# Cascade Pixel Transformer with Distance-Driven Spatial Fusion for Hyperspectral Image Classification Using Limited Training Samples

Biri Chanakya Reddy, and Deivalakshmi Subbian

***Abstract*— Classification of hyperspectral images is a crucial component of remote sensing, as it facilitates characterization of land surfaces. The limited availability of labeled data necessitates the design of architectures that can train with a minimal number of samples. Existing pixel- and patch-based methods advanced the field; however, they struggle to provide optimal results with limited training samples due to architectural limitations in learning complex spectral features and spatial contextual information from a small number of samples. To overcome these limitations, the Cascade Pixel Transformer Network (CPTNet) is proposed in this paper. CPTNet introduces a novel pixel-level transformer architecture that comprises transformer units in a cascade configuration to capture long-range spectral features. This work proposes an innovative input feature embedding strategy that combines PCA with a fully connected layer with pruned weights. CPTNet includes a distance-driven spatial fusion block that effectively aggregates the spectral features of all neighboring pixels in the input patch, which results in efficient spatial contextual information usage without relying on overfitting spatial patterns. Extensive experiments involving three benchmark datasets demonstrated the superiority of CPTNet as compared to state-of-the-art methods. The proposed CPTNet improves overall accuracies for three benchmark datasets by 3.31%, 1.71%, and 0.84% compared to the respective second-best approaches. The work proposed in the paper advances the field of hyperspectral classification by robust integration of PCA, weight pruning, pixel-level transformer units, and a distance-driven spatial fusion strategy. The source code repository for CPTNet can be found in https://github.com/profCRB/CPTNet.**

**Link to graphical and video abstracts, and to code:**
**https://latamt.ieeer9.org/index.php/transactions/article/view/10534**

***Index Terms*—Hyperspectral images, computer vision, deep learning, classification, transformer, pixel-level transformer, sparse feature embedding, and spatial fusion.**

## I. Introduction

HYPERSPECTRAL imaging involves capturing any given surface through several spectral bands. Hyperspectral (HS) images have significantly better spectral resolution than their spatial resolution. The complex spectral characteristics provided by HS images allow for the thorough analysis and characterization of any physical surface. Classification of remote sensing HS images have shown a profound impact on fields like mineral resource mining [1], urban development and planning [2], estimation and detection of crops [3], [4], vegetation disease detection [5], and monitoring of environment and forests [6]. In comparison to ordinary color image classification, classification of remote sensing (RS) hyperspectral images constitutes a distinct challenge. Color image classification requires determining the class associated with the entire image or a certain region of the image. Each pixel of remote sensing HS images indicates reflectance of a large area of the earth's surface. As a result, each pixel, rather than the entire image or region, has its own class label. Furthermore, it is possible that nearby pixels belong to different classes. Hyperspectral image (HSI) classification required an alternative approach compared to general classification, as it concentrated on determining the class of individual pixels rather than determining the class of the entire image. Techniques for HSI classification need to deal with data scarcity and class imbalance; most pixels in a subregion of an HS image are enclosed by pixels of the same class, as opposed to pixels on the boundary of two class regions. To address these challenges, HSI classification techniques use just a limited number of samples for training and will be tested on a large number of samples to assess their effectiveness for real-world applications that need classification of a large number of unknown samples. Artificial neural networks became an integral part of modern classification methods related to financial systems [7] and computer vision applications. Deep learning (DL) techniques, especially convolution neural networks (CNN)-based ones, bring significant advances in the field of image processing and computer vision due to CNN's effective and computationally efficient feature-extracting capabilities. Further emergence of the transformer [8] and its self-attention mechanism leads to a new era in computer vision. Unsurprisingly, deep learning architecture-based classification algorithms have had a significant effect on the field of HSI classification since DL architectures have proven effective at extracting complex spatial and spectral characteristics. DL-based classification techniques are often categorized into pixel-based and patch-based methods. Pixel-based methods capitalize on spectral dependencies within spectral values of pixels, which need to be classified. Such methods operate on single pixels with all spectral band values by taking advantage of DL architectures like autoencoders [9], recurrent neural networks (RNN) [10], and convolutional neural networks [11]. Patch-based approaches operated on square spatial patches that formed around pixels whose class needed to be determined. Patch-based approaches estimate the

The associate editor coordinating the review of this manuscript and approving it for publication was Mariel Alfaro-Ponce (*Corresponding author: Biri Chanakya Reddy*).

Biri Chanakya Reddy, and D. Subbian are with the National Institute of Technology Tiruchirappalli, Tiruchirappalli, Tamil Nadu, India, 620015 (e-mails: 408919002@nitt.edu, and deiva@nitt.edu).

class of the center pixel of the patch that they operate on. These methods rely on spectral-spatial patterns to be exploited using deep learning architectures such as CNNs and transformers. Both types of classification methods have drawbacks. As pixel-based techniques rely on a single pixel, they could not take advantage of additional information offered by neighboring pixels. CNN-based patch-based strategies are ineffective for capturing global dependencies, while transformer-based techniques are inadequate for exploiting local dependencies. Furthermore, patch-based approaches have a tendency to adapt overfitting spatial patterns from small training data. An attempt was made to deal with shortcomings of patch- and pixel-based techniques in this paper. As opposed to pixel-based techniques, the proposed CPTNet operates on the target pixel along with its surrounding pixels. As opposed to patch-based techniques, the proposed CPTNet has no dependence on spatial patterns exploited by either CNNs or transformers, but it performs spatial fusion of spectral features of pixels, which are extracted by pixel-level transformers. The approach suggested uses the distance between the centre pixel and the other pixels in the patch to spatially fuse the pixel's spectral features. The potential advantages of distance based fusion of spectral features in spatial dimensions are outlined as follows: Class prediction is improved by combining spectral features of the surrounding pixels with those of the centre pixel. In general, pixels that are near the centre pixel demonstrate higher spectral similarities with the centre pixel than pixels located farther away. This translates into superior class prediction of the centre pixel since similar neighbor pixels contribute more to spatial feature fusion than dissimilar ones.

Existing CNN-based architectures place a greater emphasis on spatial feature modeling through spatial convolutions. This leads to suboptimal spectral feature modeling, as well as models being prone to learning and overfitting spatial patterns. Further, many transformer-based methods incorporate CNNs in their architectures, thereby inheriting the drawbacks of CNNs into their architecture. Furthermore, these methods' tokenization processes involve early feature mixing across spatial-spectral dimensions, which may not produce optimal inputs for transformers with powerful modeling capabilities. In contrast, CPTNet incorporates a completely different architectural design by focusing on modelling complex spectral dependencies at the pixel level. In addition, CPTNet does not rely on CNNs; instead, it incorporates a PCA-based approach to generate effective embedded features which can act as inputs to transformer units. Further, spatial contextual information is incorporated into centre pixel features by using a non-parametric spatial fusion approach rather than relying on heavy parametric structures like CNNs or attention modules. These architectural design choices make CPTNet highly suitable for hyperspectral classification tasks under limited training samples, especially with low spatial resolution datasets. The main novel contributions of the proposed work are outlined below:

- A novel **PCA and weight pruning-based tokenization framework** is proposed for generating effective input feature embeddings. This tokenization strategy eliminates the requirement for convolution preprocessing and does not involve feature mixing across spatial dimensions.
- A **pixel-level transformer framework** is adopted for modeling global spectral dependencies of all pixels in the input patch. In contrast to existing transformer-based approaches that are heavily focused on exploiting spatial dependencies, the proposed method's transformer architecture is focused on learning complex spectral dependencies of individual pixels in input patches.
- A **non-parametric distance-based spatial fusion mechanism** is proposed to enrich the centre pixel features by incorporating contextual information from neighbourhood pixels. Unlike CNNs or attention modules, the proposed spatial fusion approach does not increase model complexity, as it involves any learnable parameters.
- A novel integrated architecture that combines PCA and weight pruning-based embedding, pixel-level modeling by transformer, and distance-based spatial fusion, which provides effective classification results under limited training samples.

Even though individual components like PCA and transformers used in the proposed CPTNet are well-established, the novelty of the proposed CPTNet lies in the tailor-made integration of such components for spectral modeling at the pixel level, followed by non-parametric spatial fusion. The rest of this paper is structured as follows: Section II discusses the existing hyperspectral classification methods. Section III presents the proposed method. Elaborate experimental analysis is presented in Section IV. Limitations and future directions are discussed in Section V. And the conclusion is presented in Section VI.

## II. Related Works

Existing patch-based techniques predominately incorporate deep learning feature extractors like CNNs and transformers in their model architectures. CNN-based methods exploit various combinations of 2D-CNNs and 3D-CNNs to generate spatial and spectral features for classification. In [12], a method that fuses the features extracted by parallel CNN units is proposed. Zhong et al. [13] employed multiple CNN-based residual units to generate spatial and spectral dependencies. In [14], parallel 3D-CNNs are used to generate rich spectral features, followed by spatial feature generation by 2D-CNNs. Ghaderizadeh et al. [15] exploit depthwise convolution operations to generate spectral features along with spatial features. An emerging deep learning strategy called the attention mechanism has shown a profound impact on the classification tasks [16], [17] due to its feature-enhancing capabilities. For hyperspectral classification, attention mechanisms can be incorporated into CNN-based methods. Such methods can be found in [18] and [19]. Dosovitskiy et al. [20] proposed the vision transformer (ViT), which is an alternative to CNNs regarding feature extraction. Compared to CNN, ViT is more effective in extracting global dependencies. Many recent techniques for HSI classification integrate transformers in the feature extraction process. Feature extraction networks that incorporate transformers along with CNNs are proposed in [21] and [22]. Roy et al. [23] introduces

a feature extraction network that uses a transformer alongside morphological CNNs. Further, Zheng et al. [24] implemented spatial feature aggregation by utilizing transformer-like symmetry calculations. Transformer architecture can be explored for pixel-based and patch-based hyperspectral classification techniques. Hong et al. [25] proposed transformer-based architectures for both patch-based and pixel-based approaches. Spectral values of HSI pixels contain a lot of redundancy. Numerous techniques in literature employed dimensionality reduction algorithms like local similarity projection (LSP) and principal component analysis (PCA) to reduce the spectral dimension of HSI pixels, which further results in lesser computational requirements for classification. In [26], a method that employs lightweight CNN architecture to extract features from PCA-transformed HSI samples is presented. Ahmad et al. [27] integrated PCA-based preprocessing with 3D-CNN-based architecture for quicker feature extraction. Xu et al. [28] exploited discrete wavelet transform (DWT) and PCA in the preprocessing stage to generate and forward effective low-dimensional pixel representations to the CNN-based network. Cihan et al. [29] exploited alternative convolution operations called involutions in the feature extraction stage. Further, HSI classification methods that adopted state-space mamba architectures are explored in [30] and [31]. PCA is also being used in transformer-based approaches to reduce computations and improve transformer capabilities with limited training samples. One such method is proposed in [32], which integrates PCA along with a light transformer architecture. Limitations of all types of existing methods are systematically discussed in Table I.

