Spectral–Spatial Morphological Attention Transformer for Hyperspectral Image Classification

**Abstract:**

In recent years, convolutional neural networks (CNNs) have drawn significant attention for the classification of hyperspectral images (HSIs). Due to their self-attention mechanism, the vision transformer (ViT) provides promising classification performance compared to CNNs. Many researchers have incorporated ViT for HSI classification purposes. However, its performance can be further improved because the current version does not use spatial–spectral features. In this article, we present a new morphological transformer (morphFormer) that implements a learnable spectral and spatial morphological network, where spectral and spatial morphological convolution operations are used (in conjunction with the attention mechanism) to improve the interaction between the structural and shape information of the HSI token and the CLS token. Experiments conducted on widely used HSIs demonstrate the superiority of the proposed morphFormer over the classical CNN models and state-of-the-art transformer models. The source will be made available publicly at https://github.com/mhaut/morphFormer .

**SECTION I.**

Introduction

Hyperspectral images (HSIs) contain information in contiguous wavelengths [1], [2], [3]. HSIs have been adopted in many application areas of remote sensing (RS) and Earth observation (EO), such as urban planning, vegetation monitoring, and crop management [4], [5], [6]. HSIs have particularly been used in EO tasks, such as desertification or climate change studies. In addition to land cover classification tasks [2], [7], [8], [9], other areas in which HSIs have been widely exploited include forestry [10], target/object detection, mineral exploration, and mapping [11], [12], environmental monitoring [13], disaster risk management, and biodiversity conservation. The popularity of HSIs is due to rich spectral and spatial information [14].

From the point of view of RS imaging technology, the affinity of spectral and spatial resolution is quite critical [15]. Spatial resolution is often limited by the very high spectral resolution of HSIs, and it may negatively affect land cover classification for complex scenes. For example, hyperspectral (HS) data do not provide proper information about the elevation and size of different structures of interest in particular application domains [14], [16]. Most conventional classifiers often process HSIs depending on spectral information and disregard spatial information among adjacent pixels. To solve this issue, different techniques can be implemented to incorporate both spatial and spectral information. With spatial processing, the size and shape of different objects can be determined resulting in better classification accuracy. In the following, we summarize some of the most relevant methods for exploiting HSI data, outlining their pros and cons.

In HSI classification, conventional classifiers have been widely utilized, even in the presence of limited training samples [3], [17], [18]. In general, these techniques include two stages. First, they reduce the dimensionality of the HSI data and extract some informative features. Then, spectral classifiers are fed with such features for classification purposes [2], [7], [19], [20], [21], [22]. In scenarios with limited training samples, support vector machines (SVMs) with nonlinear kernels have been widely used [23]. Moreover, the extreme learning machine (ELM) has been broadly used to extract features from unbalanced training sets. Li et al. [24] implemented an ELM to classify HSIs by extracting local binary patterns (LBPs) for classification. They demonstrated that ELMs can obtain better classification results than SVMs. The random forest (RF) was also utilized for the classification of HSIs due to its discriminative power [2]. However, the aforementioned classifiers face challenges when the training data are not representative, suffering from data fitting problems. This is because these classifiers consider HSIs as an assembly of measurements in the spectral domain, without considering their arrangement in the spatial domain. Classifiers based on spatial–spectral information significantly enhance the results of spectral-based classifiers with the inclusion of spatial data, such as the size and shape of various objects. In addition, spectral-based classifiers are more sensitive to noise compared to their spatial–spectral counterparts [2], [25].

Deep learning (DL) methods have attracted significant attention for multimodal data integration [26] in RS data classification [27]. A wide variety of fragmented datasets can be intelligently analyzed with DL methods. More recently, a unified and general DL framework was developed by Hong et al. [28] for classification of RS imagery. 1-D convolutional neural networks (CNNs) (CNN1Ds) [29], 2-D CNNs (CNN2Ds) [30], and 3-D CNNs (CNN3Ds) [31] have demonstrated success in the classification of HSI data.

Residual networks (ResNets) were introduced by He et al. [32]. These models have a minimum loss of information after each operation of the convolutional layers to reduce the gradient vanishing problem [32]. Zhong et al. [33] introduced a spatial–spectral ResNet (SSRN) for utilizing both spatial and spectral information to obtain enhanced classification performance. Roy et al. adopted a lightweight paradigm with the extraction of spatial and spectral features via the squeeze-and-excitation ResNet that can be added with a bag-of-features learning mechanism to accurately obtain the final classification results [34], [35]. Zhu et al. [36] incorporated other channel and spatial attention layers inside the SSRN architecture for extracting discriminative features. To take full advantage of ResNets, they can be extended to form even more complex models, such as the inclusion of adaptive kernels [17], lightweight spatial–spectral attention based on squeeze-and-excitation [35], and pyramidal ResNets [37]. Rotation-equivariant CNNs [38], gradient centralized convolutions [1], [39], and lightweight heterogeneous kernel convolutions [40] also enable efficient classification and feature extraction. Generative adversarial networks (GANs), on the other hand, may help with mitigating the class-imbalance problem in HSI classification [41], [42].

Despite their apparent ability to extract contextual information in the spatial domain, CNNs cannot easily sequentially incorporate attributes, in particular, long- and middle-term dependencies. As a consequence, their performance in HSI classification may be affected by the presence of classes with similar spectral signatures, making it difficult to extract diagnostic spectral attributes. The spectral signatures in HSIs can also be modeled using recursive neural networks (RNNs), which accumulate them in a band-by-band fashion. This is important to learn long-term dependencies, as the gradient vanishing problem may further complicate the interpretation of spectrally salient changes [43]. However, RNNs are not suitable for the simultaneous training of models because HSIs generally contain many samples, which limits classifier performance. Our work addresses the aforementioned limitations by rethinking HSI data classification using transformers.

