbf{X}} + MSA(LN(\hat{\mathbf{X}})) \\ \mathbf{Z} &= \mathbf{Y} + MLP(LN(\mathbf{Y})) \\ \hat{\mathbf{Y}} &= Img2Seq(Transpose(Seq2Img(\hat{\mathbf{Y}}))) \end{aligned} \quad (6)$$

where  $Transpose(\cdot)$  operation involves swapping the order of the three axes in the image latent features, facilitating their conversion into sequence data along different directions.

**SSA+CNN-mixer:** This architecture is designed by integrating a CNN module alongside the SSA-mixer. The goal is to explore potential improvements in the vision Transformer model's HSI classification performance by introducing the CNN module. The structure can be outlined as follows:

$$\begin{aligned} \mathbf{Y} &= \mathbf{X} + MSA(LN(\mathbf{X})) + Img2Seq(CN(Seq2Img(\mathbf{X}))) \\ \mathbf{Z} &= \mathbf{Y} + MLP(LN(\mathbf{Y})) \end{aligned} \quad (7)$$

**CSA+CNN-mixer:** This architecture is formulated by integrating a CNN module alongside the CSA-mixer. The intention is to explore potential improvements in the vision Transformer model's HSI classification efficacy with the inclusion of the CNN module. The configuration can be detailed as follows:

$$\begin{aligned} \hat{\mathbf{X}} &= Img2Seq(Transpose(Seq2Img(\mathbf{X}))) \\ \mathbf{Y} &= \hat{\mathbf{X}} + MSA(LN(\hat{\mathbf{X}})) \\ &\quad + Img2Seq(Transpose(CN(Seq2Img(\mathbf{X})))) \\ \hat{\mathbf{Y}} &= \mathbf{Y} + MLP(LN(\mathbf{Y})) \\ \mathbf{Z} &= Img2Seq(Transpose(Seq2Img(\hat{\mathbf{Y}}))) \end{aligned} \quad (8)$$

### C. Representation of the training process

In the evaluation of HSI classification models, especially when contrasting vision Transformer and CNN models, it is common for researchers to focus on performance metrics. They often employ reverse engineering and empirical analysis to emphasize the strength of specific methods. However, to our knowledge, few studies have seriously investigated the unique attributes of vision Transformer models from a model training perspective. This paper delves into the distinctions between vision Transformer and CNN models during the

HSI classification training phase, analyzing them through the 'best' pretrained weight disturbance robustness and the largest eigenvalue of the Hessian. Specifically, the 'best' pretrained weight refers to the training weight achieved after completing 300 epochs on the training dataset, while the maximum eigenvalue of the Hessian is calculated using the Hessian matrix. This matrix is constructed from the second-order partial derivatives of the neural network's loss function. It effectively describes the local curvature of a multi-variable function. In the realm of deep learning training, the 'loss landscape' refers to the visualization or portrayal of the loss function across the parameter space of a network [67], [68]. This landscape offers critical insights into the evolution of the loss function as network parameters change during training. It reveals useful insights about the model's behavior relative to the loss during its training phase. Notably, a smoother loss surface in proximity to the closest point tends to improve the model's generalization capabilities. However, given the huge number of parameters in deep learning models, capturing the intricacies of the loss landscape with a simple three-dimensional representation during the training process is a challenging task.

Building on the technique to produce three-dimensional loss landscapes, we can develop a new understanding of the model's training process. By introducing random disturbances along two unique vector directions with different magnitudes, based on the 'best' pretrained weights, the response of the loss value to these shifts can be assessed. This offers an avenue to analyze the robustness of various models to disturbances in post-training. To depict the three-dimensional loss surface subsequent to the disturbance, the model's loss value can be illustrated as follows:

$$V(w_x, w_y) = Loss(\Theta^* + w_x \nu_x + w_y \nu_y) \quad (9)$$

where  $\Theta^*$  is the 'best' pretrained weight after training, which is stored in the format of the dictionary.  $w_x$  and  $w_y$  are scale parameters ranging from -1 to 1 [68]. The vectors  $\nu_x$  and  $\nu_y$  are the basis vectors associated with the  $x$ -axis and  $y$ -axis, respectively. The procedures outlined in [67] are established through the following two steps. Initially, two new dictionaries are created based on the function  $randn(\cdot)$ , and these dictionaries are initialized with the same attributes as  $\Theta^*$ . Next, the weights and biases of each item in these dictionaries are normalized separately.

In a given deep learning model, the 'best' pretrained weights act as a baseline, with  $w_x$  and  $w_y$  serving as the horizontal axes. By varying the values of  $w_x$  and  $w_y$ , introducing different levels of perturbations to the weight, the corresponding loss values of the model on the training dataset can be determined. From the data derived from this set of three-dimensional points, the associated three-dimensional loss surface can be constructed. The framework of calculating the loss value with varying magnitude of disturbance on the 'best' pretrained weight is shown as Algorithm 1.

To further investigate local flatness and convergence properties, a qualitative analysis using the maximum eigenvalue

---

**Algorithm 1:** Framework of calculating loss value with varying magnitude disturbance of the best pretrained weight

---

**Input :** 'Best' pretrained weight  $\Theta^*$ .

**Output:** Loss value array  $V_{array}$ .

```

1 /* The loss value is calculated on the training dataset. */
2 V = []
3  $w_x \leftarrow np.linspace(-1, 1, n)$  // n represents the number of sampling points.
    $w_y \leftarrow np.linspace(-1, 1, n)$ 
4 Initialize two random normal vectors  $\nu_x^{ini}$  and  $\nu_y^{ini}$ 
   //  $\nu_x^{ini}$  and  $\nu_y^{ini}$  are same shape with  $\Theta^*$ .
5 Normalize  $\nu_x^{ini}$  and  $\nu_y^{ini}$ :  $\{\nu_x[m, n], \nu_y[m, n]\} \leftarrow$ 
    $\{\frac{\nu_x^{ini}[m, n]}{\|\nu_x^{ini}[m, n]\|} \|\Theta^*[m, n]\|, \frac{\nu_y^{ini}[m, n]}{\|\nu_y^{ini}[m, n]\|} \|\Theta^*[m, n]\|\}$ 
   //  $\Theta^*[m, n]$  denotes the m-th filter corresponding to the n-th layer.
    $V_0 = LabelSmoothingCrossEntropy(\Theta^*) + weight\_decay * L_2$  //  $L_2$  represents regularization.
6 for  $i \leftarrow 0$  to  $n - 1$  do
7   for  $j \leftarrow 0$  to  $n - 1$  do
8      $\Theta_{dis}^* = \Theta^* + w_x[i] * \nu_x + w_y[j] * \nu_y$ 
9      $V_{align} = LabelSmoothingCrossEntropy(\Theta_{dis}^*) + weight\_decay * L_2 - V_0$ 
10     $V.append(V_{align})$ 
11   end for
12 end for
13  $V_{array} = np.array(V).reshape(n, n)$ 
14 return  $V_{array}$ 

```

---

of the Hessian is necessary [69]. This explores the local characteristics of the loss surface, highlighting both flat and steep regions. These insights are pivotal in identifying areas that might either impede or aid convergence. The eigenvalue of the Hessian at a given point play a pivotal role in revealing the inherent characteristics of the model's loss function at that specific location. They are instrumental in discerning whether the point under consideration is a local minimum, a local maximum, or a saddle point. Furthermore, they offer valuable insights into the function's curvature in various directions. A negative eigenvalue in the Hessian is indicative of the curvature being concave along at least one direction. In practical terms, this means that a slight movement in the direction of the corresponding eigenvector would lead to an increase in the function's value, signifying that the point in question is not situated in a convex region of the function. Conversely, a scenario in which all the Hessian's eigenvalues at a specific point are positive denotes that the function exhibits local convexity at that juncture, categorizing the point as a local minimum [69]. Based on the 'best' pretrained weight after the training process, this paper conducts a thorough analysis of the distribution of the maximum eigenvalue of the Hessian. In the distribution curve representing the maximum eigenvalue,

TABLE I: Number of samples for each class of Houston 2013 dataset.

Class	Training	Validation	Testing	Total
1: Healthy grass	63	62	1126	1251
2: Stressed grass	62	63	1129	1254
3: Synthetic grass	35	35	627	697
4: Trees	62	62	1120	1244
5: Soil	62	62	1118	1242
6: Water	17	16	292	325
7: Residential	63	64	1141	1268
8: Commercial	62	62	1120	1244
9: Road	63	62	1127	1252
10: Highway	61	62	1104	1227
11: Railway	62	61	1112	1235
12: Parking lot 1	61	62	1110	1233
13: Parking lot 2	23	24	422	469
14: Tennis court	22	21	385	428
15: Running track	33	33	594	660

the ideal situation is for the horizontal coordinate of the curve’s peak to not only exceed zero but also remain in close proximity to it. This scenario is indicative of an augmented level of local smoothness in the vicinity of the ‘best’ pretrained point, a state achieved in post-training. This is indicative of the model’s generalization ability, showcasing its superior performance capabilities. In this paper, the *PyHessian* tool [70] is employed to compute the maximum eigenvalue of the Hessian. Notably, if model parameter gradients are absent, they are excluded from consideration. The derived maximum eigenvalue of the Hessian then becomes the foundation for applying the *KernelDensity* function from the *sklearn* library, paired with a Gaussian kernel, to shape a distribution curve.

To this end, an in-depth representation of the distinctions between CNN and vision Transformer models, as well as the impact of different mixer modules on the vision Transformer model in HSI classification, can be illustrated by combining performance metrics and training process analysis.

#### IV. EXPERIMENTAL SETUP AND RESULTS

##### A. Dataset description and implementation detail

The performance of the proposed vision Transformer models for HSI classification is evaluated using three commonly analyzed HSI datasets: Houston 2013, Botswana, and Pavia University [71], [72].

1) **Houston 2013**: Houston 2013 airborne hyperspectral data consist of 144 spectral bands. The dataset was collected over the University of Houston campus and the surrounding urban area. It comprises a total of  $349 \times 1905$  pixels, with each pixel of the orthorectified dataset having a spatial resolution of 2.5m. The dataset has 15 thematic classes. It was partitioned into three subsets for the analysis: a training set (5%), a validation set (5%), and a test set (90%). The class information and the number of training, validation, and testing samples for each class are presented in Table I. The false color image and ground reference map of the Houston 2013 dataset are shown in Fig. 2.

2) **Botswana**: The Botswana dataset, acquired by the Hyperion sensor on the EO-1 satellite over the Okavango Delta, consists of 242 spectral bands. After eliminating the noisy

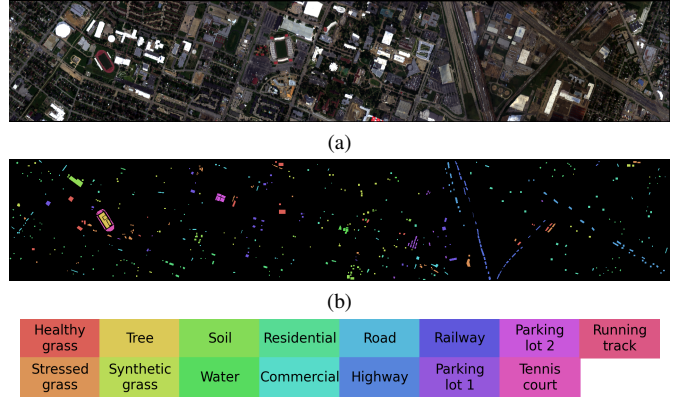


Fig. 2: Houston 2013 dataset. (a) False color image (band R: 60, G: 45, B: 20). (b) Ground truth map.

TABLE II: Number of samples for each class of Botswana dataset.

Class	Training	Validation	Testing	Total
1: Water	27	27	216	270
2: Hippo grass	10	10	81	101
3: Floodplain grasses 1	25	25	201	251
4: Floodplain grasses 2	22	21	172	215
5: Reeds	27	27	215	269
6: Riparian	27	27	215	269
7: Firescar	26	26	207	259
8: Island interior	21	20	162	203
9: Acacia woodlands	31	32	251	314
10: Acacia shrublands	24	25	199	248
11: Acacia grasslands	30	31	244	305
12: Short mopane	18	18	145	181
13: Mixed mopane	26	27	215	268
14: Exposed soils	10	9	76	95

and water absorption features bands, the dataset has 145 bands. Each pixel in the imagery has a spatial resolution of 30m. 14 classes were identified in the scene. The dataset was partitioned into three subsets for the analysis: a training set (10%), a validation set (10%), and a test set (80%). The class information and the number of training, validation, and testing samples for each class are detailed in Table II. The false color image and ground reference map of the Botswana dataset are shown in Fig. 3.