## III. Proposed Method

### A. Preprocessing

Let's consider the HSI image $X_{\text{HSI}} \in \mathbb{R}^{H\times W\times B}$, where $H$ is height, $W$ is width, and $B$ is the number of spectral bands. Each pixel in $X_{\text{HSI}}$ is considered as a 1D array of length $B$. Let's divide these arrays of length $B$ into non-overlapping subarrays of length 10. As a result, each array of length B can be represented by a 2D array of shape $(\lfloor \frac{B}{10} \rfloor \times 10)$. Let's assume $b$ is a total number of subarrays, where $b = \lfloor \frac{B}{10} \rfloor$. The overall HSI image $X_{\text{HSI}}$ can be reshaped into a 4D array of shape $(H \times W \times b \times 10)$. The reshaped $X_{\text{HSI}} \in \mathbb{R}^{H\times W\times b\times 10}$ consists of $(H\times W\times b)$ arrays of length 10. Further principal component analysis (PCA) is performed on all these $H\times W\times b$ arrays, and only 5 dominant components are chosen to represent arrays of length 10. PCA is applied to achieve the following objectives: dimensionality reduction and representing all arrays with principal components that are not associated with any specific spectral positions. After PCA application, the new dimension of $X_{\text{HSI}}$ will be $(H\times W\times b\times 5)$. Further, $X_{\text{HSI}}$ can be reshaped into a 4D array of shape $(H \times W \times 5 \times b)$ by transposing the last two axes. As a result, any pixel at spatial position $(x, y)$ is represented by 5 vectors of length $b$. These 5 vectors are independent of spectral positions, as each vector corresponds to a PCA eigenvector.

The next stage of preprocessing is sample patch generation. To classify a pixel at spatial position $(c_a, c_b)$, that pixel and its neighborhood pixels at positions $(c_a \pm u, c_b \pm v)$ are selected. Here $u, v \in (0, \frac{w-1}{2})$, where $w$ is the height and width of patches. So to classify any pixel at position $(c_a, c_b)$, an HSI 4D cube of shape $(w \times w \times 5 \times b)$ is sliced from $X_{\text{HSI}} \in \mathbb{R}^{H\times W\times 5\times b}$. HSI patch cubes $X_P \in \mathbb{R}^{w\times w\times 5\times b}$ will be inputs to the proposed model that determine the class associated with the central pixel of the patch.

### B. Proposed Architecture

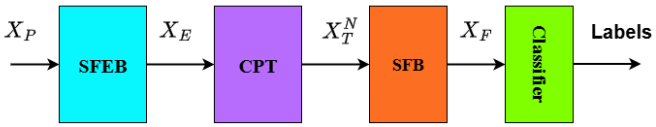


Fig. 1. Architecture of the Proposed Method. SFEB stands for sparse feature embedding block. CPT stands for cascade pixel transformer. SFB stands for spatial fusion block.

The proposed architecture is shown in Fig. 1. It consists of four blocks. The first block is the sparse feature embedding block (SFEB). The second block, CPT, stands for cascade pixel transformer (CPT); the third block is the spatial fusion block (SFB); and the final block is the classifier block. The proposed network operates on input HSI patch $X_P \in \mathbb{R}^{w\times w\times 5\times b}$, and its target labels are one-hot-encoded vectors of class labels.

### C. Sparse Feature Embedding Block

This block generates more effective feature embeddings from the input patch $X_P \in \mathbb{R}^{w\times w\times 5\times b}$. A fully connected layer with sparse weights is employed on $X_P$ to produce sparse embedded features. L1 regulation along with pruning operation is used to achieve sparse weights for fully connected layer. Weight pruning operations are carried out iteratively during training between epochs 20 and 30. From training epoch 20 onwards, the binary mask sets 25% of the weights with the least absolute magnitude to zero. The masked weights make no contribution to the forward pass. The pruning mask is fixed after epoch 30, and no further pruning is performed for the remaining training epochs. The output features $X_S \in \mathbb{R}^{w\times w\times 5\times b}$ of the fully connected layer are described in the below equation.

$$X_S(m,n,l,i) = \sum_{j=0}^{b-1} X_P(m,n,l,j) W_S(j,i) + b_v(i) \quad (1)$$

In the above equation, $W_S \in \mathbb{R}^{b\times b}$ is the weight matrix and $b_v \in \mathbb{R}^b$ is the bias vector of the fully connected layer. A further leaky ReLU layer is applied on $X_S$ to generate effective embedding features $X_E$, which will be forwarded to the transformer structure.

$$X_E = \text{LeakyReLU}(X_S) \quad (2)$$

### D. Cascade Pixel Transformer

The heart of the proposed model is the CPT block. All operations in CPT are performed at the pixel level. The structure of the CPT block is shown in Fig. 2. The fundamental

TABLE I
ARCHITECTURAL LIMITATIONS OF VARIOUS CATEGORIES OF EXISTING HYPERSPECTRAL CLASSIFICATION METHODS

| Category | Limitations |
|---|---|
| CNN-based Methods | • Spatial convolutions utilized in these methods lead to feature smoothing and may learn overfitting spatial patterns, especially when trained with a limited number of samples.<br>• Inconsistent classification performance against various datasets.<br>• Inadequacy in modeling global feature dependencies. |
| Pixel-based Transformer (SpectralFormer [25]) | • Tokens having redundant information due to overlapping spectral grouping.<br>• Computationally expensive for datasets with a large number of bands, as each band will have its own token.<br>• The pixel-variant approach is unable to use additional context information provided by neighborhood pixels. |
| Patch-based Transformer (SSFTT [22], morphFormer [23], SSBFNet [32]) | • Inadequate at exploiting local range decencies.<br>• **Spatial-Spectral Feature Imbalance**: Tokenization of most existing transformer methods involves generating tokens that correspond to spatial positions or combinations of spatial positions in an input patch. That leads to the transformer's powerful self-attention mechanism being focused on extracting spatial dependencies rather than spectral dependencies, which are essential for hyperspectral classification.<br>• Most transformer methods depend on CNNs in their architecture for preliminary local feature extraction. Utilization of CNNs leads to spatial feature smoothing and overfitting.<br>• Not having additional emphasis on central pixel features.<br>• Tokenization of some transformer-based methods involves generating tokens, which are linear combinations of vectors that correspond to all spatial positions. This results in loss of distinct positional information, as each token consists of aggregated information of all spatial positions. Further positional encoding is added to such tokens, which are not associated with any specific spatial position. |
| Patch-based Mamba (MHSSM [30], SSMamba [31]) | • Feature modeling for low spatial-resolution datasets is not as effective as compared to high-resolution datasets.<br>• Moderate classification performance under limited training samples. |

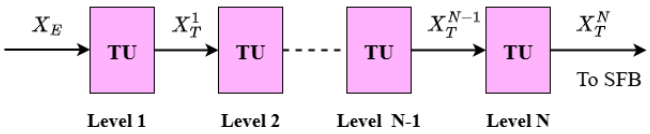


Fig. 2. Architecture of Cascade Pixel Transformer. TU in architecture stands for transformer unit.

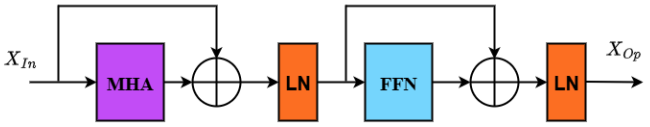


Fig. 3. Architecture of Transformer Unit. MHA stands for multi-head attention layer. LN stands for layer normalization. FFN stands for feedforward network.

computational component of CPT is the transformer unit (TU). CPT employs multiple transformer units in a cascaded manner. As shown in the figure, CPT extracts features at multiple levels. Each feature extraction level consists of one transformer unit.

Let's consider TU(.) as the operation of the transformer unit on its input, and then the extracted features at the first level are described by the following equation.

$$X_T^1 = \text{TU}^1(X_E) \tag{3}$$

Here $\text{TU}^1$ is the transformer unit of the first level. In addition, features generated at various levels are described in the following equation.

$$X_T^i = \text{TU}^i(X_T^{i-1}) \tag{4}$$

Where $i = 2$ to $N$, N is the total number of levels in CPT, $X_T^i$ is the output of $i^{th}$ level, and $\text{TU}^i$ is the transformer unit of the $i^{th}$ level. The inner functional details of the transformer unit are presented in Fig. 3. Features extracted by the final level of the CPT block are transferred to the spatial fusion block.

**Transformer Unit:** Fig. 3 presents the architecture of the transformer unit that is used in the proposed method. And Fig. 4 presents an important layer of the transformer unit, called the

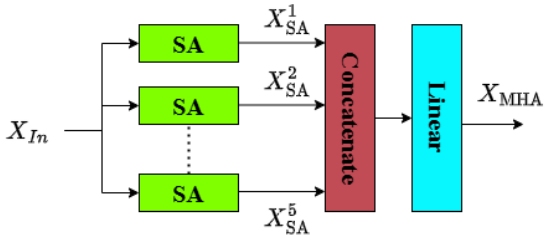


Fig. 4. Architecture of Multi-Head Attention Layer. SA in architecture stands for self-attention layer.

multi-head attention (MHA) layer. The MHA layer employs multiple self-attention (SA) layers to extract the relationships between its input vectors.

Let's assume $X_{In} \in \mathbb{R}^{w \times w \times 5 \times b}$ is input to the transformer unit. Then the 2D sub-array associated with the pixel at spatial position $(x, y)$ is represented by $X_{In}(x, y) \in \mathbb{R}^{5 \times b}$. Further, 2D sub-array $X_{In}(x, y)$ associated with pixel position $(x, y)$ can be viewed as 5 vectors of length $b$. As specified, all operations of the transformer unit are performed at the pixel level. For each position $(x, y)$ that transformer unit is operated on, its inputs are 5 vectors of length $b$ that correspond to $X_{In}(x, y)$. And transformer operations are repeated for all pixels. Mathematical operations related to the SA layers of the

transformer unit are according to the transformer architecture proposed by Vaswani et al. [8]. Let $a_h$ be the number of SA layers in the MHA layer, and let $Q_M^i \in \mathbb{R}^{b\times\left\lfloor\frac{b}{a_h}\right\rfloor}$, $K_M^i \in \mathbb{R}^{b\times\left\lfloor\frac{b}{a_h}\right\rfloor}$, and $V_M^i \in \mathbb{R}^{b\times\left\lfloor\frac{b}{a_h}\right\rfloor}$ be three learnable matrices associated with the $i^{th}$ self-attention layer. Then queries, keys, and values of input $X_{In}(x,y) \in \mathbb{R}^{5\times b}$ for the $i^{th}$ SA layer are determined by the following equation.

$$Q_{x,y}^i = X_{In}(x,y)Q_M^i \tag{5}$$

$$K_{x,y}^i = X_{In}(x,y)K_M^i \tag{6}$$

$$V_{x,y}^i = X_{In}(x,y)V_M^i \tag{7}$$

Here $Q_{x,y}^i, K_{x,y}^i, V_{x,y}^i \in \mathbb{R}^{5\times\left\lfloor\frac{b}{a_h}\right\rfloor}$. The result of the $i^{th}$ self-attention layer on $X_{In}(x,y)$ is described by the following equation.

