

DESSA-Net Model: Hyperspectral Image Classification Using an Entropy Filter With Spatial and Spectral Attention Modules on DeepNet

Javad Mahmoodi ¹, Dariush Abbasi-Moghadam ¹, Alireza Sharifi ¹, Hossein Nezamabadi-Pour, Mohammad Esmaeili ¹, and Alireza Vafaeinejad ¹

Abstract—Recent advancements in remote sensing technology have significantly expanded the exploration of natural resources and enabled the detection of materials in inaccessible areas. Hyperspectral images (HSIs) are a valuable data source due to their distinctive properties in various applications. However, several problems, including noise, band correlation, ineffectively extracted features, and most notably, a lack of sufficient labeled samples, reduce the accuracy of HSI classification. To improve the performance of such a system, we propose an effective method with the capability of paying attention to spectral and spatial features. The raw HSI data are first preprocessed using a principal component analysis (PCA) operation because of the redundancy and correlation between HSI bands. Then, the entropy base informative module is designed to add entropy information to the selected spectral features by PCA. We also use spectral and spatial attention modules in the proposed model. Moreover, a hybrid neural network that uses both 3-D convolutional neural networks (CNNs) and 2-D CNNs with skip connections is exploited to reduce the complexity of the network compared to 3-D CNNs. The spatial attention module called depthwise spatial attention block can inherently highlight spatial information. The spectral attention module named reshape softmax attention can capture useful spectral regions of feature maps. Meticulous HSI classification tests are conducted over the University of Pavia, Indian Pines, Salinas, and Houston 2013 to evaluate the effectiveness of our approach. Our experiments show higher accuracy compared to other deep learning methods.

Index Terms—Deep learning, depth-wise convolutional neural network (CNN), entropy filter, hyperspectral image (HSI) classification, remote sensing, reshape softmax attention (RSA), spatial-spectral features.

I. INTRODUCTION

ONE type of remote sensing image is hyperspectral images (HSIs), which give valuable information about the

Manuscript received 2 April 2024; revised 26 May 2024, 17 June 2024, and 12 July 2024; accepted 1 August 2024. Date of publication 6 August 2024; date of current version 26 August 2024. (Corresponding authors: Alireza Sharifi; Dariush Abbasi-Moghadam.)

Javad Mahmoodi is with the Department of Electrical Engineering, Kerman Branch, Islamic Azad University, Kerman 76351-68111, Iran (e-mail: javad.mahmoodi@iauk.ac.ir).

Dariush Abbasi-Moghadam, Hossein Nezamabadi-Pour, and Mohammad Esmaeili are with the Electrical Engineering Department, Shahid Bahonar University of Kerman, Kerman 76169-14111, Iran (e-mail: abbasimoghadam@uk.ac.ir; nezam@uk.ac.ir; m-esmaeili@eng.uk.ac.ir).

Alireza Sharifi and Alireza Vafaeinejad are with the Department of Surveying Engineering, Faculty of Civil, Water and Environmental Engineering, Shahid Beheshti University, Tehran 16589-53571, Iran (e-mail: asharif.sbu.ir@gmail.com; a_vafaei@sbu.ac.ir).

Digital Object Identifier 10.1109/JSTARS.2024.3439592

spatial distribution and composition of materials within a scene [1]. These images are captured by imaging spectrometers on various space platforms. HSIs have high spectral resolution and can contain hundreds of continuous bands or channels in the nanometer range [2], [3]. HSIs have many applications in various domains, including environmental monitoring [4], [5], [6], detecting anomalies in HSIs [7], HSI classification [8], and others. HSI classification has various applications, such as agriculture [9], exploring geological features [10], and other purposes [11], [12]. There are two main approaches for HSI classification: manually extracting features and automatically extracting features using deep learning techniques.

In the past, HSI classification systems employed traditional machine learning methods to analyze spatial-spectral features. A new method called multiscale joint representation with local adaptation was proposed in [13], which reduced the negative effects of irrelevant pixels on classification accuracy. Another significant method was the refined diffusion model and discontinuity-preserving relaxation [14], which preprocessed each pixel, computed related statistical measures, and effectively combined spatial texture and spectral signatures. Gao et al. [15] selected important bands through optimization-based sparse self-representation to improve the classification procedure. In [16], a technique combined the correlation coefficient with sparse representation to enhance classification accuracy. Moreover, Tu. et al. [17] examined the applications of multiscale superpixels and guided filtering for efficient feature extraction and classification. In [18], an unsupervised band selection technique based on Boltzmann and entropy was proposed to improve the classification performance. Moreover, the complexity of high-dimensional HSI makes it difficult to achieve optimal classification results using the earlier-mentioned methods [19]. Despite the rich spectral data in the HSI domain, the lack of labeled samples posed challenges for learning better feature representations and increased the risk of overfitting. To address these issues, several schemes have been proposed, including feature extraction [20], [21], reducing dimensionality [22], [23], [24], and augmenting data [25].

Although the previous approaches have yielded appropriate results in some cases, they relied on manual feature extraction. The classification results of these techniques rely on the reliability of the hand-crafted features. Furthermore, there is

often significant spectral variability in HSI, resulting in large differences within the same class and strong similarities across different classes. Thus, manually designed patterns are not suitable for addressing these challenges [26].

Deep learning methods, especially convolutional neural networks (CNNs), have obtained remarkable results in computer vision tasks, such as image classification [27], action recognition [28], and violence detection [29], [30]. The CNN architecture has two main stages: feature extraction and classification. The feature extraction network uses convolutional and pooling layers to extract hierarchical representations of input data, which are then utilized in the subsequent classification stage. The convolutional layer efficiently captures local patterns by applying kernels to the input features. Convolutional networks include local or global pooling layers to reduce feature map resolution. In addition, convolution layers utilize activation functions to extract nonlinear features. Finally, fully connected layers and a Softmax operator are used in the classification stage.

The capability of CNNs to process spatial HSI patches as input motivated researchers to propose complicated CNN-based techniques for HSI classifications. For example, the authors in [31] and [32] proposed a CNN-based model for classifying HSIs. Makantasis et al. [33] utilized a CNN to capture both spectral and spatial information of pixels, and a multilayer perceptron to perform the classification task. Ben Hamida et al. [34] proposed a model based on 3-D CNN and 1-D CNN and they achieved successful results in HSI classification. CNNs are increasingly used in HSI classification for capturing spatial-spectral features. However, their ability to model sample relations is limited. To address this, graph convolutional networks have been successfully applied to irregular data representation and analysis [35].

Two or multibranch architectures process input data through different pathways, capturing a wider range of features, such as spatial or spectral aspects. These architectures have been exploited for HIS classification. In [36], two branches of CNN models were introduced. One branch was utilized to extract spectral information, while the other was specifically designed for the extraction of spatial features. However, using multibranch architectures may result in longer training times and require more computational resources compared to single-branch networks.

Although CNN-based methods have made significant advancements, they still face challenges in efficiently utilizing spectral and spatial association information. Therefore, Yang et al. [37] introduced a multiscale wavelet 3D-CNN to exploit the correlation in the spectral and spatial domains. Another method exploited the 3D–2D hierarchical CNN model [38], in which 3D-CNN layers were employed to analyze spectral information, and 2D-CNN layers were exploited to focus on texture and contextual spatial information. In [39], a combination of CNN models and spatial-spectral morphological attention mechanisms was proposed to enhance feature extraction in HSI. Another method was the spatial-spectral residual network (ResNet) [40]. The authors suggested using two consecutive residual blocks to separately learn spectral and spatial representations, enabling the extraction of more discriminative features.

The deep learning-based approaches usually exploit 2-D and 3-D convolutional layers to process the spatial and spectral features. Although 2-D convolutional layers are designed to process spatial data, they miss spectral features of HSIs. The spectral dimension of HSIs contains distinct wavelengths that correspond to different material properties. When 2-D convolutional layers are applied to HSIs, they treat the spectral dimension as spatial, applying the same filters across the spectral bands. This approach can lead to a loss of critical spectral information because the filters are not specifically tuned to recognize the unique spectral signatures of different materials. The 3-D convolutional layers are also able to capture spectral-spatial information to some extent, but they have drawbacks in modeling complex spectral-spatial relationships. These challenges primarily stem from the high dimensionality of hyperspectral data and the complex interband relationships that are not fully captured by CNN architectures.

In the pursuit of advancing HSI classification, we introduce the entropy base informative module (EBIM), a novel component that infuses entropy information into input images. Unlike conventional methods, the EBIM augments the local complexity and diversity of the image and highlights spatial features that are often overlooked. This module is particularly adept at enhancing the local discriminative features, which is crucial for accurate classification. The innovation lies in the integration of entropy filtering with attention mechanisms, tailored to enhance feature extraction in the presence of noisy data, band correlation, and ineffective feature representation, which are common challenges in the HSI analysis. Moreover, we utilize a combined deep learning structure to leverage the strengths of both 2-D CNNs and 3-D CNNs. In addition, we introduce spatial and spectral attention mechanisms to enhance the features extracted by 3-D and 2-D CNNs and to address the concerns raised.

- 1) Noise reduction: DESSA-Net employs an entropy filter and a spatial attention module that effectively mitigates noise by emphasizing informative features while suppressing random variations. Entropy filters are a powerful tool for noise reduction because they can differentiate between the structured information of the images and the unstructured randomness of noise. Spatial attention is a sophisticated process that can focus on the most relevant features for given tasks. It can concentrate on certain areas of images when making decisions.
- 2) Band correlation: HSIs contain hundreds of spectral bands, many of which are highly correlated. By focusing on the appropriate bands, deep learning models can focus on the most relevant features for the classification task. The spectral attention module is designed to capture the interband relationships, thus enhancing the discriminative power of correlated spectral bands.
- 3) Feature extraction: By leveraging deep learning architectures, DESSA-Net efficiently extracts features that are crucial for accurate classification, even from complex hyperspectral data. By focusing on the most informative regions and bands, spatial and spectral mechanisms can enhance the feature extraction process, leading to better

performance of the deep model. Once the bands have been weighted appropriately, the deep model can be trained with the refined data. This leads to improved accuracy and robustness in the model's prediction.

II. RELATED WORK

New deep learning models efficiently utilize both spectral and spatial characteristics to overcome the limitations of traditional machine learning algorithms, significantly enhancing the effectiveness of HSI classification. Several advanced models have utilized two- or multibranch networks for HSI classification. In [41], a two-branch residual neural network (ResNet) was introduced to incorporate spectral and spatial information. This method had two branches: one for obtaining spectral characteristics and the other for extracting spatial features. Ge et al. [42] utilized 2-D and 3-D CNNs in a multibranch architecture. Methods based on two- or multibranch increase the number of parameters and the overall complexity of the model. This complexity can lead to longer training time, increased memory requirements, and higher computational costs.

Some other methods used generative adversarial networks (GANs) to tackle the class imbalance difficulty. However, GANs have difficulties in modeling long- and mid-term dependencies and extracting discriminative spectral features between classes with similar spectral signatures [43], [44]. To address the issue of small samples in HSI classification, adversarial representation learning based on generative adversarial networks (ARL-GAN) was introduced in [45]. An HSI block generator was developed to extract more distinct features from the input feature vector. To measure errors between the actual image and the generated image, the class probability distance was measured instead of the mean square error. Moreover, the combination of GAN and conditional entropy was exploited to alleviate the challenge of small sample sizes in HSI classification. GAN-based methods might not work well on different hyperspectral datasets, thereby reducing their usefulness in practical applications.

Learning long-term relationships is enabled by the band-by-band accumulation of spectral features in HSIs. Recurrent neural networks (RNNs) are commonly used to achieve this task. For example, the attention-based long short-term memory (LSTM) model improved HSI classification performance by effectively capturing spatial-spectral dependencies [46]. However, RNNs are not appropriate for learning spectral and spatial characteristics simultaneously. Combining attention strategies with RNNs has been used to tackle this challenge. Attention-based models have demonstrated good performance by effectively exploring both spatial and spectral features [47], [48]. In such models, hyperparameters can significantly impact model performance, requiring extensive experiments and computational resources to find the optimal configuration.

The usage of transformers in computer vision [49] has led to the development of numerous novel transformer types in recent years [50]. Transformer models are advanced models designed to process and analyze sequential (or time series) data. Transformers exploit self-attention techniques to do this task [51]. In [52],

a spatial-spectral feature labeling transformer (SSFLT) method was proposed to learn spectral-spatial features and high-level semantic ones. In this structure, a unique module extracted low-level and shallow features. SSFLT had a Gaussian weighted feature marker to transform extracted features to a transform encoder. At last, the sample labels were identified through a linear layer. Ahmad et al. [53] introduced WaveFormer, a new transformer-based approach that integrates wavelet transforms for invertible downsampling. This method maintains data integrity while allowing for attention learning. The WaveFormer effectively combines downsampling with wavelet transforms to decompress feature maps without loss. However, integrating wavelet transforms increases the complexity of the model and may require additional computational resources. Zhao et al. [54] proposed a group-separable convolutional vision transformer network. This approach utilized a group separable convolution (GSC) module to significantly reduce the number of convolutional kernel parameters. In addition, it incorporated a simple point layer with an advanced skip connection mechanism instead of a multilayer perceptron layer, which facilitated better feature fusion. However, the limited training samples posed a risk of overfitting, increased the complexity of the model, and led to higher computational costs. Transformers are indeed effective at capturing spectral information, which is sequential data in nature. However, traditional transformer models may not fully exploit the spatial information present in HSI. To address this, researchers have developed specialized frameworks such as the spatial-spectral transformer (SST) [55]. In addition, the multiscale SST has been proposed to handle global dependencies among multiscale features and better utilize the spatial-spectral information inherent in HSIs [56]. While the self-attention mechanism in the transformer architecture enables the model to capture long-range dependencies and model complex relationships between pixels for more accurate predictions, these models have a large number of parameters and require large training samples.