3) **Pavia University**: This scene was collected by the ROSIS sensor during a flight campaign over Pavia, northern Italy. There are 103 bands with 1.3m spatial resolution in this  $610 \times 340$  image, for which 9 classes have been identified. The dataset was partitioned into three subsets for the analysis: a training set (2%), a validation set (2%), and a test set (96%). The class information and the number of training, validation, and testing samples for each class are presented in Table III. The false color image and ground reference map of the Pavia University dataset are shown in Fig. 4.

Hyperspectral data are targeted for specific projects. Airborne data are expensive to acquire, and high dimensional. The data are standard common testbed data sets for algorithms, and we did not undertake any additional processing on the data. The benchmark data sets we analyze are widely used to compare classification methods. The Houston data covers the University of Houston and some of the city of Houston. The

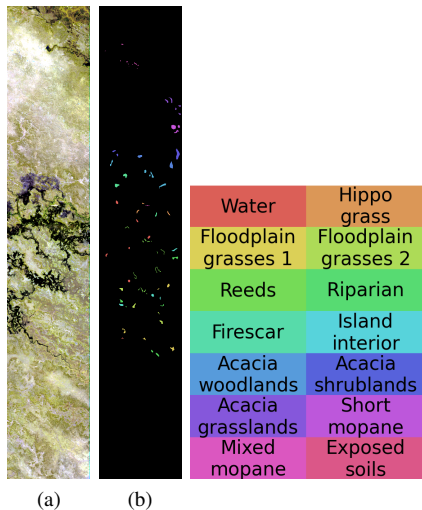


Fig. 3: Botswana dataset. (a) False color image (band R: 60, G:45, B: 15). (b) Ground truth map.

focus was to acquire information over a range of targets in an urban area with different spatial and spectral characteristics. Pavia University dataset is high resolution and covers a small area with less diversity in the classes and where the spatial representation of structures such as buildings was uniform and are easy to indicate in the ground reference. Botswana dataset was totally different both in terms of the sensor (30m data from space) and as a natural environment. The ground reference information was obtained using small homogeneous patches obtained on the ground and by interpretation of high resolution remotely sensed imagery. Thus, our analysis covers three totally different scenarios.

The proposed method, along with other established common methods, was implemented in Pytorch. The network was implemented on an NVIDIA Quadro RTX 6000 GPU with 22 GB RAM. The corresponding versions of Pytorch and CUDA were 1.10.1 and 10.2, respectively. The training process consisted of 300 epochs, with a batch size of 64. In this paper, the proposed algorithms utilized the Stochastic Gradient Descent (SGD) optimizer, configured with a learning rate of 0.001, momentum at 0.9, and a weight decay parameter of 0.0001. Parameters for the seven popular algorithms evaluated for comparison are consistent with those in the original papers. For the loss function, all algorithms employed label smoothing cross-entropy, ensuring a consistent methodological approach across the comparative analysis. To provide a quantitative comparison of the proposed method’s performance with other classical methods, the evaluation metrics employed were overall accuracy (OA), average accuracy (AA), and kappa coefficient ( $\kappa$ ). Each reported accuracy value represents an average obtained from training with five different random seeds.

### B. Comparison (baseline) methods

In the comparison study, several representative baseline methods are evaluated, including DFFN [11], CNN3D [12], M3D-DCNN [13], RSSAN [14], SpectralFormer [18], SSFTT

TABLE III: Number of samples for each class of Pavia University dataset.

Class	Training	Validation	Testing	Total
1: Asphalt	132	133	6366	6631
2: Meadows	373	373	17903	18649
3: Gravel	42	42	2015	2099
4: Trees	62	61	2941	3064
5: Painted metal sheets	27	27	1291	1345
6: Bare soil	100	101	4828	5029
7: Bitumen	27	26	1277	1330
8: Self-blocking bricks	73	74	3535	3682
9: Shadows	19	19	909	947

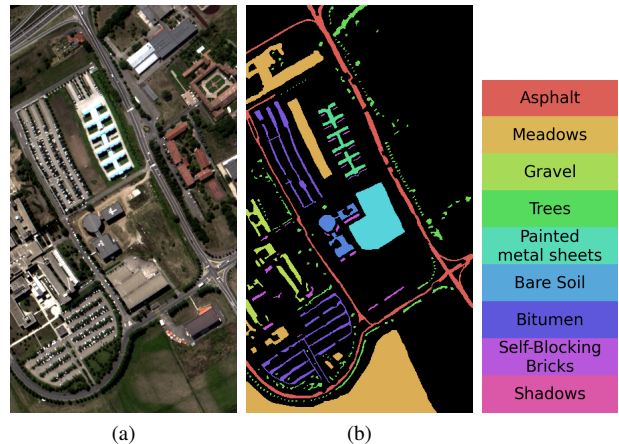


Fig. 4: Pavia University dataset. (a) False color image (band R: 40, G: 30, B: 20). (b) Ground truth map.

[52], GroupTransformer [19]. The DFFN utilizes residual learning to construct a deep 2D-CNN network. The CNN3D integrates traditional CNN architecture with 3D convolution operations. Similarly, the M3D-DCNN jointly learns both 2D multi-scale spatial features and 1D spectral features through a multiscale 3D deep convolutional neural network. The RSSAN combines a spectral-spatial residual attention network with long-short term memory (LSTM) to extract more discriminative spectral and spatial features. In SpectraFormer, which extends the vanilla vision Transformer architecture, a cross-layer skip connection is introduced to merge features across different layers. The SSFTT integrates a 3D convolution layer, a 2D convolution layer, and a vision Transformer module to construct a hybrid CNN-Transformer model for HSI classification. The GroupTransformer introduces a hierarchical Transformer alongside a 2D group convolution network for HSI classification. Thus, the aforementioned comparison of methods includes the common 2D and 3D CNNs, as well as the vision Transformer network. To ensure that each class of interest is adequately represented, stratified random sampling was employed for the dataset split. This technique consists of forcing the distribution of the target variables among the different splits to be the same. The strategy results in training on the same population in which it is being evaluated, achieving better predictions. Is implemented by the function of `sklearn.model_selection.train_test_split()`.



TABLE IV: Parameter size and FLOPs of different models.

Datasets	Complexity	CNN-based method				Transformer-based method							
		CNN3D	DFFN	M3D-DCNN	RSSAN	SpectralFormer	SSFTT	GroupTransformer	CNN-mixer	SSA-mixer	CSA-mixer	SSA+CNN-mixer	CSA+CNN-mixer
Houston 2013	Parameters (M)	0.52	0.51	0.68	0.09	0.24	0.67	0.97	1.05	0.47	1.02	1.23	2.54
	FLOPs (M)	6054.43	3979.24	3341.64	615.34	2429.33	2261.94	8628.74	8172.74	4760.42	5801.27	10636.20	16192.77
Botswana	Parameters (M)	0.11	0.51	0.17	0.09	0.23	0.68	0.98	1.10	0.48	0.36	1.28	0.91
	FLOPs (M)	1230.20	1611.91	996.46	250.53	2327.45	447.00	3241.43	3340.80	1694.47	1544.98	4189.52	3268.84
Pavia University	Parameters (M)	0.25	0.51	0.28	0.07	0.18	0.48	0.93	0.84	0.37	1.36	0.97	2.62
	FLOPs (M)	4343.47	3933.49	2282.17	523.19	1462.65	1615.03	8271.87	6508.44	3919.84	5324.28	8579.87	15089.90

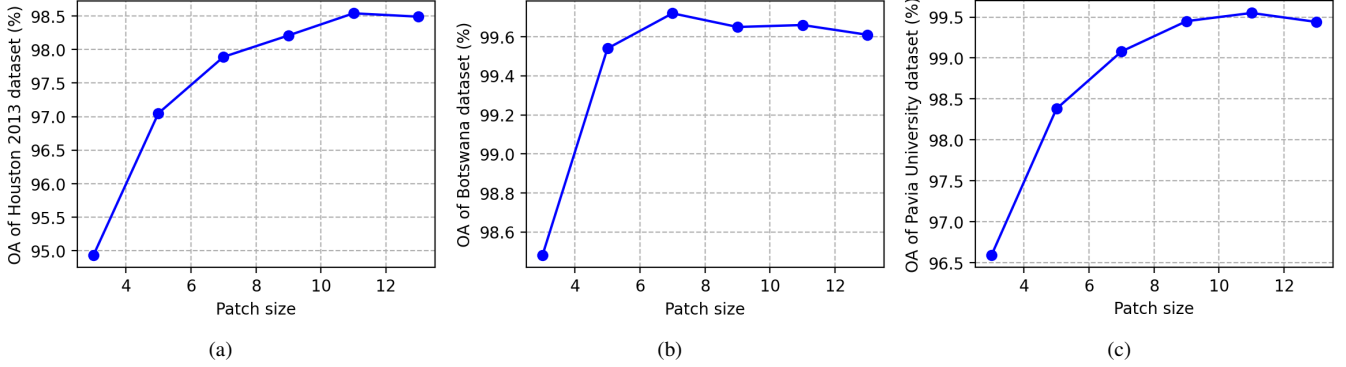


Fig. 5: Training patch size effect on the overall accuracy. (a) Houston 2013 dataset. (b) Botswana dataset. (c) Pavia University dataset.

### C. Model structure and complexity analysis

Given the dataset variations outlined in Section IV-A, tailoring model structural parameters considering the data characteristics is crucial in HSI classification. Based on the proposed unified architecture, the number of blocks per layer  $[s_1, s_2, s_3, s_4]$  for the Houston 2013, Botswana, and Pavia University datasets were set to  $[3, 2, 4, 2]$ ,  $[3, 3, 2, 2]$ , and  $[2, 2, 6, 2]$ , respectively. The dimension of the features per layer  $[c_1, c_2, c_3, c_4]$  were set to  $[96, 64, 32, 16]$ ,  $[96, 64, 32, 32]$ , and  $[96, 64, 32, 16]$ , respectively. In the joint tuning process of the number of blocks per layer and the dimension of features per layer, the initial setting for the number of blocks per layer was established as  $[2, 2, 6, 2]$ , with the Swin Transformer serving as a reference. Simultaneously, the dimensions of the features per layer were set to  $[96, 64, 32, 16]$ , ensuring a gradual decrease in feature dimension as layer depth increased. With these initial settings, the models were constructed to ensure the parameter size comparable to the baseline methods. The models were further optimized by adjusting the number of blocks per layer and the dimension of the last layer's features, with careful consideration to avoid significant changes in the model's overall parameter size. It should be noted that changes to the dimensions of the features in the first three layers were avoided, as they have a greater impact on the size of the parameter size. Furthermore, the selection of patch sizes for each dataset was determined by an analytical comparison study and evaluation of the spatial resolution of the data relative to that of the scale of spatial information in the image. As shown in Fig. 5, the optimal patch sizes were determined to be 11, 7, and 11 for the Houston 2013, Botswana, and Pavia University datasets, respectively.

Two metrics were introduced to represent the complexity of the model, the size of the parameter set and FLOPs. The

results are shown in Table IV. All measurement results use a patch cube as input, with a batch size that matches the training batch, which was set at 64. In the Houston 2013 and Pavia University datasets, the models built on SSA-mixer have the smallest number of parameters and FLOPs among the five proposed models. The number of parameters is even less than that of the SSFTT and GroupTransformer algorithms. In the Botswana dataset, the model built on the CSA-mixer has the smallest number of parameters and FLOPs. This is because the patch size in the Houston 2013 and Pavia University datasets was set to 11, which is significantly larger than the patch size of 7 set for the Botswana dataset. As the patch size increases, the number of features in the models built on CSA-mixer increases significantly, leading to a considerable increase in the number of model parameters and FLOPs. Furthermore, it is also observed that the models based on the CNN-mixer, despite their simple construction, do not have the smallest number of parameters and FLOPs among the five proposed models. In addition, after adding a CNN branch to the MSA, the two hybrid vision Transformer models (SSA+CNN-mixer and CSA+CNN-mixer) have more parameters and FLOPs than the classical Transformer models, resulting in greater computational costs.

### D. Experimental results

1) Analysis of classification performance: The mean and standard deviation of each criterion index across the three datasets are presented in Table V-VII. The highest value is highlighted in bold.

The first comparison experiment was conducted on the Houston 2013 dataset. Table V reports the prediction results on the test dataset in terms of OA, AA,  $\kappa$ , and the accuracy of each class. Among the four CNN-based models, DFFN

TABLE V: Classification results on the Houston 2013 dataset.