$$X_{\text{SA}}^i(x,y) = \text{Softmax}(\frac{Q_{x,y}^i {K_{x,y}^i}^T}{\sqrt{d}})V_{x,y}^i \tag{8}$$

In the above equation, $d$ is the dimensionality of key vectors, ${K_{x,y}^i}^T$ is the transpose of $K_{x,y}^i$, and $X_{\text{SA}}^i \in \mathbb{R}^{5\times\left\lfloor\frac{b}{a_h}\right\rfloor}$. Further results of all SA layers are concatenated and forwarded to the learnable linear layer, as shown in Fig. 4. The purpose of the linear layer is to transform the concatenated features into the shape of $X_{In}(x,y)$. Let $L_{\text{MHA}} \in \mathbb{R}^{a_h\left\lfloor\frac{b}{a_h}\right\rfloor\times b}$ be the learnable matrix of the linear layer. Then the result of MHA layer operations on input $X_{In}(x,y)$ is indicated by the following equation.

$$X_{\text{CC}}(x,y) = \text{Concatenate}(X_{\text{SA}}^1(x,y), ..., X_{\text{SA}}^{a_h}(x,y)) \tag{9}$$

$$X_{\text{MHA}}(x,y) = X_{\text{CC}}(x,y)L_{\text{MHA}} \tag{10}$$

In the above equation $X_{\text{CC}}(x,y) \in \mathbb{R}^{5\times a_h\left\lfloor\frac{b}{a_h}\right\rfloor}$. Further inputs are added with output features of MHA layers, and layer normalization is applied on resultant features as shown in Fig. 3. These steps are described by the following equation.

$$X_{\text{AN}}(x,y) = \text{LayerNorm}(X_{\text{MHA}}(x,y) + X_{In}(x,y)) \tag{11}$$

These features, $X_{\text{AN}}$, are transferred to the feedforward network (FFN) as shown in Fig. 3. The purpose of FFN is to exploit non-linear relations among features of vectors. It consists of a series of two fully connected (FC) layers. In this proposed model, the number of output nodes of the first FC layer is chosen as $a_h b$ with ReLU as the activation function. The number of output nodes of the second FC layer is chosen as $b$ with the linear activation function. $X_{\text{AN}}(x,y) \in \mathbb{R}^{5\times b}$ can be treated as 5 vectors of length $b$. The FFN layer performs its operations on each of these vectors separately. Let's assume weights of the first FC layer are $W_{\text{FC}}^1 \in \mathbb{R}^{b\times a_h b}$, and weights of the second FC layer are $W_{\text{FC}}^2 \in \mathbb{R}^{a_h b\times b}$; then the operation of FFN on $X_{\text{AN}}$ is described by the following equation.

$$X_{\text{FFN}}(x,y) = \text{ReLU}(X_{\text{AN}}(x,y)W_{\text{FC}}^1 + bias_1)W_{\text{FC}}^2 + bias_2 \tag{12}$$

In the above equation, $bias_1$ and $bias_2$ are bias values associated with the first and second FC layers. Further output of FFN is added with $X_{\text{AN}}(x,y)$, and layer normalization is applied on combined features as shown in Fig. 3. Therefore, the output of the transformer unit on a pixel at $(x,y)$ is described by the following equation.

$$X_{Op}(x,y) = \text{LayerNorm}(X_{\text{FFN}}(x,y) + X_{\text{AN}}(x,y)) \tag{13}$$

All the operations that are described in equations (5-13) are performed at the pixel level. So these operations are repeated for pixels at all spatial positions.

### *E. Spatial Fusion Block (SFB)*

This block performs distance-based global average pooling (GAP) in spatial dimensions on features $X_T^N$, which are provided by CPT. The output of SFB, $X_F \in \mathbb{R}^{1\times1\times5\times b}$, will be determined as per the following equations.

$$X_F(1,1,k,l) = \frac{\sum_{x=0}^{w-1}\sum_{y=0}^{w-1} X_T^N(x,y,k,l)D_W(x,y)}{w^2} \tag{14}$$

$$D_W(x,y) = \frac{Max_{Dist} + 1 - Dist(x,y)}{Max_{Dist} + 1} \tag{15}$$

$$Dist(x,y) = \max(\text{abs}(x - \frac{w-1}{2}), \text{abs}(y - \frac{w-1}{2})) \tag{16}$$

$$Max_{Dist} = \frac{w-1}{2} \tag{17}$$

In the above equation, $D_W$ is the distance weight that is calculated based on the distance between (x,y) and the center pixel position $(\frac{w-1}{2}, \frac{w-1}{2})$. $Dist(x,y)$ is the distance between the pixel at position (x,y) and the center pixel. $Max_{Dist}$ is the maximum distance possible with a patch size of $w$. As shown in the above equations (14-17), the weightage of pixels that are near to the center pixel is more compared to pixels that are far from the center pixel. For most of the extracted patches, pixels that are near to the center pixel have more spectral similarities with the center pixel compared to pixels that are far from the centre pixel. This results in effective feature aggregation of pixels, as similar pixels to the center pixel have more weightage in feature aggregation.

Table II presents the relationship between the distance of neighborhood pixels from the center pixel and the class similarity between the center pixel and its neighboring pixels. All possible patches from the entire dataset were used to compute these values. The percentage values reported in the table represent the average across all individual patches. As evident from Table II, the percentage of neighborhood pixels having the same class as the center pixel decreases with increasing distance. Since class similarity is closely aligned with spectral similarity, Table II empirically supports the intuition that pixels closer to the centre are spectrally more similar than those farther away.

TABLE II
CLASS SIMILARITY ANALYSIS OF NEIGHBOURHOOD PIXELS WITH CENTRE PIXEL

| Distance | Class Similarity (%) | | |
|---|---|---|---|
| | Indian pines | Salinas | Pavia Centre |
| 0 | 100 | 100 | 100 |
| 1 | 90.25 | 96.52 | 90.42 |
| 2 | 81.62 | 93.49 | 82.71 |
| 3 | 73.74 | 90.28 | 76.18 |
| 4 | 66.54 | 87.2 | 70.86 |
| 5 | 60.06 | 84.16 | 66.64 |
| 6 | 54.02 | 81.25 | 63.33 |

### F. Classifier Block

Features, $X_F \in \mathbb{R}^{1\times1\times5\times b}$, from SFB are forwarded to the classifier block to predict the class label of the center pixel of the input HSI patch $X_P$. The classifier block consists of one flatten layer and one fully connected layer. The Flatten layer converts aggregated features $X_F$ into a 1D array. These 1D features are connected to the FC layer. The number of output nodes of the FC layer is the same as the number of classes of the HSI dataset. To produce effective predictions, weight pruning operations and L1 regularization were applied on the FC layer. The FC layer utilizes the softmax activation function to produce output labels. The final output $Y_p$ is a vector of length L, where $L$ is the number of classes of the HSI dataset. During training, categorical cross-entropy loss is used as a loss function to update learnable parameters of the proposed model.

### G. Distinct Architectural Features of CPTNet

Proposed CPTNet leverages the transformer's powerful self-attention mechanism to extract global spectral dependencies by employing transformer units at the pixel level, unlike existing transformer-based methods, which are highly focused on extracting global spatial dependencies. For classification of hyperspectral images, especially of low spatial-resolution images, optimal spectral feature extraction is more important compared to spatial feature extraction, so leveraging a transformer's self-attention for spectral dependencies is a better choice compared to using a transformer for extracting spatial dependencies or spatial-spectral dependencies. The CPTNet gives additional emphasis to centre pixel features by giving maximum weight in the distance-based spatial fusion block. As discussed in preprocessing III-A, CPTNet does not use unnecessary positional encoding, as vectors that correspond to principal components that do not associate with specific spectral positions are applied to transformer units. Unlike existing methods that predominantly employ CNNs before tokenization, the CPTNet utilizes PCA on sub-arrays of pixels to generate input features to transformer units. Neighborhood pixel information is integrated with centre pixel features by using a simple intuitive non-parametric distance-based spatial fusion strategy, instead of employing CNNs before tokenization or employing transformers for finding spatial dependencies.

It is important to observe that even though individual components like PCA and transformers are well-known computational concepts. Nevertheless, the novelty of the proposed CPTNet does not lie in these individual components, but it is in their tailor-made integration in the CPTNet architecture, whereas (i) PCA-based embedding does not require convolutional preprocessing; (ii) transformer architecture focuses solely on spectral dependency modeling at the pixel level; and (iii) spatial contextual information is incorporated via a non-parametric spatial fusion mechanism rather than parametric modules such as CNNs or attention modules.

## IV. EXPERIMENTS AND ANALYSIS

### A. Datasets

The proposed method was trained and tested on three remote sensing hyperspectral datasets so that performance can be analyzed. These three datasets are the Pavia Centre dataset [33], Salinas dataset [34], and Indian Pines dataset [35]. The Pavia Centre dataset contains 102 relevant spectral bands recorded by the ROSIS sensor over Pavia, Italy. The total number of samples available for classification is 148,152, with nine classes representing distinct urban surfaces. The Salinas dataset was acquired by the AVIRIS sensor over the Salinas Valley region in the United States. This dataset comprises 204 useful spectral bands for analysis. It comprises 54,129 samples categorized into 16 vegetation categories. The Indian Pines dataset contains 10,249 samples divided into 16 classes representing various vegetation types. It has 200 usable spectral bands out of 224 total spectral bands. It comprises a single scene recorded by the AVIRIS sensor in Northwest Indiana, USA.

Hyperspectral datasets generally have the problem of class imbalance, which means some majority classes have a large number of samples compared to minority classes that have few samples. If class imbalance also presents a selected training dataset, classification models are optimized for majority classes compared to minority classes, which results in suboptimal performance for minority classes. The problem of class imbalance can be avoided by selecting a fixed number of samples from each class for training. For the Pavia Centre and Salinas datasets, 20 samples per category were selected for training, 20 samples per category were selected for validation, and the remaining samples were selected for testing. Since the fixed number of training samples per class is used for the Pavia Centre and Salinas datasets, both minority and majority classes are equally represented in the training process. This balanced setup prevents the training process from being dominated by majority classes and helps maintain good performance on minority classes. As Indian Pines consists of very few samples for some categories, approximately $5\%$ of samples are selected for training, $5\%$ of samples are selected for validation, and the remaining samples are used for testing. For minority classes like class 7 and class 9, 10 samples per class are selected for training. Tables III, IV, and V present training, validation, and testing distribution for the Indian Pines, Salinas, and Pavia Centre datasets, respectively.

