Pyramid Hierarchical Spatial-Spectral Transformer for Hyperspectral Image Classification

Muhammad Ahmad, Muhammad Hassaan Farooq Butt, Manuel Mazzara, Salvatore Distefano, Adil Mehmood Khan, and Hamad Ahmed Altuwaijri

***Abstract*—The transformer model encounters challenges with variable-length input sequences, leading to efficiency and scalability concerns. To overcome this, we propose a pyramid-based hierarchical spatial-spectral transformer (PyFormer). This innovative approach organizes input data hierarchically into pyramid segments, each representing distinct abstraction levels, thereby enhancing processing efficiency. At each level, a dedicated transformer encoder is applied, effectively capturing both local and global context. Integration of outputs from different levels culminates in the final input representation. In short, the pyramid excels at capturing spatial features and local patterns, while the transformer effectively models spatial-spectral correlations and long-range dependencies. Experimental results underscore the superiority of the proposed method over state-of-the-art approaches, achieving overall accuracies of 96.28% for the Pavia University dataset and 97.36% for the University of Houston dataset. In addition, the incorporation of disjoint samples augments robustness and reliability, thereby highlighting the potential of PyFormer in advancing hyperspectral image classification (HSIC).**

***Index Terms*—Pyramid network, spatial-spectral transformer (SST), hyperspectral image classification (HSIC).**

I. Introduction

HYPERSPECTRAL image classification (HSIC) is crucial in diverse domains [1], [2], [3]. CNNs [4], [5], [6], [7], [8], [9], [10], [11], [12] specifically Pyramid-CNN (PCNN) [13], [14], [15], [16], [17] and transformers [18], [19], [20], [21] have shown success in computer vision tasks, there is a growing interest in exploring these models for advancing HSI analysis.

Received 2 August 2024; revised 31 August 2024; accepted 13 September 2024. Date of publication 17 September 2024; date of current version 7 October 2024. This work was supported by King Saud University, Riyadh, Saudi Arabia under Grant RSPD2024R848. *(Corresponding author: Muhammad Ahmad.)*

Muhammad Ahmad is with the Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad, Chiniot- Faisalabad Campus, Chiniot 35400, Pakistan (e-mail: mahmad00@gmail.com).

Muhammad Hassaan Farooq Butt is with the School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu 611756, China (e-mail: hassaanbutt67@gmail.com).

Manuel Mazzara is with the Institute of Software Development and Engineering, Innopolis University, 420500 Innopolis, Russia (e-mail: m.mazzara@innopolis.ru).

Salvatore Distefano is with the Dipartimento di Matematica e Informatica—MIFT, University of Messina, 98121 Messina, Italy (e-mail: sdistefano@unime.it).

Adil Mehmood Khan is with the School of Computer Science, University of Hull, HU6 7RX Hull, U.K. (e-mail: a.m.khan@hull.ac.uk).

Hamad Ahmed Altuwaijri is with the Department of Geography, College of Humanities and Social Sciences, King Saud University, Riyadh 11451, Saudi Arabia (e-mail: Haaltuwaijri@ksu.edu.sa).

Code is available online at https://github.com/mahmad00/PyFormer.

Digital Object Identifier 10.1109/JSTARS.2024.3461851

PCNN incorporates multiscale processing by using multiple convolutional branches, allowing the network to capture features at different scales and levels of abstraction [22]. PCNN has been extensively studied for HSIC, and several innovative solutions have been proposed [14], [15]; however, PCNN has several limitations; first, high computational cost due to the high dimensionality, e.g., the number of convolutional branches increases with the number of scales [22]. Second, the multiple branches increase the model's complexity, i.e., a high number of parameters are required. Moreover, the complex models are prone to overfitting, particularly when the training data is limited [23]. Third, PCNN primarily focuses on the multiscale processing of spectral information, but it may not fully exploit the spatial context. Spatial context is important as neighboring pixels often exhibit strong correlations. PCNN may not fully capture these dependencies, potentially limiting its performance for HSIC [24], [25].

Whereas, the spatial-spectral Transformers (SSTs) [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36] excel in capturing global contextual information via self-attention mechanisms [37], [38], [39], [40], facilitating simultaneous consideration of relationships between all HSI regions [41]. Unlike PCNN, SSTs demonstrate strong scalability to high-resolution HSIs, effectively handling large datasets without complex pooling operations. Their adaptability contributes to widespread applicability in HSIC [42]. Furthermore, SSTs learn stratified representations directly from raw pixel values, simplifying the model-building process and often leading to improved performance [34], [43].

Recent advancements in HSIC have explored CNN-Transformer-based architectures [44], [45] that harness the complementary strengths of CNNs and Transformers. In these architectures, CNNs are utilized to capture local spatial features, while transformers preserves long-range dependencies and global context. Such approaches effectively integrate the detailed spatial information provided by CNNs with the global contextual understanding offered by transformers, enhancing overall classification performance. Tan et al. [46] developed the transformer-in-transformer module, which constructs a deep network model tailored for HSIC by incorporating extended morphological contour features. Tang et al. [47] introduced a ViT-based backbone network for HSIC that integrates a stack of spectral attention and spatial attention layers to enhance feature representation. Ma et al. [48] utilized a deep CNN to extract spatial features, followed by a densely connected ViT to capture spectral relationships within the data sequence. Song et al. [49] proposed a dual-branch framework combining a 3-D CNN with

a spatial-spectral transformer network to extract local and global features from HSI data jointly. Zhao et al. [50] developed a convolution transformer fusion splicing network for HSIC, incorporating a residual splicing convolution block to serialize HSI data. However, many CNN-transformer-based methods face limitations, as they often use low-level CNN features as inputs to transformers, leading to a deficiency in capturing semantic information.

To overcome the aforementioned limitations, Sun et al. [51] introduced the spectral–spatial feature Tokenization transformer (SSFTT), which incorporates a hybrid CNN module to capture local features and employs a Gaussian-weighted tokenizer to extract high-level semantic information. Zhang et al. [52] proposed the convolution transformer mixer (CT Mixer), which combines the strengths of ViTs and CNNs, leveraging a local-global multihead self-attention (MHSA) mechanism to enhance classification accuracy. Zhao et al. [53] developed a convolutional transformer network, utilizing CNNs for local feature extraction (LFE) and transformers for global feature extraction, and introduced a center position encoding technique to integrate spectral features with pixel positions. Yang et al. [54] designed a ViT incorporating the spectral adaptive 3-D convolution projection and the convolution permutator to capture spectral-spatial information. Li et al. [55] proposed a multigranularity ViT that employs semantic tokens to learn features at multiple granularities, aiming to enhance accuracy. This framework's LFE module is specifically designed for LFE. Ouyang et al. [56] introduced the HybridFormer network, which utilizes CNNs for extracting shallow features and a spectral-spatial attention-based transformer encoder to capture semantic features.

Despite their success, SSTs have limitations, for instance, training large SSTs can be computationally demanding [57], [58]. The self-attention mechanism introduces quadratic complexity with respect to sequence length, potentially hindering scalability [59], [60]. Unlike CNNs, which inherently possess translation invariance, SSTs may struggle to capture spatial relationships invariant to small translations in the input [61], [62]. Moreover, the tokenization process of dividing input images into fixed-size patches may not efficiently capture fine-grained details [63], [64]. Furthermore, optimal performance often requires substantial training data, and training on smaller datasets may lead to overfitting, limiting effectiveness [19], [65].