Class	CNN-based method				Transformer-based method							
	CNN3D	DFFN	M3D-DCNN	RSSAN	SpectralFormer	SSFTT	GroupTransformer	CNN-mixer	SSA-mixer	CSA-mixer	SSA+CNN-mixer	CSA+CNN-mixer
1	92.43±3.09	96.16±1.84	94.01±2.73	97.50±1.31	92.22±2.88	<b>98.72±2.04</b>	97.96±2.38	97.50±2.24	97.99±1.25	98.35±1.85	98.38±2.64	98.15±1.60
2	99.68±0.16	99.42±0.20	99.27±0.84	98.42±0.70	96.35±2.09	<b>99.61±0.20</b>	98.87±0.85	99.03±0.62	99.19±0.45	98.57±0.62	99.47±0.21	99.33±0.52
3	<b>100.00±0.00</b>	99.81±0.19	<b>100.00±0.00</b>	99.87±0.16	99.04±0.62	99.71±0.43	99.84±0.17	99.94±0.13	99.78±0.13	99.65±0.42	99.94±0.08	99.90±0.13
4	99.57±0.19	99.45±0.49	99.18±0.54	98.79±1.09	96.12±2.02	99.27±0.78	99.11±0.67	99.34±0.58	99.34±0.35	<b>99.66±0.47</b>	99.21±0.56	99.14±0.57
5	99.18±0.44	99.86±0.20	99.19±0.37	98.30±1.27	98.07±0.66	<b>100.00±0.00</b>	<b>100.00±0.00</b>	99.98±0.04	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	99.95±0.11
6	87.88±4.52	93.49±6.08	86.78±5.35	82.05±2.57	87.40±5.19	95.89±5.31	96.03±5.10	95.48±5.00	94.25±5.28	94.45±5.07	95.41±4.12	<b>96.44±5.11</b>
7	95.34±0.78	98.23±0.82	96.69±1.14	96.39±0.57	91.15±3.14	98.07±1.11	<b>99.54±0.33</b>	99.28±0.53	99.12±0.66	99.16±0.71	99.04±0.76	99.51±0.54
8	84.38±3.36	92.04±2.68	88.88±3.05	89.95±1.98	87.68±3.47	95.09±0.91	93.91±1.44	94.39±0.64	94.07±1.28	94.91±1.24	<b>95.43±0.35</b>	94.52±0.75
9	91.54±1.95	95.21±0.68	92.83±1.34	93.58±0.83	86.14±0.98	97.04±0.88	97.52±0.91	<b>98.01±1.03</b>	97.59±0.87	96.95±1.06	97.37±1.16	97.64±1.06
10	93.04±1.11	98.79±0.77	96.12±1.22	96.78±2.11	91.52±2.54	99.26±0.41	99.33±0.82	99.78±0.19	99.80±0.31	99.80±0.20	<b>99.87±0.25</b>	99.49±0.85
11	91.76±1.74	97.45±1.02	93.47±2.23	96.46±0.85	89.46±2.44	99.30±0.96	98.87±1.17	99.59±0.54	99.51±0.97	99.78±0.29	<b>99.86±0.25</b>	99.51±0.75
12	92.29±3.47	97.26±1.53	95.17±0.73	95.24±1.31	93.21±2.31	97.21±1.78	97.15±1.39	97.37±2.11	97.75±0.97	97.68±1.37	97.03±2.21	<b>98.25±1.02</b>
13	91.04±1.88	97.39±1.42	90.81±2.36	89.05±3.08	69.19±5.99	97.87±1.49	<b>98.77±1.60</b>	98.34±1.68	97.30±2.56	96.64±2.22	97.73±2.27	98.67±1.74
14	99.32±0.45	99.74±0.52	99.53±0.42	98.81±0.71	95.69±1.44	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>
15	99.93±0.13	<b>100.00±0.00</b>	99.87±0.27	98.48±1.13	96.73±1.07	99.97±0.07	99.56±0.88	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>
OA	94.41±0.65	97.59±0.39	95.67±0.40	95.97±0.23	91.99±1.07	98.47±0.41	98.38±0.29	98.54±0.38	98.48±0.26	98.50±0.25	98.64±0.39	<b>98.68±0.21</b>
AA	94.49±0.70	97.62±0.32	95.45±0.66	95.31±0.23	91.33±1.28	98.47±0.48	98.43±0.41	98.53±0.48	98.38±0.28	98.37±0.26	98.58±0.39	<b>98.70±0.30</b>
Kappa	93.95±0.70	97.40±0.42	95.32±0.44	95.64±0.25	91.34±1.16	98.34±0.44	98.25±0.31	98.43±0.41	98.36±0.28	98.38±0.27	98.53±0.42	<b>98.57±0.23</b>

TABLE VI: Classification results on the Botswana dataset.

Class	CNN-based method				Transformer-based method							
	CNN3D	DFFN	M3D-DCNN	RSSAN	SpectralFormer	SSFTT	GroupTransformer	CNN-mixer	SSA-mixer	CSA-mixer	SSA+CNN-mixer	CSA+CNN-mixer
1	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	99.91±0.19	99.91±0.19	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>
2	96.54±2.39	<b>100.00±0.00</b>	<b>100.00±0.00</b>	99.75±0.49	98.52±1.44	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>
3	98.11±1.49	99.80±0.77	99.50±0.63	<b>100.00±0.00</b>	96.12±2.56	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>
4	98.60±1.31	99.88±0.23	99.30±0.93	97.79±2.13	97.91±1.86	99.88±0.23	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	99.88±0.23	<b>100.00±0.00</b>	99.88±0.23
5	89.40±3.85	98.05±2.27	96.28±2.56	95.35±3.59	78.79±3.12	95.53±4.27	99.35±0.47	98.79±1.43	98.70±1.62	99.53±0.93	97.86±2.62	<b>99.63±0.35</b>
6	78.23±7.61	98.33±3.77	92.84±3.79	98.23±0.74	93.95±1.72	98.23±1.12	97.40±2.57	98.42±1.60	98.42±1.37	98.42±1.46	97.12±3.16	<b>98.70±1.15</b>
7	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	99.32±0.66	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	99.71±0.58	<b>100.00±0.00</b>	<b>100.00±0.00</b>
8	95.06±5.05	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	96.30±4.40	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>
9	93.94±4.57	99.76±0.20	97.61±3.02	98.73±1.81	92.59±4.54	98.49±2.05	99.84±0.32	<b>100.00±0.00</b>	<b>100.00±0.00</b>	99.92±0.16	<b>100.00±0.00</b>	99.92±0.16
10	97.09±0.80	<b>100.00±0.00</b>	99.60±0.59	<b>100.00±0.00</b>	97.99±0.95	<b>100.00±0.00</b>	99.70±0.40	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	99.90±0.20
11	96.39±0.88	97.95±2.07	99.84±0.20	99.75±0.33	98.69±0.40	99.51±0.98	99.43±0.71	99.92±0.16	99.67±0.66	<b>100.00±0.00</b>	99.43±0.80	99.75±0.49
12	99.31±0.44	99.03±0.83	<b>100.00±0.00</b>	98.07±1.60	95.45±2.33	<b>100.00±0.00</b>	99.86±0.28	<b>100.00±0.00</b>	99.59±0.83	<b>100.00±0.00</b>	99.86±0.28	99.72±0.55
13	99.81±0.23	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	99.53±0.42	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>	<b>100.00±0.00</b>
14	91.05±2.55	96.32±7.37	97.63±4.74	97.11±3.94	84.74±5.17	98.95±2.11	96.84±5.03	98.42±2.11	96.84±3.87	97.37±3.63	<b>99.47±0.64</b>	97.63±3.16
OA	95.21±0.78	99.28±0.40	98.67±0.30	98.98±0.38	95.24±0.92	99.25±0.38	99.53±0.24	99.72±0.19	99.59±0.24	<b>99.74±0.20</b>	99.51±0.45	99.73±0.08
AA	95.25±0.81	99.20±0.68	98.76±0.32	98.91±0.52	94.99±1.05	99.32±0.40	99.45±0.40	<b>99.68±0.17</b>	99.49±0.33	99.65±0.29	99.55±0.39	99.65±0.19
Kappa	94.81±0.85	99.22±0.44	98.56±0.33	98.89±0.41	94.85±0.99	99.18±0.42	99.49±0.26	99.69±0.17	99.56±0.26	<b>99.72±0.22</b>	99.47±0.48	99.71±0.08

TABLE VII: Classification results on the Pavia University dataset.

Class	CNN-based method				Transformer-based method							
	CNN3D	DFFN	M3D-DCNN	RSSAN	SpectralFormer	SSFTT	GroupTransformer	CNN-mixer	SSA-mixer	CSA-mixer	SSA+CNN-mixer	CSA+CNN-mixer
1	96.79±0.50	96.44±2.98	97.89±0.48	99.46±0.32	93.57±1.77	99.38±0.27	99.85±0.10	99.73±0.25	99.80±0.18	99.85±0.20	99.59±0.39	<b>99.88±0.15</b>
2	99.46±0.30	95.66±4.63	99.89±0.07	99.82±0.17	99.67±0.20	99.89±0.07	99.89±0.08	<b>99.97±0.02</b>	99.95±0.02	99.96±0.01	99.91±0.08	99.92±0.06
3	76.36±3.86	99.25±0.62	89.71±3.64	96.38±1.40	85.86±2.63	98.00±1.22	97.70±1.70	98.26±0.86	99.31±0.37	98.02±1.60	<b>99.47±0.81</b>	98.97±0.54
4	97.86±0.34	97.32±3.75	<b>98.48±0.66</b>	96.38±0.99	93.67±1.68	98.40±0.76	96.89±0.42	98.00±0.16	97.27±0.49	97.38±0.49	97.97±0.39	98.78±0.26
5	99.95±0.09	99.23±0.58	99.78±0.20	99.89±0.14	99.91±0.19	99.85±0.18	99.88±0.18	99.98±0.03	99.97±0.04	<b>100.00±0.00</b>	99.80±0.40	99.98±0.03
6	94.75±0.96	81.09±8.77	99.18±0.39	99.67±0.28	96.51±3.66	99.95±0.05	99.96±0.03	99.81±0.31	99.98±0.03	<b>100.00±0.00</b>	99.90±0.11	99.76±0.36
7	83.65±4.21	93.39±3.17	93.91±2.71	96.13±1.92	78.20±4.94	99.15±0.73	99.58±0.21	99.51±0.74	99.87±0.18	99.73±0.33	99.81±0.34	<b>99.94±0.06</b>
8	95.09±1.26	81.41±2.70	96.17±1.20	96.08±1.03	88.48±3.13	98.55±0.46	97.13±1.81	99.10±0.70	<b>98.28±1.51</b>	97.92±1.46	98.14±1.96	98.34±1.26
9	99.71±0.24	87.82±2.71	99.85±0.09	99.12±0.89	95.93±1.96	97.40±1.34	98.37±1.31	97.62±2.50	97.27±2.67	97.73±1.25	<b>98.13±1.62</b>	99.08±0.49
OA	96.40±0.30	93.53±0.35	98.38±0.23	98.88±0.19	95.54±0.53	99.43±0.17	99.29±0.13	<b>99.55±0.15</b>	99.50±0.15	99.44±0.11	99.50±0.14	99.54±0.10
AA	93.74±0.78	93.58±0.62	97.21±0.56	98.10±0.29	92.42±0.62	98.95±0.30	98.80±0.21	99.11±0.47	99.08±0.35	98.96±0.19	99.19±0.32	<b>99.29±0.11</b>
Kappa	95.21±0.41	93.01±0.38	97.86±0.31	98.52±0.25	94.07±0.72	99.24±0.22	99.06±0.17	<b>99.40±0.20</b>	99.34±0.19	99.25±0.14	99.34±0.18	<b>99.40±0.14</b>

stands out with an OA of 97.59%, marking a 3.18% increase in performance compared to the CNN3D algorithm, which has the lowest OA in this group. As for the three classical vision Transformer algorithms, SpectralFormer, SSFTT, and GroupTransformer have OA values of 91.99%, 98.47%, and 98.38%, respectively. In comparison to the SpectralFormer algorithm based on the vanilla vision Transformer, the latter two show significant improvements in classification accuracy. Utilizing the unified hierarchical Transformer architecture proposed in this paper, five mixer-based HSI classification models demonstrated exceptional OA, ranging from 98.48% to 98.68%. On this dataset, the model built upon the CSA+CNN-mixer outperforms other classical CNN and vision Transformer models, with accuracy improvements of 0.21% and 6.69%, respectively. The corresponding prediction map is shown in Fig. 6. In the prediction maps for (a), (c), (d), and

(e), there is a comparatively higher occurrence of spurious anomalies. The prediction map outcomes for (h) through (l), which represent the five models developed using a unified architecture, show a remarkable similarity.