In addition to previous methods, some techniques combined the input data with samples extracted from specific filters or mathematical operations, such as morphology and entropy. Among the recently proposed models, we can mention the work by the authors in [57] and [58], which combines HSI data with extracted samples from morphological operations in a multibranch structure with an attention mechanism. Moreover, Esmaeili et al. [59] proposed a method that extracted morphological features using morphological mathematics by four morphological operators. Then, they extracted environmental features, edges, and structures of shapes and regions in HSI and injected them between the layers of a deep network. This method improved representation and classification by injecting morphological features into the model layers and enabling end-to-end learning in deep networks. Integrating morphological operations with deep learning networks enhances feature extraction in HSI classification. However, this approach increases computational complexity. In addition, morphological operations are generally applied in a predefined manner, which may not be optimal for all types of HSI data. This lack of adaptability can result in the loss of important information that is crucial for accurate

TABLE I
COMPARISON RESULTS OF OTHER METHODS IN TERMS OF ACCURACY, BENEFITS, AND DRAWBACKS

Methods	References	Drawbacks (⊗) and benefits (⊙)	The overall accuracy				
			Ref	IP	HT	PU	SA
Two or multi-branches	[41], [42]	⊙ Increase the number of parameters and the complexity of the model ⊙ Enhanced feature extraction	[41]	-	97.61	95.86	-
			[42]	96.07	-	99.52	99.94
			[43]	79.79	-	90.29	-
GANs-based methods	[43], [44], [45]	⊙ Tackle the class imbalance difficulty ⊙ Low accuracy in some HSI datasets	[44]	93.97	-	97.26	98
			[45]	98.25	-	99.83	99.97
			[46]	95	-	98.48	-
RNNs and LSTMs-based methods	[46], [47], [48]	⊙ Ability to process sequential data ⊗ Requiring extensive experiments and computational resources to find the optimal configuration	[47]	97.56	-	96.85	-
			[48]	99.66	99.17	99.97	-
			[53]	-	96.54	95.66	-
Transformer models	[53], [54], [55]	⊙ Exploit self-attention techniques ⊙ Demand for large training sample ⊙ Large number of hyperparameters	[54]	96.98	-	-	97.15
			[55]	91.2	-	93.73	96.83
			[57]	86	-	-	-
Diffusion models	[57], [58], [59]	⊙ ability to extract more features ⊙ Increase computational complexity	[58]	87.45	86.51	95.51	-
			[59]	97.81	98.67	99.33	99.71
			[66]	96.31	-	-	-
Lightweight models	[66], [67]	⊙ Minimize the number of parameters ⊙ Decrease computational complexity ⊙ Limited feature extraction	[67]	98.5	99.04	99.51	-

classification. Therefore, while morphological operations can improve the feature extraction process, they need to be carefully designed and integrated into deep learning models to avoid potential drawbacks.

Some researchers have proposed models to reduce training time and parameters while also improving accuracy. In [60], convolution kernels of 1×1 and 3×3 were employed to effectively classify data by extracting spectral and spatial properties through dense connections. Others suggested the SC-FR feature multiplexing module with 1×1 convolution kernels and two coupled cross layers [61]. The coupled cross layer improved the flow and utilization of feature information. However, it increased the depth of the model. Han et al. [62] proposed a ResNet and exploited pyramidal bottleneck residual units [63]. Moreover, Dang et al. [64] designed a classification model based on the suggested techniques in [62] and [63]. Their model improved the proposed method [55] by exploiting depthwise separable convolution instead of simple convolutions in the residual block [65]. Moreover, a lightweight hybrid convolutional neural network (Lite-HCNet) was proposed to minimize the number of parameters and decrease computational complexity [66]. In this model, a new attention module was combined with a strategy to design a lightweight network. In [67], a method based on a three-branch CNN was proposed to reduce the number of parameters. They used three different branches for their network: the first one employed a compression and stimulation network (SENet), the second one combined three-dimensional CNN and two-dimensional DSC, and the third one used only DSC. The main purpose of using this structure was to enhance the extracted features from HSIs. However, a lightweight architecture may have difficulty capturing fine details and subtle changes, potentially leading to suboptimal classification performance. Furthermore, lightweight architectures typically aim to strike a balance between the size of the model and its performance. Although a reduction in model size can lead to improved memory and computational efficiency, it may also result in decreased classification accuracy. The tradeoff between

model size and performance must be carefully considered based on the specific requirements and limitations of HSI classification. Table I summarizes a comparative analysis of other methods, including the overall accuracy (OA), their benefits, and drawbacks.

Motivated by the above successful methods, we have also proposed ideas related to HSI classification, which we will discuss further in these contributions. Our contribution can be summarized as follows.

- 1) We propose an EBIM to add entropy information to input images. The EBIM enhances the local complexity and diversity of the image, which can help to detect suitable variations in the spectral and spatial features. DESSA-Net employs an entropy filter that effectively mitigates noise by emphasizing informative features while suppressing random variations.
- 2) The depthwise spatial attention (DSA) block, a spatial attention module, is introduced. It utilizes the softmax function along with 2-D depthwise convolutions. The softmax function adds probability information to the feature maps and 2-D depthwise convolutions preserve the spatial information and the channelwise correlations of the input image, which can improve the feature extraction and representation.
- 3) The reshape softmax attention (RSA) block is a spectral attention module that employs reshape layers and the softmax function in its architecture. It also adds probability information to the important bands by exploiting the softmax function. The spectral attention modules are designed to capture the interband relationships, thus enhancing the discriminative power of correlated spectral bands.
- 4) Our deep neural network (DNN) has a hybrid architecture with a single branch and three residual connections. A 2-D global average pooling (2-D GAP) and a hybrid structure are also used in place of a flattened layer to decrease the number of trainable parameters and the computational complexity of the network. In addition, residual

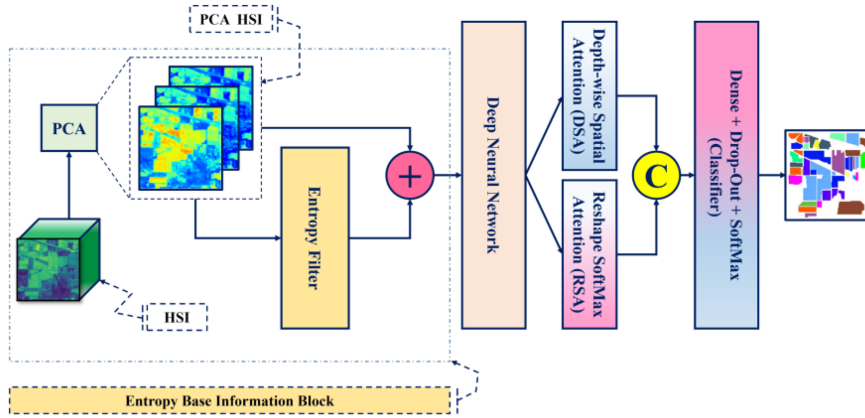


Fig. 1. Architecture of the proposed methodology.

TABLE II
PSEUDOCODE: PROPOSED METHOD (DESSA-NET)

<p>The objective is to classify and visualize four datasets using the raw HSI data $G \in R^{W \times H \times S}$ and the ground truth $Y \in R^{W \times H}$.</p> <p>Step 1: Reduce the dimension of $G \in R^{W \times H \times S}$ with PCA and obtain $P \in R^{W \times H \times C}$.</p> <p>Step 2: EF= Compute Entropy of $P \in R^{W \times H \times C}$.</p> <p>Step 3: EBIM=EF+P.</p> <p>Step 4: HSI EBIM-Cube to expand and set HSI EBIM -Patch $\in R^{s \times s \times b}$, with Win-Size $s \times s$.</p> <p>Step 5: Apply EBIM-Cube to DNN and obtain the input of the RSA module ($RSA_{input} \in R^{a \times b \times c}$), the input of the DSA module ($DSA_{input} \in R^{a \times b \times c}$).</p> <p>Step 6: Apply spatial and spectral attention modules to feature map and obtain the output of these modules (RSA_{output} and DSA_{output}).</p> <p>Step 7: apply 2DGAP to RSA_{output} and DSA_{output}.</p> <p>Step 8: Concatenate the output of two 2DGAPs.</p> <p>Step 9: Obtain the output classification maps.</p>

connections are also employed to overcome overfitting. We also exploit 2-D and 3-D convolutional layers in the deep network. The 2-D CNNs efficiently extract spatial features from images, while 3-D CNNs can provide context by analyzing different bands. While 3-D CNNs are computationally more intensive, integrating them with 2-D CNNs can help reduce computational complexity by strategically applying 3-D convolutions only where necessary. By leveraging deep learning architectures, DESSA-Net efficiently extracts features that are crucial for accurate classification, even from complex hyperspectral data.

III. DESCRIPTION OF THE METHODOLOGY

Our proposed HSI classification system consists of four parts: EBIM, hybrid DNN, DSA block, and RSA block. Fig. 1 depicts the architecture of the proposed methodology. Moreover, the pseudocode of the proposed methodology is given in Table II.

At first, the raw HSI data are preprocessed using the PCA operation. Analyzing hyperspectral data can be time-consuming because of the extensive number of bands, their significant intercorrelation, and the presence of redundant information. To alleviate these problems, the dimension of HSIs should be

reduced. In this study, we employ the PCA to reduce the dimensionality of the input data. This decision is driven by the need to find a low-dimensional representation that maintains as much information as possible. Furthermore, many researchers utilized this technique to decrease the dimensionality of the input data [42], [59], [68], [69]. Firat et al. [70] presented a comprehensive discussion on different dimension reduction techniques for HSI classification.

We suggest using the EBIM to enhance the HSIs with entropy information. To do this, we apply an entropy filter to the selected bands by PCA. Then, we add this information to the HSIs to provide more information to them. Indeed, entropy is the term used in information theory to quantify how much information is contained in data or how uncertain an event is. In image processing, the image entropy measures how complex or random the image is, and it is often used as an indicator of its texture.

Our proposed DNN consists of skip connections and a hybrid architecture using 2-D and 3-D convolutional layers. By using 3-D CNNs only on a subset of spectral bands and then applying 2-D CNNs on the output, hybrid CNNs can balance efficiency and accuracy [38]. The 2-D CNN alone is not sufficient to extract highly discriminative features from the spectral dimensions. Comparably, the computational complexity of a 3-D CNN is higher. Furthermore, 3-D CNNs often exhibit unsatisfactory performance when dealing with classes with similar textures across multiple spectral bands [38]. Skip connections are incorporated into the network architecture to create a residual network. This helps DNNs to learn features more effectively and avoid the problem of vanishing or exploding gradients [71]. These connections connect the input of a layer to the output of a further layer, bypassing some layers in between. This way, the network can learn the residual function, which is the difference between the input and the output, rather than the direct mapping. Indeed, there are two ways of using a skip connection. The first one uses adding layers in the architecture such as ResNet [71]. While the second one exploits concatenate layers such as DenseNet [72]. We can point to the fact that concatenative skip connections are a popular alternative for ensuring feature reusability of the same dimensions from the earlier layers.

HSIs often exhibit spectral variability, which means that the same class may have different spectral signatures in distinct images due to variations in illumination, viewing angle, atmospheric conditions, or sensor characteristics [73], [74], [75]. This makes the classification of such images challenging as the spectral features may not be consistent or discriminative across different images. Therefore, spectral attention modules are often needed to find the most relevant and robust spectral features for classification. The RSA block is our attention block that learns the importance of each spectral feature. This block consists of reshape layers and a softmax function. We exploit the softmax function to add probability information to the most important spectral features. The output of the RSA block is a spectral attention maps, which assign a weight to each spectral feature according to its relevance for the classification task.

The spatial attention module is applied to the HSI data to learn the importance of each spatial location. Our spatial attention module is named the DSA block. This block consists of 2-D depthwise convolutions and softmax functions. We use depthwise to pay attention to spatial regions and exploit the softmax activation function to add the probability information to the important spatial feature. The output of the spatial attention module is a spatial attention map, which assigns a weight to each spatial location according to its relevance for the classification task.

Our classification system consists of four parts: EBIM, DSA, RSA, and hybrid CNNs with skip connections. The performance of the system is boosted by the interaction of these modules. Specifically, the EBIM adds entropy information to the HSIs. The DSA and RSA blocks are used to make spatial and spectral attention mechanisms. Our spatial and spectral attentions are based on the softmax function, which is a mathematical function to convert a vector of real numbers into a probability distribution. We add the probability information to the spectral and spatial information. In the spatial attention module, we utilize depthwise convolution to emphasize spatial information. Unlike traditional 2-D convolution, 2-D depthwise convolution conducts the convolution operation independently for each input channel. This method preserves the distinctiveness of each channel and prevents the mixing of information during the convolution process. As a result, it is more effective in capturing spatial features. In addition, the proposed DNN architecture with residual blocks and the combination of 2-D CNNs with 3-D CNNs enhances the overall performance of the system. We go into more detail about the proposed technique in this section.

A. Entropy Base Information Module

The architecture of the EBIM is shown in Fig. 2. The HSI is denoted as $G \in R^{W \times H \times S}$, with $W \times H$ denoting the spatial dimensions, and S representing the number of spectral dimensions. The HSIs are often characterized by their large and complex data due to the presence of hundreds or even thousands of spectral bands. This poses challenges for data storage, processing, and analysis. It also leads to the curse of dimensionality meaning that the data become sparse and noisy in high-dimensional spaces, and the distance between data points becomes less meaningful

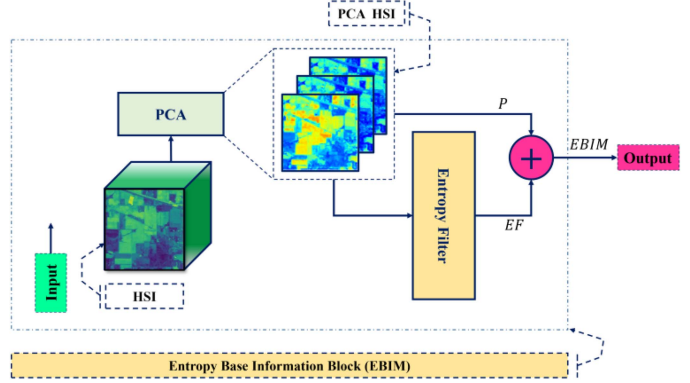


Fig. 2. Architecture of the EBIM.

[76], [77]. To mitigate the challenges associated with the size and complexity of HSI data, it is often necessary to employ dimensionality reduction techniques. These techniques aim to reduce the data size and complexity while ensuring that the relevant information is preserved. The initial data as $G \in R^{W \times H \times S}$ are processed by eliminating the redundancy of spectral bands. Here, we have $P \in R^{W \times H \times C}$, which is the output of the PCA block. After applying PCA, P is sent to the entropy filter block for further processing.

The entropy provides quantitative information about the structure and complexity of images, which can be useful in image processing. The Shannon entropy formula, which is based on the probability distribution of the pixel intensities throughout the image, is used to compute it. A higher global entropy indicates higher disorder or complexity in the image. In the context of grayscale images, higher entropy might suggest a more textured or complex scene. The Shannon entropy can be written as follows:

$$Y = - \sum_{i=0}^{255} p_i \log_2(p_i). \quad (1)$$

In image processing, p_i is equal to $\text{hist}(L_i)$. Therefore, the global entropy of an image can be written as follows:

$$Y = - \sum_{i=0}^{255} \text{hist}(L_i) \log(\text{hist}(L_i)) \quad (2)$$

where L_i represents the intensity levels of the input image and $\text{hist}(L_i)$ is the normalized histogram of the image. Therefore, the following equation can be written for $\text{hist}(L_i)$:

$$\sum_{i=0}^{255} \text{hist}(L_i) = 1. \quad (3)$$

The local entropy is calculated independently for each local region, typically using a sliding window or kernel. It provides a map of entropy values across the image and highlights regions of interest that have varying levels of complexity or texture. High local entropy in a region might indicate the presence of edges, textures, or other patterns. We use an entropy filter with the kernel size of 3×3 to generate an output that each pixel contains the local entropy value of the 3-by-3 neighborhood

around the corresponding pixel in $P \in R^{W \times H \times C}$. Let EF be the output of the entropy filter. The last step of the EBIM is the summation of $P \in R^{W \times H \times C}$ and EF. Therefore, the output of the entropy filter can be expressed as follows:

$$\text{EBIM} = \text{EF} + P. \quad (4)$$

Fig. 3 shows the output of the EBIM that highlights regions in the HSIs.