### B. Ablation Studies

Fig. 5 presents CPTNet performance against various numbers of transformer levels in its architecture. As is evident,

TABLE III
SAMPLES DISTRIBUTION FOR INDIAN PINES DATASET

| No. | Class | Training | Validation | Testing |
|---|---|---|---|---|
| 1 | Alfalfa | 10 | 10 | 26 |
| 2 | Corn notill | 71 | 71 | 1286 |
| 3 | Corn mintill | 41 | 41 | 748 |
| 4 | Corn | 11 | 11 | 215 |
| 5 | Grass pasture | 24 | 24 | 435 |
| 6 | Grass trees | 36 | 36 | 658 |
| 7 | Grass pasture mowed | 10 | 10 | 8 |
| 8 | Hay windrowed | 23 | 23 | 432 |
| 9 | Oats | 10 | 5 | 5 |
| 10 | Soybean notill | 48 | 48 | 876 |
| 11 | Soybean mintill | 122 | 122 | 2211 |
| 12 | Soybean clean | 29 | 29 | 535 |
| 13 | Wheat | 10 | 10 | 185 |
| 14 | Woods | 63 | 63 | 1139 |
| 15 | Buildings Drives Grass Trees | 19 | 19 | 348 |
| 16 | Stone Steel Towers | 10 | 10 | 73 |

TABLE IV
SAMPLES DISTRIBUTION FOR SALINAS DATASET

| No. | Class | Training | Validation | Testing |
|---|---|---|---|---|
| 1 | Brocoli green weeds 1 | 20 | 20 | 1969 |
| 2 | Brocoli green weeds 2 | 20 | 20 | 3686 |
| 3 | Fallow | 20 | 20 | 1936 |
| 4 | Fallow rough plow | 20 | 20 | 1354 |
| 5 | Fallow smooth | 20 | 20 | 2638 |
| 6 | Stubble | 20 | 20 | 3919 |
| 7 | Celery | 20 | 20 | 3539 |
| 8 | Grapes untrained | 20 | 20 | 11231 |
| 9 | Soil vinyard develop | 20 | 20 | 6163 |
| 10 | Corn senesced green weeds | 20 | 20 | 3238 |
| 11 | Lettuce romaine 4wk | 20 | 20 | 1028 |
| 12 | Lettuce romaine 5wk | 20 | 20 | 1887 |
| 13 | Lettuce romaine 6wk | 20 | 20 | 876 |
| 14 | Lettuce romaine 7wk | 20 | 20 | 1030 |
| 15 | Vinyard untrained | 20 | 20 | 7228 |
| 16 | Vinyard vertical trellis | 20 | 20 | 1767 |

TABLE V
SAMPLES DISTRIBUTION FOR PAVIA CENTRE DATASET

| No. | Class | Training | Validation | Testing |
|---|---|---|---|---|
| 1 | Water | 20 | 20 | 65931 |
| 2 | Trees | 20 | 20 | 7558 |
| 3 | Asphalt | 20 | 20 | 3050 |
| 4 | Self-Blocking Bricks | 20 | 20 | 2645 |
| 5 | Bitumen | 20 | 20 | 6544 |
| 6 | Tiles | 20 | 20 | 9208 |
| 7 | Shadows | 20 | 20 | 7247 |
| 8 | Meadows | 20 | 20 | 42786 |
| 9 | Bare Soil | 20 | 20 | 2823 |

TABLE VI
ABLATION STUDIES ON SPATIAL FUSION BLOCK (SFB)

| Dataset | SFB (GAP based) | | | SFB (distance-based GAP) | | |
|---|---|---|---|---|---|---|
| | AA (%) | OA (%) | KC×100 | AA (%) | OA (%) | KC×100 |
| Pavia Centre | 97.2±0.9 | 98.34±0.7 | 97.65±0.9 | **97.77±0.7** | **98.72±0.3** | **98.2±0.4** |
| Salinas | **98.59±0.2** | **96.71±0.5** | **96.34±0.6** | 98.35±0.3 | 96.07±0.8 | 95.63±0.9 |
| Indian Pines | 97.57±0.3 | 97.35±0.2 | 96.97±0.3 | **98.1±0.4** | **98±0.3** | **97.72±0.3** |

TABLE VII
SALINAS DATASET - ABLATION STUDIES ON SFB WITH VARIED PATCH SIZES

| Patch Size | SFB (GAP based) | | | SFB (distance-based GAP) | | |
|---|---|---|---|---|---|---|
| | AA (%) | OA (%) | KC×100 | AA (%) | OA (%) | KC×100 |
| 11 | **98.59±0.2** | **96.71±0.5** | **96.34±0.6** | 98.35±0.3 | 96.07±0.8 | 95.63±0.9 |
| 19 | 99.06±0.3 | 98±0.7 | 97.78±0.8 | **99.12±0.3** | **98.01±0.6** | **97.79±0.7** |
| 23 | 98.93±0.3 | 98.05±0.4 | 97.83±0.4 | **99.25±0.2** | **98.36±0.5** | **98.17±0.6** |

TABLE VIII
ABLATION STUDIES ON SPARSE FULLY CONNECTED LAYERS

| Dataset | With FC layers | | | With Sparse FC layers | | |
|---|---|---|---|---|---|---|
| | AA (%) | OA (%) | KC×100 | AA (%) | OA (%) | KC×100 |
| Pavia Centre | 97.68±0.9 | 98.61±0.5 | 98.04±0.7 | **97.77±0.7** | **98.72±0.3** | **98.2±0.4** |
| Salinas | 97.98±0.5 | 95.21±1.2 | 94.67±1.3 | **98.35±0.3** | **96.07±0.8** | **95.63±0.9** |
| Indian Pines | 97.95±0.4 | 97.61±0.3 | 97.27±0.4 | **98.1±0.4** | **98±0.3** | **97.72±0.3** |

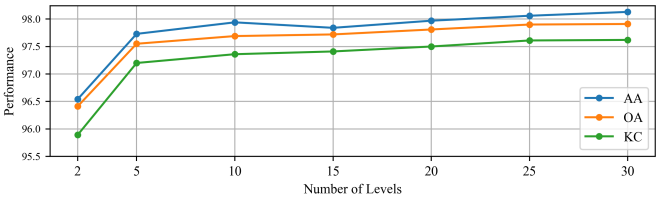


Fig. 5. Classification results against various number of levels in CPTNet.

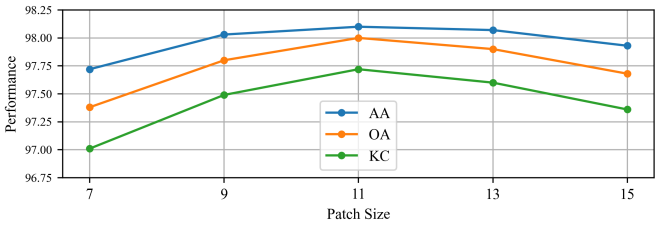


Fig. 6. Classification results against various patch sizes.

the performance of CPTNet is improved as the number of transformer units increases, and its performance saturates from 25 transformer levels. Fig. 6 presents classification results against various patch sizes for CPTNet. The results in Fig. 5 are obtained with a fixed patch size of 13. As demonstrated in Fig. 6, the performance of CPTNet will improve with an initial increase in patch size because of additional spatial contextual information. However, eventually performance will decline with larger patch size because of the increasing effect of other class pixels. The results in Fig. 6 are obtained with transformer levels at 25. The results in Fig. 5 and Fig. 6 were acquired from experimentation on the challenging Indian Pines dataset.

Table VI provides ablation studies regarding the spatial fusion block. As it demonstrated, utilization of distance-based GAP in SFB block results in better classification performance for Indian Pines and Pavia Centre datasets. It is due to distance-based spatial fusion being more robust to the influence of unrelated neighborhood patch pixels compared to regular GAP-based spatial fusion. For the Indian Pines dataset, utilization of the SFB block results in a 0.65% improved overall accuracy. According to the Wilcoxon test, the

improvement for the Indian Pines dataset is highly significant, with a p-value of 0.002. For the Pavia Centre dataset, SFB brought a 0.38% improvement in overall accuracy. According to the Wilcoxon test, the improvement for the Pavia Centre dataset is statistically significant with a p-value of 0.037. In contrast, GAP-based spatial fusion provides better results only on the Salinas dataset, where most pixels in the majority of input patches belong to the same class as the center pixel. As presented in Table II, for the Salinas dataset, the percentage of neighborhood pixels having the same class as the center pixel patch is higher even for higher distances as compared to the Indian Pines and Pavia Center datasets. As a result, distance-based spatial fusion did not bring improvements for the Salinas dataset. To ensure model architecture consistency across all datasets, distance-based GAP was used in the final model, including the Salinas dataset. Even though distance-based GAP is slightly inferior to regular GAP for the Salinas dataset at patch size 11, this is due to high spatial homogeneity in small patches for this dataset, where the majority of pixels belong to the same class as the centre pixel, and uniform weights may be sufficient. However, as patch size increases, the percentage of pixels that belong to classes other than the centre pixel increases, and distance-based GAP will dominate regular GAP as it suppresses the influence of other class pixels, mixed-class pixels, and noisy pixels. Table VII supports the notion that distance-based GAP eventually outperforms regular GAP as patch size increases against the Salinas dataset. Further adapting distance-based GAP as a unified aggregation method avoids the dataset-specific tuning of the model and improves the robustness of the proposed model. Distance-based GAP achieves 96.07% OA for Salinas, which is still better than other methods. We chose distance-based GAP for the spatial fusion block in the final model for all datasets because it provides stable performance across varying spatial contexts, which is desirable flexibility for practical applications. Table VIII presents ablation studies on the impact of using sparse layers in the SFEB and classifier block. In the first case, a typical fully connected layer was used in the classifier block and SFEB block instead of a pruned fully connected layer. As demonstrated, usage of sparse weights results in better classification performance against all the datasets in terms of all three metrics.

TABLE IX
STUDY ON CLASSIFICATION PERFORMANCE AGAINST VARIOUS TRAINING SAMPLES PER CLASS

| **Dataset** | **Metric** | Training Samples per Class | | | | |
|---|---|---|---|---|---|---|
| | | 5 | 10 | 15 | 20 | 25 |
| Salinas | OA | 92.16±2 | 93.98±1.2 | 95.29±1.1 | 96.07±0.8 | 96.45±1 |
| Pavia Centre | OA | 97.51±0.8 | 97.97±0.9 | 98.18±0.7 | 98.72±0.3 | 98.78±0.7 |

TABLE X
ABLATION STUDIES ON WEIGHT PRUNING PERCENTAGES

| Pruning % | | AA | OA | KC |
|---|---|---|---|---|
| SFEB | Classifier | (%) | (%) | (×100) |
| 25 | 25 | 98.11±0.5 | 97.89±0.4 | 97.59±0.4 |
| 25 | 50 | 98.1±0.4 | **98±0.3** | **97.72±0.3** |
| 50 | 25 | **98.11±0.3** | 97.85±0.2 | 97.54±0.2 |
| 50 | 50 | 98.07±0.4 | 97.83±0.4 | 97.53±0.4 |

Table IX presents the classification performance of the proposed CPTNet with various training samples per class for the Salinas and Pavia Centre datasets. For the Salinas dataset, CPTNet produced competitive results even with 0.15% (equivalent to 5 samples per class) of training samples. CPTNet produced effective results even with 0.03% (equivalent to 5 samples per class) of training samples for the Pavia Centre dataset. The results from Table IX demonstrate CPTNet's ability to learn from a very fraction of samples. Table X presents an ablation study on various weight pruning ratios in SFEB and classifier blocks. As shown in results, SFEB with 25% pruned weights and classifier block with 50% pruned weights give better results compared to other combinations of pruning ratios. The ablation results of Table X are based on the Indian Pines dataset.