As cutting-edge backbone networks, transformers utilize self-attention techniques to process and analyze sequential data more efficiently [44]. In recent years, several new transformer models have been developed including SpectralFormer [45], which is capable of learning spectral information by creating a transformer encoder module and utilizing adjacent bands. Transformers excel at characterizing spectral signatures, yet they are not able to model local semantic elements or utilize spatial information effectively. He et al. [46] proposed a bidirectional encoder representation for a transformer that incorporates flexible and dynamic input regions for pixel-based classification of HSIs. Zhong et al. [47] proposed a factorized architecture search (FAS) framework, which enables a stable and fast spectral–spatial transformer architecture search subject to find out the optimal architecture settings for the HSI classification task. To further improve the classification performance of HSIs, Sun et al. [48] introduced spatial and spectral tokenization of feature representations in the encoder, which helps to extract local spatial information and establish long-range relations between neighboring sequences. Yang et al. [49] utilize an adaptive 3-D convolution projection module to incorporate spatial–spectral information in an HSI transformer classification network. The above transformer models are designed based on HSI data and utilize spectral–spatial feature representation mechanisms. Roy et al. [50] recently developed a multimodal fusion transformer (MFT) to extract features from HSIs and fuse them with a CLS token derived from light detection and ranging (LiDAR) data to enhance the joint classification performance.

Mathematical morphology (MM) is a theory to analyze geometrical structures, based on topology, lattice theory, set theory, and random functions. Researchers have utilized MM-based techniques such as attribute profiles (APs) and extended morphological profiles (EPs) to extract spatial features and classify HSI data more accurately [16], [51], [52]. Rasti et al. [53] applied total variation component analysis for feature fusion to improve the joint extraction of EPs. Merentis et al. [54] used an RF classifier to classify HSI data with an automated fusion approach. By exploiting APs and EPs, MM has been successfully applied to extract features from RS data [55], [56], [57], [58]. In EPs and APs, several handcrafted characteristics are collected by sequentially performing dilation and erosion operations using an extensive set of structuring elements (SEs). There are a few limitations common to both EPs and APs, however. Specifically, the shape of the SE is fixed. In addition, the SEs can only obtain information about the size of existing objects but are unable to collect information about the shape of arbitrary item boundaries in complicated environments. To circumvent these restrictions, Roy et al. [3] introduced a spectral–spatial CNN based on morphological erosion and dilation operations for HSI classification. In this work, a spatial and spectral morphological block was created for extracting discriminative and robust spatial and spectral information from HSIs using its own trainable SEs in the erosion and dilation layers.

Although MM has been successfully applied in RS for extracting the spatial information based on techniques such as EPs or APs, the SEs are nontrainable [55], [56], [57], [58] and unable to capture dynamic features. If the EPs or APs are replaced with learnable MM operations, the resulting networks can be more capable of learning subtle features. Conventional transformer models use self-attention to highlight the most important features. If MM operations are combined with the transformer, the model may be able to learn intrinsic shape information and use this information in the self-attention block for better feature extraction, leading to higher classification accuracies.

With the aforementioned rationale in mind, a new morphological fusion transformer encoder is introduced in this work, where the input patch is passed through two different morphological blocks simultaneously. The results provided by these blocks are concatenated, and the CLS token is added to the concatenated patch. The objective of our morphological transformer (morphFormer) model is to learn the spectral–spatial information from the patch embeddings of the HSI inputs, as well as to enrich the description of the abstract provided by the CLS token without adding significant computational complexity.

The main contributions of this work can be summarized as follows.

1. We provide a new learnable classification network based on a spectral–spatial morphFormer that conducts spatial and spectral morphological convolutions via dilation and erosion operators.
2. We introduce a new attention mechanism for efficiently fusing the existing CLS tokens and information obtained from HSI patch tokens into a new token that carries out morphological feature fusion.
3. We conduct experiments on four public HSI datasets by comparing the proposed network with other state-of-the-art approaches. The obtained results reveal the effectiveness of the proposed approach.

The remainder of this article is organized as follows. Section II describes the proposed method in detail. Section III discusses our experimental results. Section IV concludes this article.

**SECTION II.**

Proposed Method

**A. Convolutional Networks for Feature Learning**

CNNs exhibit promising performance in HSI classification due to their ability to automatically extract contextual features. Since HSIs have numerous spectral bands, it is possible to take advantage of CNNs for controlling the depth of the output feature maps. CNNs have already been proved to be effective in capturing high-level features independently of the data source modality. Our proposed model uses CNNs for extracting high-level abstract features to be used by the transformer. The spectral dimensions of the HSI are reduced by the CNN.

Our proposed model utilizes sequential layers of Conv3D and HetConv2D for extracting robust and discriminative features from HSIs. The original data are arranged in subcubes XHSI (with dimensionality 11×11×B ) that are reshaped into (1×11×11×B ) and used as input to a Conv3D layer with kernel size (3×3×9 ) and padding (1×1×0 ). Padding is used so that the spatial size of the output image is the same as that of the input image. The HetConv2D block follows the Conv3D layer and consists of two Conv2D layers working in parallel. One of the Conv2D layers is used for groupwise convolution, and the other one is used for pointwise convolution. HetConv2D utilizes two kernels of different sizes to extract multiscale information. The outputs obtained from these two convolutions are combined in an elementwise fashion (⊕ ) and returned as output

Xin=Xout=Reshape(Conv3D(Reshape(XHSI)))Conv2D(Xin,k1,g1,p1)⊕Conv2D(Xin,k2,g2,p2)(1)

View SourceRight-click on figure for MathML and additional features.where k1=3 , g1=4 , p1=1 , k2=1 , g2=1 , and p2=0 . The output shape of the Conv3D layer is (8×11×11×(B−8) ), and that of the HetConv2D block is (11×11×64 ). Batch normalization (BN) [59] and ReLU activation layers are used after the Conv3D layer and the HetConv2D block. If only a few limited training samples are available, the overfitting phenomenon may arise. To address this issue and accelerate the training performance, we use a BN. ReLU also helps in smoothing the back-propagation of the loss by introducing nonlinearity.

