Pyramid Hierarchical Transformer for Hyperspectral Image Classification

Muhammad Ahmad, Muhammad Hassaan Farooq Butt, Manuel Mazzara, Salvatore Distifano

***Abstract*—The traditional Transformer model encounters challenges with variable-length input sequences, particularly in Hyperspectral Image Classification (HSIC), leading to efficiency and scalability concerns. To overcome this, we propose a pyramid-based hierarchical transformer (PyFormer). This innovative approach organizes input data hierarchically into segments, each representing distinct abstraction levels, thereby enhancing processing efficiency for lengthy sequences. At each level, a dedicated transformer module is applied, effectively capturing both local and global context. Spatial and spectral information flow within the hierarchy facilitates communication and abstraction propagation. Integration of outputs from different levels culminates in the final input representation. Experimental results underscore the superiority of the proposed method over traditional approaches. Additionally, the incorporation of disjoint samples augments robustness and reliability, thereby highlighting the potential of our approach in advancing HSIC. The source code is available at https://github.com/mahmad00/PyFormer.**

***Index Terms*—Pyramid Network; Hierarchical Transformer; Hyperspectral Image Classification.**

I. Introduction

HYPERSPECTRAL IMAGE CLASSIFICATION (HSIC) is crucial in diverse domains, including remote sensing [1], earth observation [2], urban planning [3], agriculture [4], forestry [5], target/object detection [6], mineral exploration [7], environmental monitoring [8], [9], climate change [10], food processing [11], [12], bakery products [13], bloodstain identification [14], [15], and meat processing [16], [17]. The abundance of spectral data in Hyperspectral Images (HSIs) presents challenges and opportunities for classification [18]. While CNNs [19], [20] and its variants [21]–[23], and Transformers [24]–[26] have shown success in computer vision tasks, there is a growing interest in exploring Transformer models for advancing HSI analysis.

The vision and spatial-spectral transformers (SSTs) [27]–[34] excel in capturing global contextual information via self-attention mechanisms, facilitating simultaneous consideration of relationships between all HSI regions [35]. Unlike CNNs, SSTs demonstrate strong scalability to high-resolution HSIs, effectively handling large datasets without complex pooling operations. Their adaptability contributes to widespread applicability in HSIC. Furthermore, SSTs learn stratified representations directly from raw pixel values, simplifying the model-building process and often leading to improved performance [36]. The interpretability of attention maps generated by SSTs aids in understanding model decision-making, highlighting influential image regions [37].

Despite their success, SSTs have limitations, for instance, training large SSTs can be computationally demanding [38], [39]. The self-attention mechanism introduces quadratic complexity with respect to sequence length, potentially hindering scalability [40]. Unlike CNNs, which inherently possess translation invariance, SSTs may struggle to capture spatial relationships invariant to small translations in the input [41]. Moreover, the tokenization process of dividing input images into fixed-size patches may not efficiently capture fine-grained details [42], [43]. The quadratic scaling of self-attention poses challenges, particularly with long sequences. Furthermore, optimal performance often requires substantial training data, and training on smaller datasets may lead to overfitting, limiting effectiveness in scenarios with limited labeled data [25].

Therefore, this work proposed a pyramid 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 multi-level processing, ensuring efficient capture of local and global context. Information flow occurs both spatially and spectrally within the hierarchy, fostering communication and abstraction propagation. Integration of Transformer outputs from different levels yields the final output maps. In short, the following contributions are made;

1) **Hierarchical Segmentation:** Input sequences are divided into hierarchical segments, each representing varying levels of abstraction or granularity.
2) **Pyramid Organization:** These segments adopt a pyramid-like structure, wherein the lowest level retains detailed information while higher levels convey increasingly abstract representations.
3) **Multi-level Processing:** Transformer modules are independently applied at each level of the hierarchy, facilitating efficient capture of both local and global context.