The second comparative experiment was carried out using the Botswana dataset. The prediction results are listed in Table VI. On this dataset, the CNN3D algorithm has a significant decrease in accuracy, differing by at least 3.46% compared to the other three common CNN algorithms. Among these, DFFN is the top-performing CNN algorithm, attaining a maximum accuracy of 99.28%. Similar to the results on the Houston 2013 dataset, among the three typical vision Transformer algorithms, SpectralFormer achieved a lower accuracy at 95.24%, in contrast to GroupTransformer achieved an OA value of 99.53%. Among the five models employing different mixers, the one utilizing the CSA-mixer outperforms

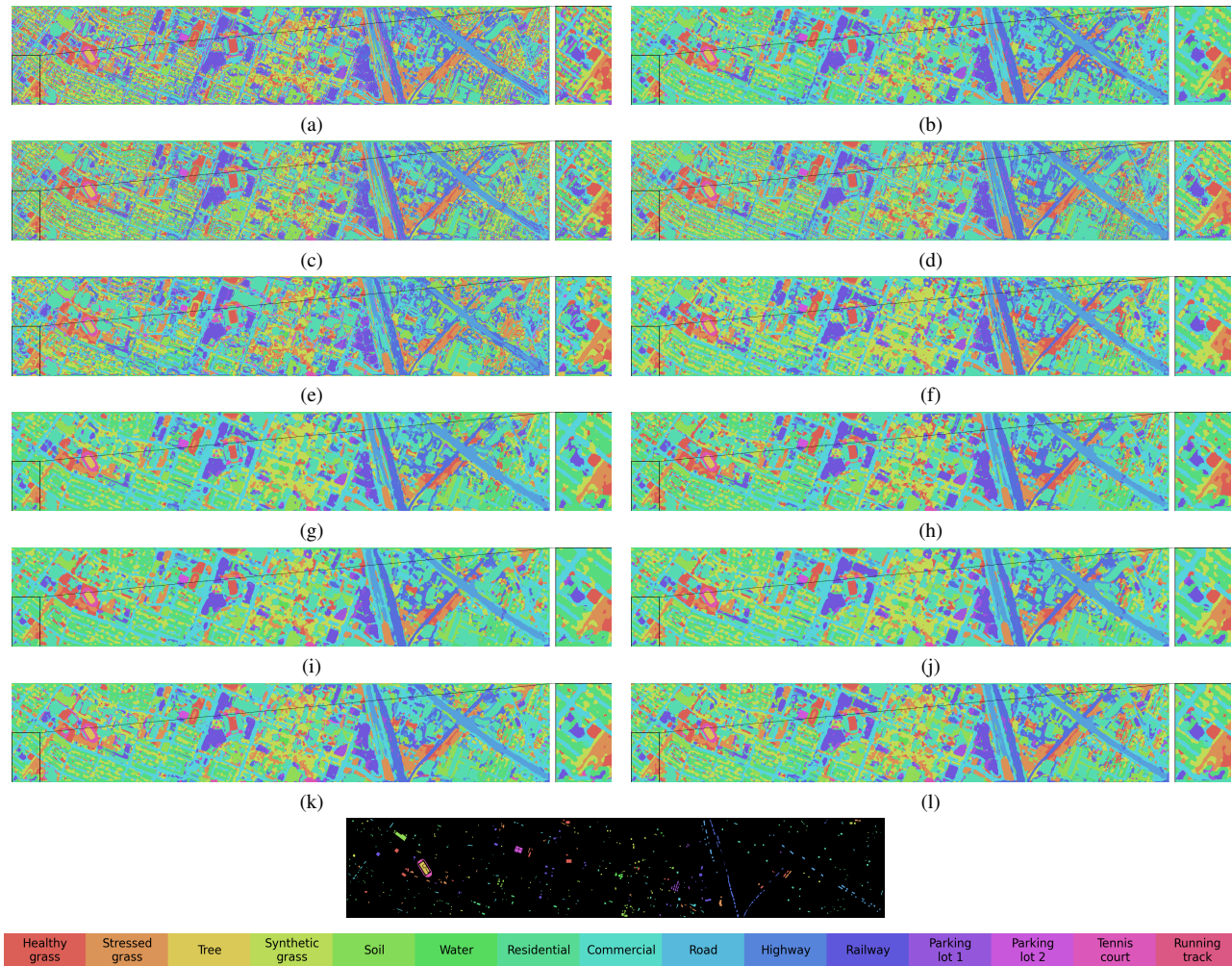


Fig. 6: Prediction map on Houston 2013 dataset. (a) CNN3D, (b) DFFN. (c) M3D-DCNN. (d) RSSAN. (e) SpectralFormer. (f) SSFTT. (g) GroupTransformer. (h) Proposed CNN-mixer. (i) Proposed SSA-mixer. (j) Proposed CSA-mixer. (k) Proposed SSA+CNN-mixer. (l) Proposed CSA+CNN-mixer. (m) Ground truth.

the rest, achieving an OA of 99.74%. The SSA+CNN-mixer-based method performs the worst. However, its OA value is only 0.02% lower than that of the GroupTransformer. The prediction map is depicted in Fig. 7. The maps for (a), (c), (d), (e), and (f) demonstrate suboptimal performance. Upon exploring a magnified view, it is evident that (b) shows a higher number of errors in predicting 'Riparian' compared to the proposed models based on the unified vision Transformer architecture. This phenomenon matches the metrics provided in Table VI.

The third comparative experiment utilized the Pavia dataset, and Table VII displays the prediction results obtained by different methods. Among the four prevalent CNN methods, DFFN obtains an OA of 93.53%, which is comparatively lower, while RSSAN distinguishes itself with a higher OA of 98.88%. Within the three popular vision Transformer-based methods, SpectralFormer lags slightly behind the other two algorithms. Notably, in contrast to the previous two datasets, the SSFTT algorithm outperforms the other two popular vision Transformer methods, reaching an accuracy

of 99.43% and surpassing the GroupTransformer by 0.14%. Meanwhile, the five HSI classification algorithms proposed in this paper consistently demonstrate the highest accuracy, ranging from 99.44% to 99.55%, with the CNN-mixer-based algorithm arriving at 99.55%. Overall, the differences in OA among these five algorithms remain relatively small. The prediction map is depicted in Fig. 8. It illustrates that the five models built on the unified Transformer architecture demonstrate superior classification accuracy for the categories 'Gravel' and 'Bitumen'.

In summary, the five models utilizing the unified hierarchical vision Transformer architecture demonstrate the highest OA across three testbed datasets. Differences in accuracies achieved by the five algorithms are generally insignificant. This also implies that the performance of the proposed HSI classification models predominantly hinges on the *unified architecture*, rather than the specific mixer modules that attracted attention in previous research.

2) Analysis of the training process: Based on traditional evaluation methods such as OA metric and prediction maps,

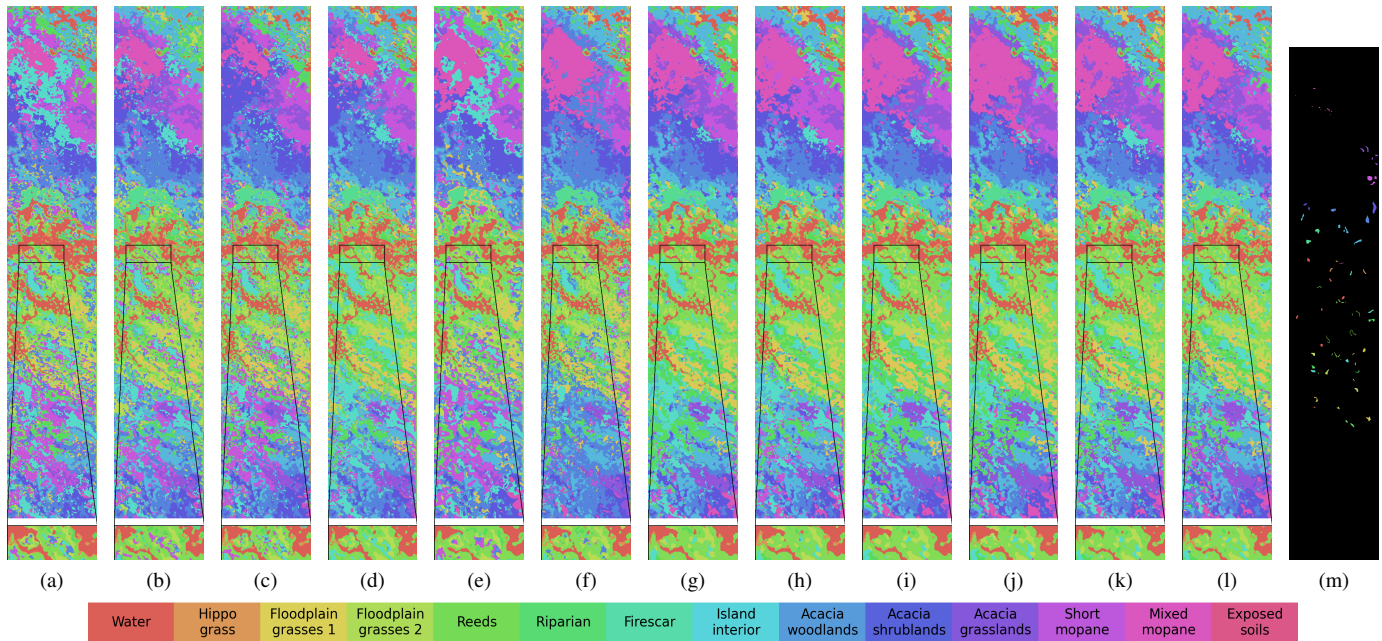


Fig. 7: Prediction map on Botswana dataset. (a) CNN3D. (b) DFFN. (c) M3D-DCNN. (d) RSSAN. (e) SpectralFormer. (f) SSFTT. (g) GroupTransformer. (h) Proposed CNN-mixer. (i) Proposed SSA-mixer. (j) Proposed CSA-mixer. (k) Proposed SSA+CNN-mixer. (l) Proposed CSA+CNN-mixer. (m) Ground truth.

the following two challenges remain difficult to address comprehensively: (1) Is MSA indeed the pivotal module in vision Transformers for enhancing HSI classification? (2) What are the critical differences in model characteristics between vision Transformer-based models and CNN models regarding training on hyperspectral datasets? To address these questions, as noted previously, the macro and micro-level characteristics of the models during the training process were investigated.

Fig. 9, 10, and 11 contain plots of three-dimensional surfaces representing the loss values for different models, following the introduction of disturbance with varying amplitudes into the 'best' pretrained weights on the three datasets. (a) - (d) depict loss surface contours based on four typical CNN algorithms. From a macroscopic perspective, it is evident that introducing two directional disturbances to the 'best' pretrained weights across the three datasets results in a noticeable spike in loss values calculated with these perturbed weights, thereby accentuating the contour of the surface. It is important to note that the HSI classification model based on DFFN, when exposed to substantial disturbances, produces excessively high loss values that surpass our predefined threshold of 100. This leads to scenarios, as illustrated in (b), where loss values become undefined as the values of the  $x$  and  $y$  axes approach 1. This suggests that the stability of the DFFN model is the least resilient to disturbances in the 'best' pretrained weight. (e) - (g) depict three-dimensional loss surface contours corresponding to three representative vision Transformer models, while (i) - (l) illustrate three-dimensional loss surface contours for the five algorithms proposed in this paper. The figures clearly show that the three-dimensional surface contours based on vision Transformer algorithms are notably smoother when

compared to those based on CNN algorithms. However, in the case of (e), it can be observed that after introducing disturbances of varying magnitudes, the loss value along the  $z$ -axis for SpectralFormer changes at a notably lower rate. This suggests that the model exhibits a weak response to disturbances when starting from the 'best' pretrained weight. Interestingly, this phenomenon may not favor the model's ability to converge towards an optimal solution during training. This consequence may be attributed to the newly introduced structures such as group-wise spectral embedding and cross-layer adaptive fusion. Furthermore, by investigating (g) - (f), it is apparent that as the magnitude of disturbances increases, the model's loss value initially experiences a slight increase before eventually reaching saturation. An *ideal model* should demonstrate the characteristic of a moderate increase in loss value when disturbances are applied to the optimal training weights. It's noteworthy that (h) corresponds to the proposed algorithm based on the CNN-mixer. This model lacks the MSA module, yet the shape of its loss surface contours across the three datasets differs significantly from those corresponding to CNN models. In contrast, the loss surface contours for vision Transformer models constructed with the five different mixers exhibit remarkably similar shapes. This further indicates that the distinctive characteristics of vision Transformer models in HSI classification primarily are derived from their unified hierarchical Transformer architecture rather than the specific mixer modules. It also suggests that the MSA module is not the fundamental reason for the differences between CNN and vision Transformer models.