B. Deep Neural Network

Our proposed DNN is shown in Fig. 4. It comprises four 3-D convolutional layers, each of which is subsequently followed by batch normalization layers. The first convolutional layer utilizes a kernel size of $(3 \times 3 \times 9)$, while the second and third convolutional layers employ a kernel size of $(3 \times 3 \times 7)$ and $(3 \times 3 \times 5)$, respectively. The last one has a kernel size of $(3 \times 3 \times 3)$. In addition, all the activation functions of convolutional layers are RELU. From the first 3-D convolution to the final convolution layer, we reduce the kernel size. It is a widely used approach in which the kernel size gradually reduces from the first layer to the last. This reduction in the kernel size has multiple purposes within the network architecture. In the initial layers of a deep network, larger kernel sizes are often employed. These larger kernels enable the extraction of low-level features and local patterns from the input data [78]. By using larger receptive fields, these early layers can capture broad spatial information, such as edges, corners, and textures. However, as the network progresses deeper into subsequent layers, the focus shifts toward higher level feature extraction. Therefore, smaller kernel sizes are preferred in these later layers because they allow for the capture of fine-grained details and localized features. By reducing the kernel size, the network can concentrate on extracting intricate patterns, complex relationships, and global semantics. These smaller kernels enable the network to learn higher level representations of the input data. In addition, decreasing the kernel size in deeper layers can help manage computational complexity and alleviate the risk of overfitting. The utilization of smaller kernels in each layer helps in reducing the number of parameters. Moreover, reducing the number of parameters prevents the network from excessively memorizing the training data, promoting generalization and increasing the model's ability to generalize effectively to unseen examples.

The proposed architecture incorporates two 2-D convolutional layers with a kernel size of (3×3) . To address the issue of gradient vanishing, the architecture also includes three residual connections. These residual connections are utilized to mitigate the problem of gradients vanishing. The input and output of the residual connection can be formulated as follows:

$$R_{\text{Output}} = F(R_{\text{Input}}) + R_{\text{Input}} \quad (5)$$

where R_{Output} and R_{Input} show the residual output and input, respectively. F denotes the residual function.

After four 3-D convolution layers, we exploit a reshaped layer to make the 3-D CNN output compatible with the 2-D convolutional layer input so that the 2-D convolutional layer can receive the 3-D CNN output. We consider the input tensor

of the reshape layer as $I_{\text{Reshape}} \in R^{a \times b \times c \times d}$, where the reshape layer transfers it into the output tensor $O_{\text{Reshape}} \in R^{a \times b \times e}$. Here, e is the multiplication of c and d . In addition, our network architecture incorporates spectral and spatial attention modules, which are designed to enhance the model's focus on important spectral and spatial features within the input data. After the attention modules, we apply a 2-D GAP to transform the output of a convolutional layer before applying a softmax layer. This pooling operation aggregates spatial information across each channel, resulting in a compressed representation that captures the overall context of the features. It decreases the number of trainable parameters and the computational complexity of the network, thereby improving efficiency and preventing overfitting. In addition, it enhances the feature representation and discrimination power of the network by obtaining the global average of each feature map [79]. This reveals the importance of each feature for the classification task. Furthermore, the architecture employs a concatenate layer to merge the output of spatial and spectral attention modules. The concatenate layer combines information from different pathways or branches in the network, enabling the model to capture and utilize diverse features or representations from multiple sources.

Our proposed network also incorporates a dropout, which is a regularization technique in deep learning to mitigate overfitting and improve the generalization ability of neural networks. This technique involves randomly dropping out a fraction of the neurons or during training. Our architecture incorporates a dropout rate of 40%. In the final two layers of the architecture, there are dense layers. The first dense layer consists of 128 neurons using the RELU activation function. The output classes of each dataset determine the number of neurons in the second dense layer. In addition, the activation function of the second dense layer is the softmax.

C. Reshape SoftMax Attention

Spectral attention is a useful technique that enhances the robustness and accuracy of HSI classification by selectively focusing on the spectral information present in the input image. In our model, we incorporate a reshape layer and a softmax layer to implement the spectral attention. Fig. 5 shows the architecture of the RSA block. As shown in this figure, the input of the RSA block, $\text{RSA}_{\text{input}} \in R^{a \times b \times c}$, is transformed into two matrices using two separate branches and reshaped layers. The reshape layer processes $\text{RSA}_{\text{input}}$, resulting in two outputs: $\text{Re1}_{\text{output}} \in R^{d \times c}$ and $\text{Re2}_{\text{output}} \in R^{d \times c}$. The reshape function is applied to preserve the spectral features and merge the spatial information. To further emphasize high values and de-emphasize low values, we perform matrix multiplication between these two matrices. This multiplication serves to enhance the importance of higher values while diminishing the significance of lower values. The multiplication of these two matrices can be summarized as follows:

$$\text{Re}_{\text{Output}} = \text{Re1}_{\text{output}} \odot \text{Re2}_{\text{output}} \quad (6)$$

where \odot is the elementwise multiplication and $\text{Re}_{\text{Output}} \in R^{d \times c}$ indicates the output of the elementwise multiplication. Then, we

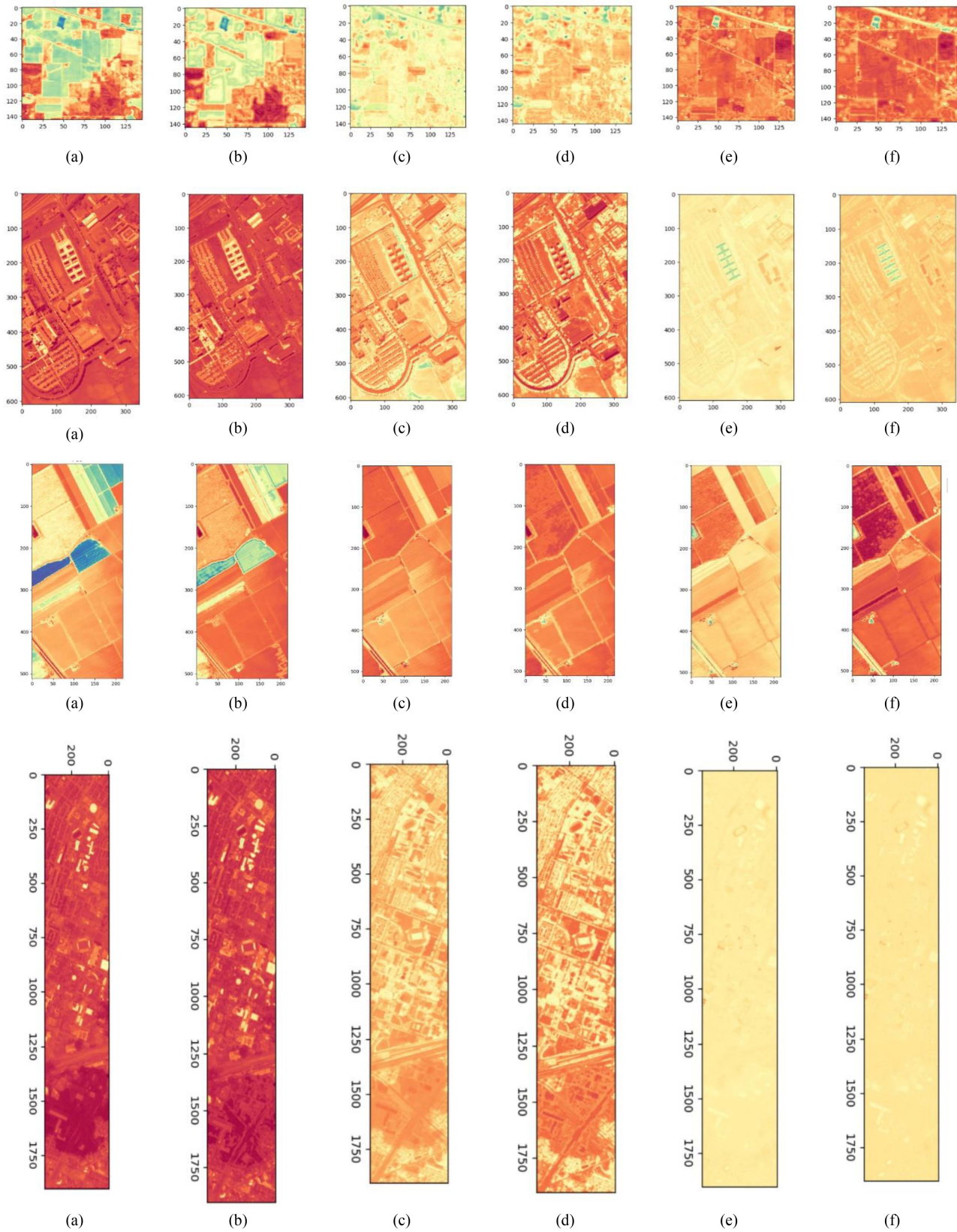


Fig. 3. Three selected bands by PCA: the three selected bands and their relative outputs for the different datasets (first row: IP; second row: PU; third row: SA; fourth row: HT). (a) First band selected by PCA. (b) Output of the EBIM for the first band. (c) Second band selected by PCA. (d) Output of the EBIM for the second band. (e) Third band selected by PCA. (f) Output of the EBIM for the third band, respectively.

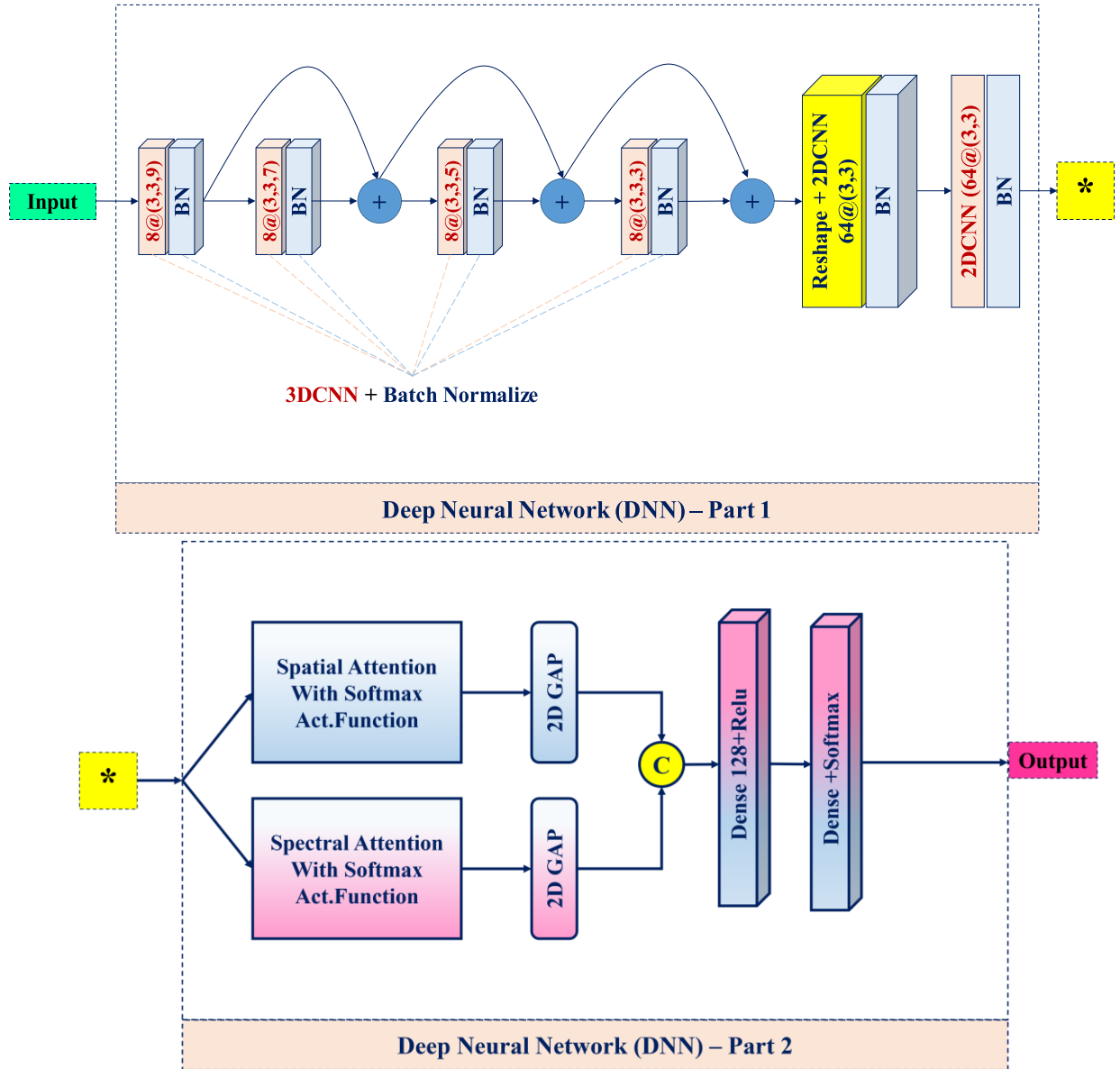


Fig. 4. Architecture of the DNN.

use a softmax layer to normalize its weights. S_{RSA} represents the output of the softmax function. The output of the softmax function can be formulated as follows:

$$S_{RSA} = \text{softmax}(Re_{Output}). \quad (7)$$

The equation mentioned above calculates the exponential of the input value and the sum of the exponential values of all the input values. The output of the softmax function corresponds to the ratio between the exponential of the input value and the sum of the exponentials of all the input values. Let us consider R_i as each element of Re_{Output} , then the softmax function can be summarized as follows:

$$\text{softmax}(R_i) = \frac{e^{R_i}}{\sum_{i=1}^k e^{R_i}}. \quad (8)$$

The output of the softmax layer is subsequently multiplied by the output of the second reshape layer. It can be formulated as follows:

$$Re3_{input} = S_{RSA} \odot Re2_{output} \quad (9)$$

where $Re3_{input}$ denotes the input of the third reshape layer. Finally, the output of the RSA block can be summarized as follows:

$$RSA_{output} = Re3_{output} + RSA_{input}. \quad (10)$$

In the above-mentioned formula, $Re3_{output} \in R^{a \times b \times c}$ shows the output of the third reshape layer and RSA_{output} denotes the RSA output.