Therefore, this work synergistically integrates a PCNN and SST, resulting in an innovative hierarchical SST for HSIC. The hierarchical structure partitions the input into segmentation, each denoting varying abstraction levels, organized in a pyramid-like manner. Transformer modules are applied at each level for multilevel processing, ensuring efficient capture of local and global context. Information flow occurs both spatially and spectrally within the hierarchy, fostering abstraction propagation. Integration of transformer outputs from different levels yields the final output maps. In short, the following contributions are made; first, the input sequences are divided into hierarchical segments, each representing varying levels of abstraction. Second, these segments adopt a pyramid structure, wherein the lowest level retains detailed information while higher levels convey increasingly abstract representations. Third, the transformer modules are independently applied at each level of the hierarchy, facilitating efficient capture of both local and global contexts. In a nutshell, the Pyramid excels at capturing spatial features and local patterns, while the transformer effectively models spatial-spectral correlations and long-range dependencies.

II. Proposed Methodology

An HSI cube $X = \{x_i, y_i\} \in \mathbb{R}^{(M\times N\times B)}$, comprises spectral vectors $x_i = \{x_{i,1}, x_{i,2}, x_{i,3}, \ldots, x_{i,L}\}$, and y_i be the corresponding class label of x_i. The cube is initially divided into overlapping 3-D patches, each centered at spatial coordinates (α, β) and spanning $S \times S$ pixels across B bands. The total count of extracted patches (m) from X is $(M - S + 1) \times (N - S + 1)$, where a patch $P_{\alpha,\beta}$ covers spatial dimensions within $\alpha \pm \frac{S-1}{2}$ and $\beta \pm \frac{S-1}{2}$. In cases, where the stride s is less than the patch size, S, which results in overlapping patches. The overlap ratio r can be defined as: $r = 1 - \frac{s}{S}$. The extracted patches, along with their central pixel labels, constitute the training X_{train}, validation X_{val}, and a test X_{test} sets, ensuring $X_{\text{train}} \cap X_{\text{val}} \cap X_{\text{test}} = \emptyset$ to prevent sample overlaps and biases. The complete model is presented in the Fig. 1.

Let (S, S, B) denote the input shape. The scale is utilized to derive the input shape for the pyramid layers, given by Input shape; $l_1 = \frac{S}{2} \times \frac{S}{2} \times B$ and $l_2 = \frac{S}{4} \times \frac{S}{4} \times B$, respectively, i.e., each pyramid level l extracts features at different scales with the scaling factor $s_l = 2^l$

$$X = \text{Downsample}(X, s_l) \in \mathbb{R}^{\frac{S}{s_l}\times\frac{S}{s_l}\times B} \tag{1}$$

where the Downsample$(., s)$ reduces the spatial dimensions by a factor of s. These Downsample patches are fed into a convolutional layer to extract spatial-spectral semantic features from HSI patches. Each patch, with dimensions $(\frac{S}{s_l} \times \frac{S}{s_l} \times B)$, undergoes processing using 3-D convolutional layers with kernel sizes $(32 \times S \times S \times B)$ and $(64 \times S \times S \times B)$, along with ReLU activation as

$$Y_i^{(1)} = \sigma\left(X_l * W_l^{(1)} + b_l^{(1)}\right) \in \mathbb{R}^{\frac{S}{s_l}\times\frac{S}{s_l}\times B\times 32} \tag{2}$$

$$Y_i^{(2)} = \sigma\left(Y_i^{(1)} * W_l^{(2)} + b_l^{(2)}\right) \in \mathbb{R}^{\frac{S}{s_l}\times\frac{S}{s_l}\times B\times 64} \tag{3}$$

where $\sigma(.)$ denotes the ReLU activation function, $*$ denotes the convolution operation, and $W_l^{(i)}$ and $b_l^{(i)}$ are the convolution kernels and biases. Later, the upsampling$(., s)$ is performed, which increases the spatial dimensions by a factor of s as

$$U_l = \text{Upsample}(Y_l^{(2)}, s_l) \in \mathbb{R}^{S\times S\times B\times 64} \tag{4}$$

where the output of each pyramid level is U_l. The feature maps from all pyramid levels are concatenated along the channel dimensions as

$$Y_{\text{Pyramid}} = \text{concat}(U_1, U_2, \ldots, U_l) \in \mathbb{R}^{S\times S\times B\times(64\times l)}. \tag{5}$$

Let $Y_{\text{Pyramid}} \in \mathbb{R}^{S\times S\times B\times(64\times l)}$ denote the input tensor to the transformer. This encoding is integrated with the input embeddings, augmenting the model with spatial arrangement details. The foundational architecture of the transformer centers around the encoder, which consists of multiple layers incorporating multimodal attention and a feedforward network. Each transformer layer incorporates multihead self-attention specifically

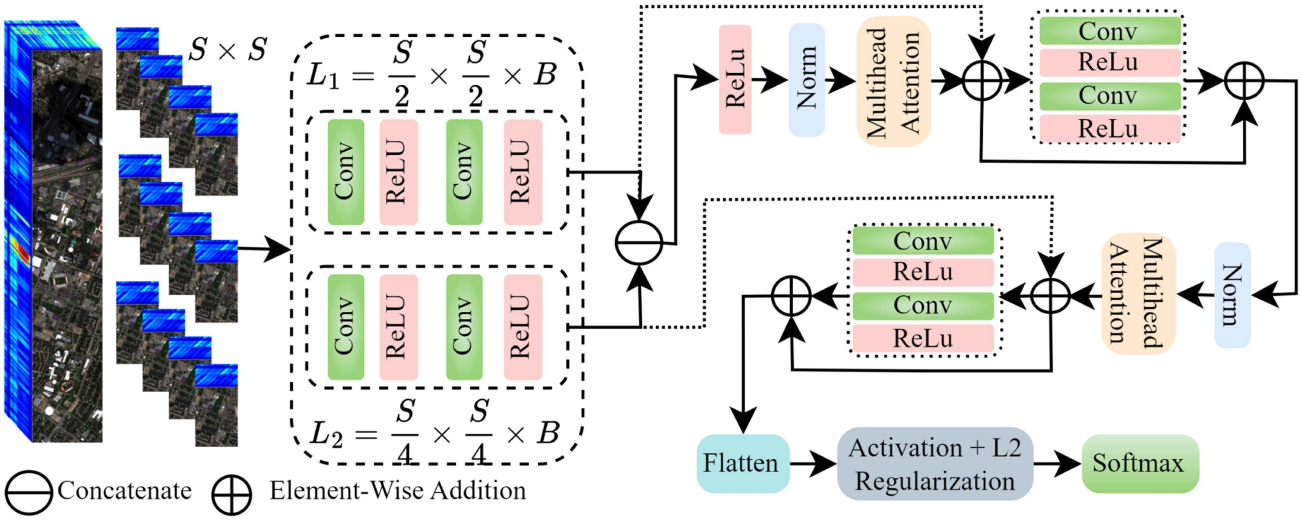

Fig. 1. Pyramid hierarchical transformer features a structured pyramid block comprising two levels. The hierarchical transformer block, receiving the learned multiscale information, consists of two layers and four multiheads. The acquired information undergoes flattening and is subsequently subjected to the ReLU activation function and L2 regularization technique. This regularization aids in mitigating overfitting by reducing weights, rendering the network less responsive to minor input variations. Finally, the output layer employs softmax activation for final maps.

adapted for HSI data. For each transformer layer i where ($i = 1, 2, \ldots,$ layers) first undergoes a layer normalization as

$$Z_i = \text{LayerNorm}(Y_{\text{pyramid}}). \tag{6}$$

The attention mechanism plays a pivotal role in enabling the model to capture intricate relationships between distinct patches using the query, key, and value matrices as

$$Q = Z_i W_Q; K = Z_i W_K; V = Z_i W_V \tag{7}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{(64\times l)\times d_k}$ and d_k is the dimensionality of the attention space. Finally, the scaled dot product attention is computed as