**B. Image Tokenization and Position Embedding**

HSIs contain spatial and spectral features which can provide highly discriminative information that can lead to higher classification accuracies. Patch tokens of shape (1×64 ) each are obtained by flattening HSI subcubes of shape [(11×11)×64] as follows:

Xflat=T(Flatten(Xout))(2)

View SourceRight-click on figure for MathML and additional features.where T(⋅) is a transpose function and Xflat∈R121×64 . The tokenization [48] operation is used to select n from 121 patches as follows:

XWa=XWb=softmax(T(Xflat.WaH))Xflat.WbH(3)

View SourceRight-click on figure for MathML and additional features.where WaH∈R64×n , WbH∈R64×64 , XWa∈Rn×121 , and XWb∈R121×64 . The tokenization operation uses two learnable weights to extract the key features

Xpatch=XWa.XWb(4)

View SourceRight-click on figure for MathML and additional features.where Xpatch∈Rn×64 . A total of (n+1) patches are obtained as described in [(5)](https://ieeexplore.ieee.org/document/#deqn5) by concatenating (⊙ ) the CLS token to the HSI patch tokens. The CLS token (Xcls ) is a learnable tensor, which is randomly initialized. To simplify the calculation of head dimensions, a size of 64 is used

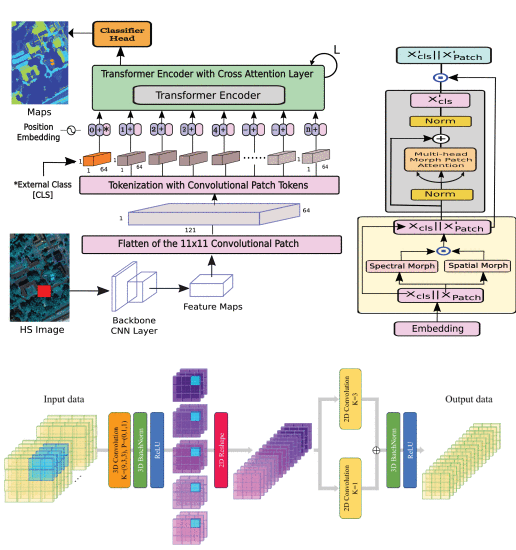
Xˆ=[Xcls⊙Xˆpatch].(5)

View SourceRight-click on figure for MathML and additional features.

The semantic textural information in the image patch tokens can be preserved by adding trainable position embeddings to the patch embeddings. Hence, a trainable position embedding is added to the created HSI patch tokens. Fig. 1 graphically illustrates the addition of position embeddings (in elementwise fashion) to the patches (1 to n+1 ). A dropout layer is used after this operation to reduce the effect of the vanishing gradient. The above procedure can be expressed as

X=DP(Xˆ⊕PE)(6)

View SourceRight-click on figure for MathML and additional features.where DP denotes a dropout layer with value of 0.1 and PE represents a learnable position embedding.

[[](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza1ab-3242346-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza1ab-3242346-large.gif)

**Fig. 1.**

In the upper row, we show the proposed HSI classification network such that the classification map of the proposed method contains (left) less noise than existing methods and (right) transformer encoder with a multihead patch attention mechanism. In the bottom row, we show the backbone of the proposed method.

[Show All](https://ieeexplore.ieee.org/document/10036472/figures)

**C. Spectral and Spatial Morphological Convolutions**

MM is a powerful technique for characterizing the intrinsic shape, structure, and size of objects in an image. The spectral and spatial morphological network presented here is designed based on dilation and erosion operations with SEs of size (s×s) .

A dilated image is produced by combining the input HSI patch tokens with SEs, selecting the pixel with the maximum value in the local neighborhood. As a result of the dilation procedure, the boundaries of the foreground objects of the HSI input patch token are broadened. In other words, the size of the kernel affects the size of the texture for various regions of an HSI patch token. The dilation process is represented by ⊞ and can be denoted by the following equation:

(Xpatch⊞Wd)(x,y)=max(i,j)∈ψ(Xpatch(x+i,y+j)+Wd(i,j))(7)

View SourceRight-click on figure for MathML and additional features.where ψ={(i,j)|i∈{1,2,3,…,s};j∈{1,2,3,…,s}} represents the elements of the kernel and Wd denotes the SEs used for the dilation operation.

Regarding the erosion operation, the output of the convolution with the SE selects the pixel with minimum value in the local neighborhood. This operation reduces the shape of the background object in the HSI patch token (as opposed to the dilation). Erosions can eliminate minor details and enlarge holes, making them distinguishable from each other in different texture regions. Let Xpatch∈Rk×k be an input HSI patch token of spatial size k×k , and let ⊟ represent the morphological erosion operation. The erosion operation can be defined as

(Xpatch⊟We)(x,y)=min(i,j)∈ψ(Xpatch(x+i,y+j)−We(i,j))(8)

View SourceRight-click on figure for MathML and additional features.where ψ={(i,j)|i∈{1,2,3,…,s};j∈{1,2,3,…,s}} represents the elements of the kernel and We denotes the SEs used for the erosion operation. It can be understood from the operations defined in [(7)](https://ieeexplore.ieee.org/document/#deqn7) and [(8)](https://ieeexplore.ieee.org/document/#deqn8) that the HSI patch tokens are shifted by i×j , as in the convolutional operation. The padding function is used to keep the input and output shapes of the objects. After applying the operations given in [(7)](https://ieeexplore.ieee.org/document/#deqn7) and [(8)](https://ieeexplore.ieee.org/document/#deqn8) to the HSI patch tokens, dilation and erosion maps can be, respectively, obtained.