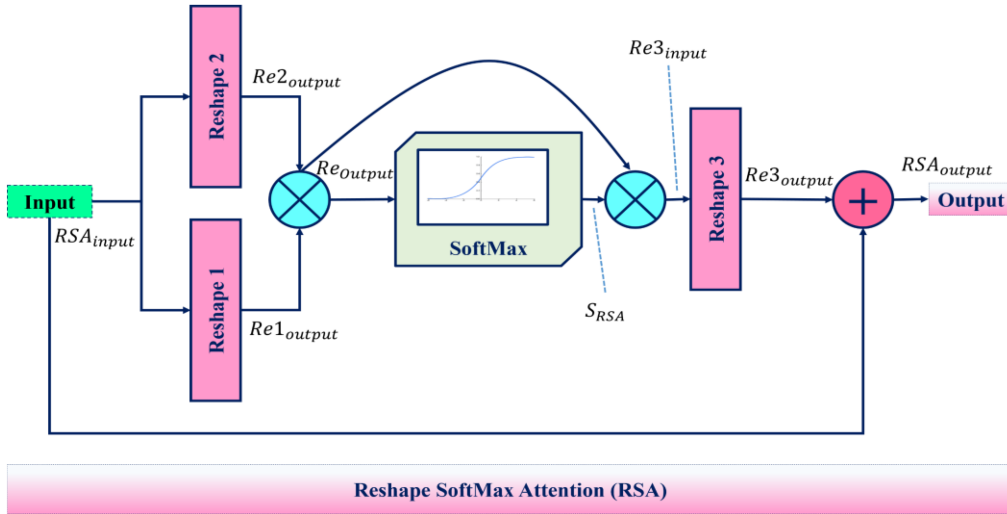


Fig. 5. Architecture of the RSA block.

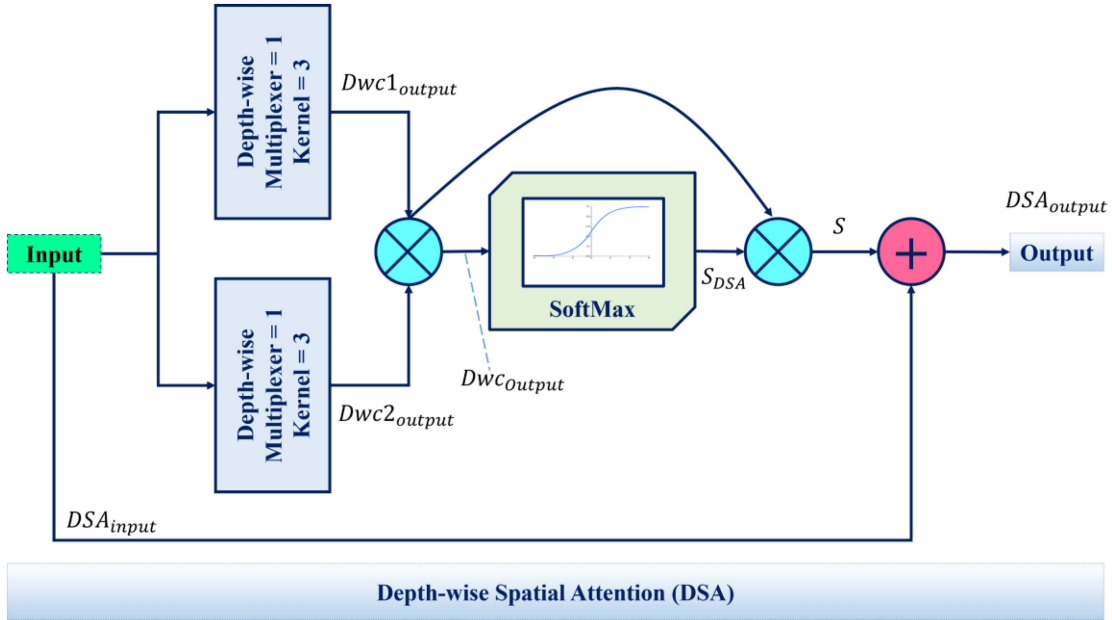


Fig. 6. Architecture of the DSA block.

D. Depthwise Spatial Attention

A spatial attention module is a technique that improves the performance of HSI classification by focusing on the most relevant spatial features and regions of the input image. Fig. 6 shows the architecture of the DSA block. In our spatial attention module, we employ 2-D depthwise convolutions along with a softmax layer. The depthwise convolution is applied to the output of the 2-D CNN. This operation performs separate convolutions for each input channel, effectively reducing the number of parameters and computational complexity of the network [80]. Simultaneously, it preserves the spatial information of the input image, which can improve the feature extraction and representation. We also use a softmax layer to normalize the weights of the

spatial attention module. The considerations for the arguments of depthwise convolutions are determined as follows: the depth multiplier parameter is set to one, indicating that the number of output channels remains the same as the number of input channels. The kernel size is set to three, implying that a 3×3 convolutional filter is applied. In addition, a stride value of one is employed, indicating that the filter moves one pixel at a time during the convolution operation.

The input of the DSA block is from the last 2-D CNN. It is considered as $DSA_{input} \in R^{a \times b \times c}$. The outputs of 2-D depthwise convolutions have the same size as the DSA input. Two depthwise convolutions are applied to the DSA_{input} , resulting in two outputs $Dwc1_{output} \in R^{a \times b \times c}$ and $Dwc2_{output} \in R^{a \times b \times c}$. To further emphasize high values and de-emphasize

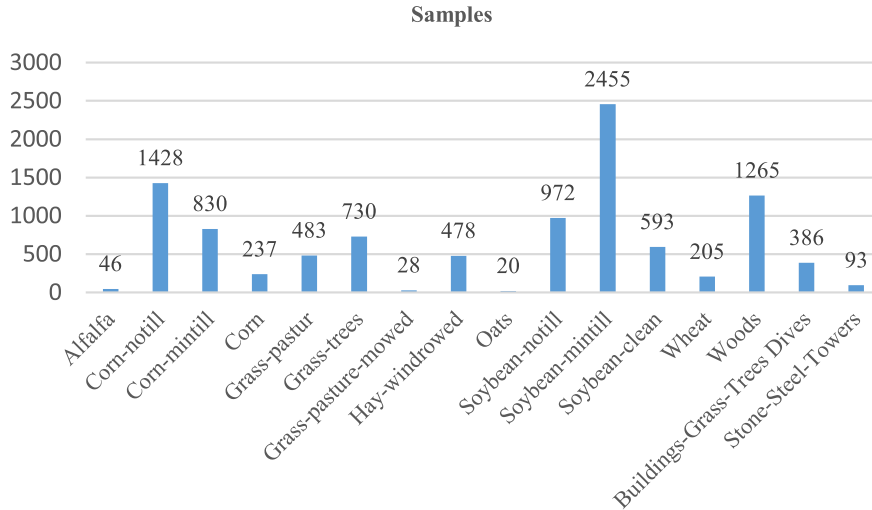


Fig. 7. Class distribution of the IP dataset.

low values, we perform multiplication between these two tensors. The multiplication of these tensors can be summarized as follows:

$$Dwc_{Output} = Dwc1_{output} \odot Dwc2_{output} \quad (11)$$

where $Dwc_{Output} \in R^{a \times b \times c}$ indicates the output of elementwise multiplication. The product of this multiplication is then passed through a softmax function. The softmax normalizes the values, typically to highlight the most significant spatial features by assigning them higher probabilities. S_{DSA} represents the output of the softmax function. It can be formulated as follows:

$$S_{DSA} = \text{softmax}(Dwc_{Output}). \quad (12)$$

The output of the softmax layer is then multiplied by the output of the second depthwise convolution. In fact, this operation is done to emphasize the element of the $Dwc1_{output}$ tensor based on the probability information. It can be formulated as follows:

$$S = S_{DSA} \odot Dwc2_{output} \quad (13)$$

where S denotes the input of the add operation in this figure. Finally, the output of the DSA block, DSA_{output} , can be summarized as follows:

$$DSA_{output} = S + DSA_{input}. \quad (14)$$

IV. EXPERIMENT AND ANALYSIS

In this section, we present a comparative analysis of our approach with methods employing different techniques. The proposed method has been assessed and evaluated on four benchmark datasets, namely Pavia University (PU), Salinas (SA), Houston 2013 (HT), and Indian Pines (IP). These datasets were selected to provide a diverse range of scenarios and contexts for testing and validating the effectiveness and performance of our approach. The methods are evaluated by three main metrics that measure the classification performance: Average accuracy (AA), OA, and kappa coefficient. Our proposed method has been

implemented on Python-Keras. The Google Collab Plus with V100 GPU is used to implement the proposed method.

A. Hyperspectral Datasets Description

We evaluated our classification system by utilizing four benchmark datasets of remote sensing. These datasets were thoughtfully chosen to cover a suitable range of classes and samples. To facilitate an accurate evaluation of our method and highlight the key characteristics of these datasets, we have explained comprehensive information about these datasets.

1) *Indian Pines Dataset*: IP contains HSIs of a landscape in Indiana, U.S., with 145×145 pixels and 200 spectral bands. The images cover different types of land cover, such as agriculture, forest, and water. The dataset has 16 classes of labels, such as corn, soybean, and alfalfa. Moreover, the spatial resolution is 20 m per pixel. Fig. 7 describes the class distribution of the IP dataset.

2) *Salinas Dataset*: The SA dataset is an image of agricultural crops and natural vegetation in Salinas Valley, California. The dataset contains a real ground image, sample class information, and a false color image. The image has a spatial resolution of 3.7 m per pixel and a size of 512×217 pixels. The dataset has 16 classes of land cover and 224 bands, such as vineyards, crops, and arid soils. The dataset was prepared by excluding 20 water absorption bands. Therefore, this dataset has 204 bands. The class distribution of the SA dataset is shown in Fig. 8.

3) *Houston Dataset*: Fig. 9 illustrates the class information of the HT dataset. The data were collected from the University of Houston (HT) campus and the surrounding metropolitan area. It contains 144 spectral bands with a spatial resolution of 2.5 m. The Houston dataset encompasses 16 classes and has an image resolution of 349×1905 pixels.

4) *Pavia University Dataset*: Fig. 10 shows the class distribution of the University of Pavia (PU). The dataset was collected from the University of Pavia in northern Italy. The dataset has

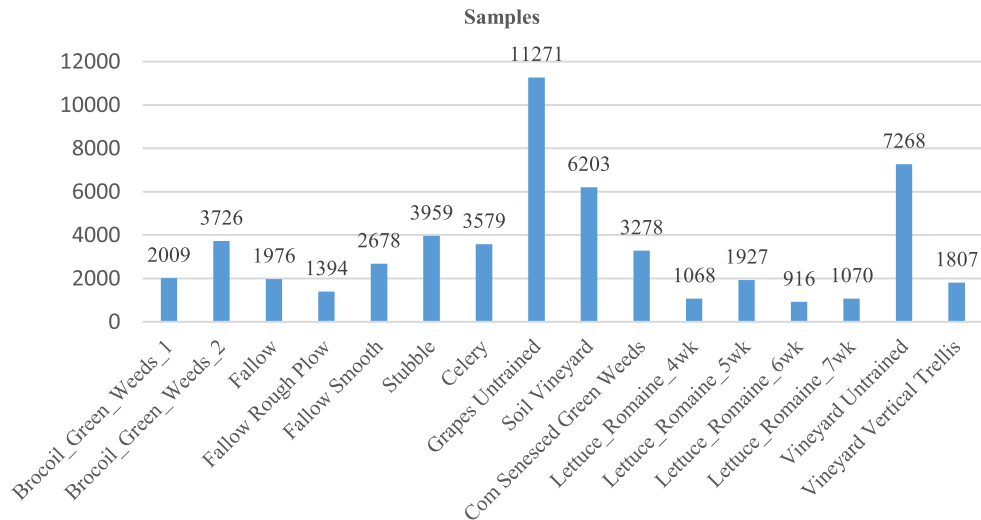


Fig. 8. Class distribution of the SA dataset.

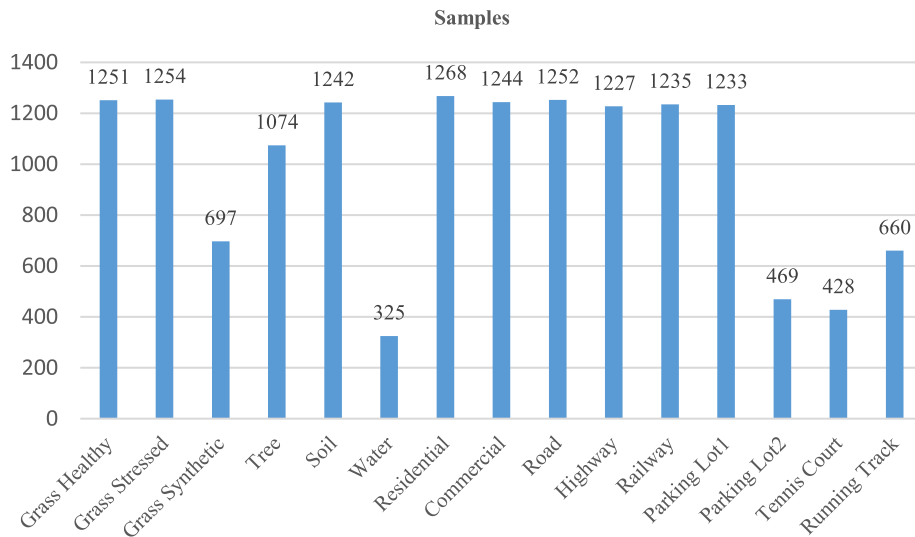


Fig. 9. Class distribution of the HT dataset.

nine urban land cover classes. The dataset comprises 115 spectral bands with a spatial resolution of 1.3 m per pixel. The image dimensions are 610×340 pixels. As part of the preprocessing stage, 12 noisy bands were eliminated from the dataset, resulting in a final set of 103 usable bands for subsequent research.

B. Experiment With Other Methods

We conduct experiments with other methods to compare and assess the performance of our proposed approach. By employing this comparative analysis, we aim to evaluate the weaknesses and strengths of our method and demonstrate its competitive results against methods, such as MCHN [58], SSLDBR [41], SSFLT [52], ARL-GAN [45], DSC-MMF [66], Lite-HCNet [67], WaveFormer [53], and GSC-ViT [54].

The 2D-CNN technique utilizes 2-D convolution kernels to capture spatial features from input images. However, the development of 3D-CNN allows for the simultaneous learning of both spectral and spatial features. Some researchers use morphological operations to boost the performance of the 3-D CNNs [58]. There is a dual-branch residual neural network on SSLDBR [41]. A GAN is exploited in ARL-GAN [45]. In SSFLT [52], a transformer framework is utilized to extract spectral and spatial information. In DSC-MMF [66], a new model based on CNNs is proposed. Three branches are exploited to build this proposed architecture. Lite-HCNet [67] is a technique that uses depthwise separable convolution. WaveFormer [53] is a technique based on the wavelet and transformer. GSC-ViT [54] presents a group-separable convolutional vision transformer network.