II. Proposed Methodology

An HSI cube, denoted as $X = \{x_i, y_i\} \in \mathcal{R}^{(M \times N \times B)}$, comprises spectral vectors $x_i = \{x_{i,1}, x_{i,2}, x_{i,3}, \ldots, x_{i,L}\}$,

M. Ahmad is with the Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad, Chiniot-Faisalabad Campus, Chiniot 35400, Pakistan, and Dipartimento di Matematica e Informatica—MIFT, University of Messina, Messina 98121, Italy; (e-mail: mahmad00@gmail.com)

M. H. F. Butt is with the Institute of Artificial Intelligence, School of Mechanical and Electrical Engineering, Shaoxing University, Shaoxing 312000, China. (e-mail: hassaanbutt67@gmail.com

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

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

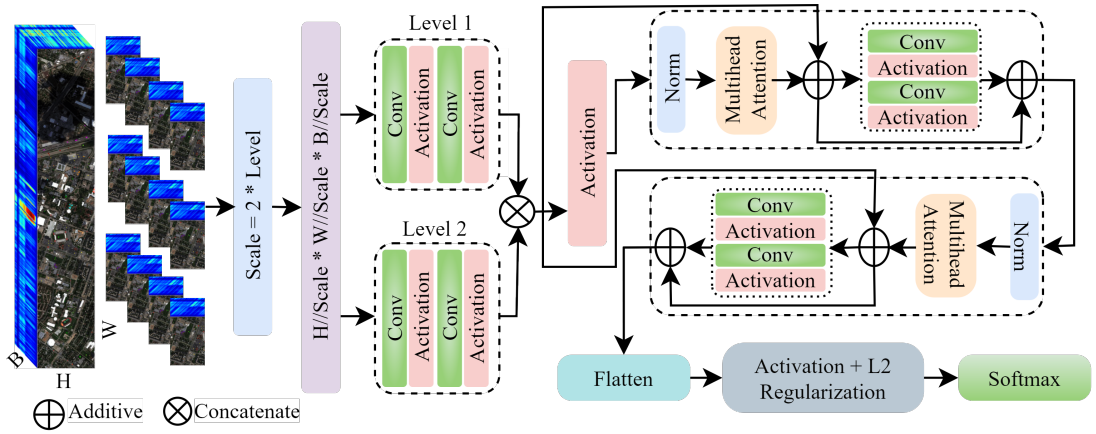


Fig. 1: The Pyramid-type Hierarchical Transformer features a structured pyramid block comprising two levels. Each level incorporates two convolutional layers with filter sizes of $(32 \times S \times S \times B)$ and $(64 \times S \times S \times B)$, respectively, followed by a residual connection and activation function. The hierarchical transformer block, receiving the learned multi-scale information, consists of two layers and four multi-heads. 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.

and y_i and corresponding class labels x_i. The cube is initially divided into overlapping 3D 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 $\beta \pm \frac{S-1}{2}$. These patches, along with their central pixel labels, constitute the training X_{train}, validation X_{val}, and a test X_{test} sets, ensuring $|X_{train}| \ll |X_{val}|$ and $|X_{train}| \ll |X_{test}|$ to prevent sample overlaps and biases. The model's overall structure is presented in Figure 1.

Let (S, S, B^*) denote the input shape, B^* where represents the reduced number of bands. The scaling factor, $Scale = 2^{Level}$, is computed. This scale is utilized to derive the input shape for the pyramid structure layers, given by $Input\ shape = (\frac{S}{Scale}, \frac{S}{Scale}, \frac{B^*}{Scale})$. This input is fed into a convolutional layer to extract spatial-spectral semantic features from HSI patches. Each patch, with dimensions $(S \times S \times B^*)$, undergoes processing using 3D convolutional layers with kernel sizes $(32 \times 1 \times S \times S)$ and $(64 \times 1 \times S \times S)$, along with ReLU activation. The activation maps for the spatial-spectral position (x, y, z) at the i-th feature map and j-th layer are denoted as $f_{i,j}^{(x,y,z)}$ [19];

$$f_{(i,j)}(x,y,z) = \sigma\Bigg(\sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{p=1}^{P} w_{(i,j,m,n,p)} \times x_{(m+x-1,n+y-1,p+z-1)+b_{i,j}}\Bigg) \quad (1)$$