A graphical visualization of the input and output images after the dilation and erosion operations (using an SE of size 3×3 ) is shown in Fig. 2(a) and (b). To obtain the morphological shape feature from the HSI patch tokens, a spatial morphological (SpatialMorph) block with primitive operations (e.g., dilation and erosion) is used. The SpatialMorph block comprises parallel branches of dilation and erosion, followed by their respective convolutional operations, and finally, the results from both branches are combined in an elementwise fashion. As morphological operations are nonlinear, they can generate a discrepancy in the learned features. In order to normalize those learned features, convolutional operations are used. The entire SpatialMorph block can be described as

FSpatMorph(Xpatch)=F2D((Xpatch⊞Wd))⊕F2D((Xpatch⊟We))(9)

View SourceRight-click on figure for MathML and additional features.where Wd and We are the weights of the (3×3) kernel and F2D is the function that represents the linear combination between the dilation and erosion feature maps obtained utilizing the 2-D convolution. To obtain the morphological spectral feature from the HSI patch tokens, a spectral morphological (SpectralMorph) block using the primitive operations is used. This block can be described using the following equation:

FSpecMorph(Xpatch)=F2D((Xpatch⊞Wd))⊕F2D((Xpatch⊟We))(10)

View SourceRight-click on figure for MathML and additional features.where Wd and We are the weights of the (1×1) kernel and F2D is the function that represents the linear combination between the dilation and erosion feature maps obtained utilizing the 2-D convolution.

[[Fig. 2. - Graphical visualization of (a) dilation and (b) erosion operations for an input image patch of size (
$7 \times 7$
), dilated, and eroded with an SE of size (
$3 \times 3$
). The resulting outputs keep the same size using a padding technique.](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza2ab-3242346-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza2ab-3242346-large.gif)

**Fig. 2.**

Graphical visualization of (a) dilation and (b) erosion operations for an input image patch of size (7×7 ), dilated, and eroded with an SE of size (3×3 ). The resulting outputs keep the same size using a padding technique.

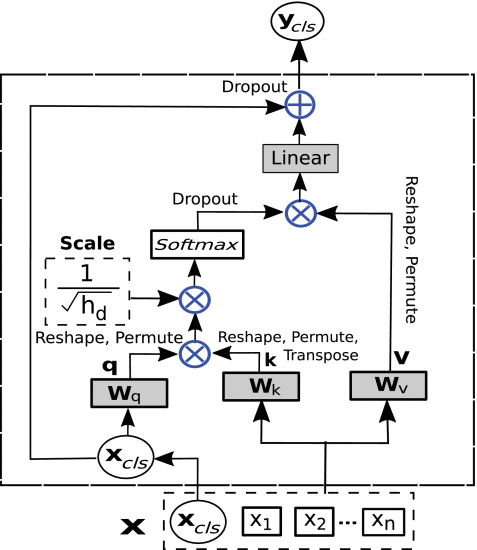
[Show All](https://ieeexplore.ieee.org/document/10036472/figures)

**D. Patch Attention Using Morphological Feature Fusion**

As shown in Fig. 3, the CLS (Xcls ) token uses the HSI patch tokens for exchanging information between each other to provide an abstract representation of the whole HSI patch. This entire operation is executed in blocks of the transformer encoder, where each transformer block consists of a spectral and spatial morphological feature extraction block and a residual multihead cross attention block. The spectral and spatial morphological feature extraction block consists of a spectral morphological layer and a spatial morphological layer, both of which take Xpatch as input. The spectral morphological layer is used to extract morphological spectral features from the HSI data, whereas the spatial morphological layer is used to extract morphological spatial features using two primitive morphological operations: dilation and erosion. The spatial and spectral morphological features allow for better attention between the intrinsic spatial and spectral characteristics of the image. The outputs from both layers are then concatenated in channelwise form (X′patch ) along with Xcls to generate the final output of the entire morphological block, as shown in Fig. 1. The output channel from both the spectral and spatial morphological blocks is half of the input Xpatch so that, after concatenating both of them, the number of channels becomes equal to that of Xpatch . The entire morphological block can be summarized as follows:

X′patch=X′=FSpatMorph(Xpatch)⊙FSpecMorph(Xpatch)Xcls⊙X′patch.(11)

View SourceRight-click on figure for MathML and additional features.

[[](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza3-3242346-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza3-3242346-large.gif)

**Fig. 3.**

Multihead patch attention module, where a query value interacts with all the other HSI patch tokens through the attention mechanism.

[Show All](https://ieeexplore.ieee.org/document/10036472/figures)

On the other hand, a layer normalization (LN) operation is used in the residual attention block. It takes the output from the morphological block as input. A self-attention layer is used after the LN operation, whose output ycls is added in elementwise fashion (⊕ ) to the input of the LN (as described in Fig. 1).

In the morphological patch attention module (MorphPAT) between Xcls and X′patch , three linear weights, i.e., Q , K , and V , are used. They are multiplied inside the morphological attention block and can be represented as

Z=softmax(QKThd−−√)V(12)

View SourceRight-click on figure for MathML and additional features.where Z∈R1×64 , hd is the embedding dimension/number of heads, Q is the query (which equals XclsWqwhereWq∈R64×64 ), K is the key (which equals X′patchWkwithWk∈R64×64 ), and V is the value (which equals X′patchWvwithWv∈R64×64 ). A dropout layer (DP ) with a value of 0.1 is used, followed by a linear projection layer (Wl∈R64×64 ) that is applied to the final output of the qkv operation. A self-attention module with a number of heads greater than one becomes a multihead self-attention module. Similarly, the MorphAT module (upon using multiple heads) becomes a multihead morphological attention module and can be represented as MMorphAT. Mathematically, the morphological attention module can be formulated as

MMorphAT(X′)=DP(WlZ).(13)

View SourceRight-click on figure for MathML and additional features.The output X′cls of the MMorphAT module for a given X′k−1 , where k is the kth transformer encoder block, can be expressed as

ycls=X′cls=MMorphAT(LN(X′k−1))LN(ycls⊕X′k−1cls)(14)

View SourceRight-click on figure for MathML and additional features.where X′cls∈R1×64 . This output X′cls is then concatenated with X′patch to yield the final output of that particular transformer encoder block, as shown in Fig. 1, and can be defined as

X′k=X′cls⊙X′patch.(15)

View SourceRight-click on figure for MathML and additional features.The proposed model uses eight heads. Finally, the CLS token is extracted from the output of the transformer encoder blocks (Xk ), and the final classification results are obtained from the CLS token via a classifier head.