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

where the output of multihead is computed as

$$\text{MHA}_i(Z_i) = \text{concat}\left(\text{head}_1, \text{head}_2, \ldots, \text{head}_l\right) W_o \tag{9}$$

where each head is computed as $\text{head}_j = \text{Attention}(Q_j, K_j, V_j)$ and $W_o \in \mathbb{R}^{(\text{heads}\times d_k)\times(63\times l)}$. Later, the model adds and normalizes the output $Y_{(i+1)}$ as follows:

$$Y_{(i+1)} = \text{LayerNorm}\left(Y_{\text{pyramid}} + \text{MHA}_i(Z_i)\right). \tag{10}$$

After the transformer layers, the model applies convolutional layers with residual connections to integrate spectral-spatial features as initial convolution, second convolution, and a residual connection

$$C_i^{(1)} = \sigma\left(Y_{i+1} * W_i^{(1)} + b_i(2)\right) \in \mathbb{R}^{S\times S\times B\times B} \tag{11}$$

$$C_i^{(2)} = \sigma\left(C_i^{(1)} * W_i^{(2)} + b_i(2)\right) \in \mathbb{R}^{S\times S\times B\times(2\times \text{mlp\_dim})} \tag{12}$$

$$Y_{(i+2)} = Y_{(i+1)} + C_i(2). \tag{13}$$

The final output of the hierarchical transformer model is flattened and passed through dense layers for classification as

$$F = \text{Flatten}(Y_{\text{Final}}) \tag{14}$$

TABLE I
OVERVIEW OF HSI DATASETS EMPLOYED IN EXPERIMENTAL EVALUATION

—	PU	UH	SA
Spectral	115	144	224
Spatial	610 × 610	340 × 1905	340 × 1905
Samples	207400	1329690	54129
Classes	9	15	16
Resolution	1.3 m	2.5 mpp	3.7 m
Wavelength (nm)	430 − 860	350 − 1050	350 − 1050
Sensor	ROSIS-03	CASI	AVIRIS
Source	Aerial	Aerial	Aerial

$$D = \sigma(FW_d + b_d) + \lambda||W_d||_2^2 \tag{15}$$

$$O = \text{Softmax}(DW_o + b_o) \tag{16}$$

where F, D, and O represent flattened, dense, and classification layers, respectively. Within the dense layer, L_2 regularization $L_2 = \lambda|W|_2^2$ is added to the loss function during training, with ReLU as the activation function and $\lambda = 0.01$ as the regularization parameter. Finally, a Softmax function is employed to generate the classification maps.

III. EXPERIMENTAL DATASETS

In this section, we introduce the experimental datasets along with their corresponding ground truths, class names, and the total number of samples in each class. Table I presents the details of each dataset used in the experiments. Here, we emphasize that the number of disjoint training, validation, and test samples, as well as their geographical distributions, are consistent across all methods employed in the experimental evaluation. This ensures unbiased and equitable assessments across the board.

The IEEE Geoscience and Remote Sensing Society published the *University of Houston* (UH) dataset–collected by the Compact Airborne Spectrographic Imager (CASI)–in 2013 as part of its Data Fusion Contest. This dataset is composed of

Fig. 2. Ground truth maps for UH dataset.

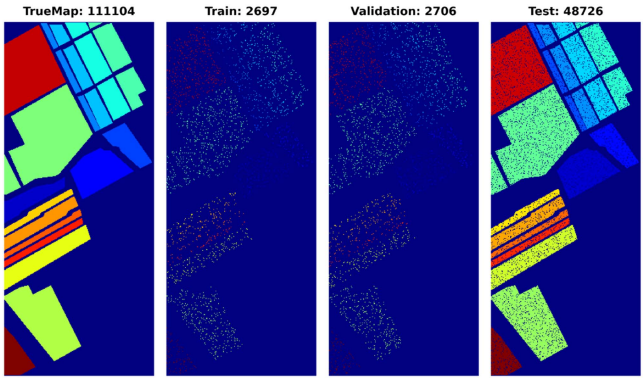

Fig. 3. Ground truth maps for SA dataset.

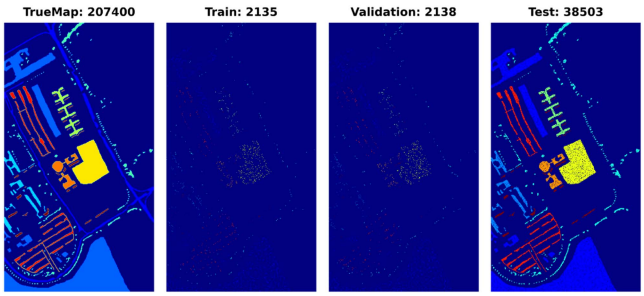

Fig. 4. Ground truth maps for PU dataset.

340 × 1905 pixels with 144 spectral bands. The spatial resolution is 2.5 meters per pixel (MPP), with wavelengths ranging from 0.38 to 1.05 μm. The ground truth comprises 15 different land-cover classes. The ground truth maps are presented in Fig. 2. Originally, the dataset comprised 664 845 samples, as shown on the true map. For training, we utilized a subset of 745 samples, while 751 samples were set aside for validation. The remaining 13 533 samples were used for testing.

Salinas (SA) scene was collected by the 224-band Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor over SA Valley, California, and is characterized by high spatial resolution with 3.7-m pixels. The area covered comprises 512 lines by 217 samples. This image is available only as at-sensor radiance data and includes vegetables, bare soils, and vineyard fields. The SA ground truth contains 16 classes. The ground truth maps are presented in Fig. 3.

Pavia University (PU) is acquired by the reflective optics system imaging spectrometer (ROSIS) sensor during a flight campaign over Pavia, northern Italy. The number of spectral bands is 103 for PU. PU is 610 × 610 pixels. The geometric resolution is 1.3 m. PU image ground truths differentiate nine classes. The ground truth maps are presented in Fig. 4.

TABLE II
PERFORMANCE ON DIFFERENT TRAIN RATIOS ACROSS DIFFERENT DATASETS

Datasets	Metrics	Data split ratio				
		5%	**10%**	**15%**	**20%**	**25%**
PU	OA	96.28	98.33	99.35	98.02	**99.80**
	AA	93.81	97.08	98.83	97.35	**99.63**
	KA	95.07	97.78	99.14	97.39	**99.73**
	F1	94.44	97.33	99.0	96.55	**99.66**
SA	OA	97.53	99.08	99.33	99.75	**99.86**
	AA	98.37	94.42	99.71	99.80	**99.83**
	KA	97.25	98.98	99.27	99.72	**99.84**
	F1	98.43	99.31	99.81	99.87	**100.0**
UH	OA	93.11	97.36	97.13	**98.19**	98.18
	AA	92.11	95.97	96.46	97.28	**98.26**
	KA	92.55	97.14	96.9	**98.05**	98.03
	F1	86.56	90.68	90.56	91.62	**92.25**

The highest accuracy values are highlighted in bold.