Where $f_{i,j}(x, y, z)$ is the activation value, σ is the activation function, $w_{i,j,m,n,p}$ are the weights of the convolutional kernel, $x_{m+x-1,n+y-1,p+z-1}$ represents the input patch values, and $b_{i,j}$ is the bias term. Let $f_{i,j} \in \mathbb{R}^{N \times D}$ denote the input tensor to the transformer, where N represents the number of patches, and D signifies the dimensionality of each patch after the convolutional process. 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 feedforward convolutional networks. The attention mechanism assumes a critical role in enabling the model to capture intricate relationships between distinct patches. Specifically, for a given input $v_{i,j}$ each layer within the transformer encoder encompasses;

$$H_{att}^{l} = Attention(H^{(l-1)}) \quad (2)$$

$$H_{FF\_Conv}^{l} = FF\_Conv(H_{att}^{l}) \quad (3)$$

$$H^{l} = H^{(l-1)} + H_{FF\_Conv}^{l} \quad (4)$$

Equations 2, 3, and 4 define the attention, feedforward convolutional neural network, and residual connections within the transformer layers, respectively. Later, H^l is flattened as $W = reshape(H^l, (m \times n \times p, 1))$, where m, n and p represent the batch size, height, and width, respectively. Subsequently, L_2 regularization $L_2Reg = \lambda|W|_2^2$ 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.

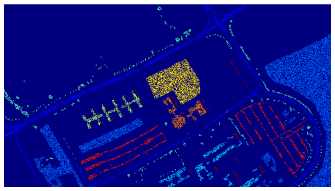
III. Experimental Results and Discussion

To ensure fairness, maintaining identical experimental conditions is crucial, including consistent geographical locations for model selection and uniform sample numbers for each

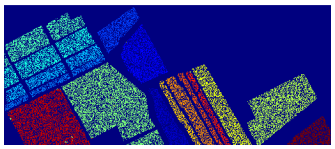
training round in cross-validation. Random sample selection can introduce variability, potentially causing discrepancies among models executed at different times. Another common issue in recent literature is the overlap of training and test samples, leading to biased models with inflated accuracy. To mitigate this, the proposed method 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 our experimental setup, the proposed PyFormer was assessed using a mini-batch size of 128, the Adam optimizer, and specific learning parameters i.e., 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.

TABLE I: Performance on different train ratios across different datasets. Figure 2 illustrates the geographical maps showcasing the best results attained in these experiments.

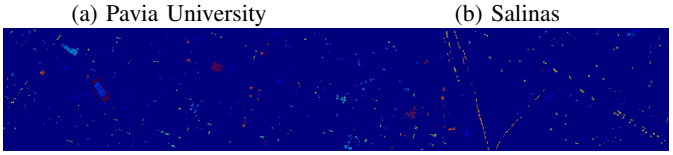
Datasets	Metrics	Data Split Ratio				
		5%	10%	15%	20%	25%
PU	OA	96.28	98.33	99.35	98.02	**99.80**
	AA	93.81	97.08	98.83	97.35	**99.63**
	KA	95.07	97.78	99.14	97.39	**99.73**
	F1	94.44	97.33	99.0	96.55	**99.66**
SA	OA	97.53	99.08	99.33	99.75	**99.86**
	AA	98.37	94.42	99.71	99.80	**99.83**
	KA	97.25	98.98	99.27	99.72	**99.84**
	F1	98.43	99.31	99.81	99.87	**100.0**
HU	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**



(a) Pavia University



(b) Salinas



(c) University of Houston

Fig. 2: The PyFormer model achieves kappa accuracies of 99.73%, 99.84%, and 98.01% on the Disjoint Training, Validation, and Test samples for the Pavia University (PU), Salinas (SA), and University of Houston (UH) datasets, respectively.

Initially, we examine four critical factors impacting the model's performance: patch sizes and training samples as shown in Tables I and II, and the Number of heads and layers in the Transformer model as shown in Tables III and IV. These factors are pivotal for performance optimization. Ensuring an adequately sized training set covering diverse

TABLE II: 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**
HU	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

TABLE III: 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**
HU	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

TABLE IV: 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
HU	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

spectral signatures and representative samples from each class is crucial. Increasing the number of labeled training samples offers the model more diverse examples, enhancing learning and generalization capabilities as shown in Table I. Sufficient samples enhance model performance, especially in accommodating variations within classes. An imbalanced distribution of training samples among classes can bias models towards 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. The choice of patch size significantly impacts the model's ability to capture spatial details and contextual dependencies. Larger patch

sizes enhance the model's understanding of global spatial relationships, facilitating improved spatial feature extraction and incorporation of broader spectral information. Conversely, smaller patch sizes focus on local details, advantageous for intricate patterns or objects with distinct spatial characteristics and spectral variations as shown in Table II.