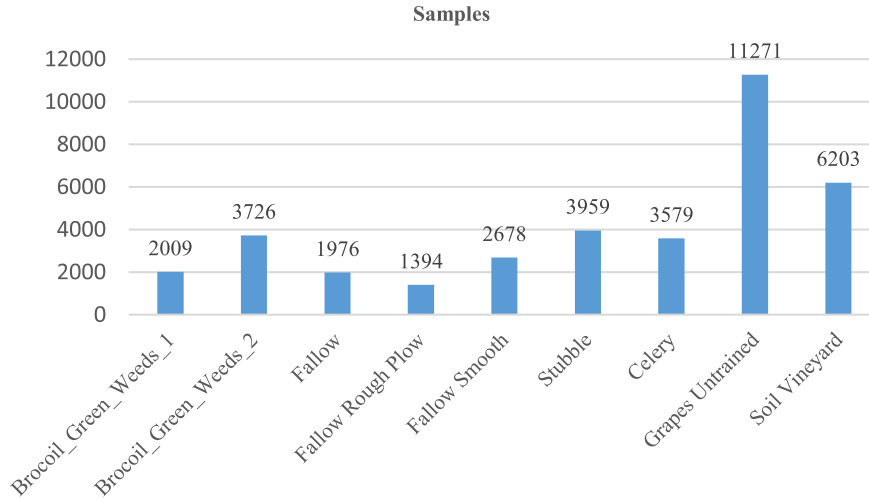


Fig. 10. Class distribution of the PU dataset.

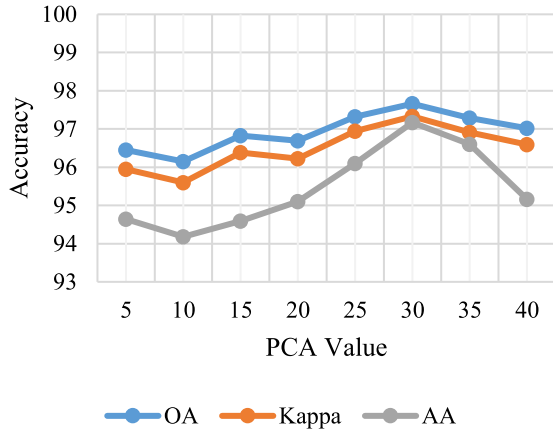


Fig. 11. Effect of the PCA value on the IP dataset (spatial size = 15).

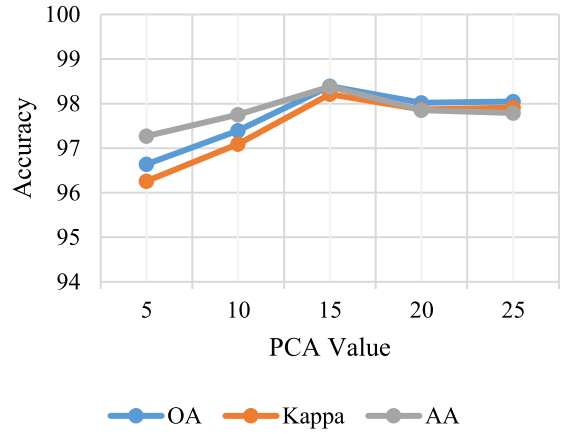


Fig. 12. Effect of the PCA value on the SA dataset (spatial size = 15).

C. Experimental Setup and Comparison

This section is dedicated to conducting experiments to determine the parameters that influence the performance of the proposed method. In particular, our analysis is focused on the number of PCA and the spatial size of the patches for the input of the network. In the training procedure, we employ the Adam optimization algorithm [81], with a learning rate of 0.001. The batch size of datasets is set to 256. The number of samples for training is different for each dataset. For SA and PU, it is set to 1% of samples. For IP and HT, it is set to 5% of data.

To obtain the effect of principal components on classification accuracy, different runs are conducted on four benchmark datasets. The datasets have a window size of 15 and the results for different numbers of principal components are shown in Figs. 11–14. These figures indicate that the best performance for IP is achieved with 30 principal components, while for the other datasets, it is 15. Appropriate choice of PCA value has a strong impact on the classification performance. In all the experiments, the model performance and

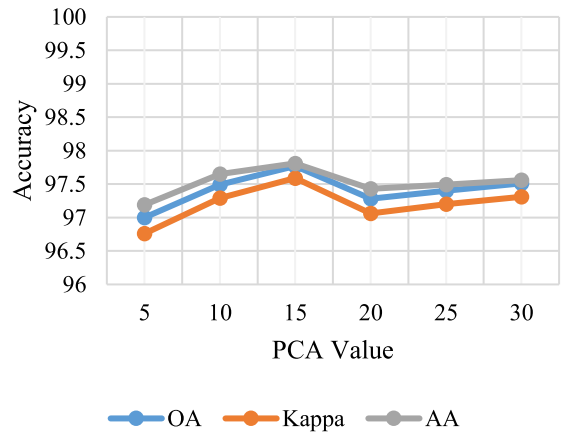


Fig. 13. Effect of the PCA value on the HT dataset (spatial size = 15).

its stability are satisfactory according to the selected PCA value.

The classifier of the HSI classification systems takes patches as inputs. We examine the impact of different spatial sizes

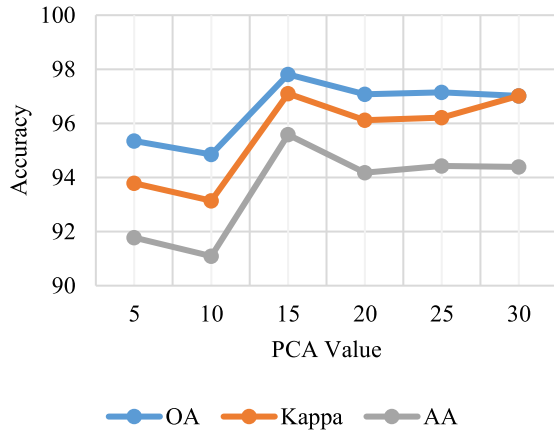


Fig. 14. Effect of the PCA value on the PU dataset (spatial size = 15).

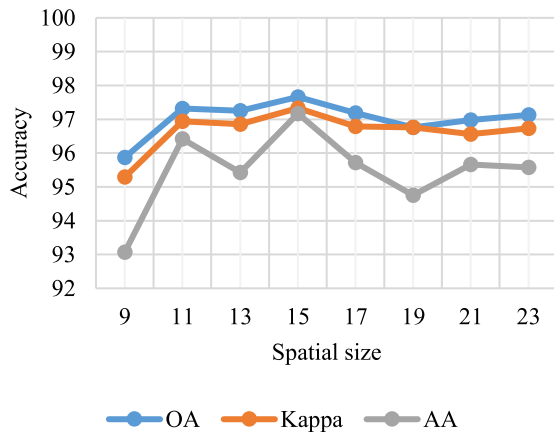


Fig. 15. Effect of spatial size on the IP dataset (PCA = 30).

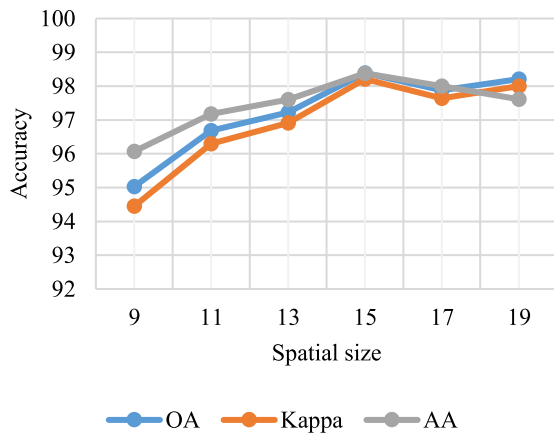


Fig. 16. Effect of spatial size on the HT dataset (PCA = 15).

on our suggested model. The comparison results for different datasets are reported in Figs. 15–18. The results indicate that the classification accuracy is influenced by the spatial size. A smaller spatial size leads to a reduced amount of information being captured from the object, resulting in lower accuracy. When the spatial size is large, the network gets more information from the object but also gets more noise and interference from other objects, which can reduce the accuracy

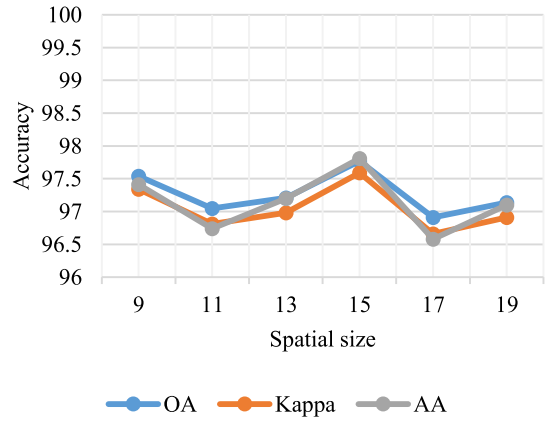


Fig. 17. Effect of spatial size on the HT dataset (PCA = 15).

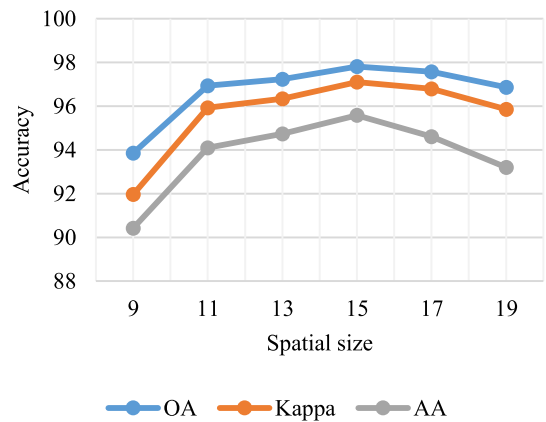


Fig. 18. Effect of spatial size on the PU dataset (PCA = 15).

[68]. Therefore, there is an optimal range of spatial size that can improve the classification accuracy significantly for four datasets.

In conclusion, the selection of setup parameters plays a vital role in evaluating the classification performance effectively. Factors, such as the optimization algorithm, appropriate train/test ratio as well as the spatial size of patches, and the selection of appropriate PCA values, are all important and can affect the efficiency and accuracy of the proposed model. Through meticulous parameter selection, we can obtain enhanced performance and accuracy in HSI classification.

We evaluate the proposed method with deep learning methods. To assess the performance of our model, we adopted AA, kappa coefficient, and OA. The OA is the percentage of correctly classified pixels. The AA is the mean value of the OAs measured over each category and the kappa coefficient is the statistical measurement of the interrater agreement among qualitative items. The AA is calculated as

$$AA = \frac{1}{n} \sum_{i=1}^n \frac{TP_i}{TP_i + FN_i} \quad (15)$$

where

- n number of classes;
- TP_i number of true positives for class i ;
- FN_i number of false negatives for class i .

TABLE III
COMPARISON RESULTS FOR THE IP DATASET (PCA = 30, SPATIAL SIZE = 15)

Classes	MCHN	SSLDBR	SSFLT	ARL-GAN	DSC_MFF	Lite_HCNet	WaveFormer	GSC-ViT	Proposed Method
C0	93.87 ± 0.93	96.00 ± 1.72	94.32 ± 0.68	91.62 ± 2.51	96.09 ± 1.68	93.84 ± 0.89	86.21 ± 4.63	82.89 ± 1.15	100.00 ± 0.0
C1	85.17 ± 0.51	94.11 ± 0.55	97.18 ± 1.36	98.04 ± 0.09	95.67 ± 0.86	97.29 ± 1.08	93.01 ± 2.86	93.72 ± 2.83	96.19 ± 0.29
C2	81.32 ± 0.82	93.25 ± 0.63	98.43 ± 0.32	94.59 ± 1.33	98.35 ± 1.05	98.37 ± 0.23	94.73 ± 3.72	97.55 ± 1.81	99.46 ± 0.11
C3	67.94 ± 5.23	86.18 ± 0.58	92.68 ± 1.30	93.56 ± 3.24	96.67 ± 3.41	92.63 ± 2.71	90.18 ± 5.34	89.45 ± 3.14	91.33 ± 1.20
C4	92.56 ± 0.25	95.44 ± 0.92	95.61 ± 0.51	95.89 ± 2.31	97.58 ± 2.66	94.31 ± 1.92	92.25 ± 3.17	92.65 ± 3.19	99.95 ± 0.09
C5	96.43 ± 0.43	98.14 ± 0.31	98.93 ± 1.14	97.39 ± 0.36	96.39 ± 1.08	95.12 ± 0.68	98.96 ± 0.72	99.09 ± 0.88	98.92 ± 0.58
C6	65.36 ± 4.36	88.41 ± 5.71	100.00 ± 0.0	94.36 ± 1.72	93.54 ± 2.18	95.01 ± 3.52	78.96 ± 2.74	93.30 ± 2.51	100.00 ± 0.0
C7	98.13 ± 0.2	99.75 ± 0.09	100.00 ± 0.0	99.18 ± 0.03	100.00 ± 0.0	96.12 ± 0.49	96.38 ± 1.08	100.00 ± 0.00	98.90 ± 0.78
C8	71.86 ± 2.93	87.23 ± 4.04	90.58 ± 4.51	92.49 ± 1.36	91.68 ± 1.89	92.58 ± 0.06	97.37 ± 0.89	79.92 ± 7.14	97.37 ± 2.63
C9	85.61 ± 0.70	93.82 ± 1.45	96.83 ± 2.17	94.86 ± 0.17	96.75 ± 0.92	94.36 ± 0.18	75.56 ± 8.42	96.41 ± 1.67	98.29 ± 0.34
C10	91.57 ± 1.23	96.34 ± 2.07	98.09 ± 0.81	99.21 ± 0.74	96.87 ± 1.82	93.76 ± 1.04	94.13 ± 2.58	98.22 ± 0.47	98.61 ± 0.31
C11	92.71 ± 0.20	96.07 ± 0.86	97.22 ± 1.64	97.35 ± 0.14	93.59 ± 3.33	94.28 ± 2.63	96.58 ± 1.86	94.98 ± 2.64	91.12 ± 0.64
C12	90.18 ± 0.87	93.70 ± 0.95	100.00 ± 0.0	94.82 ± 1.77	91.55 ± 2.22	90.92 ± 1.46	91.18 ± 2.82	99.63 ± 0.28	100.00 ± 0.0
C13	99.12 ± 0.01	98.76 ± 0.64	97.59 ± 1.44	100.00 ± 0.0	99.16 ± 0.52	96.18 ± 0.73	99.57 ± 0.40	99.42 ± 0.29	99.42 ± 0.08
C14	73.28 ± 5.83	88.34 ± 7.42	98.23 ± 0.76	98.19 ± 0.49	92.98 ± 0.39	92.41 ± 0.59	91.02 ± 3.16	97.09 ± 2.46	92.51 ± 2.25
C15	86.62 ± 1.58	93.69 ± 2.47	94.68 ± 1.32	91.42 ± 1.63	96.32 ± 2.10	93.75 ± 1.34	92.31 ± 0.73	96.18 ± 1.97	92.61 ± 4.14
OA	85.93 ± 0.68	93.60 ± 1.42	96.51 ± 1.24	95.83 ± 0.24	96.53 ± 0.98	94.68 ± 0.19	91.65 ± 0.47	94.83 ± 0.53	97.66 ± 0.08
Kappa	84.68 ± 0.49	92.76 ± 1.20	95.38 ± 0.68	95.12 ± 0.90	94.86 ± 1.51	94.19 ± 0.72	92 ± 1.82	94.12 ± 0.48	97.33 ± 0.10
AA	85.73 ± 0.39	93.7 ± 1.09	96.89 ± 0.54	95.81 ± 1.06	95.82 ± 0.76	94.43 ± 0.98	91.77 ± 0.49	94.4 ± 0.76	97.17 ± 0.26

Bold values represent the best significant results.