Through the aforementioned loss surface contours, the overall global features of different models after disturbances can be

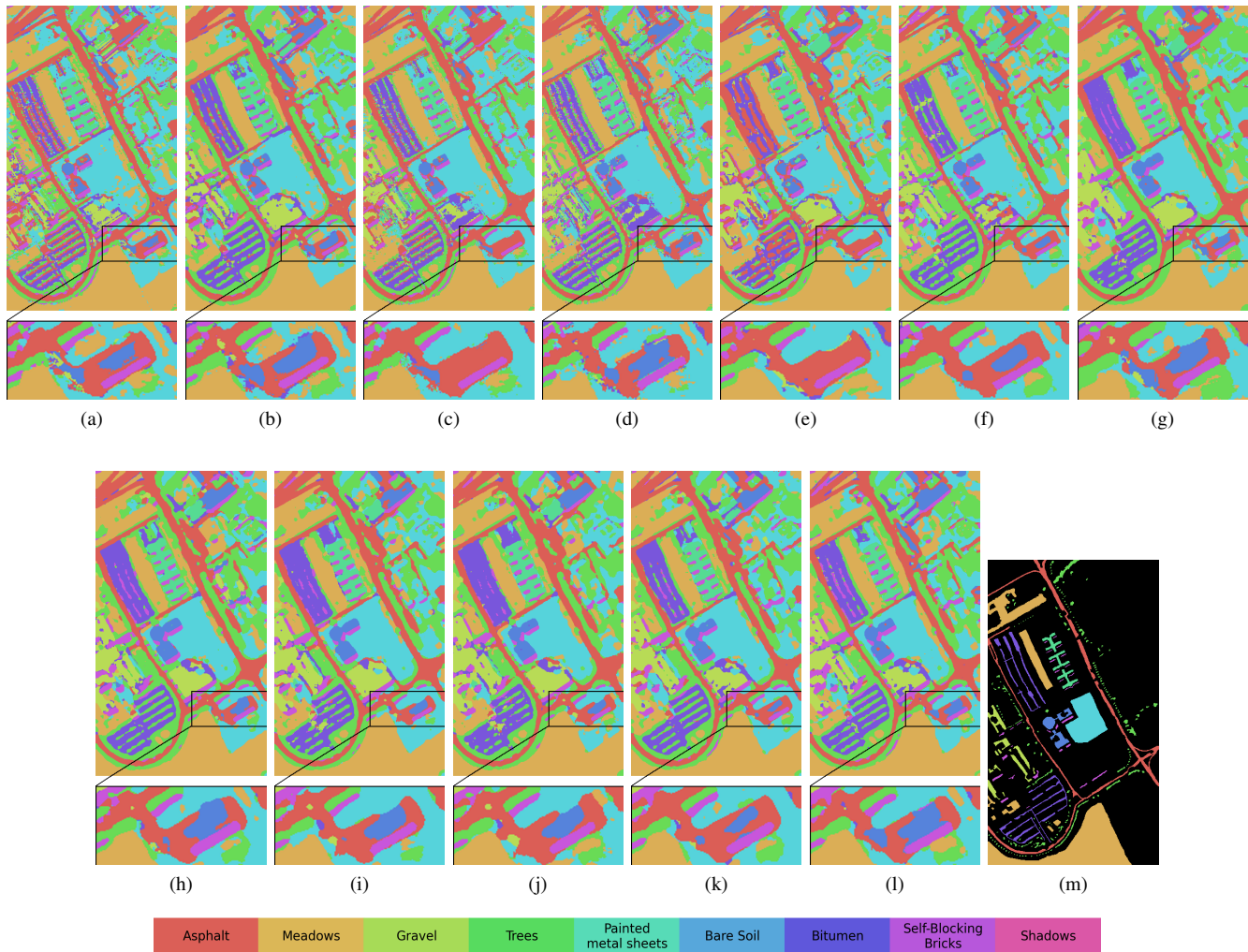


Fig. 8: Prediction map on Pavia University dataset. (a) CNN3D. (b) DFFN. (c) M3D-DCNN. (d) RSSAN. (e) SpectralFormer. (f) SSFTT. (g) GroupTransformer. (h) Proposed CNN-mixer. (i) Proposed SSA-mixer. (j) Proposed CSA-mixer. (k) Proposed SSA+CNN-mixer. (l) Proposed CSA+CNN-mixer. (m) Ground truth.

intuitively visualized. However, these contours do not offer a microscopic analysis of the model’s local characteristics in the neighborhood of the ‘best’ pretrained weight. To address this gap, this paper introduces the distribution of the maximum eigenvalue of the Hessian, aiming to quantitatively analyze the model’s gradient properties from a local perspective in the vicinity of the ‘best’ pretrained weight. Fig. 12, 13, and 14 depict the distribution of the maximum eigenvalue of the Hessian for different models across the three datasets. Among the four typical CNN models, CNN3D and M3D-DCNN exhibit similar curves for the maximum eigenvalue of the Hessian. The magnitude of the maximum eigenvalue of the Hessian approaches zero, and negative values are virtually absent. In contrast, for DFFN, the magnitude of the maximum eigenvalue of the Hessian is relatively larger, especially on the Houston 2013 and Botswana datasets. Consequently, in general, CNN3D and M3D-DCNN, exhibit smoother local behavior around the ‘best’ pretrained weights among the four classical CNN-based models. Interestingly, on the Pavia

University dataset, the magnitude of the maximum eigenvalue of the Hessian for all four CNN algorithms is similar, and none of them exhibit negative values. This implies that all four CNN algorithms demonstrate remarkably smooth local behavior around the optimal points as they approach the end of model training. (e), (f), and (g) represent three classical vision Transformer models. Among these models, it can be observed that SpectralFormer’s distribution of the maximum eigenvalue of the Hessian approaches the  $x = 0$  axis. This indicates that the model exhibits a highly flat behavior in the vicinity of optimal weights, resulting in minimal responsiveness to local perturbations. Conversely, for SSFTT, the maximum eigenvalue of the Hessian is partly situated on the  $x < 0$  side across all three datasets. This implies non-convexity in the model’s local behavior near this point, potentially hindering its ability to search for optimal weights during training. Within the GroupTransformer algorithm, the magnitude of the maximum eigenvalue of the Hessian is notably higher on the Botswana dataset compared to Houston 2013 and

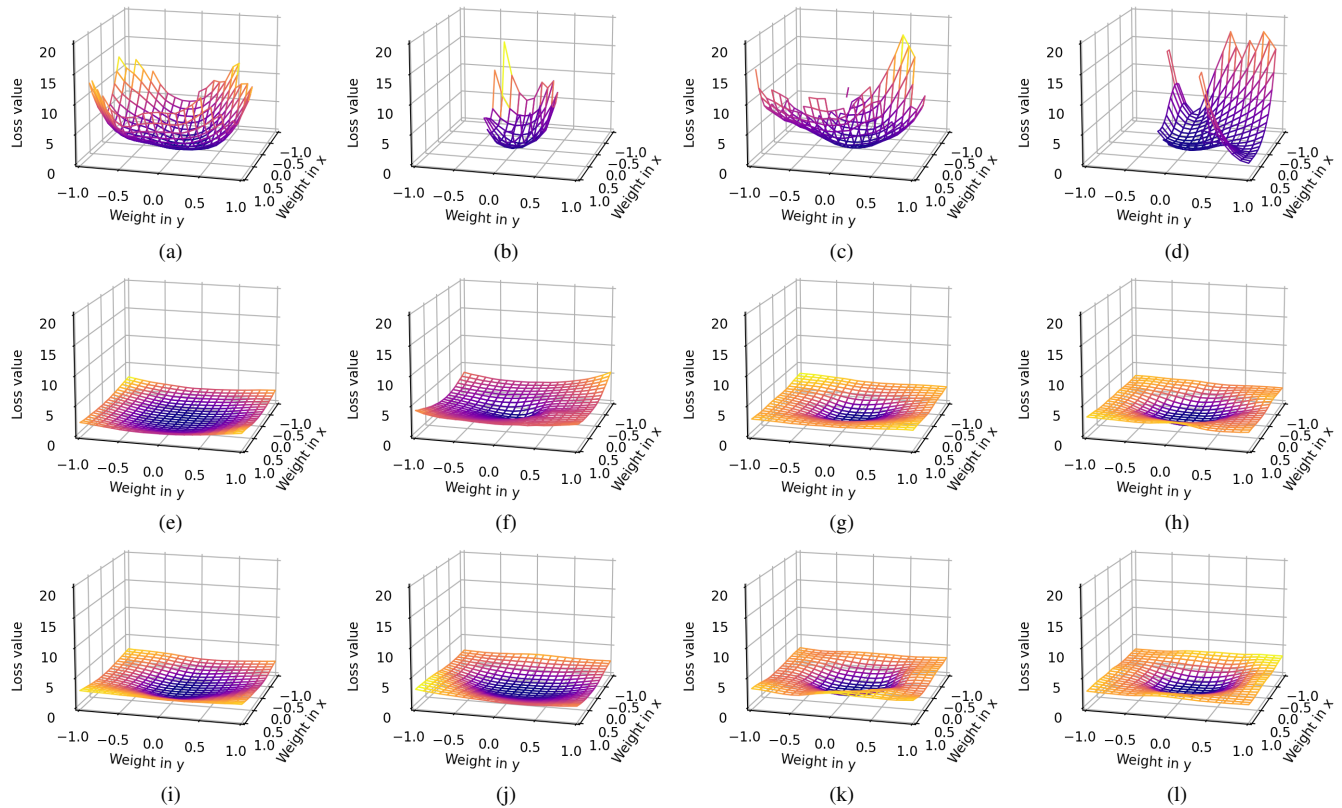


Fig. 9: Disturbance robustness visualization based on Houston 2013 dataset. (a) CNN3D. (b) DFFN. (c) M3D-DCNN. (d) RSSAN. (e) SpectralFormer. (f) SSFTT. (g) GroupTransformer. (h) Proposed CNN-mixer. (i) Proposed SSA-mixer. (j) Proposed CSA-mixer. (k) Proposed SSA+CNN-mixer. (l) Proposed CSA+CNN-mixer.

Pavia University. This leads to a sharper high-dimensional loss surface near the optimal point. This discrepancy may be attributed to the relatively smaller number of training samples per class in Botswana, indicating a need for improved model generalization. Upon comparing the five vision Transformer models proposed, it is evident that the horizontal coordinate of the peak value in the maximum eigenvalue distribution of the Hessian consistently exceeds 0. This signifies their capacity to maintain convexity in the vicinity of optimal points. Moreover, when compared to the other three pure (CNN-mixer, SSA-mixer, and CSA-mixer) models, the magnitude of the Hessian eigenvalue for the two hybrid-mixer (SSA+CNN-mixer and CSA+CNN-mixer) models approaches 0, indicating that hybrid-mixer models tend to exhibit smoother behavior near the optimal point. However, as shown in Fig. 9, 10, and 11, the smoothness of loss values after disturbance is already relatively high. Consequently, while hybrid-mixer models can further enhance local smoothness, they do not wield a decisive influence on optimizing the entire model, given the closely matched performance of all five models across the three datasets.