Tables V and VI provide a summary of OA, AA, and kappa, along with the average for each class. The best performance in each row is highlighted in bold. For comparison, we have chosen several SOTA models, Transformer [44], Spectralformer [45], HiT [46], CSiT [47], and WaveFormer (WF) [25]. The transformer architecture utilizes a standard ViT [48] that has been adapted for pixel-wise HSI input, consisting of five encoder blocks. CSiT is evaluated without a Cross-Spectral Attention Fusion module. All results are obtained using the specified parameter configurations from the original papers to facilitate direct comparison.

TABLE V: **Pavia University:** PyFormer is compared against other SOTA models. All models are evaluated with training/validation/test samples distributed as 5%/5%/90%, respectively.

Class	SF [45]	ViT [44]	WF [25]	CSiT [47]	HiT [46]	**PyFormer**
Asphalt	92.67%	95.67%	**96.21%**	93.84%	95.21%	95.82%
Meadows	92.79%	88.37%	99.33%	95.23%	92.54%	**99.47%**
Gravel	**90.60%**	73.71%	81.31%	88.79%	81.18%	85.61%
Trees	98.15%	98.03%	96.77%	96.19%	97.21%	**98.79%**
Painted	98.28%	99.01%	**100%**	99.18%	**100%**	**100%**
Soil	93.29%	89.26%	91.55%	91.99%	91.93%	**97.18%**
Bitumen	83.01%	79.40%	89.55%	92.06%	92.75%	**92.88%**
Bricks	84.50%	85.54%	89.34%	82.25%	87.59%	**89.97%**
Shadows	**99.77%**	99.65%	96.47%	99.19%	99.47%	95.61%
OA	92.30%	89.32%	95.66%	93.35%	91.35%	**96.28%**
AA	88.86%	87.39%	93.39%	90.48%	85.07%	**93.81%**
κ	89.66%	86.60%	94.22%	91.13%	88.94%	**95.07%**

TABLE VI: **University of Houston:** PyFormer is compared against other SOTA models. All models are evaluated with training/validation/test samples distributed as 10%/10%/90%, respectively.

Class	SF [45]	ViT [44]	WF [25]	CSiT [47]	HiT [46]	**PyFormer**
Healthy grass	93.31%	90.28%	98.90%	93.39%	97.26%	**98.98%**
Stressed grass	97.81%	98.21%	97.60%	99.54%	97.29%	**99.63%**
Synthetic grass	**100%**	96.90%	99.82%	**100%**	98.74%	99.71%
Trees	**100%**	100%	99.19%	98.09%	95.78%	99.48%
Soil	98.16%	96.89%	99.79%	97.70%	98.41%	**100%**
Water	100%	98.21%	98.07%	100%	91.36%	96.74%
Residential	87.83%	82.82%	90.64%	90.96%	94.60%	**97.35%**
Commercial	85.91%	79.83%	97.38%	89.18%	91.82%	96.83%
Road	75.33%	76.88%	97.40%	90.62%	92.39%	**97.36%**
Highway	82.52%	80.31%	97.75%	93.22%	90.61%	**99.18%**
Railway	79.19%	83.17%	96.76%	87.91%	89.09%	**98.84%**
Parking Lot 1	72.76%	69.42%	**97.56%**	83.15%	94.18%	95.72
Parking Lot 2	79.49%	63.08%	65.33%	**84.11%**	82.51%	78.39%
Tennis Court	93.90%	90.36%	98.83%	97.21%	91.55%	**100%**
Running Track	97.50%	94.40%	99.05%	**100%**	96.72%	**100%**
OA	88.45%	86.45%	96.54%	93.09%	93.06%	**97.36%**
AA	87.81%	86.60%	95.60%	92.06%	86.61%	**95.97%**
κ	87.50%	85.35%	96.26%	92.53%	92.50%	**97.14%**

The detailed results of the aforementioned models can be found in Tables V and VI. In summary, the proposed PyFormer model exhibits outstanding performance, surpassing 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 spatial-spectral Transformer 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 crucial, especially considering the limited availability of training samples.