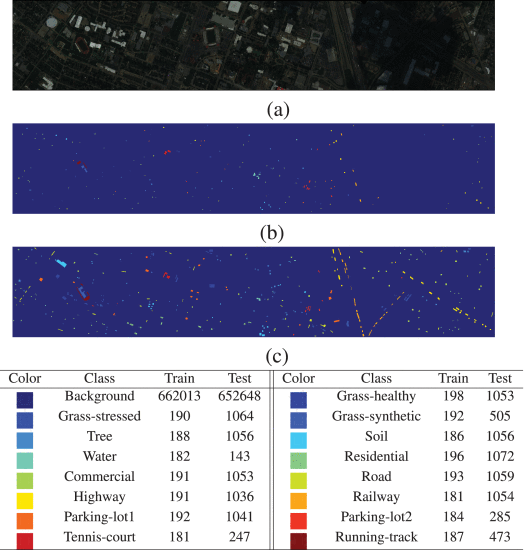
**SECTION III.**

Experiments

For evaluating the classification performance of the proposed morphFormer, we have considered four different datasets and compared our approach with other state-of-the-art techniques. The datasets utilized in experiments were collected from the University of Houston (UH), the University of Southern Mississippi Gulfpark (MUUFL), and the cities of Trento and Augsburg.

**A. Image Datasets**

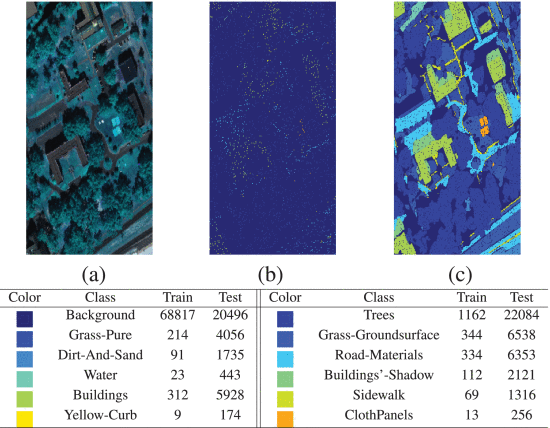
1. In the experiments, four HSI datasets, i.e., UH, MUUFL, Trento, and Augsburg are used. The UH data were gathered by the Compact Airborne Spectrographic Imager (CASI) in 2013 and published by the IEEE Geoscience and Remote Sensing Society. The image consists of 340×1905 pixels and 144 different spectral bands. Its wavelength range is 0.38–1.05μm with a spatial resolution of 2.5 meters per pixel (MPP). Its ground truth consists of 15 distinct classes. Total samples are separated into 15 distinct classes by disjoint train and test samples. Fig. 4 lists the disjoint train and test samples for each of the 15 distinct classes of land cover.
2. The MUUFL data were obtained in November 2010 around the region of the University of Southern Mississippi Gulf Park, Long Beach, MS, USA, by using the Reflective Optics System Imaging Spectrometer (ROSIS) sensor [60], [61]. It is made up of 325×220 pixels along with 72 spectral bands, and LiDAR data are also available and made up of elevation data from two rasters. The first and last eight bands are deleted owing to noise, resulting in 64 bands in total. There are 53687 ground-truth pixels with 11 different types of classes for urban land cover. 5% of the samples are randomly selected for training from each of the 11 classes, as shown in Fig. 5.
3. The Trento data were collected around rural areas in the south of Trento, Italy, by utilizing the AISA eagle sensor. The corresponding LiDAR data were obtained by the Optech ALTM 3100EA sensor. The HS image comprises 63 different spectral channels with wavelengths ranging from 0.42 to 0.99μm , whereas the LiDAR data have two rasters with elevation data. The HSI data include 600×166 pixels with 6 mutually exclusive land-cover classes of vegetation, with a spatial resolution of 1 MPP and spectral resolution of 9.2nm. Furthermore, the total samples are separated into six groups of disjoint train and testing samples. The information regarding the number of samples per class is given in Fig. 6.
4. The Augsburg scene includes three distinct data sources, including an HSI, a dual-Pol synthetic aperture radar (SAR) image, and a digital surface model (DSM) [62]. The HSI and DSM data were acquired by DLR, whereas the SAR data were collected from the Sentinel-1 platform over the city of Augsburg, Germany. The information was gathered with the HySpex sensor [63], the Sentinel-1 sensor, and the DLR-3 K system [64], respectively. For proper multimodal fusion, the spatial resolutions of all datasets were downsampled to a uniform spatial resolution of 30-m ground sampling distance (GSD). It has 332×485 pixels in the HSI, with 180 spectral channels that ranges from 0.4 to 2.5μm . In the ground truth, there are 15 distinct classes of land cover. The train and testing sets are demonstrated in detail in Fig. 7.

[[](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza4abc-3242346-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza4abc-3242346-large.gif)

**Fig. 4.**

UH data. (a) Pseudocolor image using bands 64, 43, and 22. (b) Disjoint train samples. (c) Disjoint testing samples. The table shows land-cover types for each class along with the number of disjoint train and test samples.