The kappa coefficient is calculated as

$$\text{Kappa coefficient} = \frac{p_o - p_e}{1 - p_e}$$

where

- p_o observed accuracy (OA);
- p_e expected accuracy by chance, calculated as

$$p_e = \sum_{i=1}^n \frac{(TP_i + FN_i) \times (TP_i + FP_i)}{N^2}.$$

In the aforementioned expression

- 1) FP_i is the number of false positives for class i .
- 2) N is the total number of observations.

The OA is calculated as

$$\text{OA} = \frac{\sum_{i=1}^n TP_i}{N}.$$

These metrics are essential for evaluating the performance of classification models, particularly in multiclass scenarios. We repeated the training process ten times to ensure the reliability and robustness of the reported results. Then, we reported the mean and standard deviation (SD) of AA, kappa, and OA. Tables III and IV list the comparison results of IP and SA datasets, respectively. Similarly, the comparison results of HT and PU datasets are presented in Tables V and VI, respectively. The results demonstrate the superior performance of our suggested method compared to others. On the IP dataset, we achieved an OA of 97.66% with an SD of 0.08%. Its Kappa

and AA were 97.33%, 97.17% with 0.1%, and 0.26% SD, respectively. MCHN had an inferior result. It had 85.93% OA, 84.68% Kappa, and 85.73% AA, respectively. Our approach achieved high performance on the SA dataset, with 98.39% OA, 98.21% Kappa, and 98.38% AA. The SD of these metrics was 0.53%, 0.6%, and 0.83%, respectively. SSFLT had the lowest OA among other methods on the SA dataset. Its OA was 95.24%. The MCHN with 94.61% Kappa and SSFLT with 95.12% AA had the lowest Kappa and AA. For the HT dataset, we obtained 97.77% OA, 97.59% Kappa, and 97.81% AA with an SD of 0.5%, 0.54%, and 0.43%, respectively. For the HT dataset, MCHN had the lowest accuracy compared to others. It had 91.13% OA, 89.98% Kappa, and 90.73% AA. Notably, the PU dataset was 97.81% OA, 97.1% Kappa, and 95.58% AA. The SD of these metrics was 0.12%, 0.16%, and 0.09%, respectively. Lite-HCNet has the worst results with 93.27% OA, 92.75% Kappa, and 93.03% AA. Figs. 19–22 display the visualization results of different datasets, including the ground truth and the corresponding classification outcomes and Fig. 23 shows the class color for ground truth and visualization results of different datasets.

Although hybrid DNNs are capable of capturing global spectral and spatial information, their ability to classify HSIs boosts when we add and exploit three proposed modules. Our method surpasses others in terms of OA, Kappa, and AA. Our method primarily utilizes the EBIM that adds entropy information to the input data. Specifically, the entropy filter identifies areas with high entropy, which typically contain more detail or texture, and areas with low entropy, which are more uniform or smooth.

TABLE IV
COMPARISON RESULTS FOR THE SA DATASET (PCA = 15, SPATIAL SIZE = 15)

Classes	MCHN	SSLDBR	SSFLT	ARL-GAN	DSC_MFF	Lite_HCNet	WaveFormer	GSC-ViT	Proposed Method
C0	96.26 ± 2.61	97.35 ± 2.07	95.79 ± 2.49	98.23 ± 1.27	98.46 ± 0.46	100.00 ± 0.0	98.19 ± 2.27	99.44 ± 0.28	99.81 ± 0.32
C1	93.41 ± 4.32	95.92 ± 1.88	98.17 ± 1.04	98.26 ± 0.46	96.58 ± 1.97	96.19 ± 1.19	99.49 ± 0.51	100.00 ± 0.00	100.00 ± 0.0
C2	95.73 ± 1.19	96.84 ± 2.06	95.33 ± 2.73	99.12 ± 0.27	97.89 ± 1.93	99.46 ± 0.07	95.48 ± 0.22	98.36 ± 1.15	100.00 ± 0.0
C3	92.96 ± 4.38	92.82 ± 3.13	94.16 ± 0.88	99.74 ± 0.15	98.57 ± 0.59	91.33 ± 3.38	98.96 ± 0.34	98.97 ± 1.24	97.93 ± 2.12
C4	100.00 ± 0.0	99.26 ± 0.21	97.82 ± 1.23	96.83 ± 1.09	91.34 ± 4.29	99.95 ± 0.01	98.42 ± 0.83	98.25 ± 2.14	98.15 ± 1.64
C5	100.00 ± 0.0	99.06 ± 0.42	98.26 ± 1.07	100.00 ± 0.0	94.58 ± 1.66	98.92 ± 0.88	99.48 ± 0.59	99.86 ± 1.34	99.06 ± 1.72
C6	94.37 ± 0.72	96.54 ± 1.61	95.25 ± 2.91	100.00 ± 0.0	96.49 ± 2.37	100.00 ± 0.0	99.90 ± 0.09	99.58 ± 0.19	99.88 ± 0.14
C7	95.57 ± 1.24	97.53 ± 1.83	98.11 ± 0.43	91.43 ± 3.81	96.73 ± 0.98	98.90 ± 0.94	92.49 ± 2.73	93.27 ± 2.99	99.25 ± 1.05
C8	90.43 ± 2.51	93.49 ± 2.08	92.67 ± 3.61	98.76 ± 1.07	100.00 ± 0.0	97.37 ± 1.25	99.43 ± 0.41	99.87 ± 0.23	99.25 ± 1.34
C9	93.21 ± 2.63	94.35 ± 2.71	91.56 ± 4.82	91.59 ± 2.83	97.18 ± 1.55	98.29 ± 0.64	95.82 ± 2.76	98.12 ± 1.98	99.04 ± 0.90
C10	95.24 ± 1.15	96.08 ± 1.92	94.39 ± 0.72	93.47 ± 1.19	91.63 ± 1.67	98.61 ± 1.09	96.23 ± 3.82	98.88 ± 1.27	99.36 ± 0.60
C11	90.87 ± 4.93	91.19 ± 1.49	91.58 ± 0.56	98.14 ± 0.92	92.48 ± 1.56	91.12 ± 3.97	99.34 ± 0.28	99.64 ± 0.87	98.29 ± 1.89
C12	94.83 ± 0.72	96.60 ± 0.98	94.98 ± 1.44	97.63 ± 0.43	97.90 ± 0.83	100.00 ± 0.0	99.00 ± 1.38	98.53 ± 2.43	90.92 ± 8.27
C13	100.00 ± 0.0	99.62 ± 0.08	99.43 ± 0.27	92.51 ± 4.84	96.51 ± 0.76	99.42 ± 0.32	98.72 ± 0.37	99.32 ± 0.13	99.59 ± 0.43
C14	92.43 ± 0.94	91.93 ± 3.49	90.85 ± 5.13	88.69 ± 5.92	98.43 ± 0.51	92.51 ± 1.29	93.08 ± 2.61	91.01 ± 3.72	93.53 ± 1.61
C15	93.36 ± 1.37	92.96 ± 1.09	92.91 ± 2.06	95.41 ± 1.26	97.58 ± 0.28	92.61 ± 2.83	96.38 ± 1.42	96.73 ± 3.18	99.78 ± 0.34
OA	96.38 ± 0.45	96.43 ± 0.73	95.24 ± 1.31	96.34 ± 0.71	96.42 ± 0.36	97.66 ± 0.25	97.03 ± 0.76	97.92 ± 0.38	98.39 ± 0.53
Kappa	94.61 ± 1.57	95.66 ± 1.08	95.04 ± 0.97	95.18 ± 2.16	95.64 ± 1.03	97.33 ± 0.40	97.12 ± 0.81	98.1 ± 0.62	98.21 ± 0.60
AA	94.91 ± 1.03	95.72 ± 1.16	95.12 ± 1.02	96.23 ± 0.39	96.39 ± 0.22	97.17 ± 0.29	97.53 ± 0.94	98.11 ± 0.40	98.38 ± 0.83

Bold values represent the best significant results.

TABLE V
COMPARISON RESULTS FOR THE HT DATASET (PCA = 15, SPATIAL SIZE = 15)

Classes	MCHN	SSLDBR	SSFLT	ARL-GAN	DSC_MFF	Lite_HCNet	WaveFormer	GSC-ViT	Proposed Method
C0	85.62 ± 0.56	95.33 ± 1.18	96.14 ± 2.91	95.95 ± 0.27	96.19 ± 1.85	95.17 ± 1.08	99.42 ± 0.41	99.59 ± 0.38	98.22 ± 1.41
C1	86.79 ± 0.78	98.75 ± 0.35	97.34 ± 1.09	98.05 ± 0.59	97.24 ± 0.73	94.98 ± 2.23	82.64 ± 5.91	98.24 ± 1.79	97.82 ± 0.31
C2	98.17 ± 1.06	98.85 ± 0.61	94.72 ± 2.31	95.10 ± 0.65	98.37 ± 0.28	96.42 ± 0.73	96.67 ± 1.89	99.47 ± 0.5	99.35 ± 0.34
C3	94.51 ± 0.61	99.14 ± 0.52	96.59 ± 1.26	95.57 ± 1.53	98.16 ± 1.20	95.67 ± 1.24	95.38 ± 2.43	100.00 ± 0.00	98.73 ± 0.98
C4	100.00 ± 0.0	99.59 ± 0.09	99.39 ± 0.39	96.80 ± 1.08	98.59 ± 0.51	97.05 ± 0.69	84.36 ± 7.81	87.28 ± 10.44	99.94 ± 0.11
C5	100.00 ± 0.0	98.09 ± 0.43	99.24 ± 0.44	97.58 ± 0.89	96.38 ± 0.98	93.89 ± 1.36	81.37 ± 10.97	89.09 ± 4.59	96.80 ± 0.73
C6	90.72 ± 3.12	88.23 ± 6.94	91.63 ± 3.22	95.69 ± 3.13	92.06 ± 1.19	90.52 ± 4.61	99.52 ± 0.38	99.31 ± 0.37	91.63 ± 1.21
C7	75.16 ± 2.46	95.78 ± 1.39	97.43 ± 0.61	93.60 ± 3.54	97.42 ± 1.54	95.81 ± 1.93	89.27 ± 7.32	97.29 ± 3.61	98.63 ± 0.79
C8	87.24 ± 10.32	96.43 ± 1.52	95.86 ± 3.12	94.92 ± 1.39	95.48 ± 0.83	92.07 ± 5.38	94.63 ± 3.72	91.02 ± 1.67	95.72 ± 1.67
C9	65.38 ± 5.98	97.48 ± 0.98	95.08 ± 2.43	94.66 ± 2.81	98.31 ± 0.45	95.98 ± 0.62	96.83 ± 1.73	99.19 ± 0.72	99.43 ± 0.65
C10	91.48 ± 3.17	98.23 ± 0.33	97.38 ± 1.95	97.16 ± 0.62	97.85 ± 1.31	96.11 ± 1.13	96.95 ± 2.17	96.22 ± 3.60	98.93 ± 1.44
C11	95.42 ± 0.49	94.98 ± 2.41	96.41 ± 0.37	95.82 ± 1.16	97.52 ± 0.99	95.28 ± 1.46	98.12 ± 1.67	99.51 ± 0.2	97.77 ± 0.75
C12	92.27 ± 1.25	96.58 ± 1.85	96.13 ± 0.61	94.52 ± 2.38	95.19 ± 2.74	91.83 ± 2.59	98.27 ± 3.08	99.42 ± 0.16	95.43 ± 3.20
C13	100.00 ± 0.0	98.73 ± 1.07	99.73 ± 0.22	98.57 ± 0.74	96.91 ± 0.38	96.42 ± 0.83	90.42 ± 5.71	99.53 ± 0.18	99.80 ± 0.26
C14	98.16 ± 4.13	97.43 ± 0.73	98.44 ± 0.83	97.41 ± 1.03	98.46 ± 0.52	95.76 ± 1.22	91.2 ± 5.71	99.1 ± 0.18	98.90 ± 0.80
OA	91.13 ± 1.78	96.84 ± 0.85	96.72 ± 0.19	96.43 ± 0.52	97.07 ± 0.38	95.43 ± 0.67	93.92 ± 1.27	96.35 ± 1.75	97.77 ± 0.50
Kappa	89.98 ± 2.41	95.68 ± 1.12	95.39 ± 0.88	95.21 ± 0.91	96.36 ± 1.23	94.87 ± 0.96	93.24 ± 1.62	96.05 ± 1.89	97.59 ± 0.54
AA	90.73 ± 2.03	96.93 ± 0.99	96.76 ± 0.57	96.10 ± 0.75	96.94 ± 0.75	94.86 ± 1.17	93.07 ± 1.48	96.95 ± 2.95	97.81 ± 0.43

Bold values represent the best significant results.

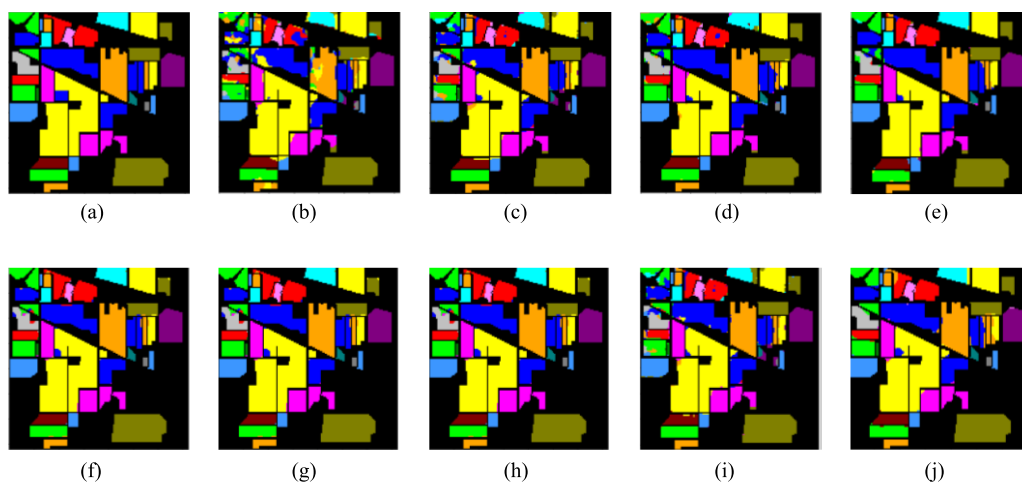


Fig. 19. Ground truth and visualization results of the IP dataset. (a) GT. (b) MCHN. (c) SSLDBR. (d) SSFLT. (e) ARL-GAN. (f) DSC_MFF. (g) Lite_HCNet. (h) WaveFormer. (i) GSC-ViT. (j) Proposed method.