IV. Conclusions

This paper introduces PyFormer, a novel approach that leverages the strengths of Pyramid and Vision Transformer for Hyperspectral Image Classification (HSIC). By extracting multi-scale 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 pre-training and network optimizations to further enhance PyFormer's performance, especially in scenarios with limited data availability.

References

[1] M. Ahmad, S. Shabbir, S. K. Roy, D. Hong, X. Wu, J. Yao, A. M. Khan, M. Mazzara, S. Distefano, and J. Chanussot, "Hyperspectral image classification—traditional to deep models: A survey for future prospects," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2021.

[2] V. Lodhi, D. Chakravarty, and P. Mitra, "Hyperspectral imaging for earth observation: Platforms and instruments," *Journal of the Indian Institute of Science*, vol. 98, pp. 429–443, 2018.

[3] Y. Li, D. Hong, C. Li, J. Yao, and J. Chanussote, "Hd-net: High-resolution decoupled network for building footprint extraction via deeply supervised body and boundary decomposition," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 209, pp. 51–65, 2024.

[4] B. Lu, P. D. Dao, J. Liu, Y. He, and J. Shang, "Recent advances of hyperspectral imaging technology and applications in agriculture," *Remote Sensing*, vol. 12, no. 16, p. 2659, 2020.

[5] T. Adão, J. Hruška, L. Pádua, J. Bessa, E. Peres, R. Morais, and J. J. Sousa, "Hyperspectral imaging: A review on uav-based sensors, data processing and applications for agriculture and forestry," *Remote sensing*, vol. 9, no. 11, p. 1110, 2017.

[6] C. Li, B. Zhang, D. Hong, J. Yao, and J. Chanussot, "Lrr-net: An interpretable deep unfolding network for hyperspectral anomaly detection," *IEEE Transactions on Geoscience and Remote Sensing*, 2023.

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

[8] C. Weber, R. Aguejdad, X. Briottet, J. Avala, S. Fabre, J. Demuynck, E. Zenou, Y. Deville, M. S. Karoui, F. Z. Benhalouche *et al.*, "Hyperspectral imagery for environmental urban planning," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018, pp. 1628–1631.

[9] M. B. Stuart, A. J. McGonigle, and J. R. Willmott, "Hyperspectral imaging in environmental monitoring: A review of recent developments and technological advances in compact field deployable systems," *Sensors*, vol. 19, no. 14, p. 3071, 2019.

[10] C. B. Pande and K. N. Moharir, "Application of hyperspectral remote sensing role in precision farming and sustainable agriculture under climate change: A review," *Climate Change Impacts on Natural Resources, Ecosystems and Agricultural Systems*, pp. 503–520, 2023.

[11] M. H. Khan, Z. Saleem, M. Ahmad, A. Sohaib, H. Ayaz, M. Mazzara, and R. A. Raza, "Hyperspectral imaging-based unsupervised adulterated red chili content transformation for classification: Identification of red chili adulterants," *Neural Computing and Applications*, vol. 33, no. 21, pp. 14 507–14 521, 2021.

[12] M. H. Khan, Z. Saleem, M. Ahmad, A. Sohaib, H. Ayaz, and M. Mazzara, "Hyperspectral imaging for color adulteration detection in red chili," *Applied Sciences*, vol. 10, no. 17, p. 5955, 2020.

[13] Z. Saleem, M. H. Khan, M. Ahmad, A. Sohaib, H. Ayaz, and M. Mazzara, "Prediction of microbial spoilage and shelf-life of bakery products through hyperspectral imaging," *IEEE Access*, vol. 8, pp. 176 986–176 996, 2020.

[14] M. H. F. Butt, H. Ayaz, M. Ahmad, J. P. Li, and R. Kuleev, "A fast and compact hybrid cnn for hyperspectral imaging-based bloodstain classification," in *2022 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2022, pp. 1–8.

[15] M. Zulfiqar, M. Ahmad, A. Sohaib, M. Mazzara, and S. Distefano, "Hyperspectral imaging for bloodstain identification," *Sensors*, vol. 21, no. 9, p. 3045, 2021.