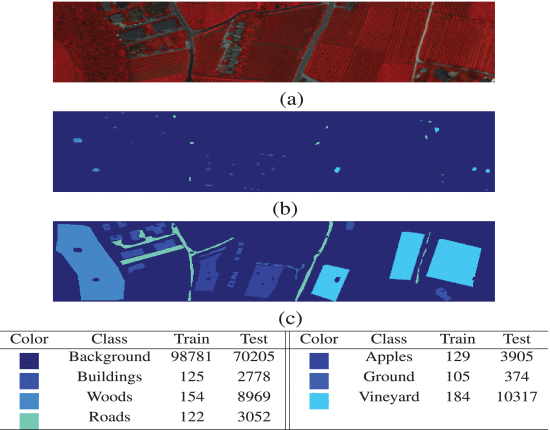
[Show All](https://ieeexplore.ieee.org/document/10036472/figures)

[[](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza5abc-3242346-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza5abc-3242346-large.gif)

**Fig. 5.**

MUUFL data. (a) Pseudocolor image using bands 40, 20, and 10. (b) Disjoint train samples. (c) Disjoint testing samples. The table shows land-cover types for each class along with the number of disjoint train and test samples, where the train samples represent 5% of the available ground truth, and the test samples represent the remaining 95% of the ground truth.

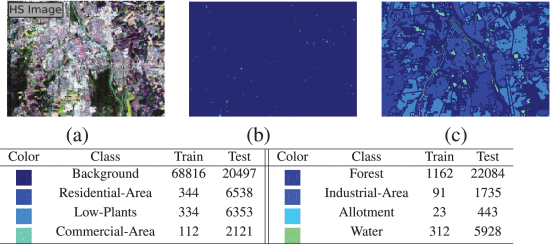
[Show All](https://ieeexplore.ieee.org/document/10036472/figures)

[[](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza6abc-3242346-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza6abc-3242346-large.gif)

**Fig. 6.**

Trento data. (a) Pseudocolor image using bands 40, 20, and 10. (b) Disjoint train samples. (c) Disjoint testing samples. The table shows land-cover types for each class along with the number of disjoint train and test samples.

[Show All](https://ieeexplore.ieee.org/document/10036472/figures)

[[](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza7abc-3242346-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza7abc-3242346-large.gif)

**Fig. 7.**

Augsburg data. (a) Pseudocolor image using bands 40, 20, and 10. (b) Disjoint train samples. (c) Disjoint testing samples. The table shows land-cover types for each class along with the number of disjoint train and test samples.

[Show All](https://ieeexplore.ieee.org/document/10036472/figures)

**B. Experimental Setting**

Extensive experiments have been performed using the proposed morphFormer model, and the results have been compared with that of traditional and state-of-the-art models to assess the performance of the proposed model.

The compared techniques include traditional classifiers, such as RF [2], K-nearest neighbors KNN [65], and SVM [23], in addition to classical CNN methods, such as CNN1D [66], CNN2D [30], CNN3D [31], and RNN [67]. We also included state-of-the-art transformer-based techniques, such as vision transformer (ViT) [68] and SpectralFormer [45].

For testing, a CPU with a Red Hat Enterprise Server (Release 7.6) has been used that consists of the ppc64le architecture, 40 cores consisting of four threads in each core, and 377 GB of memory. The GPU utilized is a single Nvidia Tesla V100 having VRAM of 32510 MB.

During our experiments, the number of HSI patch tokens (n) obtained from the tokenization process is taken as four. During training and testing, batch sizes of 64 and 500 were, respectively, utilized. Patches with a size of 11×11×B are taken from the HSI and used as input to the model. Aside from KNN, RF, SVM, and RNN, the Adam optimizer [69], [70] has been used to train the models, with a weight decay of 5e−3 and learning rate of 5e−4 . In addition, these methods (considering also the RNN) used a step scheduler with a gamma of 0.9, steps of size 50, and trained during 500 epochs. The average and standard deviation of each experiment have been calculated based on three repetitions. Python 3.7.7 and PyTorch 1.5.0 were used to implement the coding of the proposed morphFormer.

Different widely utilized quantitative measurement methods, such as the overall accuracy (OA), average accuracy (AA), and kappa coefficients (*k* appa), are utilized for assessing the performance. The experiments have been performed on spectrally and spatially disjoint sets of train and testing samples [71] such that there is no interaction between the respective samples. In addition, varying percentages or train samples have been considered for validating the performance of the considered techniques.

**C. Performance Analysis With Disjoint Train/Test Samples**

A quantitative assessment of classification performance is presented in Tables I–IV. The best classification values are displayed in bold. The results show that the proposed approach is superior to all other techniques in terms of OA, AA, and *k* s, and exhibits better performance in most cases in terms of classwise accuracy.

**TABLE I**Classification Performance (in %) on the UH HSI Dataset

[[Table I- 
Classification Performance (in %) on the UH HSI Dataset](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza.t1-3242346-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza.t1-3242346-large.gif)

**TABLE II**Classification Performance (in %) on the MUUFL HSI Dataset

[[Table II- 
Classification Performance (in %) on the MUUFL HSI Dataset](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza.t2-3242346-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza.t2-3242346-large.gif)

**TABLE III**Classification Performance (in %) on the Trento HSI Dataset

[[Table III- 
Classification Performance (in %) on the Trento HSI Dataset](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza.t3-3242346-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza.t3-3242346-large.gif)

**TABLE IV**Classification Performance (in %) on the Augsburg HSI Dataset

[[Table IV- 
Classification Performance (in %) on the Augsburg HSI Dataset](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza.t4-3242346-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza.t4-3242346-large.gif)

It is worth noting that conventional classifiers, such as KNN, RF, or SVM, exhibit similar performance. An exception is the KNN with the MUFFL and Trento datasets, which provides inferior accuracies than those provided by RF and SVM. In addition, the performance of DL-based classifiers, such as CNN1D, CNN2D, CNN3D, and RNN, is generally superior to that of conventional classifiers, except for RF in UH and MUFFL datasets (which is better than CNN2D and CNN3D). Transformer methods, such as ViT and SpectralFormer, provide better performance due to the incorporation of the sequential mechanism. However, the incorporation of the spatial–spectral information in the proposed morphFormer leads to better classification performance in terms of OA, AA, and *k* in all considered datasets.