3) Impact of training ratio on OA: To investigate the impact of the number of training samples on the overall accuracy of HSI classification, experiments were conducted using different numbers of training samples in the three datasets with the proposed five approaches based on the unified hierarchical

vision Transformer architecture. For the Houston 2013 dataset, proportions of 1%, 3%, 5%, 7%, 9%, and 11% of the data were selected for training using a stratified sampling strategy. As shown in Fig. 15(a), when only 1% of the sample points were used as the training dataset, the OA ranges from 87.98% to 89.98%. At this stage, the model based on the CSA-mixer exhibits an OA of 87.98%, while the model using CSA+CNN-mixer achieves an OA of 89.98%, the highest accuracy among the models. As the number of training samples increases, the OA accuracy of all the five mixer models exhibits an increasing trend. When the training sample proportion reaches 11%, the model accuracy approaches saturation. At this point, the CSA+CNN mixer model reaches a notable accuracy of 99.56%, demonstrating a slight enhancement in performance with an accuracy improvement of less than 0.14% relative to four other models. For the Botswana dataset, a stratified sampling strategy was also employed to select training data in proportions of 2%, 6%, 10%, 14%, 18%, and 22%. As illustrated in Fig. 15(b), when only 2% of the data was utilized as training samples, the CNN-mixer-based model records an OA of 90.54%. It is at least 0.44% less than the accuracies achieved by the other four algorithms. Additionally, as the number of training samples grows, all the five models initially exhibit significant improvements in OA accuracy, which gradually plateau as they approach saturation. When the training sample proportion reaches 22% of the total dataset, the

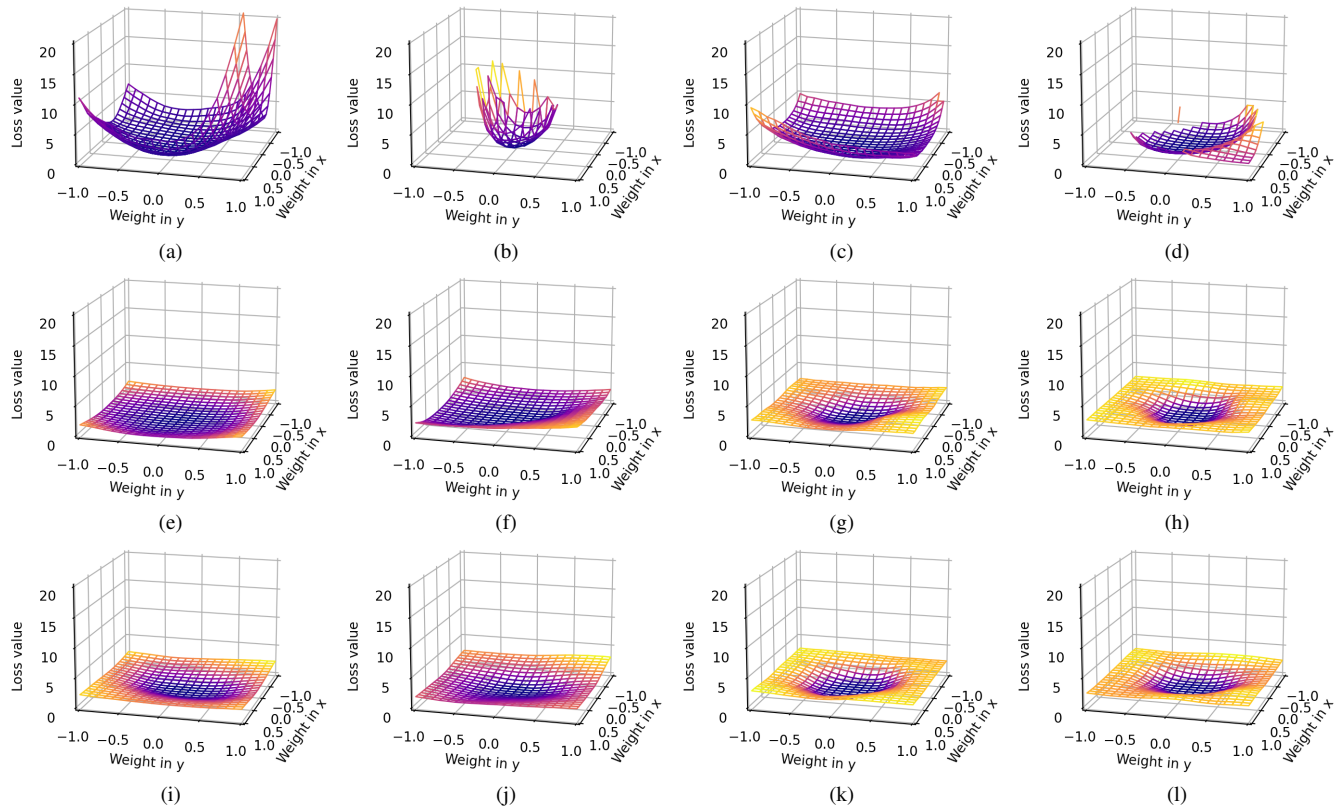


Fig. 10: Disturbance robustness visualization based on Botswana dataset. (a) CNN3D. (b) DFFN. (c) M3D-DCNN. (d) RSSAN. (e) SpectralFormer. (f) SSFTT. (g) GroupTransformer. (h) Proposed CNN-mixer. (i) Proposed SSA-mixer. (j) Proposed CSA-mixer. (k) Proposed SSA+CNN-mixer. (l) Proposed CSA+CNN-mixer.

accuracy of the five models ranges from 99.88% to 99.95%, with an OA accuracy fluctuation of no more than 0.07% among them. For the Pavia University dataset, only 0.5%, 1%, 1.5%, 2%, 2.5%, and 3% of the data were selected as training samples. This is because the number of sample points in this dataset is significantly greater than that of the previous two datasets. As shown in Fig. 15(c), the OA accuracy curves for the five mixer algorithms exhibit similar trends to those observed in the previous two datasets. For example, when the training samples increase from 0.5% to 3%, the OA accuracy improves dramatically from 90.99% to 99.75%, showing an impressive growth of 8.76%. With 2% of the data used as the training set, the OA accuracy of the five mixer models ranges from 99.44% to 99.55%. Overall, under the constraint of limited annotated samples, the quantity of training samples has a particularly noticeable impact on the accuracy of HSI classification. Furthermore, based on the proposed unified hierarchical vision Transformer architecture, different HSI classification models constructed with various mixers exhibit comparable performance across different datasets. This provides additional empirical support that for HSI vision Transformer classification algorithms, performance primarily relies on the unified hierarchical vision Transformer architecture rather than specific MSA or other mixer modules, especially under conditions where the proportion of training data is sufficiently substantial.

## V. CONCLUSIONS

A novel unified hierarchical vision Transformer architecture is developed for HSI classification. Five different vision Transformer models are constructed by configuring different mixers within the proposed unified architecture. Experiments on three commonly analyzed hyperspectral benchmark data sets with different characteristics reveal that the proposed methods outperform traditional CNN-based or vision Transformer-based HSI classification methods. Furthermore, an in-depth analysis conducted from two perspectives, disturbance robustness and the distribution of the maximum eigenvalue of the Hessian, implies that the effectiveness of vision Transformer-based HSI classification models primarily depends on the holistic unified architecture, rather than the commonly presumed MSA module. This paper provides insights into the design of vision Transformer-based neural networks for future research in HSI classification. Further work is warranted to incorporate self-supervised learning and analyze the frequency characteristics of feature space extraction in various mixer modules within the vision Transformer architecture through a self-supervised pre-training paradigm.

## VI. ACKNOWLEDGMENTS

This work was supported by the NASA grant #80NSSC22K1163.

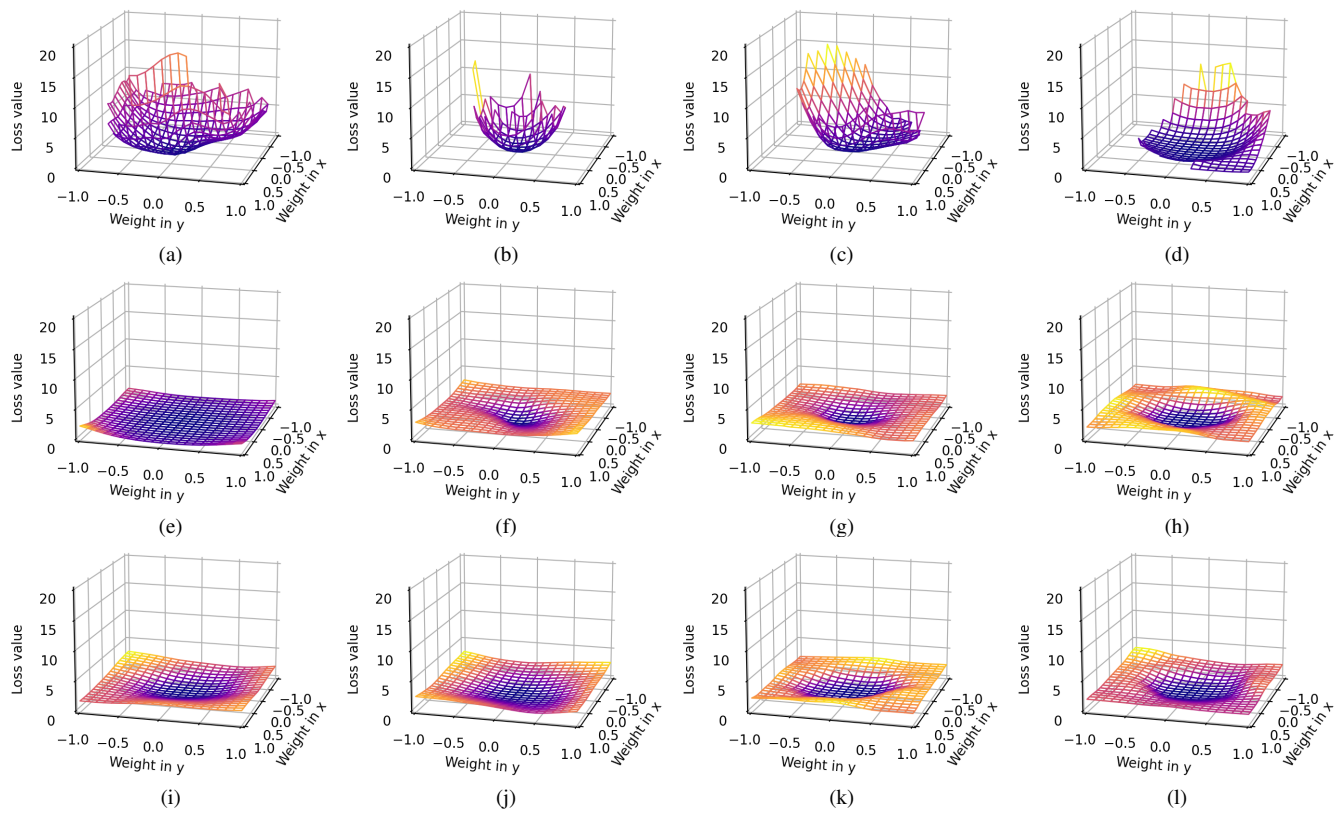


Fig. 11: Disturbance robustness visualization based on Pavia University dataset. (a) CNN3D. (b) DFFN. (c) M3D-DCNN. (d) RSSAN. (e) SpectralFormer. (f) SSFTT. (g) GroupTransformer. (h) Proposed CNN-mixer. (i) Proposed SSA-mixer. (j) Proposed CSA-mixer. (k) Proposed SSA+CNN-mixer. (l) Proposed CSA+CNN-mixer.