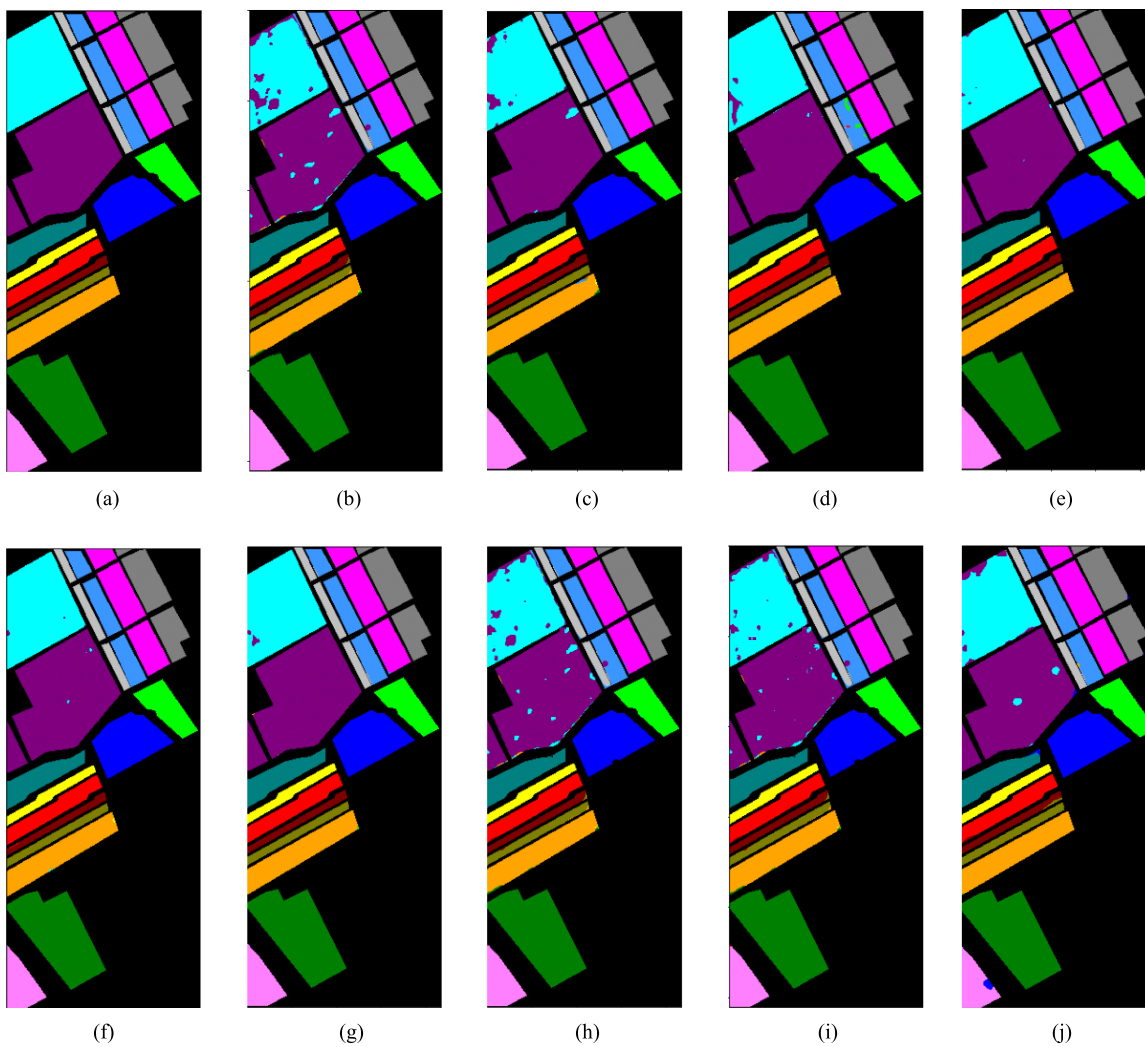


Fig. 20. Ground truth and visualization results of the SA dataset. (a) GT. (b) MCHN. (c) SSLDBR. (d) SSFLT. (e) ARL-GAN. (f) DSC_MFF. (g) Lite_HCNet. (h) WaveFormer. (i) GSC-ViT. (j) Proposed method.

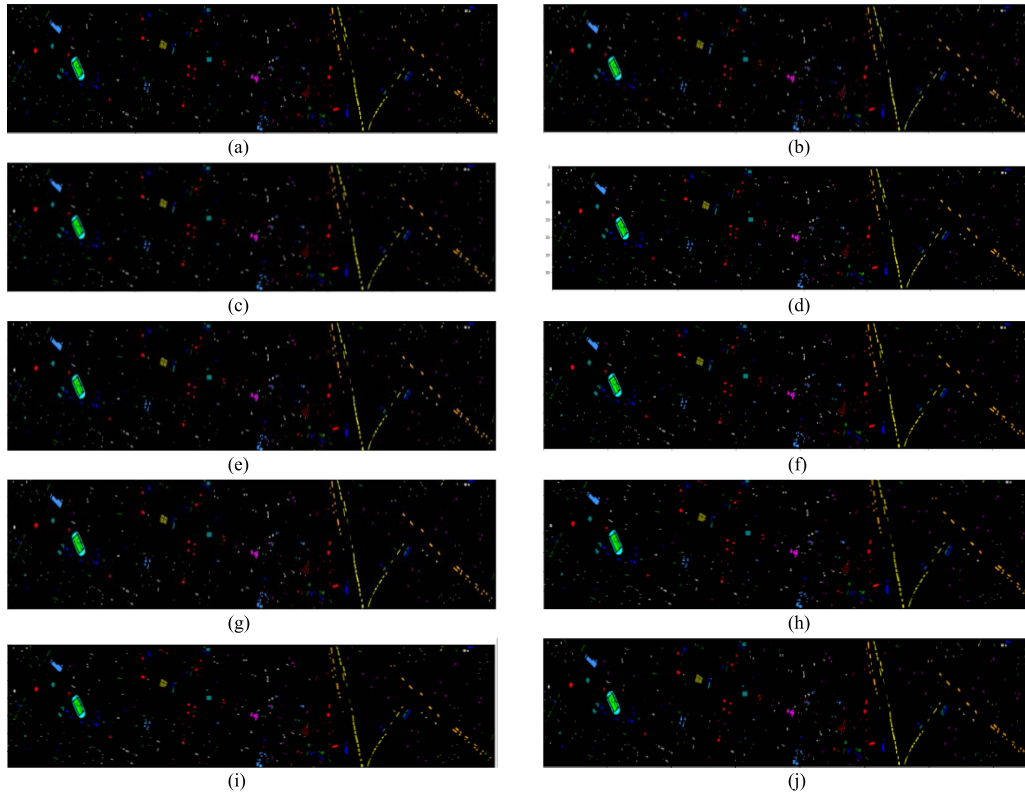


Fig. 21. Ground truth and visualization results of the HT dataset. (a) GT. (b) MCHN. (c) SSLDBR. (d) SSFLT. (e) ARL-GAN. (f) DSC_MFF. (g) Lite_HCNet. (h) WaveFormer. (i) GSC-ViT. (j) Proposed method.

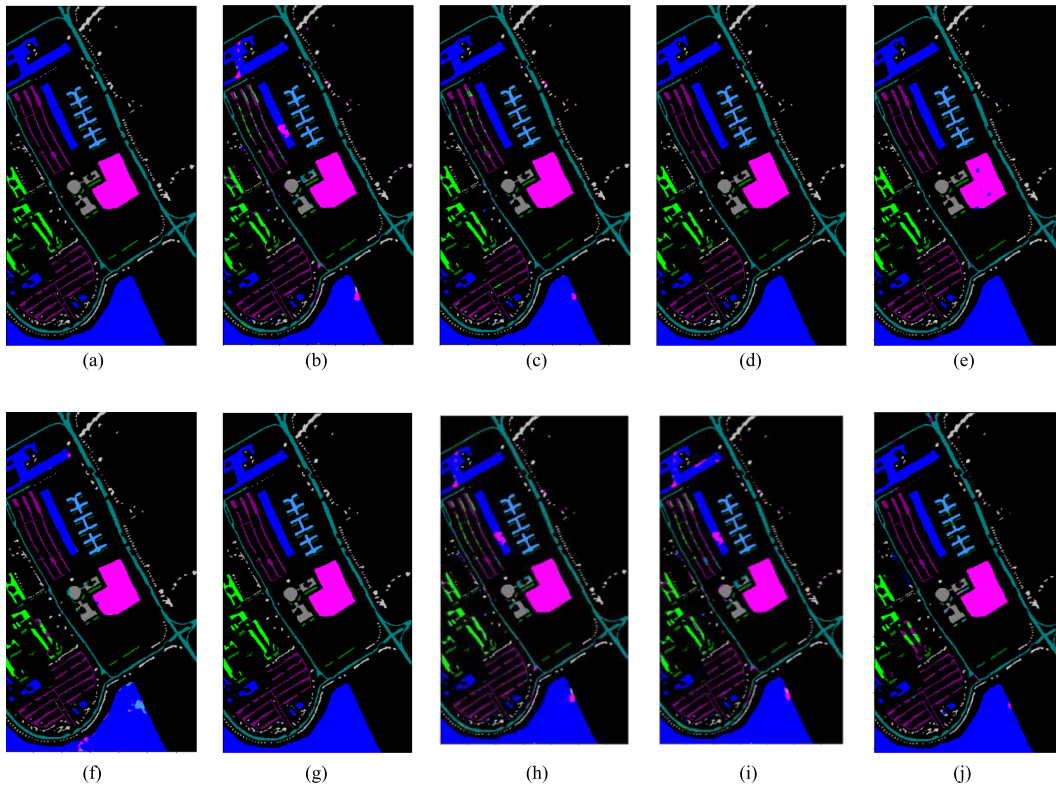


Fig. 22. Ground truth and visualization results of the PU dataset. (a) GT. (b) MCHN. (c) SSLDBR. (d) SSFLT. (e) ARL-GAN. (f) DSC_MFF. (g) Lite_HCNet. (h) WaveFormer. (i) GSC-ViT. (j) Proposed method.

TABLE VI
COMPARISON RESULTS FOR THE PU DATASET (PCA = 15, SPATIAL SIZE = 15)

Classes	MCHN	SSLDBR	SSFLT	ARL-GAN	DSC_MFF	Lite_HCNet	WaveFormer	GSC-ViT	Proposed Method
C0	91.98 ± 2.91	93.82 ± 1.04	92.53 ± 0.67	98.01 ± 0.41	96.81 ± 2.47	93.58 ± 0.83	98.08 ± 0.39	99.50 ± 0.27	97.64 ± 0.40
C1	97.15 ± 1.08	97.68 ± 1.22	100.00 ± 0.0	97.85 ± 1.06	97.53 ± 1.51	96.34 ± 1.06	96.25 ± 2.73	94.13 ± 4.09	99.45 ± 0.15
C2	93.57 ± 2.53	87.32 ± 8.34	91.86 ± 2.07	91.58 ± 3.19	96.48 ± 0.35	92.47 ± 2.34	88.92 ± 8.64	88.26 ± 6.22	95.97 ± 0.65
C3	93.29 ± 0.27	93.27 ± 0.14	92.19 ± 1.19	93.41 ± 0.85	92.98 ± 4.13	90.15 ± 3.56	97.95 ± 1.23	98.44 ± 0.53	93.50 ± 0.56
C4	99.31 ± 0.06	99.94 ± 0.09	96.11 ± 0.72	97.69 ± 0.19	96.23 ± 2.03	93.28 ± 0.91	89.64 ± 2.59	82.16 ± 4.89	96.11 ± 0.58
C5	95.48 ± 1.43	97.92 ± 2.01	94.83 ± 1.58	95.34 ± 1.26	97.52 ± 1.47	96.72 ± 0.99	96.62 ± 3.24	98.24 ± 1.62	99.68 ± 0.56
C6	91.91 ± 3.19	92.31 ± 2.19	91.14 ± 3.61	92.71 ± 1.38	92.18 ± 1.87	90.23 ± 2.03	99.28 ± 0.18	99.82 ± 0.25	91.14 ± 1.83
C7	94.43 ± 2.73	90.20 ± 4.82	96.47 ± 2.53	90.18 ± 5.14	95.23 ± 2.54	93.17 ± 1.19	92.01 ± 2.67	91.81 ± 4.09	96.47 ± 1.04
C8	90.46 ± 1.16	95.31 ± 0.73	90.18 ± 4.81	93.52 ± 2.22	91.33 ± 1.11	91.25 ± 2.43	90.16 ± 5.74	87.30 ± 7.17	90.18 ± 1.34
OA	95.37 ± 1.47	94.52 ± 0.39	93.95 ± 0.18	94.83 ± 0.72	96.32 ± 0.93	93.27 ± 0.34	94.72 ± 1.03	93.89 ± 0.17	97.81 ± 0.12
Kappa	94.08 ± 2.03	93.91 ± 0.77	93.57 ± 0.32	93.67 ± 1.03	95.61 ± 1.13	92.75 ± 0.94	94.29 ± 0.98	93.22 ± 0.23	97.10 ± 0.16
AA	94.18 ± 0.12	94.20 ± 0.54	93.92 ± 0.24	94.51 ± 0.98	95.15 ± 1.84	93.03 ± 0.48	94.32 ± 1.38	93.30 ± 1.05	95.58 ± 0.09

Bold values represent the best significant results.

TABLE VII
CLASSIFICATION ACCURACY ON IP DATASET (ADDING GAUSSIAN NOISE WITH ZERO MEAN AND DIFFERENT VARIANCE)

Noise	Without Noise	Variance = 2	Variance = 4	Variance = 6	Variance = 8
OA	97.66	97.4	97.18	96.96	96.75
KAPPA	97.33	97.04	96.78	96.53	96.29
AA	97.17	96.43	96.69	94.04	94

TABLE VIII
COMPUTATIONAL COMPLEXITY OF THE PROPOSED METHOD AND STATE-OF-THE-ART METHODS

Methods	Max-FLOPS	Parameters	Computational Complexity Formula	Assessment
MCHN	0.1G	500 K	$(Mn)^2 + K$	Low
SSLDBR	4G	1 M	$M(n^3 + n^2 + n) + K$	High
SSFTT	0.5G	4 M	$n(M^2 + Nn) + K$	High
ARL-GAN	1G	1 M	$(M + N)n^2 + K$	Medium
DSC-MFF	10G	5 M	$((M + N)n^2)^3 + K$	High
Lite-HCNet	70M	3 K	$Mn^2 + K$	Low
Proposed Method	1.2G	250 K	$n(Mn + N) + K$	Medium

We also use an efficient DNN with residual connection and spatial and spectral attention modules. In addition, the spectral attention module (RSA) in our approach is designed to make attention to informative bands. The spatial attention module (DSA) is based on employing 2-D depthwise convolution and can make attention to spatial regions with effective information. By combining both spectral and spatial attention, our approach can effectively capture this information in HSIs, leading to superior performance compared to methods that only use one type of attention.