TABLE XI
ABLATION STUDIES ON SUB-ARRAY LENGTH AND PCA COMPONENTS

| Sub array length | PCA components | AA (%) | OA (%) | KC (×100) |
|---|---|---|---|---|
| 10 | 3 | **98.15±0.4** | 97.99±0.4 | 97.7±0.4 |
| 10 | 5 | 98.1±0.4 | **98±0.3** | **97.72±0.3** |
| 20 | 10 | 97.68±0.5 | 97.47±0.3 | 97.14±0.4 |
| 20 | 5 | 97.43±0.7 | 97.1±0.6 | 96.69±0.7 |

TABLE XII
ABLATION STUDIES ON INCLUSION OF PCA

| | AA (%) | OA (%) | KC (×100) |
|---|---|---|---|
| Without PCA | 97.67±1.1 | 97.41±1 | 97.05±1.1 |
| With PCA | **98.1±0.4** | **98±0.3** | **97.72±0.3** |

Table XI presents the ablation studies on the length of the pixel sub-array and the number of PCA components retained. As evident in the results, retention of 5 PCA components from sub-arrays of length 10 gives better results compared to other combinations of sub-array length and PCA component retentions. Table XII represents ablation studies on inclusion of PCA. As evident in the result application of PCA, results in better classification perform as PCA produces better feature representations. Ablation results presented in Tables XI and XII are obtained using the Indian Pines dataset.

### *C. Networks Settings*

The number of sub-arrays $b$, which is discussed in Sec. III-A, is chosen as 20 for both the Indian Pines and Salinas datasets, and it is chosen as 10 for the Pavia Centre dataset. The length of each sub-array will be 10, and after the application of PCA, only 5 components are selected. Selection of these values is based on experiments conducted on various combinations of sub-array lengths and number of retained principal components, which were represented in Table XI. The number of self-attention units in each MHA unit is chosen as 5. The number of levels in CPTNet is chosen as 25. The choice of 25 levels is supported by ablation results represented in Fig. 5. The proposed model employs the Adam optimizer with a learning rate of 0.0001 to update the model weights by

back-propagating the loss provided by the categorical entropy loss function. L1 and L2 regularization are adopted to prevent overfitting on small training samples. L2 regularization with a regularization factor of 10E-4 is adopted for layers in transformer units, whereas L1 regularization with the factor of 10E-3 is adopted for fully connected layers in SFEB and classifier blocks. For the fully connected layer in the SFEB unit, 25% of weights are pruned, whereas 50% of weights are pruned for the fully connected layer in the classifier block. The selection of these pruning ratios is based on ablation experiments represented by Table X. Additionally, the early stopping technique is adopted to stop the training process when the model stops improving against validation samples continuously for 10 epochs.

### *D. Comparison Studies*

Seven state-of-the-art methods are selected to compare with the proposed method so that its effectiveness can be assessed. These seven methods are LCDCNN [26], FCCNN [27], DWTD [28], IRSN [29], MHSSM [30], SSMamba [31], and SSBFNet [32]. DWTD [28] incorporates DWT and PCA preprocessing techniques to select useful features and to reduce spectral dimension. The other methods employ PCA in the preprocessing stage. We did not include SpectralFormer [25] in this comparison study because its experimental setting is different from the methods considered above. SpectralFormer [25] is primarily a pixel-based method that uses raw spectral band values from individual pixels as input, whereas the compared methods are patch-based approaches that employ PCA during preprocessing. Three well-used metrics of classification in literature are selected to study and compare the proposed method along with state-of-the-art techniques. These metrics are Kappa Coefficient (KC), Average Accuracy (AA), and Overall Accuracy (OA). In the results tables , 100 times the Kappa coefficient is presented, whereas percentage values are indicated for overall accuracy and average accuracy.

For the purpose of fair comparison between the proposed method and other selected methods, the following measures are taken: Classification results are aggregated after running each method 10 times. The size of the patch $(11 \times 11)$ is selected for all methods. Model parameters like optimizer, learning rate, and loss functions are selected according to references LCDCNN [26], FCCNN [27], DWTD [28], IRSN [29], MHSSM [30], SSMamba [31], and SSBFNet [32]. All methods use identical training, validation, and testing samples for each of the 10 experiments. For each of these 10 experiments, random seeds between 100 and 109 are used; the resultant training, validation, and testing samples will be reproducible and the same for all methods, which ensures a fair comparison. All models are trained using the early stopping strategy with a maximum of 1000 epochs; thus, models continue to learn until they cease improving their performance over the validation sample after 10 consecutive epochs of learning. All the experiments were carried out on the Google Colab platform with the L4-GPU run time environment (RTE). The L4-GPU RTE provides 53GB RAM along with 22.5GB GPU RAM.

The experimental results for the Indian Pines dataset are reported in Table XIII. The proposed CPTNet outperformed every other method in all three metrics. It outperforms the second-best approach, SSBFNet [32], by 3.31% in terms of overall accuracy. The third and fourth best results are produced by FC-CNN [27] and IRSN [29] respectively. Additionally, the results from DWTD [28] and MHSSM [30] were modest. LC-DCNN [26] and SSMamba [31] were not as effective as other approaches. Further, Class 9 and 7 are minority classes in the Indian Pines dataset according to Table III. As evident in Table XIII, the proposed CPTNet produces the best classification performance on these minority classes.

In Table XIV, the classification results on the Salinas dataset are presented. The proposed CPTNet outperformed all the other methods in terms of all three metrics. It achieves 1.71% better overall accuracy against the second-best method, SSBFNet [32]. The third and fourth best results are obtained by IRSN [29] and FC-CNN [27], respectively. Further, DWTD [28] and SSMamba [31] produced moderate results. Compared to other methods, LC-DCNN [26] and MHSSM [30] performed poorly. Furthermore, Table IV shows that Class 13 and 11 are minority classes in the Salinas dataset. As presented in Table XIV, the proposed CPTNet produces the best classification results on these minority classes.

In Table XV, results from experiments on the Pavia Centre dataset are laid out. The proposed CPTNet surpassed all the other methods in terms of every single one of the metrics. It produces 0.84% better overall accuracy compared to the second-best method, SSBFNet [32]. CPTNet produced better classification performance compared to SSBFNet [32] even in terms of average accuracy and kappa coefficient. Compared to SSBFNet [32], CPTNet produced better results for 8 classes out of a total of 9 classes. This shows consistency of CPTNet across individual classes. According to the Wilcoxon test, CPTNet's performance is highly significant compared to the second-best SSBFNet [32], with a p-value of 0.0098. The third and fourth best results are obtained by FC-CNN [27] and DWTD [28]. Additionally, both IRSN [29] and SSMamba [31] achieved modest results. When compared to other methods, MHSSM [30] and LC-DCNN [26] were performed inadequately. Additionally, Class 4 and 9 are minority classes in the Pavia dataset according to Table V. As demonstrated in Table XV, the proposed CPTNet achieves the best classification performance on these minority classes.

Figures 7, 8, and 9 represent the predicted class maps for CPTNet along with other methods. And figures 10, 11, and 12 represent the zoomed-in predicted class maps for CPTNet along with other methods. These figures are obtained by zooming in on original class maps from figures 7, 8, and 9. Further, Table XVI represents the average of class-wise F1-scores corresponding to classification maps presented in 7, 8, and 9. Based on outcomes presented in Tables XIII, XIV, and XV, the proposed CPTNet obtained better performance against all other methods in terms of all three metrics. LC-DCNN [26] provided comparable results against Pavia Centre datasets, but its performance against Salinas and Indian Pines is not remarkable. FC-CNN [27] achieved the third-best results for the Pavia Centre dataset, and it produced

TABLE XIII
CLASS-WISE AND OVERALL CLASSIFICATION PERFORMANCE ON THE INDIAN PINES DATASET

| Class | Classification Accuracy of Individual Classes | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | LC-DCNN [26] | FC-CNN [27] | DWTD [28] | IRSN [29] | MHSSM [30] | SSMamba [31] | SSBFNet [32] | CPTNet |
| 1 | 36.54 | 95.38 | 98.85 | 96.15 | 91.54 | 71.54 | 98.08 | 98.08 |
| 2 | 48.52 | 87.36 | 85.9 | 88.86 | 80.83 | 61.83 | 95.06 | 97.54 |
| 3 | 17.22 | 84.21 | 83.68 | 91.32 | 81.34 | 52.75 | 93.21 | 97.67 |
| 4 | 39.53 | 67.63 | 69.67 | 89.26 | 60.56 | 16.19 | 91.77 | 97.35 |
| 5 | 62.69 | 93.31 | 91.24 | 92.34 | 87.08 | 75.49 | 91.72 | 94.44 |
| 6 | 67.71 | 98.39 | 96.84 | 97.89 | 95.62 | 91.98 | 95.87 | 98.56 |
| 7 | 61.25 | 100 | 100 | 100 | 98.75 | 95 | 100 | 100 |
| 8 | 87.94 | 98.36 | 98.84 | 99.86 | 96.99 | 97.34 | 98.45 | 99.93 |
| 9 | 50 | 98 | 100 | 100 | 96 | 86 | 100 | 100 |
| 10 | 24.35 | 86 | 86.44 | 88.73 | 84.17 | 66.23 | 90.14 | 95.75 |
| 11 | 95.67 | 89.26 | 90.61 | 95.05 | 89.66 | 76.67 | 95.99 | 98.66 |
| 12 | 32.43 | 80.93 | 77.57 | 83.14 | 70.62 | 44.88 | 88.99 | 97.76 |
| 13 | 43.35 | 98.16 | 96.86 | 99.41 | 91.51 | 74.92 | 97.14 | 98.05 |
| 14 | 98.05 | 95.94 | 97.97 | 97.92 | 96.16 | 88.99 | 98.44 | 99.84 |
| 15 | 54.2 | 85.09 | 80.03 | 91.72 | 82.33 | 58.59 | 92.53 | 97.39 |
| 16 | 27.12 | 99.86 | 97.53 | 98.9 | 97.26 | 73.7 | 97.4 | 98.63 |
| AA (%) | 52.91±11 | 91.12±1.6 | 90.75±1.5 | 94.41±0.8 | 87.53±4.3 | 70.76±4.3 | 95.3±1.2 | **98.1±0.4** |
| OA (%) | 63.84±6.2 | 89.52±1.9 | 89.33±1.3 | 93.11±0.7 | 86.72±4.1 | 71.18±4.3 | 94.69±1.2 | **98±0.3** |
| KC (×100) | 57±7.8 | 88.04±2.1 | 87.8±1.5 | 92.13±0.8 | 84.84±4.7 | 67.01±2.9 | 93.94±1.4 | **97.72±0.3** |