TABLE III
PERFORMANCE ON DIFFERENT PATCH SIZES ACROSS DIFFERENT DATASETS

Datasets	Metrics	Different patch size				
		2 × 2	4 × 4	6 × 6	8 × 8	10 × 10
PU	OA	95.88	98.72	**99.45**	99.22	96.28
	AA	94.62	98.61	99.0	**99.25**	95.71
	KA	94.53	98.3	**99.27**	98.97	95.05
	F1	95	98.55	99.0	**99.33**	94.88
SA	OA	94.5	98.53	99.29	98.46	**99.45**
	AA	96.26	99.44	**99.71**	98.81	99.51
	KA	93.88	98.36	99.21	98.28	**99.38**
	F1	95.87	99.5	**99.68**	98.93	**99.68**
UH	OA	88.36	88.24	97.13	**98.19**	90.19
	AA	86.86	84.34	96.46	**97.28**	89.03
	KA	87.41	87.26	96.9	**98.05**	89.03
	F1	81.68	79.33	90.56	**91.62**	83.81

The highest accuracy values are highlighted in bold.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

Random sample selection can introduce variability, potentially causing discrepancies among models executed at different times. Another common issue in recent literature is the overlap of training and test samples, leading to biased models with inflated accuracy. To mitigate this, the PyFormer ensures that while training, validation, and test samples are randomly selected, efforts are made to prevent any overlap between these sets, thereby reducing biases introduced by overlapping samples. In the experimental setup, the proposed PyFormer was assessed using a mini-batch size of 128, the Adam optimizer with a learning rate of 0.0001, and a decay rate of 1e-06 over 50 epochs. We systematically tested various configurations to comprehensively evaluate the proposed model. This exploration aimed to thoroughly understand the model's performance under diverse training scenarios and spatial resolutions. Initially, we examine four critical factors impacting the model's performance: patch sizes and training samples, as shown in Tables II and III, and the Number of heads and layers in the transformer model as shown

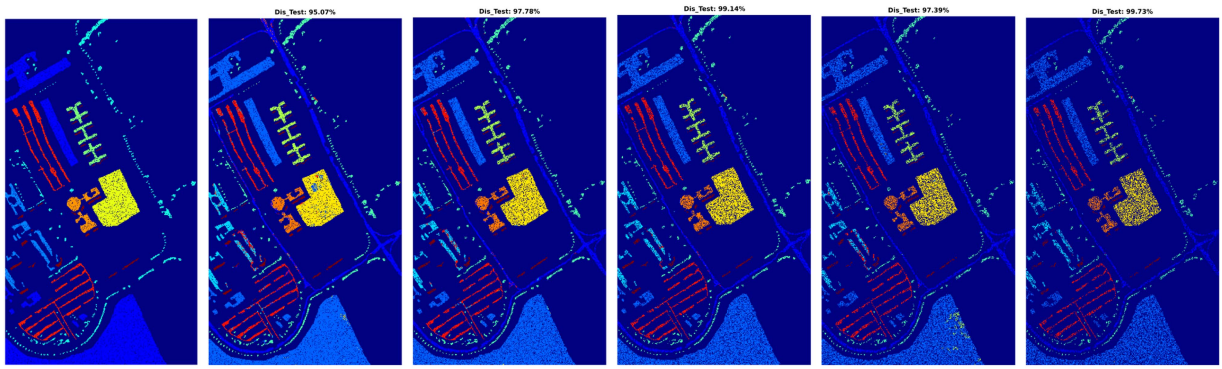

Fig. 5. PyFormer model achieves kappa accuracies of 99.73%, on 25% disjoint test samples for the PU dataset. (a) Disjoint test GTs. (b) 5% training. (c) 10% training. (d) 15% training. (e) 20% training. (f) 25% training.

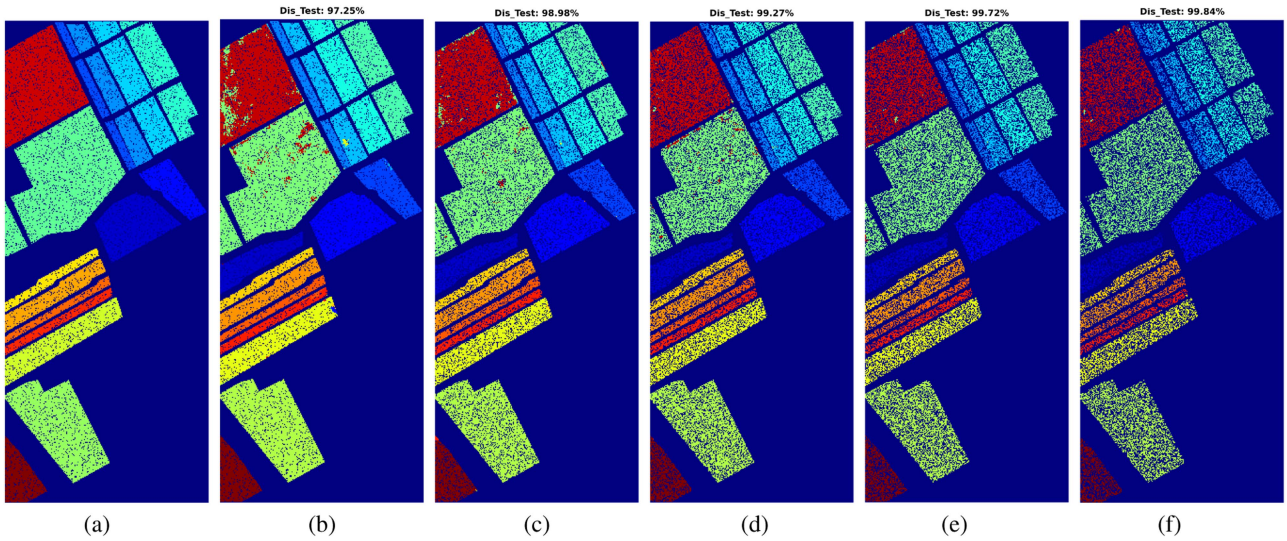

Fig. 6. PyFormer model achieves kappa accuracies of 99.84%, on 25% disjoint test samples for the SA dataset. (a) Disjoint test GTs. (b) 5% training. (c) 10% training. (d) 15% training. (e) 20% training. (f) 20% training.

TABLE IV
PERFORMANCE ON DIFFERENT NUMBER OF HEADS ACROSS DIFFERENT DATASETS

Datasets	Metrics	Different number of heads				
		2	4	6	8	10
PU	OA	99.33	95.93	98.38	99.11	**99.36**
	AA	**98.83**	95.99	96.89	98.42	98.61
	KA	99.12	94.65	97.86	98.82	**99.15**
	F1	**98.88**	95.55	97.4	98.66	98.77
SA	OA	98.67	99.5	99.33	99.33	**99.61**
	AA	98.13	99.61	98.98	99.56	**99.66**
	KA	98.52	99.44	99.25	99.26	**99.57**
	F1	97.62	99.56	99.12	99.56	**99.75**
UH	OA	97.49	97.67	**97.82**	84.47	96.77
	AA	96.8	96.57	**97.32**	85.92	96.39
	KA	97.29	97.48	**97.65**	83.25	96.51
	F1	96.87	97.26	**97.56**	84.4	96.53

The highest accuracy values are highlighted in bold.

TABLE V
PERFORMANCE ON DIFFERENT NUMBER OF LAYERS ACROSS DIFFERENT DATASETS

Datasets	Metrics	Different number of layers				
		2	4	6	8	10
PU	OA	98.79	99.37	**99.44**	99.11	97.94
	AA	98.22	98.83	**99.19**	99.0	95.43
	KA	98.4	99.17	**99.38**	99.26	97.27
	F1	98.11	99	**99.33**	99.22	96.33
SA	OA	**99.47**	99.07	98.89	99.33	99.27
	AA	99.68	99.53	**99.87**	99.24	98.63
	KA	99.42	98.86	**99.65**	98.76	98.07
	F1	99.81	99.68	**99.87**	97.31	98.87
UH	OA	97.69	**98.1**	97.63	97.28	97.81
	AA	96.55	**97.47**	97.03	96.50	97.13
	KA	97.5	**97.95**	97.44	97.06	97.64
	F1	97	**97.75**	97.4	96.6	97.26

The highest accuracy values are highlighted in bold.

in Tables IV and V. These factors are pivotal for performance optimization. Ensuring an adequately sized training set covering diverse spectral signatures and representative samples from each class is crucial.