Table I shows that the RF provides better performance in the UH dataset in comparison to other conventional classifiers, but it cannot provide better performance than transformer methods. The proposed technique exhibits a performance that is superior to that of all compared methods due to its capacity to learn spatial and spectral information. The morphFormer shows mean OA, AA, and *k* of 87.85%, 89.66%, and 86.81% having a standard deviation of 0.20%, 0.39%, and 0.22%, respectively.

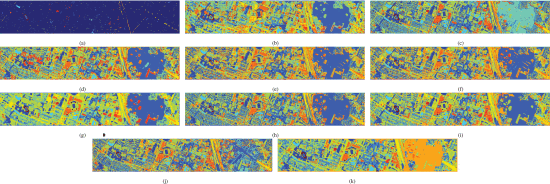
Table II shows the generalization ability of the MUUFL dataset for disjoint train and test samples. Both RNN and RF exhibit comparable accuracies and outperform the remaining conventional classifiers. The morphFormer shows better accuracy than that of all other techniques, including transformer-based approaches, with OA, AA, and *k* of 93.84 ± 0.10%, 80.55 ± 0.27%, and 91.84 ± 0.13%, respectively.

Table III lists the classification results on the Trento dataset. RF outperforms other conventional classifiers, and CNN3D shows better accuracy than other DL-based methods. The morphFormer shows better classification accuracy than all other methods with OA, AA, and *k* of 96.73 ± 0.58%, 93.68 ± 1.28%, and 95.62 ± 0.77%, respectively.

Table IV shows the classification results on the Augsburg dataset. RNN exhibits lower accuracies than other conventional classifiers, while RF is the best conventional classifier, and CNN3D outperforms other DL-based approaches. The transformer ViT method outperforms our approach in terms of AA. However, the morphFormer outperforms all other methods in terms of OA and *k*.

**D. Visual Comparison**

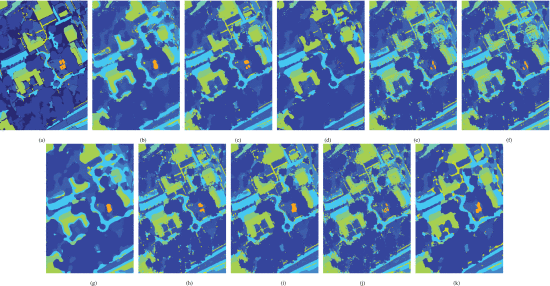
Figs. 8–11 show the obtained classification maps. Our goal is to perform a qualitative evaluation of the compared methods. Conventional classifiers, such as KNN, RF, and SVM, provide classification maps with salt and pepper noise around the boundary areas because they only exploit spectral information. In addition, the DL methods produce better classification noise in comparison to conventional classifiers. Specifically, the maps produced by CNN1D, CNN2D, and CNN3D are smoother because the boundaries between land-use and land-cover classes can be separated in a better way. ViT can extract more abstract information in sequential representation, so it provides better classification maps. Compared to ViT and SpectraFormer, the proposed morphFormer exhibits better classification maps. In other words, our newly proposed morphFormer can enhance classification performance by considering spatial-contextual information and positional information across different layers. As a result, it characterizes texture and edge details better than other transformer-based techniques.

[[](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza8abcdefghijk-3242346-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza8abcdefghijk-3242346-large.gif)

**Fig. 8.**

Classification maps for the Houston (UH) HSI dataset. (a) Ground truth. (b) KNN (69.48%). (c) RF (74.87%). (d) SVM (68.13%). (e) CNN1D (63.04%). (f) CNN2D (65.85%). (g) CNN3D (70.26%). (h) RNN (65.20%). (i) ViT (83.23%). (j) SpectralFormer (76.35%). (k) morphFormer (87.85%).

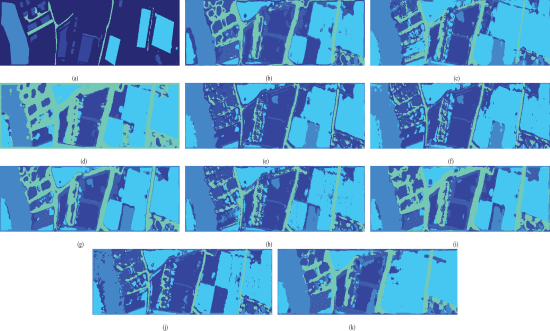
[Show All](https://ieeexplore.ieee.org/document/10036472/figures)

[[](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza9abcdefghijk-3242346-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza9abcdefghijk-3242346-large.gif)

**Fig. 9.**

Classification maps for the MUUFL HSI dataset. (a) Ground truth. (b) KNN (75.80%). (c) RF (89.85%). (d) SVM (84.30%). (e) CNN1D (81.17%). (f) CNN2D (82.95%). (g) CNN3D (77.59%). (h) RNN (88.60%). (i) ViT (91.99%). (j) SpectralFormer (86.68%). (k) morphFormer (93.84%).

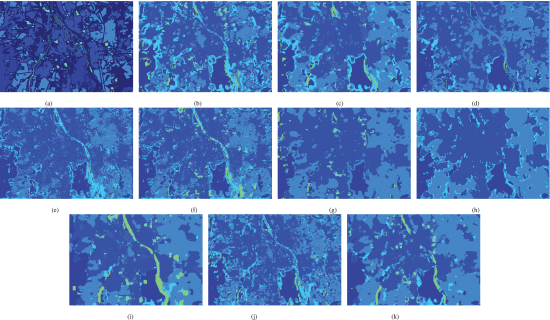
[Show All](https://ieeexplore.ieee.org/document/10036472/figures)

[[](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza10abcdefghijk-3242346-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza10abcdefghijk-3242346-large.gif)

**Fig. 10.**

Classification maps for the Trento HSI dataset. (a) Ground truth. (b) KNN (86.42%). (c) RF (94.73%). (d) SVM (88.55%). (e) CNN1D (93.02%). (f) CNN2D (92.31%). (g) CNN3D (96.14%). (h) RNN (86.83%). (i) ViT (94.62%). (j) SpectralFormer (88.42%). (k) morphFormer (96.73%).