TABLE XIV
CLASS-WISE AND OVERALL CLASSIFICATION PERFORMANCE ON THE SALINAS DATASET

| Class | Classification Accuracy of Individual Classes | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | LC-DCNN [26] | FC-CNN [27] | DWTD [28] | IRSN [29] | MHSSM [30] | SSMamba [31] | SSBFNet [32] | CPTNet |
| 1 | 59.88 | 99.86 | 99.96 | 99.96 | 98.71 | 98.06 | 99.56 | 100 |
| 2 | 59.23 | 99.89 | 99.26 | 99.98 | 99.13 | 98.27 | 99.93 | 99.98 |
| 3 | 59.02 | 99.03 | 96.96 | 98.61 | 90.38 | 98.56 | 99.83 | 99.61 |
| 4 | 78.44 | 98.1 | 97.86 | 96.98 | 95.69 | 95.3 | 97.77 | 99.84 |
| 5 | 68.2 | 96.91 | 98.62 | 96.45 | 85.58 | 94.12 | 99.25 | 97.82 |
| 6 | 88.23 | 99.51 | 99.75 | 99.74 | 97.2 | 98.24 | 99.49 | 100 |
| 7 | 78.16 | 99.56 | 98.12 | 99.88 | 93.48 | 95.43 | 99.4 | 100 |
| 8 | 33.35 | 75.93 | 76.83 | 79.45 | 68.68 | 74.16 | 82.53 | 87.21 |
| 9 | 48.87 | 99.38 | 98.36 | 99.95 | 98.7 | 99.42 | 100 | 99.85 |
| 10 | 82.65 | 93.96 | 91.12 | 94.74 | 92.27 | 90.47 | 95.85 | 96.57 |
| 11 | 68.56 | 97.27 | 96.41 | 97.25 | 91.08 | 98.45 | 99.74 | 99.98 |
| 12 | 68.48 | 99.47 | 98.98 | 97.54 | 94.35 | 97.05 | 98.77 | 100 |
| 13 | 89.37 | 99.1 | 99.47 | 99.06 | 96.95 | 95.88 | 99.19 | 99.86 |
| 14 | 78.33 | 98.59 | 98.35 | 99.14 | 88.25 | 95.48 | 98.43 | 99.69 |
| 15 | 86.67 | 86.4 | 79.64 | 81.84 | 86.01 | 85.51 | 89.79 | 93.59 |
| 16 | 77.13 | 98.3 | 98.19 | 98.34 | 91.71 | 95.82 | 98.5 | 99.6 |
| AA (%) | 70.29±9.1 | 96.33±0.3 | 95.49±0.8 | 96.18±0.4 | 91.76±10.6 | 94.39±0.8 | 97.38±0.5 | **98.35±0.3** |
| OA (%) | 64.01±11.6 | 92.19±0.9 | 91.02±1.5 | 92.35±0.8 | 88.08±9.4 | 90.46±1 | 94.36±1.1 | **96.07±0.8** |
| KC (×100) | 60.58±12.4 | 91.33±1 | 90.03±1.7 | 91.49±0.9 | 86.84±10.3 | 89.42±1.2 | 93.73±1.2 | **95.63±0.9** |

TABLE XV
CLASS-WISE AND OVERALL CLASSIFICATION PERFORMANCE ON THE PAVIA CENTRE DATASET

| Class | Classification Accuracy of Individual Classes | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | LC-DCNN [26] | FC-CNN [27] | DWTD [28] | IRSN [29] | MHSSM [30] | SSMamba [31] | SSBFNet [32] | CPTNet |
| 1 | 88.28 | 99.45 | 98.09 | 99.85 | 99.28 | 99.41 | 99.61 | 99.24 |
| 2 | 84.83 | 91.25 | 89.77 | 90.84 | 83.38 | 89.13 | 92.3 | 95.02 |
| 3 | 93.88 | 93.08 | 92.33 | 90.85 | 89.5 | 87.21 | 92.68 | 95.67 |
| 4 | 95.91 | 95.68 | 94.36 | 92.88 | 93.95 | 93.43 | 97.83 | 99.16 |
| 5 | 88.66 | 87.72 | 91.32 | 86.16 | 83.56 | 85.76 | 93.4 | 95.5 |
| 6 | 94.76 | 97.75 | 94.94 | 92.22 | 86.26 | 89.12 | 96.94 | 99.39 |
| 7 | 83.35 | 92.57 | 92.17 | 93.14 | 90.86 | 89.84 | 96.12 | 97.03 |
| 8 | 87.98 | 99.1 | 98.68 | 96.24 | 94.36 | 93.28 | 97.82 | 99.35 |
| 9 | 98.29 | 99.55 | 98.27 | 92.16 | 86.87 | 87.89 | 96.81 | 99.62 |
| AA (%) | 90.66±6.5 | 95.13±1.4 | 94.44±1.3 | 92.7±1.3 | 89.78±9 | 90.56±2.3 | 95.94±1.3 | **97.77±0.7** |
| OA (%) | 88.64±15.4 | 97.77±0.8 | 96.87±1.1 | 96.48±0.8 | 94.59±5.7 | 94.81±1.4 | 97.88±0.8 | **98.72±0.3** |
| KC (×100) | 85.81±17.7 | 96.84±1.1 | 95.59±1.5 | 95.02±1.1 | 92.4±8 | 92.72±1.9 | 97±1.1 | **98.2±0.4** |

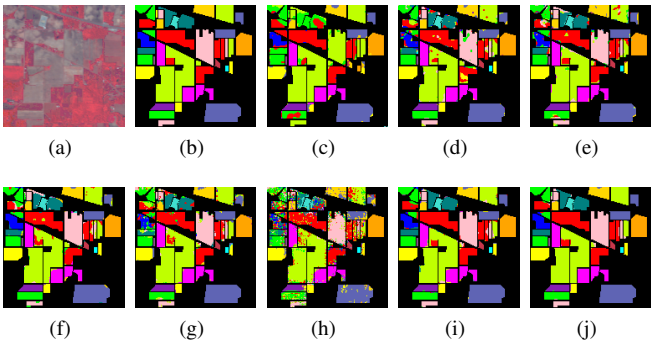

Fig. 7. Classification Maps for Indian Pines Dataset. (a) False Color Image (b) Ground Truth (c) LC-DCNN [26] (d) FC-CNN [27] (e) DWTD [28] (f) IRSN [29] (g) MHSSM [30] (h) SSMamba [31] (i) SSBFNet [32] (j) CPTNet.

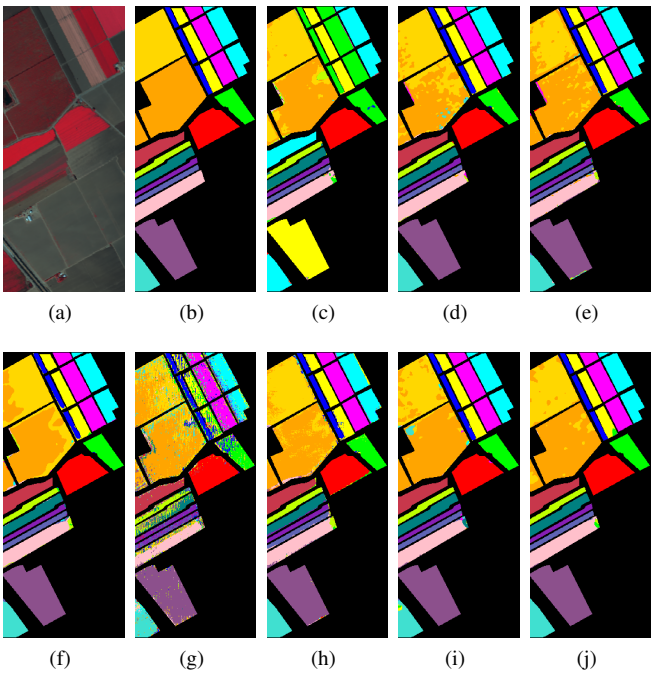

Fig. 8. Classification Maps for Salinas Dataset. (a) False Color Image (b) Ground Truth (c) LC-DCNN [26] (d) FC-CNN [27] (e) DWTD [28] (f) IRSN [29] (g) MHSSM [30] (h) SSMamba [31] (i) SSBFNet [32] (j) CPTNet.

TABLE XVI
AVERAGE CLASS-WISE F1 SCORES COMPUTED FROM GROUND-TRUTH AND PREDICTED CLASSIFICATION MAPS

| **Method** | Average of class-wise F1 Scores | | |
|---|---|---|---|
| | Indian Pines | Salinas | Pavia Centre |
| LCDCNN [26] | 0.6249 | 0.5625 | 0.8319 |
| FCCNN [27] | 0.8982 | 0.9586 | 0.9556 |
| DWTD [28] | 0.9081 | 0.9605 | 0.9435 |
| IRSN [29] | 0.9305 | 0.9497 | 0.897 |
| MHHSM [30] | 0.9005 | 0.6791 | 0.9145 |
| SSMamba [31] | 0.7317 | 0.9333 | 0.8843 |
| SSBFNet [32] | 0.9455 | 0.9699 | 0.9654 |
| CPTNet | 0.982 | 0.9827 | 0.9818 |

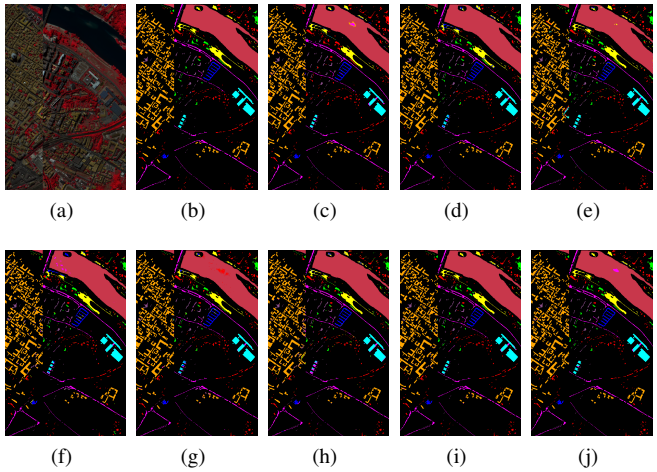

Fig. 9. Classification Maps for Pavia Centre Dataset. (a) False Color Image (b) Ground Truth (c) LC-DCNN [26] (d) FC-CNN [27] (e) DWTD [28] (f) IRSN [29] (g) MHSSM [30] (h) SSMamba [31] (i) SSBFNet [32] (j) CPTNet.

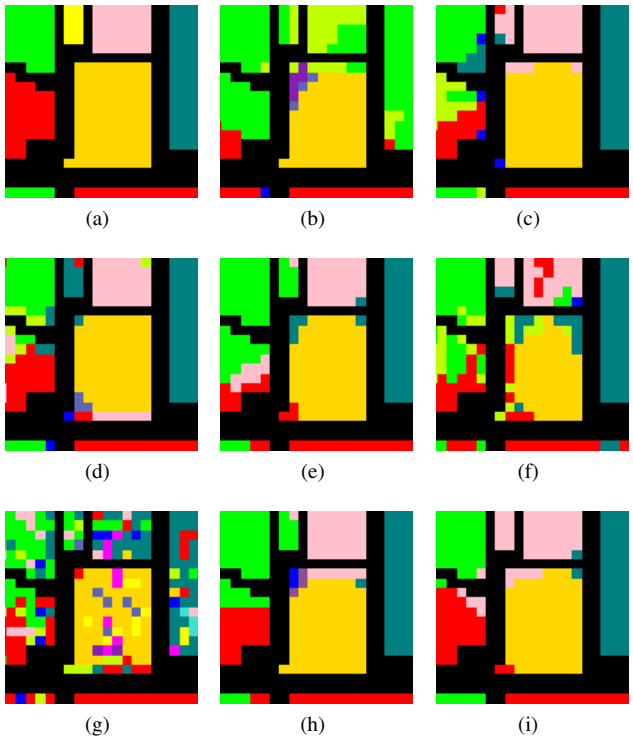

Fig. 10. Zoomed-In Classification Maps for Indian Pines Dataset. (a) Ground Truth (b) LC-DCNN [26] (c) FC-CNN [27] (d) DWTD [28] (e) IRSN [29] (f) MHSSM [30] (g) SSMamba [31] (h) SSBFNet [32] (i) CPTNet.