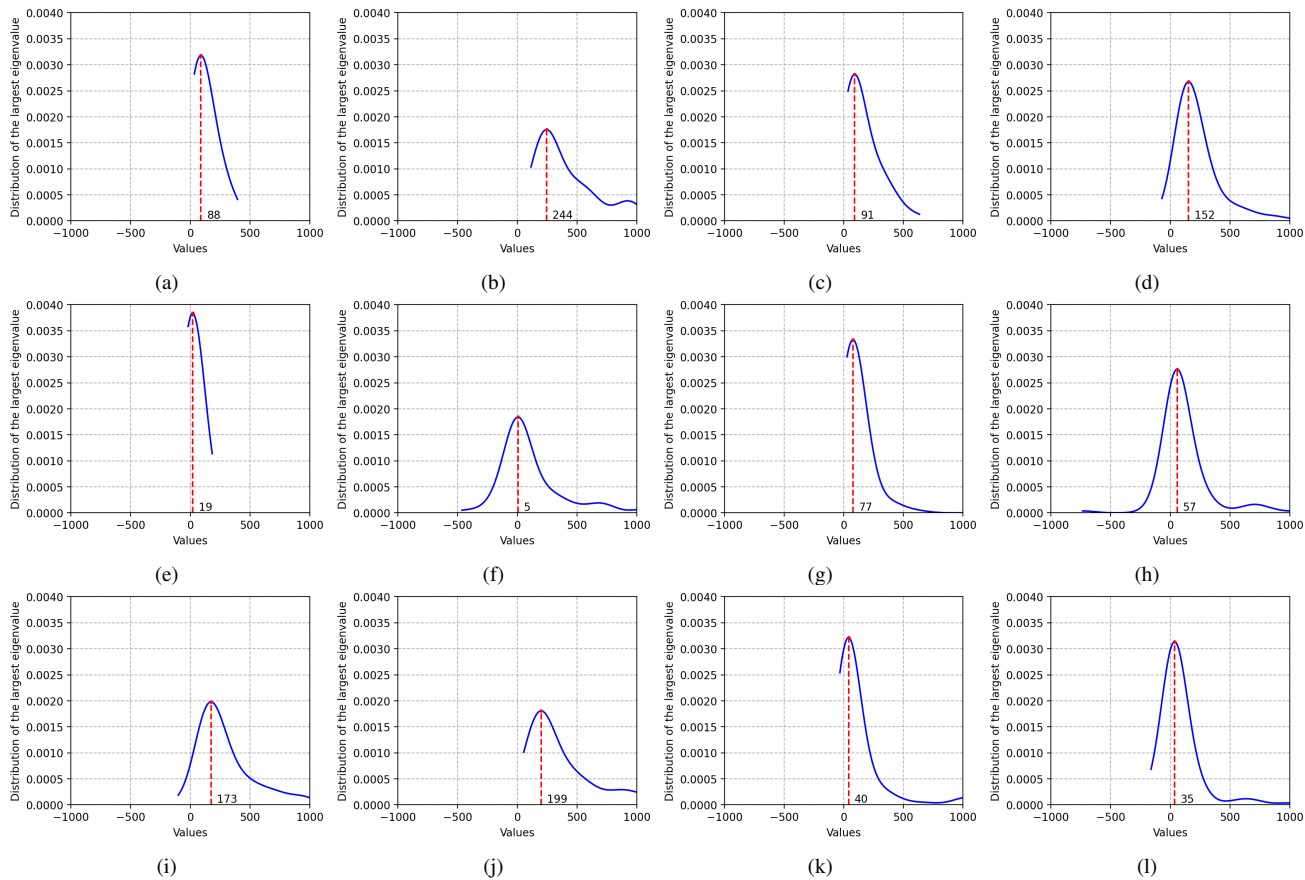


Fig. 12: Largest eigenvalue of the Hessian based on Houston 2013 dataset. (a) CNN3D. (b) DFFN. (c) M3D-DCNN. (d) RSSAN. (e) SpectralFormer. (f) SSFTT. (g) GroupTransformer. (h) Proposed CNN-mixer. (i) Proposed SSA-mixer. (j) Proposed CSA-mixer. (k) Proposed SSA+CNN-mixer. (l) Proposed CSA+CNN-mixer.

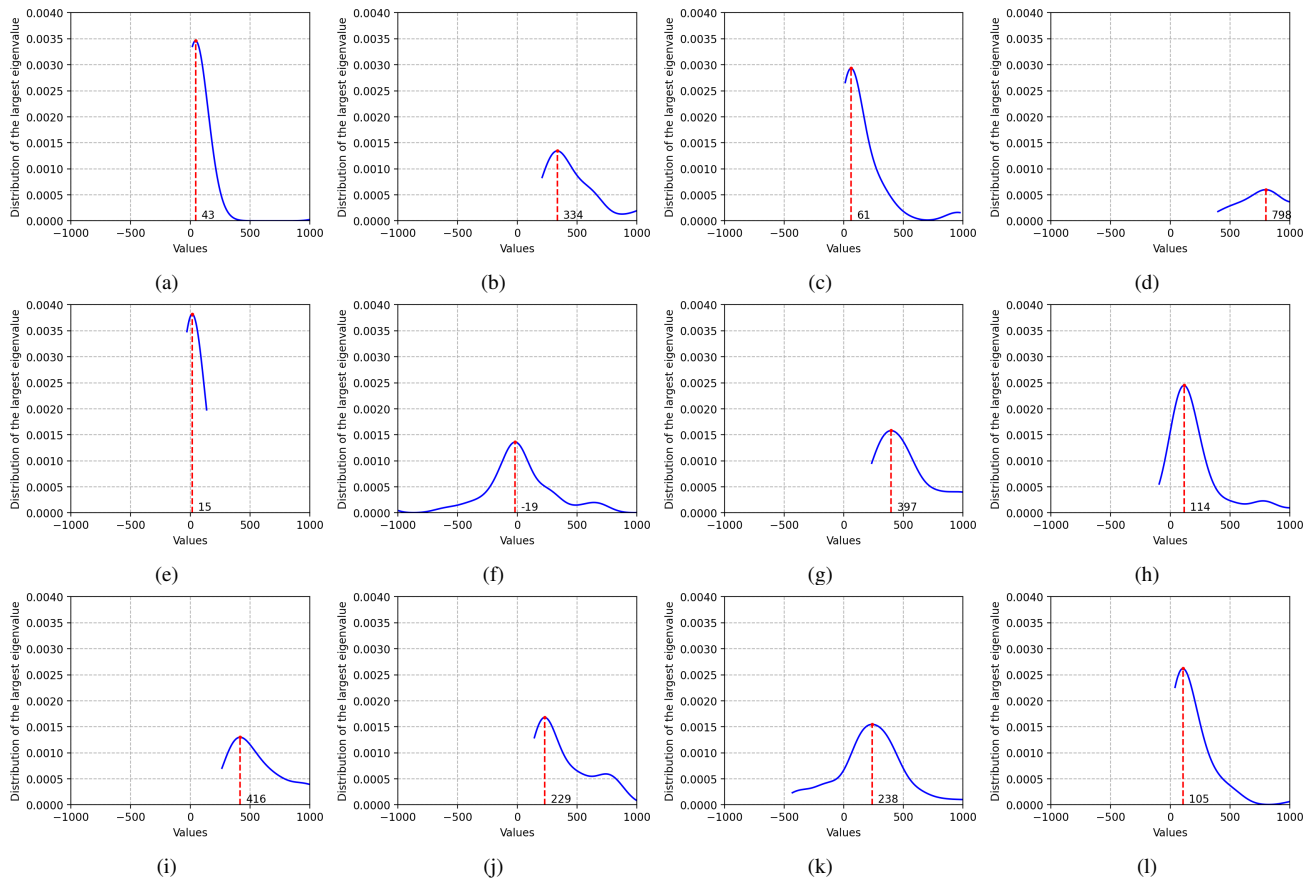


Fig. 13: Largest eigenvalue of the Hessian based on Botswana dataset. (a) CNN3D. (b) DFFN. (c) M3D-DCNN. (d) RSSAN. (e) SpectralFormer. (f) SSFTT. (g) GroupTransformer. (h) Proposed CNN-mixer. (i) Proposed SSA-mixer. (j) Proposed CSA-mixer. (k) Proposed SSA+CNN-mixer. (l) Proposed CSA+CNN-mixer.

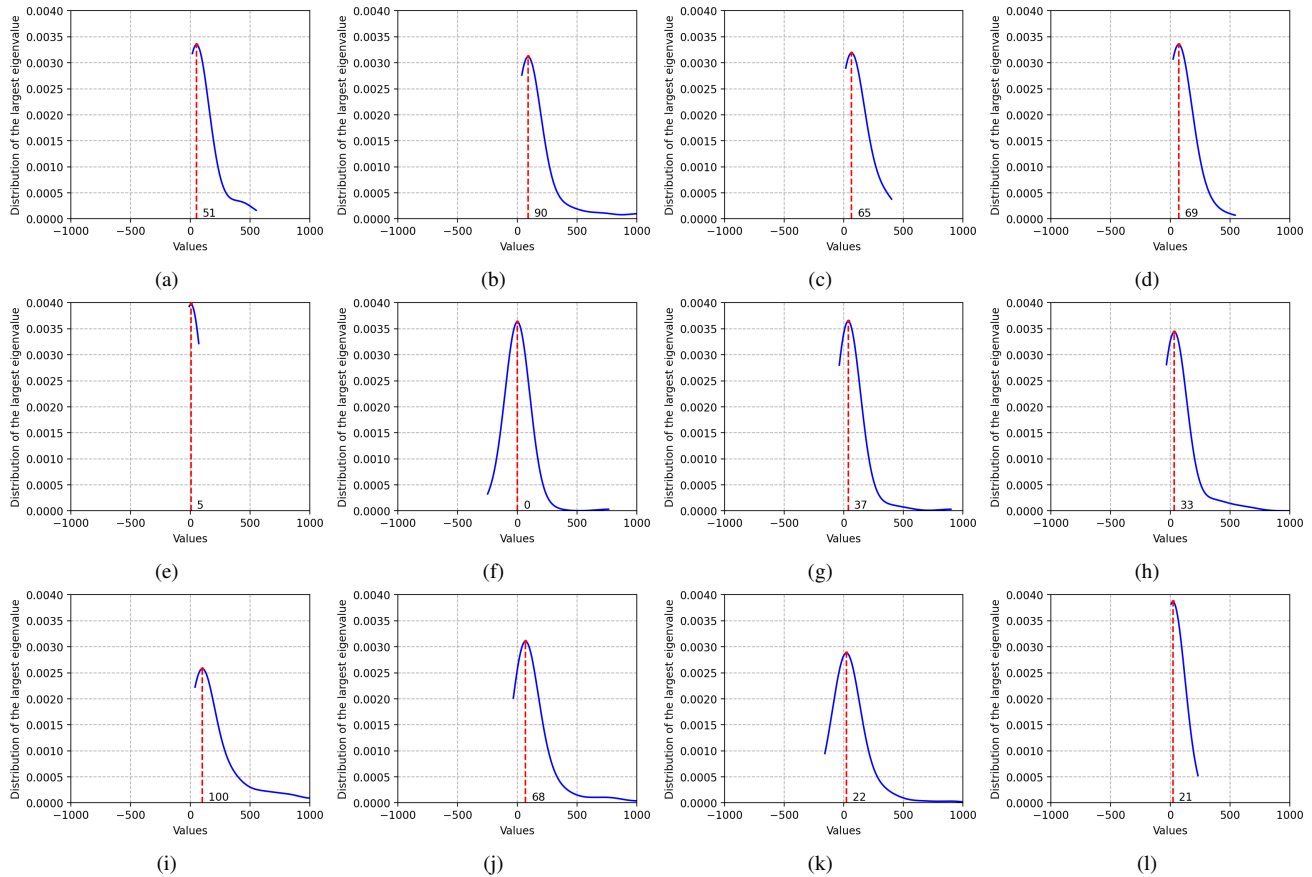


Fig. 14: Largest eigenvalue of the Hessian based on Pavia dataset. (a) CNN3D. (b) DFFN. (c) M3D-DCNN. (d) RSSAN. (e) SpectralFormer. (f) SSFTT. (g) GroupTransformer. (h) Proposed CNN-mixer. (i) Proposed SSA-mixer. (j) Proposed CSA-mixer. (k) Proposed SSA+CNN-mixer. (l) Proposed CSA+CNN-mixer.

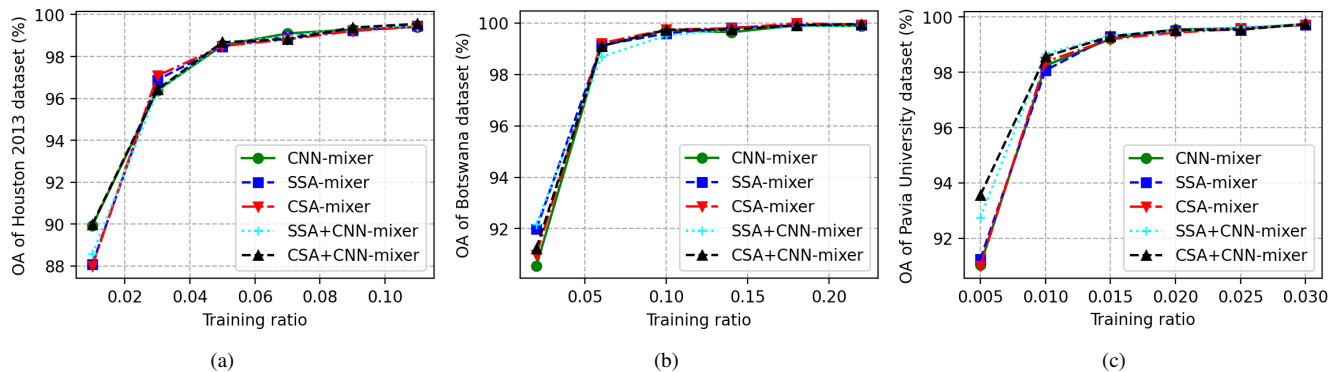


Fig. 15: Training ratio effect on the overall accuracy. (a) Houston 2013 dataset. (b) Botswana dataset. (c) Pavia University dataset.