[16] H. Ayaz, M. Ahmad, M. Mazzara, and A. Sohaib, "Hyperspectral imaging for minced meat classification using nonlinear deep features," *Applied Sciences*, vol. 10, no. 21, p. 7783, 2020.

[17] H. Ayaz, M. Ahmad, A. Sohaib, M. N. Yasir, M. A. Zaidan, M. Ali, M. H. Khan, and Z. Saleem, "Myoglobin-based classification of minced meat using hyperspectral imaging," *Applied Sciences*, vol. 10, no. 19, p. 6862, 2020.

[18] D. Hong, B. Zhang, X. Li, Y. Li, C. Li, J. Yao, N. Yokoya, H. Li, P. Ghamisi, X. Jia, A. Plaza, P. Gamba, J. A. Benediktsson, and J. Chanussot, "Spectralgpt: Spectral remote sensing foundation model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024, dOI:10.1109/TPAMI.2024.3362475.

[19] 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*, 2020.

[20] U. Ghous, M. S. Sarfraz, M. Ahmad, C. Li, and D. Hong, "(2+1)d extreme xception net for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pp. 1–14, 2024.

[21] M. Ahmad and M. Mazzara, "Scsnet: Sharpened cosine similarity-based neural network for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 21, pp. 1–4, 2024.

[22] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 7, pp. 5966–5978, 2021.

[23] A. Jamali, S. K. Roy, D. Hong, P. M. Atkinson, and P. Ghamisi, "Attention graph convolutional network for disjoint hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, pp. 1–1, 2024.

[24] 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 Transactions on Geoscience and Remote Sensing*, 2023.

[25] M. Ahmad, U. Ghous, M. Usama, and M. Mazzara, "Waveformer: Spectral–spatial wavelet transformer for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 21, pp. 1–5, 2024.

[26] X. Huang, M. Dong, J. Li, and X. Guo, "A 3-d-swin transformer-based hierarchical contrastive learning method for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.

[27] 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 Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–13, 2024.

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

[29] 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. 11 836–11 849, 2024.

[30] Y. Ma, Y. Lan, Y. Xie, L. Yu, C. Chen, Y. Wu, and X. Dai, "A spatial-spectral transformer for hyperspectral image classification based on global dependencies of multi-scale features," *Remote Sensing*, vol. 16, no. 2, 2024.

[31] J. Lian, L. Wang, H. Sun, and H. Huang, "Gt-had: Gated transformer for hyperspectral anomaly detection," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2024.

[32] 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 Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–16, 2024.

[33] 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 Geoscience and Remote Sensing Letters*, vol. 21, pp. 1–5, 2024.

[34] 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 Transactions on Image Processing*, vol. 33, pp. 738–752, 2024.

[35] 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.

[36] 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, 2022.

[37] J. Chen, C. Yang, L. Zhang, L. Yang, L. Bian, Z. Luo, and J. Wang, "Tccu-net: Transformer and cnn collaborative unmixing network for hyperspectral image," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pp. 1–20, 2024.

[38] 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 Transactions on Geoscience and Remote Sensing*, pp. 1–1, 2024.

[39] M. Ye, J. Chen, F. Xiong, and Y. Qian, "Adaptive graph modeling with self-training for heterogeneous cross-scene hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–15, 2024.

[40] W. Huang, Y. Deng, S. Hui, Y. Wu, S. Zhou, and J. Wang, "Sparse self-attention transformer for image inpainting," *Pattern Recognition*, vol. 145, p. 109897, 2024.

[41] 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," *Information Fusion*, vol. 104, p. 102163, 2024.

[42] T. Kim, J. Kim, H. Oh, and J. Kang, "Deep transformer based video inpainting using fast fourier tokenization," *IEEE Access*, vol. 12, pp. 21 723–21 736, 2024.

[43] Y. Shi, J. Xia, M. Zhou, and Z. Cao, "A dual-feature-based adaptive shared transformer network for image captioning," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–13, 2024.

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[45] 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.

[46] 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.

[47] W. He, W. Huang, S. Liao, Z. Xu, and J. Yan, "Csit: A multiscale vision transformer for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 9266–9277, 2022.

[48] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.