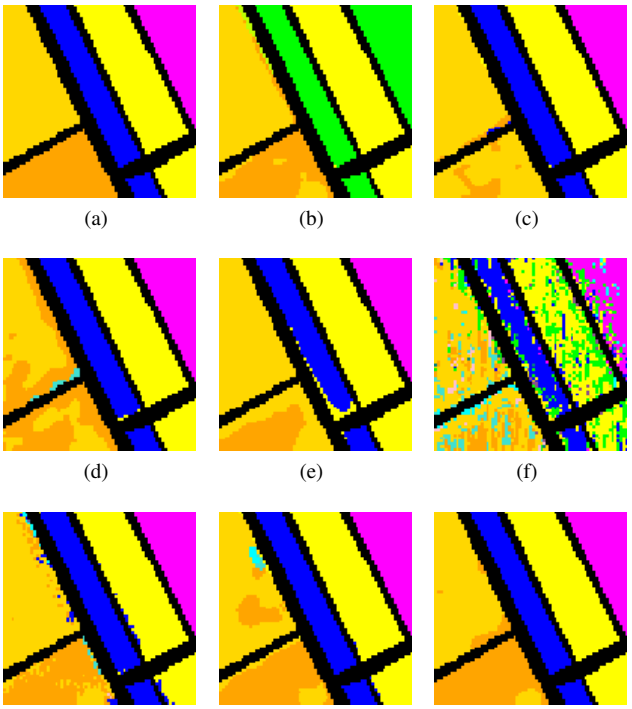

Fig. 11. Zoomed-In Classification Maps for Salinas Dataset. (a) Ground Truth (b) LC-DCNN [26] (c) FC-CNN [27] (d) DWTD [28] (e) IRSN [29] (f) MHSSM [30] (g) SSMamba [31] (h) SSBFNet [32] (i) CPTNet.

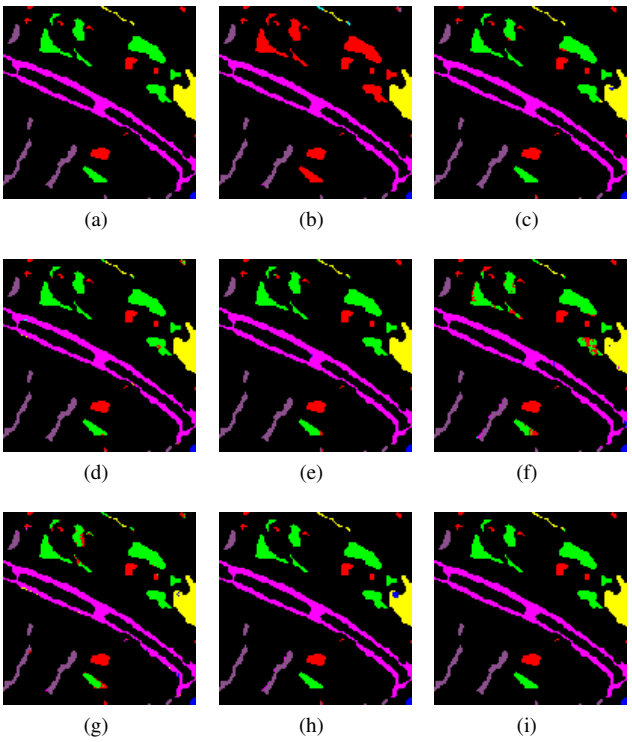

Fig. 12. Zoomed-In Classification Maps for Pavia Centre Dataset. (a) Ground Truth (b) LC-DCNN [26] (c) FC-CNN [27] (d) DWTD [28] (e) IRSN [29] (f) MHSSM [30] (g) SSMamba [31] (h) SSBFNet [32] (i) CPTNet.

competitive results for other datasets. Wavelet-based DWTD [28] obtained the fourth-best results against the Pavia Centre dataset, and it produced moderate results against the Salinas and Indian Pines datasets. Inverse convolution-based IRSN [29] provided the third-best results for the Indian Pines and Salinas datasets, along with moderate results for the Pavia Centre dataset. The state space mamba model MHSSM [30] achieved moderate results for Indian Pines dataset. Another state space-based method, SSMamba [31], obtained competitive results for the Pavia Centre and Salinas datasets. Transformer-based SSBFNet [32] achieved second-best results for all three datasets. Its improved performance is due to its lightweight transformer architecture, which allows for efficient exploration of complex global relationships from input hyperspectral samples. Compared to the respective second-best methods, the proposed CPTNet achieved 1.71% more overall accuracy for the Salinas dataset, 3.31% better accuracy for the Indian Pines dataset, and 0.84% better for the Pavia Centre dataset. Its better classification results are due to effective extraction of spectral dependencies by pixel-level transformer units, innovative input feature embedding by combining PCA with sparse fully connected layers, and effective spatial fusion of spectral features of all pixels in the input patch.

### *E. Computational Analysis*

Table XVII presents computational aspects of existing state-of-the-art methods and the proposed CPTNet with various transformer levels. The Indian Pines dataset with patch size 11 is used to obtain all metrics presented in XVII. The proposed CPTNet achieved the highest classification performance in terms of 98% overall accuracy, while the 5-level CPTNet outperforms existing methods even with the least number of parameters and model memory. with 5 transformer levels required the least number of parameters and required minimum model memory. And 5-level CPTNet outperforms existing methods in terms of classification performance. The inference time of CPTNet models is more compared to other methods. It is primarily due to CPTNet involving multi-level transformer operations on each pixel in the input patch. As CPTNet involves feature extraction at the pixel level, transformer units are operated on 5D inputs of shape $(Batch, w, w, 5, d)$, where traditional transformer units operated on 3D inputs $(Batch, N, d)$. So CPTNet relies on custom-made transformer functions that can operate on 5D inputs instead of inbuilt transformer functions, which are highly optimized for parallel GPU operations. Further multi-head attention calculation involves operations like tensor reshaping, tensor splitting, and tensor transposing. Such tensor operations involving a 5D tensor introduce additional memory movement bottlenecks. Further CPTNet involves repeated attention calculation for pixels at spatial positions in a patch. The slow inference time of CPTNet can be attributed to these computational aspects.

The primary object of this work is extending the maximum classification performance under limited training samples. The proposal achieves this objective with a reasonable efficiency-performance trade-off. In general, for many practical hyperspectral imaging systems, especially for satellites, the primary

objective is scene capturing and sending captured data to a ground data collection centre, rather than performing complete real-time classification. So the inference times (12-41ms) for the variants of the proposed method are feasible for practical hyperspectral classification applications involving both near real-time and offline approaches. Further lighter variants of CPTNet can be derived by reducing transformer levels or by adjusting sub-array length and number of retained PCA components. These lighter versions of CPTNet serve the needs of less computational requirements without much compromising classification performance.

TABLE XVII
COMPUTATIONAL EFFICIENCY AND MODEL COMPLEXITY ANALYSIS

| Method | Parameters (Thousands) | FLOPs (Millions) | Inference Time(ms) | Model Memory(KB) | OA (%) |
|---|---|---|---|---|---|
| LCDCNN [26] | 255.39 | 24.56 | 1.87 | 997.63 | 63.84±6.2 |
| FCCNN [27] | 995.46 | 20.16 | 3.1 | 3800 | 89.52±1.9 |
| DWTD [28] | 97.34 | 38.25 | 4.09 | 380.25 | 89.33±1.3 |
| IRSN [29] | 53.04 | 1.75 | 7 | 207.18 | 93.11±0.7 |
| MHHSM [30] | 67.49 | 8.66 | 12.83 | 263.64 | 86.72±4.1 |
| SSBFNet [32] | 178.46 | 11.49 | 1.85 | 690 | 94.69±1.2 |
| CPTNet (5L) | 31.04 | 37.67 | 12.49 | 121.23 | 97.34±0.7 |
| CPTNet (10L) | 60.04 | 74.81 | 19.93 | 234.52 | 97.83±0.3 |
| CPTNet (15L) | 89.04 | 111.95 | 26.37 | 347.8 | 97.86±0.3 |
| CPTNet (20L) | 118.04 | 149.09 | 31.39 | 461.08 | 97.88±0.3 |
| CPTNet (25L) | 147.04 | 186.23 | 36.7 | 574.36 | 98±0.3 |
| CPTNet (30L) | 176.04 | 223.37 | 41.39 | 687.64 | 97.99±0.3 |

## V. FUTURE WORK

Currently, the largest hyperspectral scene on which CPTNet has been evaluated is the Pavia Centre dataset (1096×715), while the maximum number of classes considered in evaluation is 16 (Indian Pines and Salinas). Because CPTNet requires pixel-level feature extraction from all pixels in each patch, its computational feasibility must be tested on a dataset with extremely large spatial dimensions. Future research can concentrate on adapting CPTNet to very large-scale scenes with a large number of classes. Hyperspectral datasets are captured using various types of sensors, which vary in spatial resolution, spectral resolution, and number of bands. In the current experimental setup, CPTNet is trained and evaluated on each dataset independently. Therefore, its ability to generalize on combined datasets from different sensors has not been demonstrated. Future works can focus on extending CPTNet for a generalized classification framework that works on combined datasets derived from multiple sensors with different spatial-spectral characteristics. In this paper, CPTNet was evaluated on only benchmark datasets like the Indian Pines, Salinas, and Pavia Centre datasets. Because these datasets provide cleaner samples, the robustness of CPTNet during high-noise acquisition is not established. Furthermore, its performance under conditions such as noise and missing or distorted spectra must be studied. Future work can focus on potential robustness improvements after evaluating against various sensor artifacts. As CPTNet involves pixel-level feature extraction followed by distance-based spatial fusion of all pixels in a patch, it allows for pretraining with incremental patch sizes from 1 to the final patch size. CPTNet can be pretrained with patch size 1, then with patch sizes 3, 5, and 7 until the final patch size. This pretraining could improve the model's final feature extraction capabilities. As presented in Table XVII, inference time for CPTNet is high compared to other methods. Even though inference time is affected by external factors like programming libraries and platforms used to implement the model, future work can focus on reducing the inference time of CPTNet. In proposed CPTNet, cascade pixel transfer units extract features from all pixels of the input patch, and these features are aggregated using distance-based weights. Pixels of an input patch can be grouped into superpixels using unsupervised methods like K-means clustering and agglomerative hierarchical clustering. Computations can be reduced by using superpixels instead of all individual pixels for feature extraction and subsequent feature fusion.