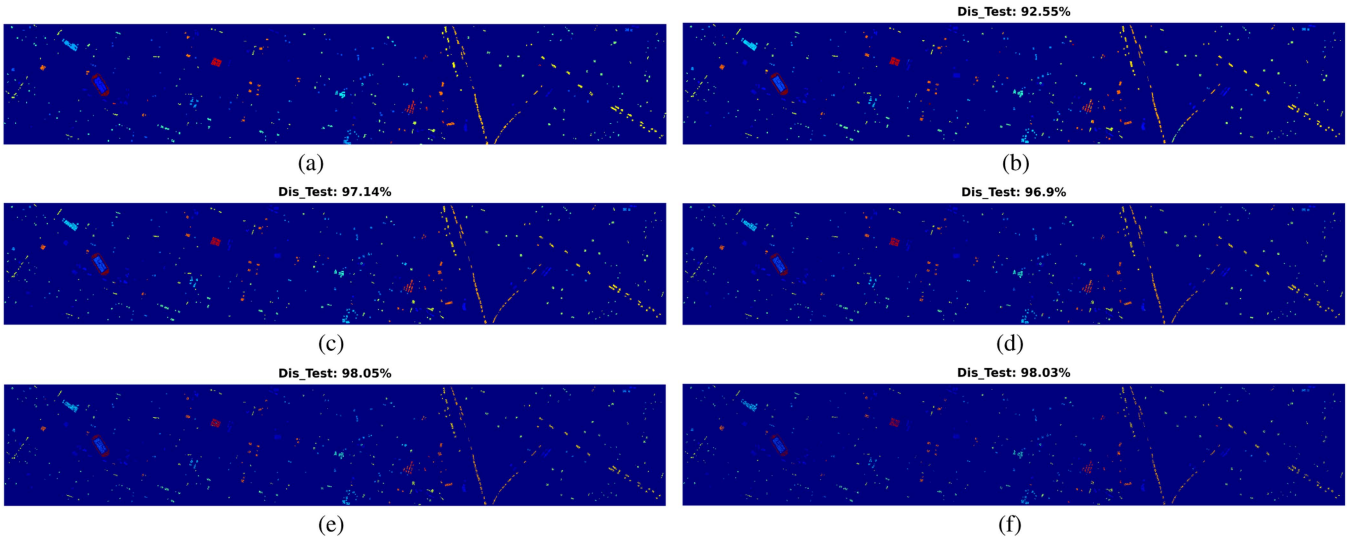

Fig. 7. PyFormer model achieves kappa accuracies of 98.03%, on 25% disjoint test samples for the UH dataset. (a) Disjoint test GTs. (b) 5% training. (c) 10% training. (d) 15% training. (e) 20% training. (f) 20% training.

Table II and Figs. 5–7 highlights the impact of varying training set sizes on performance metrics across the PU, SA, and UH datasets. Beginning with fewer labeled samples allows us to simulate real-world scenarios where data scarcity is common, helping us assess the model's robustness under constrained conditions. As the training set size gradually increases, the model benefits from exposure to a more diverse range of examples, which enhances its learning and generalization capabilities. For instance, the smaller and less complex PU dataset requires fewer training samples to achieve high performance, while the larger and more diverse UH dataset benefits from a higher percentage of training data to capture its variability. This trend is evident in consistently improving metrics such as OA, AA, KA, and F1 scores, particularly in the PU and SA datasets. However, an imbalanced distribution of training samples among classes can bias models toward dominant classes, compromising generalization. Balancing sample distribution across classes is crucial to mitigate biases and enhance the model's generalization ability across all classes.

Moreover, patch size denotes the spatial extent of input patches, crucial for capturing local spatial information and contextual relationships within HSI data as shown in Table III. Table III illustrates the performance of different patch sizes across the PU, SA, and UH datasets. Patch size is crucial as it determines the spatial extent of the input, which is essential for capturing local spatial information and contextual relationships within HSI. A gradual increase in patch size generally leads to improved performance across all metrics, with the optimal patch size varying slightly depending on the dataset. For example, the SA dataset shows the best OA with a 10×10 patch size, while the UH dataset achieves its highest OA with an 8×8 patch size. The choice of an 8×8 patch size strikes a balance between capturing sufficient spatial context and maintaining computational efficiency, making it a consistent choice for both PU and UH datasets. This approach standardizes the comparison while ensuring that the spatial characteristics of each dataset are adequately represented. The analysis underscores the importance of selecting an appropriate patch size to optimize performance while considering the unique attributes of each dataset.

Table IV presents the performance metrics—overall accuracy (OA), average accuracy (AA), Kappa (κ), and F1-score (F1)—for three different datasets. The performance is evaluated across varying numbers of heads (2, 4, 6, 8, and 10) used in the transformer model. The highest OA is 99.36, achieved with 10 heads, indicating that increasing the number of heads generally improves accuracy for the PU dataset. However, two heads also perform exceptionally well with 99.33 OA, suggesting that a simpler model can still achieve high accuracy. The best AA is observed with two heads (98.83), but it drops slightly with four heads (95.99). This indicates that while more heads can capture more features, it might lead to overfitting or increased complexity that does not generalize well. The highest κ is 99.15 with ten heads, mirroring the trend in OA. The best F1 score is 98.88 with two heads, followed closely by ten heads with 98.77. For the SA dataset, The highest OA is 99.61 with ten heads, indicating a clear benefit from increasing heads in this dataset. The best AA is 99.66 with ten heads, supporting the observation that more heads improve performance. The highest κ is 99.57 with ten heads. The best F1-score is 99.75 with ten heads, indicating that the increased heads improve the harmonic mean of precision and recall. For the UH dataset, The highest OA is 97.82 with six heads. Interestingly, eight heads result in a significant drop to 84.47, suggesting potential overfitting or instability with too many heads. The best AA is 97.32 with six heads, again showing that six heads are optimal for this dataset. The highest κ is 97.65 with six heads and the best F1-score is 97.56 with six heads.

Table V presents the performance metrics across varying numbers of layers (2, 4, 6, 8, and 10) used in the transformer model. The highest OA is 99.44 with six layers, indicating that increasing layers to a certain point improves accuracy. Beyond six layers, the performance drops. The best AA is 99.19 with six layers, suggesting that a moderate number of layers provides the