[Show All](https://ieeexplore.ieee.org/document/10036472/figures)

[[](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza11abcdefghijk-3242346-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza11abcdefghijk-3242346-large.gif)

**Fig. 11.**

Classification maps for the Augsburg HSI dataset. (a) Ground truth. (b) KNN (67.27%). (c) RF (79.96%). (d) SVM 71.60 (%). (e) CNN1D (72.00%). (f) CNN2D (73.59%). (g) CNN3D (82.89%). (h) RNN (40.26%). (i) ViT (85.90%). (j) SpectralFormer (70.81%). (k) morphFormer (88.68%).

[Show All](https://ieeexplore.ieee.org/document/10036472/figures)

**E. Performance Over Different Train Sample Sizes**

Fig. 12(a)–(i) shows the classification performance of transformer models with different percentages of training samples on three HSI datasets of Houston, MUUFL, and Trento. The training samples on these three datasets are randomly selected as 3%, 5%, 7%, and 9%.

[[Fig. 12. - Classification accuracies in terms of AA, OA, and kappa (
$\kappa $
) obtained by various techniques with different percentages of training samples randomly selected from (a), (d), (g), UH (b), (e), (h) MUUFL, and (c), (f), (i) Trento datasets.](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza12abcdefghi-3242346-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza12abcdefghi-3242346-large.gif)

**Fig. 12.**

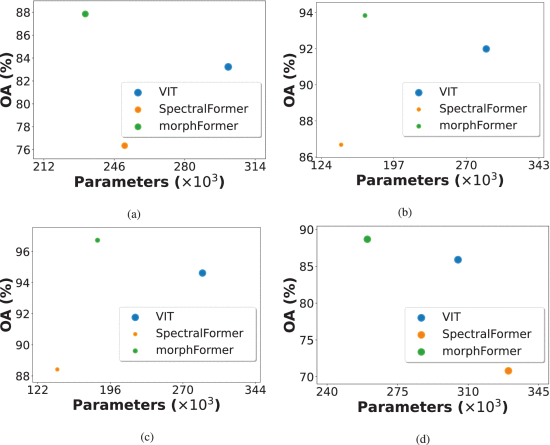
Classification accuracies in terms of AA, OA, and kappa (κ ) obtained by various techniques with different percentages of training samples randomly selected from (a), (d), (g), UH (b), (e), (h) MUUFL, and (c), (f), (i) Trento datasets.

[Show All](https://ieeexplore.ieee.org/document/10036472/figures)

In the Houston dataset, the proposed morphFormer outperforms the second-best-performing transformer model (ViT) by a margin of approximately 4% in terms of OA, AA, and *k* for all considered percentages of randomly selected samples. Although the margin is smaller for larger training sizes, the proposed morphFormer exhibits superior classification performance for all sample sizes in the other two datasets (MUUFL and Trento). It can be concluded that the proposed morphFormer exhibits significantly better classification performance than the other transformer networks, even with a limited number of training samples.

**F. Hyperparameter Sensitivity Analysis**

In terms of computing complexity, the proposed model is not only effective but also rather efficient. In Fig. 13(a)–(d), the parameters and calculations of the proposed method are compared to those of various transformer networks. Specifically, we show the OA, the number of parameters, and the number of calculations (FLOPs) for the UH, Trento, MUUFL, and Augsburg datasets. The calculations are shown by the radii of circles. The efficiency of morphFormer is clear in Houston and Augsburg datasets, where it needs the fewest parameters and FLOPS. Although the parameters and FLOPS needed by morphFormer are higher than those required by SpectralFormer in certain cases, the gain in performance compensates for that. As can be seen with the UH data, morphFormer offers an outstanding gain in OA (4.62%) over the next best model (ViT). In this case, the parameter tradeoff is justified by the significant increase in classification accuracy.

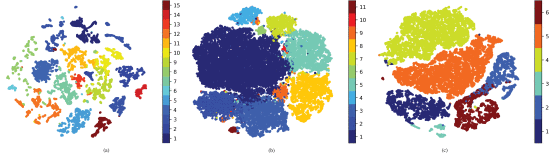
[[](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza13abcd-3242346-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza13abcd-3242346-large.gif)

**Fig. 13.**

Comparing the performance of transformer methods in terms of OA, network parameters, and FLOPs, shown by the radii of circles considered from (a) UH, (b) MUUFL, (c) Trento, and (d) Augsburg.

[Show All](https://ieeexplore.ieee.org/document/10036472/figures)

Furthermore, 2-D graphical plots depicting the features extracted by the proposed morphFormer are presented in Fig. 14(a)–(c) for Houston, MUUFL, and Trento datasets, respectively. Using the t-SNE approach [72], the features extracted by morphFormer can be analyzed. It can be observed that samples of similar categories gather together, and intraclass variance is minimized in all three datasets.

[[](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza14abc-3242346-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/36/10006360/10036472/plaza14abc-3242346-large.gif)

**Fig. 14.**

In the upper row, we show the proposed HSI classification network such that the classification map of the proposed method contains (left) less noise than existing methods and (right) transformer encoder with a multihead patch attention mechanism. In the bottom row, we show the backbone of the proposed method.

[Show All](https://ieeexplore.ieee.org/document/10036472/figures)

**SECTION IV.**

Conclusion

We present a novel morphFormer network for HSI data classification, which is based on spectral and spatial morphological convolutions. Although fusing attention and morphological characteristics are not straightforward, our approach can successfully merge attention mechanisms with morphological operations and provide superior classification performance compared to standard convolutional models and the recently developed transformer models. Our morphFormer has the potential to excel in many different classification tasks in EO and RS. It is because of its ability to apply learnable morphological operations in addition to multihead self-attention mechanisms. A general adversarial network (GAN)-based method will be investigated with the morphFormer in our future work. Moreover, the LiDAR processing problem will also be solved using a morphFormer-based approach.