## REFERENCES

- [1] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6690–6709, 2019.
- [2] N. Chen, J. Yue, L. Fang, and S. Xia, "Spectraldiff: A generative framework for hyperspectral image classification with diffusion models," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–16, 2023.
- [3] L. Fang, Y. Yan, J. Yue, and Y. Deng, "Towards the vectorization of hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–14, 2023.
- [4] F. Xu, G. Zhang, C. Song, H. Wang, and S. Mei, "Multiscale and cross-level attention learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–15, 2023.
- [5] B. Cheng, I. S. Saggi, R. Shah, G. Bansal, and D. Bharadia, "S 3 net: Semantic-aware self-supervised depth estimation with monocular videos and synthetic data," in *European Conference on Computer Vision*. Springer, 2020, pp. 52–69.
- [6] S. Li, K. Gong, C. H. Liu, Y. Wang, F. Qiao, and X. Cheng, "Metasaug: Meta semantic augmentation for long-tailed visual recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 5212–5221.
- [7] Z. Xiong, F. Qiao, Y. Zhang, and N. Jacobs, "Stereo-flowgan: Co-training for stereo and flow with unsupervised domain adaptation," *arXiv preprint arXiv:2309.01842*, 2023.
- [8] F. Lu, G. Chen, Y. Liu, Z. Qu, and A. Knoll, "Rskdd-net: Random sample-based keypoint detector and descriptor," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 297–21 308, 2020.
- [9] L. Wu, L. Fang, X. He, M. He, J. Ma, and Z. Zhong, "Querying labeled for unlabeled: Cross-image semantic consistency guided semi-supervised semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 7, pp. 8827–8844, 2023.
- [10] J. Wu, D. Zhu, L. Fang, Y. Deng, and Z. Zhong, "Efficient layer compression without pruning," *IEEE Transactions on Image Processing*, vol. 32, pp. 4689–4700, 2023.
- [11] W. Song, S. Li, L. Fang, and T. Lu, "Hyperspectral image classification with deep feature fusion network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 6, pp. 3173–3184, 2018.
- [12] A. B. Hamida, A. Benoit, P. Lambert, and C. B. Amar, "3-d deep learning approach for remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 8, pp. 4420–4434, 2018.
- [13] M. He, B. Li, and H. Chen, "Multi-scale 3d deep convolutional neural network for hyperspectral image classification," *2017 IEEE International Conference on Image Processing*, pp. 3904–3908, 2017.
- [14] M. Zhu, L. Jiao, F. Liu, S. Yang, and J. Wang, "Residual spectral-spatial attention network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 1, pp. 449–462, 2020.
- [15] W. Liu, S. Prasad, and M. Crawford, "Cnn-mixer hierarchical spectral transformer for hyperspectral image classification," in *IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2023, pp. 5946–5949.
- [16] X. Yang, W. Cao, Y. Lu, and Y. Zhou, "Hyperspectral image transformer classification networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.
- [17] D. Wang, J. Zhang, B. Du, L. Zhang, and D. Tao, "Dcn-t: Dual context network with transformer for hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 32, pp. 2536–2551, 2023.
- [18] D. Hong, Z. Han, J. Yao, L. Gao, B. Zhang, A. Plaza, and J. Chanussot, "Spectralformer: Rethinking hyperspectral image classification with transformers," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2021.
- [19] S. Mei, C. Song, M. Ma, and F. Xu, "Hyperspectral image classification using group-aware hierarchical transformer," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.
- [20] J. Zou, W. He, and H. Zhang, "Lessformer: Local-enhanced spectral-spatial transformer for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2022.
- [21] W. Qi, C. Huang, Y. Wang, X. Zhang, W. Sun, and L. Zhang, "Global-local three-dimensional convolutional transformer network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–20, 2023.
- [22] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik, and Y. Li, "Maxvit: Multi-axis vision transformer," in *European conference on computer vision*. Springer, 2022, pp. 459–479.
- [23] E. Ouyang, B. Li, W. Hu, G. Zhang, L. Zhao, and J. Wu, "When multi-granularity meets spatial-spectral attention: A hybrid transformer for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–18, 2023.
- [24] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *Advances in Neural Information Processing Systems*, vol. 34, pp. 24 261–24 272, 2021.
- [25] Y. Shao, J. Liu, J. Yang, and Z. Wu, "Spatial-spectral involution mlp network for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 9293–9310, 2022.
- [26] J. Guo, Y. Tang, K. Han, X. Chen, H. Wu, C. Xu, C. Xu, and Y. Wang, "Hire-mlp: Vision mlp via hierarchical rearrangement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 826–836.
- [27] W. Yu, C. Si, P. Zhou, M. Luo, Y. Zhou, J. Feng, S. Yan, and X. Wang, "Metaformer baselines for vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
- [28] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, "Pre-trained image processing transformer," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 299–12 310.
- [29] S. Yu, S. Jia, and C. Xu, "Convolutional neural networks for hyperspectral image classification," *Neurocomputing*, vol. 219, pp. 88–98, 2017.
- [30] J. Yang, Y. Zhao, J. C.-W. Chan, and C. Yi, "Hyperspectral image classification using two-channel deep convolutional neural network," in *2016 IEEE international geoscience and remote sensing symposium (IGARSS)*. IEEE, 2016, pp. 5079–5082.
- [31] J. Yue, L. Fang, and M. He, "Spectral-spatial latent reconstruction for open-set hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 31, pp. 5227–5241, 2022.
- [32] J. Yang, C. Wu, B. Du, and L. Zhang, "Enhanced multiscale feature fusion network for hsi classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 12, pp. 10 328–10 347, 2021.
- [33] J. Yang, B. Du, Y. Xu, and L. Zhang, "Can spectral information work while extracting spatial distribution?—an online spectral information compensation network for hsi classification," *IEEE Transactions on Image Processing*, vol. 32, pp. 2360–2373, 2023.
- [34] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 145, pp. 120–147, 2018.
- [35] X. Yang, Y. Ye, X. Li, R. Y. Lau, X. Zhang, and X. Huang, "Hyperspectral image classification with deep learning models," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 9, pp. 5408–5423, 2018.
- [36] M. Ahmad, A. M. Khan, M. Mazzara, S. Distefano, M. Ali, and M. S. Sarfraz, "A fast and compact 3-d cnn for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2020.
- [37] C. Yu, R. Han, M. Song, C. Liu, and C.-I. Chang, "A simplified 2d-3d cnn architecture for hyperspectral image classification based on spatial-spectral fusion," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 2485–2501, 2020.
- [38] Z. Ge, G. Cao, X. Li, and P. Fu, "Hyperspectral image classification method based on 2d-3d cnn and multibranch feature fusion," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 5776–5788, 2020.
- [39] Z. Xia, X. Pan, S. Song, L. E. Li, and G. Huang, "Vision transformer with deformable attention," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 4794–4803.
- [40] Z. Meng, X. Xia, and J. Ma, "Toward foundation models for inclusive object detection: Geometry-and category-aware feature extraction across road user categories," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2024.
- [41] J. Feng, Z. Zhou, R. Shang, J. Wu, T. Zhang, X. Zhang, and L. Jiao, "Class-aligned and class-balancing generative domain adaptation for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–17, 2024.
- [42] J. Feng, Z. Gao, R. Shang, X. Zhang, and L. Jiao, "Multi-complementary generative adversarial networks with contrastive learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–18, 2023.

- [43] F. Lu, G. Chen, Y. Liu, L. Zhang, S. Qu, S. Liu, R. Gu, and C. Jiang, "Hregnet: A hierarchical network for efficient and accurate outdoor lidar point cloud registration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 10, pp. 11 884–11 897, 2023.
- [44] L. Fang, Y. Jiang, Y. Yan, J. Yue, and Y. Deng, "Hyperspectral image instance segmentation using spectral–spatial feature pyramid network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–13, 2023.
- [45] H. Cao, Z. Qu, G. Chen, X. Li, L. Thiele, and A. Knoll, "Ghostvit: Expediting vision transformers via cheap operations," *IEEE Transactions on Artificial Intelligence*, 2023.
- [46] 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 Transactions on Image Processing*, vol. 31, pp. 1559–1572, 2022.
- [47] Y. Zhan, K. Wu, and Y. Dong, "Enhanced spectral–spatial residual attention network for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 7171–7186, 2022.
- [48] R. Shang, W. Li, W. Zhang, J. Feng, Y. Li, and L. Jiao, "Simplified nonlocal network based on adaptive projection attention method for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.
- [49] Y. Xu, Y. Zhang, C. Yu, C. Ji, T. Yue, and H. Li, "Residual spatial attention kernel generation network for hyperspectral image classification with small sample size," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.
- [50] H. Zhai, J. Zhao, and H. Zhang, "Double attention based multilevel one-dimensional convolution neural network for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 3771–3787, 2022.
- [51] Y. Liu, K. Cao, R. Wang, M. Tian, and Y. Xie, "Hyperspectral image classification of brain-inspired spiking neural network based on attention mechanism," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.
- [52] L. Sun, G. Zhao, Y. Zheng, and Z. Wu, "Spectral–spatial feature tokenization transformer for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.
- [53] B. Tu, X. Liao, Q. Li, Y. Peng, and A. Plaza, "Local semantic feature aggregation-based transformer for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.
- [54] Z. Xue, Q. Xu, and M. Zhang, "Local transformer with spatial partition restore for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 4307–4325, 2022.
- [55] H. Gao, H. Wu, Z. Chen, Y. Zhang, and S. Xu, "Fusion network for local and global features extraction for hyperspectral image classification," *International Journal of Remote Sensing*, vol. 43, no. 10, pp. 3843–3867, 2022.
- [56] H. Yan, E. Zhang, J. Wang, C. Leng, A. Basu, and J. Peng, "Hybrid convit network for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 20, pp. 1–5, 2023.
- [57] G. Zhao, Q. Ye, L. Sun, Z. Wu, C. Pan, and B. Jeon, "Joint classification of hyperspectral and lidar data using a hierarchical cnn and transformer," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–16, 2022.
- [58] Y. Chen, P. Liu, J. Zhao, K. Huang, and Q. Yan, "Shallow-guided transformer for semantic segmentation of hyperspectral remote sensing imagery," *Remote Sensing*, vol. 15, no. 13, p. 3366, 2023.
- [59] X. Qiao and W. Huang, "A dual frequency transformer network for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2023.
- [60] J. Yang, B. Du, and L. Zhang, "Overcoming the barrier of incompleteness: A hyperspectral image classification full model," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2023.
- [61] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.
- [62] D. Hong, B. Zhang, X. Li, Y. Li, C. Li, J. Yao, N. Yokoya, H. Li, P. Ghamisi, X. Jia *et al.*, "Spectralgpt: Spectral remote sensing foundation model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [63] Y. Chen and Q. Yan, "Lfsmim: A low-frequency spectral masked image modeling method for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, 2024.
- [64] Y. Liu, X. Li, Z. Hua, C. Xia, and L. Zhao, "A band selection method with masked convolutional autoencoder for hyperspectral image," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.
- [65] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.
- [66] Q. Zhang, Y. Xu, J. Zhang, and D. Tao, "Vitaev2: Vision transformer advanced by exploring inductive bias for image recognition and beyond," *International Journal of Computer Vision*, pp. 1–22, 2023.
- [67] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [68] N. Park and S. Kim, "How do vision transformers work?" *arXiv preprint arXiv:2202.06709*, 2022.
- [69] B. Ghorbani, S. Krishnan, and Y. Xiao, "An investigation into neural net optimization via hessian eigenvalue density," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2232–2241.
- [70] Z. Yao, A. Gholami, K. Keutzer, and M. W. Mahoney, "Pyhessian: Neural networks through the lens of the hessian," in *2020 IEEE International Conference on Big Data (Big data)*. IEEE, 2020, pp. 581–590.
- [71] (2023) Hyperspectral remote sensing scenes. Accessed: 2023-10-30. [Online]. Available: [https://www.ehu.eus/ccwintco/index.php?title=Hyperspectral\\_Remote\\_Sensing\\_Scenes](https://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes)
- [72] (2023) 2013 ieee grss data fusion contest – fusion of hyperspectral and lidar data. Hyperspectral Imaging Laboratory, University of Houston. Accessed: 2023-10-30. [Online]. Available: [https://hyperspectral.ee.uh.edu/?page\\_id=459](https://hyperspectral.ee.uh.edu/?page_id=459)