TABLE VI
PU: PYFORMER IS COMPARED AGAINST OTHER SOTA MODELS

Class	SSRN [7]	EMFFN [8]	DBMA [9]	DBDA [10]	SSGC [11]	OSDN [12]	PCIA [16]	PMCN [17]	SF [34]	ViT [35]	WF [19]	CSiT [36]	HiT [54]	**PyFormer**
Asphalt	98.45%	83.27%	95.46%	90.06%	89.36%	87.76%	96.38%	96.27%	92.67%	95.67%	**96.21%**	93.84%	95.21%	95.82%
Meadows	93.07%	89.34%	98.69%	98.99%	99.41%	98.55%	98.55%	98.09%	92.79%	88.37%	99.33%	95.23%	92.54%	**99.47%**
Gravel	52.69%	68.53%	82.96%	93.65%	83.78%	96.33%	95.53%	**98.60%**	90.60%	73.71%	81.31%	88.79%	81.18%	85.61%
Trees	99.77%	87.20%	95.66%	96.45%	97.67%	97.66%	96.93%	97.82%	98.15%	98.03%	96.77%	96.19%	97.21%	**98.79%**
Painted	86.60%	98.15%	99.06%	99.26%	99.49%	99.62%	97.02%	98.72%	98.28%	99.01%	**100%**	99.18%	**100%**	**100%**
Soil	97.83%	89.68%	99.48%	96.95%	99.31%	**99.99%**	97.79%	97.28%	93.29%	89.26%	91.55%	91.99%	91.93%	97.18%
Bitumen	34.07%	73.45%	98.16%	99.91%	**100%**	**100%**	**100%**	90.02%	83.01%	79.40%	89.55%	92.06%	92.75%	92.88%
Bricks	90.55%	75.58%	84.75%	84.49%	87.80%	83.08%	88.37%	88.82%	84.50%	85.54%	89.34%	82.25%	87.59%	**89.97%**
Shadows	98.60%	99.74%	97.83%	95.85%	98.79%	97.37%	98.39%	98.59%	**99.77%**	99.65%	96.47%	99.19%	99.47%	95.61%
OA	84.03%	86.14%	96.01%	95.33%	95.43%	95.12%	95.73%	95.88%	92.30%	89.32%	95.66%	93.35%	91.35%	**96.28%**
AA	83.52%	84.99%	92.67%	92.07%	91.07%	92.71%	93.15%	93.14%	88.86%	87.39%	93.39%	90.48%	85.07%	**93.81%**
κ	78.79%	81.25%	94.70%	93.78%	93.93%	94.82%	94.98%	94.19%	89.66%	86.60%	94.22%	91.13%	88.94%	**95.07%**

All models are evaluated with training/validation/test samples distributed as 5%/5%/90%, respectively.
The highest accuracy values are highlighted in bold.

best generalization. The highest κ is 99.38 with six layers, and the best F1-score is 99.33 with six layers for the PU dataset. The highest OA is 99.47 with two layers, indicating that fewer layers might be more beneficial for this dataset. The best AA is 99.87 with six layers, showing that for some metrics, more layers can still be advantageous. The highest κ is 99.65 with six layers, and the best F1-score is 99.87 with six layers for the SA dataset. Similarly, the highest OA is 98.10 with four layers, suggesting that an intermediate number of layers is optimal. The best AA is 97.47 with four layers. The highest κ is 97.95 with four layers, and the best F1-score is 97.75 with four layers.

The UH dataset, as depicted in the true map of Fig. 2, originally comprised 664 845 samples. For our experiments, we selected a subset of 745 samples for training, allocated 751 samples for validation, and reserved the remaining 13 533 samples for testing. Similarly, the SA dataset, illustrated in Fig. 2, consisted of 111 104 samples in the true map. From this dataset, we used 2497 samples for training, 2706 samples for validation, and the remaining 48 726 samples for testing. In addition, the ground truth maps for the PU dataset, shown in Fig. 4, initially contained 207 400 samples. For this dataset, we utilized 2135 samples for training, set aside 2138 samples for validation, and designated 38 503 samples for testing.

The comprehensive evaluation across Tables II–V reveals key insights into optimizing model performance. Table II demonstrates that increasing the training set size consistently enhances model performance metrics (OA, AA, KA, F1 Score), emphasizing the importance of adequate labeled data for robust generalization and bias mitigation. Table III indicates that larger patch sizes generally improve performance, with the 8×8 patch size being optimal for balancing spatial context and computational efficiency across datasets. Table IV shows that increasing the number of attention heads generally improves performance, with ten heads being optimal for PU and SA datasets, while six heads are best for UH, suggesting variability in effectiveness based on dataset complexity. Table V reveals that while increasing the number of layers enhances performance up to a point, the optimal number varies by dataset, with six layers being ideal for PU and SA and eight layers for UH. These findings underscore the importance of tailoring training data size, patch size, attention heads, and network depth to the specific characteristics of each dataset for achieving optimal model performance.

Maintaining a consistent experimental methodology is essential when evaluating CNN and transformer approaches. Consistency in the distribution of samples for training, validation, and testing is crucial. Each comparative model was trained and validated using 5% of the samples, with the remaining samples utilized for classification using 8×8 pixel patches. The performance of PyFormer was assessed using the UH and PU datasets, comparing it against several models: SpectralFormer (SF): rethinking HSIC with transformers [34], Attention is all you need (ViT) [35], WaveFormer (WF): spectral–spatial wavelet transformer for HSIC [19], CSiT: a multiscale vision transformer for HSIC (CSiT) [36], hyperspectral image transformer (HiT) classification networks (HiT) [54], Spectral–Spatial Residual Network (SSRN) for HSIC: A 3-D deep learning framework [7], enhanced multiscale feature fusion network (EMFFN) for HSI classification [8], double-branch multiattention (DBMA) mechanism network for HSIC [9], classification of hyperspectral image based on double-branch dual-attention (DBDA) mechanism network [10], spectral and spatial global context (SSGC) attention for HSIC [11], one-shot dense network (OSDN) with polarized attention for HSIC [12], double-branch network with pyramidal convolution and iterative attention (PCIA) for HSIC [16], and pyramidal multiscale convolutional network (PMCN) with polarized self-attention for pixel-wise HSIC [17].

The detailed results of the aforementioned models can be found in Tables VI and VII. In summary, the proposed PyFormer model exhibits outstanding performance, surpassing state-of-the-art (SOTA) ViT-based models across various evaluation metrics, including OA, AA, and κ coefficient. A comprehensive analysis of the quantitative results indicates that PyFormer consistently achieves superior performance across different categories, demonstrating significant improvements in accuracy, as illustrated in the Tables. Notably, while the performance gaps are relatively small in the PU dataset due to the abundance of samples, the UH dataset presents a considerable challenge for modeling. For instance, when evaluating the challenging UH dataset, PyFormer outperforms the baseline ViT by more than 7% and exceeds SF by approximately 4%. Moreover, the AA achieved by PyFormer surpasses that of both ViT and SpectralFormer by margins of around 5%, highlighting the potential effectiveness of spatial-spectral feature extraction. In comparison with the most recent SST and CSiT models, PyFormer consistently delivers promising results, demonstrating its proficiency in both spectral and spectral-spatial feature extraction tasks. It is noteworthy that while HiT excels in identifying land-cover classes with spectral-spatial information, PyFormer approaches similar levels of performance. In conclusion, these findings underscore the robustness and effectiveness of the PyFormer, particularly in scenarios where the extraction of spatial-spectral information is