## VI. CONCLUSION

This paper presented CPTNet, a transformer-based architecture for hyperspectral image classification developed to provide effective results with a small number of training samples. The proposed CPTNet makes use of transformers in conjunction with informative input vectors generated by PCA on sub-arrays of pixel spectral values. By integrating PCA-based spectral transformation and a sparsely fully connected layer, CPTNet generates effective input spectral embeddings for pixel-transformer learning. The cascade transformer units in CPTNet capture complex global spectral dependencies for each pixel, while distance-driven spatial fusion optimally consolidates contextual information surrounding the center pixel. Datasets such as Indian Pines, Pavia Centre, and Salinas are employed to evaluate the performance of the proposed CPTNet. The classification performance of the proposed CPTNet is a significant improvement as compared to other methods in terms of metrics like Kappa coefficient (KC), Overall Accuracy (OA), and Average Accuracy (AA). Notably, CPTNet produced robust classification results after training on a small number of training samples and testing on a large number of testing samples, demonstrating its suitability for practical hyperspectral applications with limited labeled data.

## REFERENCES

[1] S. Peyghambari and Y. Zhang, "Hyperspectral remote sensing in lithological mapping, mineral exploration, and environmental geology: An updated review," in *Journal of Applied Remote Sensing*, vol. 15, no. 3, p. 031501, 2021, doi: 10.1117/1.JRS.15.031501.

[2] U. Heiden, W. Heldens, S. Roessner, K. Segl, T. Esch, and A. Mueller, "Urban structure type characterization using hyperspectral remote sensing and height information," in *Landscape and Urban Planning*, vol. 105, no. 4, pp. 361–375, 2012, doi: 10.1016/j.landurbplan.2012.01.001.

[3] W. Yang, T. Nigon, Z. Hao, G. D. Paiao, F. G. Fernández, D. Mulla, and C. Yang, "Estimation of corn yield based on hyperspectral imagery and convolutional neural network," in *Computers and Electronics in Agriculture*, vol. 184, p. 106092, 2021, doi: 10.1016/j.compag.2021.106092.

[4] T. Ishida, J. Kurihara, F. A. Viray, S. B. Namuco, E. C. Paringit, G. J. Perez, Y. Takahashi, and J. J. Marciano, "A novel approach for vegetation classification using UAV-based hyperspectral imaging," in *Computers and Electronics in Agriculture*, vol. 144, pp. 80–85, 2018, doi: 10.1016/j.compag.2017.11.027.

[5] K. Nagasubramanian, S. Jones, A. K. Singh, S. Sarkar, A. Singh, and B. Ganapathysubramanian, "Plant disease identification using explainable 3D deep learning on hyperspectral images," in *Plant Methods*, vol. 15, p. 98, 2019, doi: 10.1186/s13007-019-0479-8.

[6] P. Duan, X. Kang, P. Ghamisi, and S. Li, "Hyperspectral remote sensing benchmark database for oil spill detection with an isolation forest-guided unsupervised detector," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–11, 2023, doi: 10.1109/TGRS.2023.3268944.

[7] S. Armoogum, D. A. Dewi, V. Armoogum, N. Melanie, and T. B. Kurniawan, "Deep Learning-Based Loan Approval Prediction Using Artificial Neural Network (ANN) and Feature Importance Analysis," in *J. Digit. Mark. Digit. Curr.*, vol. 3, no. 1, pp. 38–56, Feb. 2026. doi: 10.47738/jdmdc.v3i1.55.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All You Need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Long Beach, CA, USA, 2017, pp. 6000–6010, doi: 10.48550/arXiv.1706.03762.

[9] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2094–2107, 2014, doi: 10.1109/JSTARS.2014.2329330.

[10] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3639–3655, 2017, doi: 10.1109/TGRS.2016.2636241.

[11] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 844–853, 2017, doi: 10.1109/TGRS.2016.2616355.

[12] H. Lee and H. Kwon, "Going deeper with contextual CNN for hyperspectral image classification," in *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 4843–4855, 2017, doi: 10.1109/TIP.2017.2725580.

[13] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 847–858, 2018, doi: 10.1109/TGRS.2017.2755542.

[14] Z. Ge, G. Cao, X. Li, and P. Fu, "Hyperspectral image classification method based on 2D–3D CNN and multibranch feature fusion," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 5776–5788, 2020, doi: 10.1109/JSTARS.2020.3024841.

[15] S. Ghaderizadeh, D. Abbasi-Moghadam, A. Sharifi, N. Zhao, and A. Tariq, "Hyperspectral image classification using a hybrid 3D-2D convolutional neural networks," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 7570–7588, 2021, doi: 10.1109/JSTARS.2021.3099118.

[16] U. Rahardja and Q. Aini, "Enhancing Blockchain Security Through Smart Contract Vulnerability Classification Using BiLSTM and Attention Mechanism," in *Journal of Current Research in Blockchain*, vol. 3, no. 1, pp. 28–45, Feb. 2026, doi: 10.47738/jcrb.v3i1.56.

[17] M. Furqan, N. Katuk, and D. Hartama, "Multiclass Skin Lesion Classification Algorithm Using Attention-Based Vision Transformer With Metadata Fusion," in *Journal of Applied Data Sciences*, vol. 7, no. 1, pp. 203–217, 2025, doi: 10.47738/jads.v7i1.1017.

[18] Y. Dong, Q. Liu, B. Du, and L. Zhang, "Weighted feature fusion of convolutional neural network and graph attention network for hyperspectral image classification," in *IEEE Transactions on Image Processing*, vol. 31, pp. 1559–1572, 2022, doi: 10.1109/TIP.2022.3144017.

[19] M. E. Paoletti, S. Moreno-Álvarez, Y. Xue, J. M. Haut, and A. Plaza, "AAtt-CNN: Automatic attention-based convolutional neural networks for hyperspectral image classification," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–18, 2023, doi: 10.1109/TGRS.2023.3272639.

[20] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *arXiv preprint arXiv:2010.11929*, 2021, doi: 10.48550/arXiv.2010.11929.

[21] H. Yan, E. Zhang, J. Wang, C. Leng, A. Basu, and J. Peng, "Hybrid Conv-ViT network for hyperspectral image classification," in *IEEE Geoscience and Remote Sensing Letters*, vol. 20, pp. 1–5, 2023, doi: 10.1109/LGRS.2023.3287277.

[22] L. Sun, G. Zhao, Y. Zheng, and Z. Wu, "Spectral–spatial feature tokenization transformer for hyperspectral image classification," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022, doi: 10.1109/TGRS.2022.3144158.

[23] S. K. Roy, A. Deria, C. Shah, J. M. Haut, Q. Du, and A. Plaza, "Spectral–spatial morphological attention transformer for hyperspectral image classification," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–15, 2023, doi: 10.1109/TGRS.2023.3242346.

[24] X. Zheng, H. Sun, X. Lu, and W. Xie, "Rotation-Invariant Attention Network for Hyperspectral Image Classification," *IEEE Transactions on Image Processing*, vol. 31, pp. 4251–4265, 2022, doi: 10.1109/TIP.2022.3177322.

[25] D. Hong, Z. Han, J. Yao, L. Gao, B. Zhang, A. Plaza, and J. Chanussot, "SpectralFormer: Rethinking Hyperspectral Image Classification With Transformers," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022, doi: 10.1109/TGRS.2021.3130716.

[26] S. Shinde and H. Patidar, "Hyperspectral image classification for vegetation detection using lightweight cascaded deep convolutional neural network," in *Journal of the Indian Society of Remote Sensing*, vol. 51, pp. 2159–2166, 2023, doi: 10.1007/s12524-023-01754-5.

[27] 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," in *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022, doi: 10.1109/LGRS.2020.3043710.

[28] J. Xu, J. Zhao, and C. Liu, "An effective hyperspectral image classification approach based on discrete wavelet transform and dense CNN," in *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022, doi: 10.1109/LGRS.2022.3181627.

[29] M. Cihan and M. Ceylan, "IRSN: Involutional residual spectral network for hyperspectral image classification," in *IEEE Geoscience and Remote Sensing Letters*, vol. 21, pp. 1–5, 2024, doi: 10.1109/LGRS.2023.3341497.

[30] M. Ahmad, M. H. F. Butt, M. Usama, H. A. Altuwaijri, M. Mazzara, S. Distefano, and A. M. Khan, "Multi-head spatial-spectral mamba for hyperspectral image classification," in *Remote Sensing Letters*, vol. 16, no. 4, pp. 339–353, 2025, doi: 10.1080/2150704X.2025.2461330.

[31] M. Ahmad, M. Usama, M. Mazzara, and S. Distefano, "WaveMamba: Spatial-Spectral Wavelet Mamba for Hyperspectral Image Classification," in *IEEE Geoscience and Remote Sensing Letters*, vol. 22, pp. 1–5, 2025, doi: 10.1109/LGRS.2024.3506034.

[32] H. Wu, X. Yu, and Z. Zeng, "SSBFNet: A Spectral–Spatial Fusion With BiFormer Network for Hyperspectral Image Classification," in *The Visual Computer*, vol. 41, no. 8, pp. 5391–5404, 2025, doi: 10.1007/s00371-024-03728-1.

[33] P. Gamba, "A collection of data for urban area characterization," *Proc. IGARSS 2004 - IEEE Int. Geosci. Remote Sens. Symp.*, vol. 1, 2004, p. 72, doi: 10.1109/IGARSS.2004.1368947.

[34] M. Graña, M. A. Veganzons, and B. Ayerdi, "Hyperspectral Remote Sensing Scenes." www.ehu.eus. Accessed: Nov. 5, 2024. [Online.] Available: https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes.

[35] M. F. Baumgardner, L. L. Biehl, and D. A. Landgrebe, "220 Band AVIRIS hyperspectral image data set: June 12, 1992 Indian Pine test site 3," *Purdue University Research Repository (PURR)*, Sep. 2015, doi: 10.4231/R7RX991C. [Online]. Available: purr.purdue.edu/publications/1947/1.


**Biri Chanakya Reddy** received his B.Tech degree in 2012 from Jawaharlal Nehru Technological University, Hyderabad, India. He received his M.Tech degree in 2015 from IIT Guwahati, Guwahati, India. He is currently pursuing a Ph.D. in National Institute of Technology, Tiruchirappalli, India. His research interests include deep learning, computer vision, and hyperspectral image analysis.

**Subbian Deivalakshmi** (Member,IEEE ) received her Ph.D. degree from the Department of Electronics and Communication Engineering, National Institute of Technology, Tiruchirappali, Tamil Nadu, India, in 2017. She is currently an Associate Professor with Department of Electronics and Communication Engineering, National Institute of Technology, Tiruchirappalli. Her research interests span signal and image processing, biomedical imaging, machine learning, image restoration techniques, segmentation techniques for agricultural and biomedical images, remote sensing image analysis, and hyperspectral image processing.