TABLE VII
UH: PyFormer is Compared Against Other SOTA Models

Class	SSRN [7]	EMFFN [8]	DBMA [9]	DBDA [10]	SSGC [11]	OSDN [12]	PCIA [16]	PMCN [17]	SF [34]	ViT [35]	WF [19]	CSiT [36]	HiT [54]	PyFormer
Healthy grass	68.37%	85.97%	89.97%	86.73%	80.15%	87.60%	86.54%	88.11%	93.31%	90.28%	98.90%	93.39%	97.26%	**98.98%**
Stressed grass	93.18%	92.52%	87.37%	93.37%	93.95%	84.16%	87.14%	89.46%	97.81%	98.21%	97.60%	99.54%	97.29%	**99.63%**
Synthetic grass	56.72%	99.85%	**100%**	**100%**	**100%**	**100%**	99.77%	97.74%	**100%**	96.90%	99.82%	**100%**	98.74%	99.71%
Trees	68.80%	93.06%	81.08%	80.91%	78.64%	84.61%	91.93%	97.07%	**100%**	**100%**	99.19%	98.09%	95.78%	99.48%
Soil	92.03%	89.64%	99.95%	94.92%	96.08%	93.92%	92.12%	92.32%	98.16%	96.89%	99.79%	97.70%	98.41%	**100%**
Water	87.28%	98.58%	92.14%	**100%**	**100%**	**100%**	99.93%	96.36%	**100%**	98.21%	98.07%	**100%**	91.36%	96.74%
Residential	58.65%	82.19%	84.98%	77.69%	69.93%	95.93%	68.85%	78.32%	87.83%	82.82%	90.64%	90.96%	94.60%	**97.35%**
Commercial	62.56%	82.36%	99.47%	82.52%	89.35%	91.29%	**99.60%**	96.92%	85.91%	79.83%	97.38%	89.18%	91.82%	96.83%
Road	61.76%	55.76%	77.92%	75.61%	80.44%	73.43%	84.65%	81.93%	75.33%	76.88%	**97.40%**	90.62%	92.39%	97.36%
Highway	51.39%	52.68%	73.90%	88.89%	70.70%	73.32%	87.21%	89.06%	82.52%	80.31%	97.75%	93.22%	90.61%	**99.18%**
Railway	**98.93%**	70.05%	75.93%	83.62%	89.75%	81.13%	89.38%	81.74%	79.19%	83.17%	96.76%	87.91%	89.09%	98.84%
Parking Lot 1	47.89%	45.37%	79.23%	71.93%	71.61%	87.31%	66.26%	85.04%	72.76%	69.42%	**97.56%**	83.15%	94.18%	95.72
Parking Lot 2	95.40%	61.52%	93.31%	84.00%	**93.86%**	84.55%	95.89%	75.97%	79.49%	63.08%	65.33%	84.11%	82.51%	78.39%
Tennis Court	94.74%	92.79%	**100%**	**100%**	97.00%	92.92%	90.49%	90.41%	93.90%	90.36%	98.83%	97.21%	91.55%	**100%**
Running Track	85.04%	99.63%	90.61%	94.32%	96.26%	78.07%	94.38%	95.07%	97.50%	94.40%	99.05%	**100%**	96.72%	**100%**
OA	66.32%	76.37%	85.59%	85.42%	83.33%	85.19%	85.68%	87.98%	88.45%	86.45%	96.54%	93.09%	93.06%	**97.36%**
AA	74.85%	80.14%	88.36%	87.63%	87.18%	87.22%	88.96%	89.03%	87.81%	86.60%	95.60%	92.06%	86.61%	**95.97%**
κ	63.51%	74.43%	84.41%	84.23%	81.96%	83.99%	84.51%	87.01%	87.50%	85.35%	96.26%	92.53%	92.50%	**97.14%**

All models are evaluated with training/validation/test samples distributed as 10%/10%/90%, respectively.

crucial, especially considering the limited availability of training samples.

V. Conclusion

In this article, we introduced PyFormer, a novel approach that leverages the strengths of Pyramid and SST for HSIC. By extracting multiscale spatial-spectral features using Pyramid and integrating them into a transformer encoder, PyFormer can effectively capture both local texture patterns and global contextual relationships within a single, end-to-end trainable model. A key innovation is the incorporation of Pyramid convolutions within the transformer's attention mechanism, facilitating enhanced integration of spectral and structural information. Extensive experiments demonstrate that PyFormer achieves SOTA performance, particularly on challenging datasets with limited training data. In addition to superior classification accuracy, PyFormer exhibits robustness and generalizability, showing promise for addressing real-world problems. Future research could explore techniques, such as self-supervised pretraining and network optimizations, to further enhance PyFormer's performance, especially in scenarios with limited data availability.

Acknowledgment

The authors extend their appreciation to the Researchers Supporting Project number (RSPD2024R848), King Saud University, Riyadh, Saudi Arabia.

References

[1] M. Ahmad et al., "Hyperspectral image classification–traditional to deep models: A survey for future prospects," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 968–999, Dec. 2021.

[2] D. Hong et al., "SpectralGPT: Spectral remote sensing foundation model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 8, pp. 5227–5244, Aug. 2024, doi: 10.1109/TPAMI.2024.3362475.

[3] M. Ahmad, S. Distifano, M. Mazzara, and A. M. Khan, "Traditional to transformers: A survey on current trends and future prospects for hyperspectral image classification," 2024, *arXiv:2404.14955*.

[4] U. Ghous, M. S. Sarfraz, M. Ahmad, C. Li, and D. Hong, "EXNet: (2+1)D extreme xception net for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 5159–5172, Feb. 2024.

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

[6] A. Jamali, S. K. Roy, D. Hong, P. M. Atkinson, and P. Ghamisi, "Attention graph convolutional network for disjoint hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 21, Jan. 2024, Art. no. 5503005.

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

[8] J. Yang, C. Wu, B. Du, and L. Zhang, "Enhanced multiscale feature fusion network for HSI classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 12, pp. 10328–10347, Dec. 2021.

[9] W. Ma, Q. Yang, Y. Wu, W. Zhao, and X. Zhang, "Double-branch multi-attention mechanism network for hyperspectral image classification," *Remote Sens.*, vol. 11, no. 11, 2019, Art. no. 1307.

[10] R. Li, S. Zheng, C. Duan, Y. Yang, and X. Wang, "Classification of hyperspectral image based on double-branch dual-attention mechanism network," *Remote Sens.*, vol. 12, no. 3, 2020, Art. no. 582.

[11] Z. Li, X. Cui, L. Wang, H. Zhang, X. Zhu, and Y. Zhang, "Spectral and spatial global context attention for hyperspectral image classification," *Remote Sens.*, vol. 13, no. 4, 2021, Art. no. 771.

[12] H. Pan, M. Liu, H. Ge, and L. Wang, "One-shot dense network with polarized attention for hyperspectral image classification," *Remote Sens.*, vol. 14, no. 9, 2022, Art. no. 2265.

[13] C.-I. Chang, C.-C. Liang, and P. F. Hu, "Iterative Gaussian–Laplacian pyramid network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, Feb. 2024, Art. no. 5510122.

[14] Y. Tang, X. Xie, and Y. Yu, "Hyperspectral classification of two-branch joint networks based on gaussian pyramid multiscale and wavelet transform," *IEEE Access*, vol. 10, pp. 56876–56887, 2022.

[15] B. Tu, X. Liao, Q. Li, C. Zhou, and A. Plaza, "Multi-resolution pyramid enhanced non-local feature extraction for hyperspectral classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 5865–5879, Jul. 2022.

[16] H. Shi, G. Cao, Z. Ge, Y. Zhang, and P. Fu, "Double-branch network with pyramidal convolution and iterative attention for hyperspectral image classification," *Remote Sens.*, vol. 13, no. 7, 2021, Art. no. 1403.

[17] H. Ge et al., "Pyramidal multiscale convolutional network with polarized self-attention for pixel-wise hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Feb. 2023, Art. no. 5504018.

[18] J. Yao, B. Zhang, C. Li, D. Hong, and J. Chanussot, "Extended vision transformer (ExViT) for land use and land cover classification: A multimodal deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Jun. 2023, Art. no. 5514415.

[19] M. Ahmad, U. Ghous, M. Usama, and M. Mazzara, "WaveFormer: Spectral–spatial wavelet transformer for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 21, Jan. 2023, Art. no. 5502405.

[20] X. Huang, M. Dong, J. Li, and X. Guo, "A 3-D-swin transformer-based hierarchical contrastive learning method for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Aug. 2022, Art. no. 5411415.

[21] M. Ahmad, M. Usama, A. M. Khan, S. Distefano, H. A. Altuwaijri, and M. Mazzara, "Spatial spectral transformer with conditional position encoding for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 21, Jul. 2024, Art. no. 5508205.

[22] Y. Que, H. Xiong, X. Xia, J. You, and Y. Yang, "Integrating spectral and spatial bilateral pyramid networks for pansharpening," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 3985–3998, Jan. 2024.

[23] S. Karim, G. Tong, J. Li, A. Qadir, U. Farooq, and Y. Yu, "Current advances and future perspectives of image fusion: A comprehensive review," *Inf. Fusion*, vol. 90, pp. 185–217, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253522001518

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

[25] L. Wu and H. Wang, "Global and pyramid convolutional neural network with hybrid attention mechanism for hyperspectral image classification," *Geocarto Int.*, vol. 38, pp. 1–24, 2023.

[26] L. Wang, Z. Zheng, N. Kumar, C. Wang, F. Guo, and P. Zhang, "Multilevel class token transformer with cross tokenmixer for hyperspectral images classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, Feb. 2024, Art. no. 5507913.

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

[28] G. Wang, Y. Wang, Z. Pan, X. Wang, J. Zhang, and J. Pan, "Vitfsl-baseline: A simple baseline of vision transformer network for few-shot image classification," *IEEE Access*, vol. 12, pp. 11836–11849, 2024.

[29] Y. Ma et al., "A spatial-spectral transformer for hyperspectral image classification based on global dependencies of multi-scale features," *Remote Sens.*, vol. 16, no. 2, 2024, Art. no. 404.

[30] J. Lian, L. Wang, H. Sun, and H. Huang, "GT-HAD: Gated transformer for hyperspectral anomaly detection," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Feb. 12, 2024, doi: 10.1109/TNNLS.2024.3355166.

[31] S. Mei, Z. Han, M. Ma, F. Xu, and X. Li, "A novel center-boundary metric loss to learn discriminative features for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, Feb. 2024, Art. no. 5508416.

[32] A. Jamali, S. K. Roy, D. Hong, P. M. Atkinson, and P. Ghamisi, "Spatial-gated multilayer perceptron for land use and land cover mapping," *IEEE Geosci. Remote Sens. Lett.*, vol. 21, Jan. 2024, Art. no. 5502105.

[33] Y. Xiao, Q. Yuan, K. Jiang, J. He, C.-W. Lin, and L. Zhang, "TTST: A top-k token selective transformer for remote sensing image super-resolution," *IEEE Trans. Image Process.*, vol. 33, pp. 738–752, Jan. 2024.

[34] D. Hong et al., "SpectralFormer: Rethinking hyperspectral image classification with transformers," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Nov. 2021, Art. no. 5518615.

[35] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[36] W. He, W. Huang, S. Liao, Z. Xu, and J. Yan, "CSiT: A multiscale vision transformer for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 9266–9277, Oct. 2022.

[37] W. Zhang et al., "Attention-aware dynamic self-aggregation network for satellite image time series classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Oct. 2021, Art. no. 4406517.

[38] T. Arshad and J. Zhang, "Hierarchical attention transformer for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 21, Mar. 2024, Art. no. 5504605.

[39] S. Liu et al., "A shallow-to-deep feature fusion network for VHR remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Mar. 2022, Art. no. 5410213.

[40] A. A. Aleissaee et al., "Transformers in remote sensing: A survey," *Remote Sens.*, vol. 15, 2022, Art. no. 1860.

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

[42] M. H. F. Butt, J. P. Li, M. Ahmad, and M. A. F. Butt, "Graph-infused hybrid vision transformer: Advancing GEoAI for enhanced land cover classification," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 129, 2024, Art. no. 103773. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569843224001274

[43] J. Chen et al., "TCCU-Net: Transformer and CNN collaborative unmixing network for hyperspectral image," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 8073–8089, Jan. 2024.

[44] Y. Liu, J. Hu, X. Kang, J. Luo, and S. Fan, "Interactformer: Interactive transformer and CNN for hyperspectral image super-resolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Jun. 2022, Art. no. 5531715.

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

[46] X. Tan, K. Gao, B. Liu, Y. Fu, and L. Kang, "Deep global-local transformer network combined with extended morphological profiles for hyperspectral image classification," *J. Appl. Remote Sens.*, vol. 15, no. 3, 2021, Art. no. 038509, doi: 10.1117/1.JRS.15.038509.

[47] P. Tang, M. Zhang, Z. Liu, and R. Song, "Double attention transformer for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, Feb. 2023, Art. no. 5502105.

[48] Y. Ma et al., "A spatial–spectral transformer for hyperspectral image classification based on global dependencies of multi-scale features," *Remote Sens.*, vol. 16, no. 2, 2024, Art. no. 404.

[49] R. Song, Y. Feng, W. Cheng, Z. Mu, and X. Wang, "BS2T: Bottleneck spatial–spectral transformer for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Jun. 2022, Art. no. 5532117.

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

[51] L. Sun, G. Zhao, Y. Zheng, and Z. Wu, "Spectral–spatial feature tokenization transformer for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Jan. 2022, Art. no. 5522214.

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

[53] Z. Zhao, D. Hu, H. Wang, and X. Yu, "Convolutional transformer network for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, Apr. 2022, Art. no. 6009005.

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

[55] B. Li, E. Ouyang, W. Hu, G. Zhang, L. Zhao, and J. Wu, "Multi-granularity vision transformer via semantic token for hyperspectral image classification," *Int. J. Remote Sens.*, vol. 43, no. 17, pp. 6538–6560, 2022. [Online]. Available: https://doi.org/10.1080/01431161.2022.2142078

[56] 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 Trans. Geosci. Remote Sens.*, vol. 61, Feb. 2023, Art. no. 4401118.

[57] J. Fang, J. Yang, A. Khader, and L. Xiao, "MIMO-SST: Multi-input multi-output spatial-spectral transformer for hyperspectral and multispectral image fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, Feb. 2024, Art. no. 5510020.

[58] M. Ye, J. Chen, F. Xiong, and Y. Qian, "Adaptive graph modeling with self-training for heterogeneous cross-scene hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, Jan. 2024, Art. no. 5503815.

[59] Y. Sun et al., "Dual spatial–spectral pyramid network with transformer for hyperspectral image fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Sep. 2023, Art. no. 5526016.

[60] W. Huang, Y. Deng, S. Hui, Y. Wu, S. Zhou, and J. Wang, "Sparse self-attention transformer for image inpainting," *Pattern Recognit.*, vol. 145, 2024, Art. no. 109897.

[61] S.-Q. Deng, L.-J. Deng, X. Wu, R. Ran, D. Hong, and G. Vivone, "PSRT: Pyramid shuffle-and-reshuffle transformer for multispectral and hyperspectral image fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Feb. 2023, Art. no. 5503715.

[62] Y. Sun, X. Zhi, S. Jiang, G. Fan, X. Yan, and W. Zhang, "Image fusion for the novelty rotating synthetic aperture system based on vision transformer," *Inf. Fusion*, vol. 104, 2024, Art. no. 102163.

[63] T. Kim, J. Kim, H. Oh, and J. Kang, "Deep transformer based video inpainting using fast fourier tokenization," *IEEE Access*, vol. 12, pp. 21723–21736, 2024.

[64] Y. Shi, J. Xia, M. Zhou, and Z. Cao, "A dual-feature-based adaptive shared transformer network for image captioning," *IEEE Trans. Instrum. Meas.*, vol. 73, Jan. 2024, Art. no. 5009613.

[65] J. Zhang, Y. Zhang, and Y. Zhou, "Quantum-inspired spectral-spatial pyramid network for hyperspectral image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2023, pp. 9925–9934.