D. Performance of the Proposed Method Under Noise Condition

In another experiment, we add Gaussian noise with different variances and zero mean to HSIs to assess the robustness of

our classification model. By varying the variance of the noise, we simulated different levels of image quality degradation, akin to real-world scenarios where data may be affected by various noise factors. Table VII lists the results of this experiment. Our findings indicate that as the variance of the Gaussian noise increases, there is an expected decrease in classification accuracy.

E. Computational Complexity

Computational complexity is an important aspect to evaluate the efficiency and scalability of deep learning methods for HSI data processing. This can help to understand the tradeoff between model performance and resource consumption and to design and optimize models that are suitable for different applications and scenarios. The complexity of deep learning








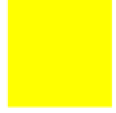







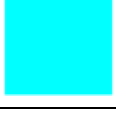
Class number	Class color		Class number	Class color	
Class 0		Black	Class 8		Purple
Class 1		Dark Sky	Class 9		Dark Green
Class 2		Blue	Class 10		Orange
Class 3		Green	Class 11		Yellow
Class 4		Silver	Class 12		Red
Class 5		Dark Blue	Class 13		Dark Red
Class 6		Pink	Class 14		Dark Yellow
Class 7		Gray	Class 15		Sky

Fig. 23. Class color for ground truth and visualization results of different datasets.

models is important for understanding model generalization, model optimization, model selection, and design. It is also essential to the computational requirements of deep learning models [82], [83], [84].

The computational complexity reflects the amount of time and space resources required by the algorithm to perform the feature extraction and integration tasks. In this section, we compare the computational complexity of our proposed method with other methods that use CNN layers, attention mechanisms, and specific blocks to classify HSI data. We use two metrics to measure the computational complexity of different methods: the number of parameters and floating-point operations per second (FLOPS). The number of parameters indicates the memory consumption of the model, while FLOPS indicates the computational cost of the model. Table VIII lists the comparison of the number of parameters and FLOPS of the competing methods. In this table, the highest value of FLOPS, the number of parameters, a formula for a better understanding of the computational complexity of each model, and finally the evaluation of the quality of the calculation complexity of each model are shown.

By considering the dimensions of the input image and the structure of the classifier model, according to the type of layer used and their arrangement, a formula can be obtained for the computational complexity of any method. We have done this by considering the dimensions of the input and the operation performed by each layer on the input data, as well as by considering the number of layers and the series and parallel structure of the blocks and branches of each model. The coefficients used in the presented formulas indicate the final operations of the layers, including the multiplication of the number of filters and kernels and addition with the output of the blocks or multiplication in parallel branches. M , N , and K coefficients will be different according to the structure of each model. The M coefficient is the result of multiplying the input dimensions with each other and the size and depth of each layer, and the N coefficient is the product of the layers and parallel branches in the models. The operation of each layer alone is also considered as N . Also, in the output of each model, the K coefficient has been considered in proportion to the number of flattened and fully connected layers.

TABLE IX
EFFECT OF REMOVING THE EBIM

Dataset	Method	OA	Kappa	AA
IP	EBIM+RSA+DSA (Proposed architecture)	97.66 ± 0.08	97.33 ± 0.10	97.17 ± 0.26
	RSA+DSA	97.15 ± 0.39	96.74 ± 0.45	95.91 ± 1.54
SA	EBIM+RSA+DSA (Proposed architecture)	98.39 ± 0.53	98.21 ± 0.60	98.38 ± 0.83
	RSA+DSA	97.15 ± 0.39	96.74 ± 0.45	95.91 ± 1.54
HT	EBIM+RSA+DSA (Proposed architecture)	97.77 ± 0.50	97.59 ± 0.54	97.81 ± 0.47
	RSA+DSA	97.59 ± 1.57	97.40 ± 1.70	97.10 ± 1.82
PU	EBIM+RSA+DSA (Proposed architecture)	97.81 ± 0.12	97.10 ± 0.16	95.58 ± 0.09
	RSA+DSA	97.68 ± 1.11	96.93 ± 1.46	95.20 ± 1.38

From Table VIII, it can be seen that our proposed method has the average number of parameters and FLOPS among all the compared methods. This is because our method uses a hybrid 3D–2D CNN structure to reduce the number of parameters as well as spatial and spectral attention modules with low parameters.

F. Ablation Studies

In this section, we conducted ablation studies on four datasets to thoroughly evaluate the effectiveness of the proposed modules. The purpose of these studies was to systematically remove the proposed modules, namely EBIM, RSA, and DSA, to examine their contributions and impact on the overall performance of the system. By carefully analyzing the results obtained from these ablation studies, we gain valuable insights into the significance and effectiveness of each module in improving the classification accuracy and robustness of the system.

1) *Effect of Removing the EBIM*: The EBIM adds entropy information to the HSIs. It enhances the local complexity and diversity of the image, which can help to detect suitable variations in the spectral and spatial features. Moreover, it improves the efficiency and accuracy of the subsequent processing steps.

Table IX illustrates the effect of removing the EBIM. The EBIM is important for the accuracy of the IP dataset, and removing it leads to a significant drop in AA, Kappa, and OA, which fall from 97.17%, 97.33%, and 97.66% to 95.91%, 96.74%, and 97.15%, respectively. Moreover, Removing the EBIM causes a significant decrease in the accuracy of the SA dataset, as the values of AA, Kappa, and OA fall from 98.38%, 98.21%, and 98.39% to 95.91%, 96.74%, and 97.15%, respectively. In fact, removing the EBIM has a strong effect on datasets with low spatial resolution. The spatial resolution is the size of the smallest feature that can be detected by the image sensor. The spatial resolution of IP is 20 m per pixel. This means that each pixel in the image represents an area of 20×20 m on the ground. The IP dataset has a low spatial resolution compared to some other hyperspectral datasets, such as PU, SA, and HT, which have spatial resolutions of 1.3, 3.7, and 2.5 m per pixel, respectively. After IP, the SA has the lowest spatial resolution than other datasets. Therefore, removing the EBIM reduces the accuracy of the IP and SA datasets considerably.

2) *Effect of Removing the RSA Block*: The RSA block is used to pay attention to spectral information. This module can enhance the performance of HSI classification by learning the spectral dependencies and discriminating the spectral signatures of different classes. The softmax function in the RSA block converts a vector of values into a probability distribution. It adds this information to the data and enhances the classification performance.

Table X lists the results of removing the RSA block and its effect on the classification performance. The effect of spectral attention on different hyperspectral datasets may vary depending on the characteristics of the datasets. The RSA block is essential for the accuracy of the IP and PU datasets, and removing it causes a large decrease. The IP dataset, which has a low spatial resolution and high spectral variability, may also benefit from spectral attention because it can help exploit the unlabeled data and select the most informative samples for labeling. The spectral attention can help learn the spectral correlations and mutual information of the pixels, and then classify them based on the principle of relevant information. The PU dataset can capture more details and features of the urban land cover than the other datasets but it also has a low contrast and a high noise level, which makes the texture features more blurred and noisier. Therefore, the accuracy of the IP and PU datasets drops significantly when the RSA block is removed. Without the RSA block, the IP dataset suffers a large loss in accuracy metrics, as AA, Kappa, and OA drop from 97.17%, 97.33%, and 97.66% to 94.25%, 95.99%, and 96.49%, respectively. Moreover, removing the RSA block causes a considerable decrease in the accuracy of the PU dataset, as the values of AA, Kappa, and OA fall from 95.58%, 97.1%, and 97.81% to 92.32%, 95.18%, and 96.36%, respectively.

3) *Effect of Removing the DSA Module*: The DSA block is used to pay attention to spatial information. It combines 2-D depthwise convolution and softmax function in its architecture. It can significantly improve the performance of HSI classification by learning the spatial dependencies and discriminating the spatial patterns of different classes.

The SA, PU, IP, and HT datasets are four different HSIs that have different spatial resolutions and challenges for classification. As given in Table XI, removing the DSA block has a strong effect on the accuracy of the IP and PU. The IP dataset has the lowest spatial resolution among the four datasets, which

TABLE X
EFFECT OF REMOVING THE RSA MODULE

Dataset	Method	OA	Kappa	AA
IP	EBIM+RSA+DSA (Proposed method)	97.66 ± 0.08	97.33 ± 0.10	97.17 ± 0.26
	EBIM+DSA	96.49 ± 1.03	95.99 ± 1.18	94.25 ± 1.98
SA	EBIM+RSA+DSA (Proposed method)	98.39 ± 0.53	98.21 ± 0.60	98.38 ± 0.83
	EBIM+DSA	98.23 ± 0.62	98.03 ± 0.69	98.32 ± 0.85
HT	EBIM+RSA+DSA (Proposed method)	97.77 ± 0.50	97.59 ± 0.54	97.81 ± 0.47
	EBIM+DSA	96.90 ± 2.64	96.65 ± 2.85	97.10 ± 2.45
PU	EBIM+RSA+DSA (Proposed method)	97.81 ± 0.12	97.10 ± 0.16	95.58 ± 0.09
	EBIM+DSA	96.36 ± 1.62	95.18 ± 2.13	92.32 ± 2.85

TABLE XI
EFFECT OF REMOVING THE SPATIAL ATTENTION MODULE (DSA)

Dataset	Method	OA	Kappa	AA
IP	EBIM+RSA+DSA (Proposed method)	97.66 ± 0.08	97.33 ± 0.10	97.17 ± 0.26
	EBIM+RSA	96.43 ± 1.03	95.92 ± 1.17	94.80 ± 2.63
SA	EBIM+RSA+DSA (Proposed method)	98.39 ± 0.53	98.21 ± 0.60	98.38 ± 0.83
	EBIM+RSA	97.74 ± 0.80	97.48 ± 0.90	98.23 ± 0.60
HT	EBIM+RSA+DSA (Proposed method)	97.77 ± 0.50	97.59 ± 0.54	97.81 ± 0.47
	EBIM+RSA	97.31 ± 0.28	97.09 ± 0.31	97.30 ± 0.40
PU	EBIM+RSA+DSA (Proposed method)	97.81 ± 0.12	97.10 ± 0.16	95.58 ± 0.09
	EBIM+RSA	96.42 ± 1.29	95.24 ± 1.72	93.25 ± 2.21

means that the IP dataset has the least details and features of the surface. The IP dataset also has a high spectral variability and a low number of labeled samples, which means that the classification models need to deal with the noise and uncertainty of the data. Therefore, removing the DSA module from the proposed architecture decreases the AA from 97.17% to 94.8%, OA from 97.66% to 96.43%, and Kappa coefficient from 97.33% to 95.92%. For the PU dataset, it has a high spatial resolution, which means that the spatial information is more important for classification. The PU dataset also has a low contrast and a high noise level, which makes the texture features more blurred and noisier. Therefore, removing the DSA block decreases the AA, Kappa, and OA from 95.58%, 97.1%, and 97.81% to 93.25%, 95.24%, and 96.42%, respectively.

V. DISCUSSION

In this article, we presented an effective approach for classifying HSI using deep learning techniques. Specifically, we proposed a hybrid CNN that incorporated different modules, such as EBIM, RSA, and DSA. Our proposed model utilizes an entropy filter to bring a significant performance improvement by adding entropy information to HSIs. In addition, we use the spatial attention module (DSA), which is based on 2-D depthwise convolutions and a softmax activation function. The spectral attention module (RSA) exploits reshaped layers and softmax activation functions in its architecture. These two modules improve the classification performance and add probability information

to feature maps. Through experimental results on benchmark datasets, our model has demonstrated superior performance compared to others. We evaluate the effect of removing different modules in the proposed model through Tables IX and X. There are four popular datasets to evaluate the performance of HSI classification systems. Each of these datasets has its characteristics and challenges for HSI classification. For example, the SA dataset has a high spectral resolution and a low spatial resolution, while the PU dataset has a high spatial resolution and a low spectral resolution. The HT dataset has a large image size and a high number of classes, which means that the classification models need to handle the scalability and complexity of the data. The IP dataset has a high spectral variability and a low number of labeled samples, which means that the classification models need to deal with the noise and uncertainty of the data. According to the characteristics of these datasets, different modules are exploited in the proposed architecture to increase the accuracy. The EBIM has a strong effect on the accuracy of the SA and IP, which have the lowest spatial resolution compared to others. This is done by enhancing the spatial information by adding entropy information to the HSIs through the EBIM. The IP and PU datasets need the RSA block for accurate results. Without the RSA block, the performance drops significantly. The IP dataset has high spectral variability. The RSA block learns the spectral correlations and mutual information of the pixels and classifies the IP dataset efficiently. The PU dataset has a low contrast and a high noise level, which makes the texture more blurred

and noisier. The RSA block efficiently pays attention to spectral bands. Therefore, removing the RSA block has a strong effect on the accuracy of these two datasets.

The DSA pays attention to spatial features. It has more effect on the accuracy of the IP and PU. The IP dataset has the lowest spatial resolution among the four datasets, which means that the IP dataset has the least details and features of the surface. Therefore, the DSA block can effectively notice informative regions. The PU dataset has a low contrast and a high noise level, which makes the texture more blurred and noisier. The DSA block has a strong effect on the accuracy of this dataset by effectively highlighting spatial regions.

VI. CONCLUSION

In this article, we proposed a novel method to classify HSIs using spatial and spectral attention mechanisms as well as the EBIM technique. Our method aims to enhance the performance and efficiency of HSI classification by exploiting the EBIM and attention modules. For spatial attention, we introduced the DSA module, which was a combination of 2-D depthwise convolution and the softmax function. It learned to generate spatial attention maps for each spectral band of the input HSI. It also highlighted the regions of interest and suppressed the background noise in the HSI. For spectral attention, we introduced the RSA module, which was a combination of reshape layers and the softmax function. It learned to generate spectral attention weights for each spectral band of the input HSI. The spectral attention weights were used to emphasize the important spectral bands and reduce the redundancy and correlation among the bands. In addition to the spatial and spectral attention modules, we also proposed the EBIM module, which was a preprocessing step that applied the entropy filter to the input HSI. The entropy filter is a function that calculates the local entropy of an image, which is a measure of the randomness or complexity of the image. The EBIM technique enhanced the quality and contrast of the HSI. We evaluated our proposed method on several benchmark HSI datasets and compared it with several state-of-the-art methods. We used an effective DNN backbone based on a hybrid structure of 2-D CNNs, 3-D CNNs, and residual connections. The experimental results showed that our method achieved superior performance in terms of OA, AA, and Kappa. We also conducted ablation studies to demonstrate the effectiveness and necessity of each component of our method. In summary, we proposed a novel method to classify HSIs using spatial and spectral attention mechanisms as well as the EBIM module. In the future, we will continue to explore new architectures and techniques to train a more reliable and compact model for HSI classification.

REFERENCES

- [1] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. M. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, Jun. 2013, doi: [10.1109/MGRS.2013.2244672](https://doi.org/10.1109/MGRS.2013.2244672).
- [2] N. Keshava and J. F. Mustard, "Spectral unmixing," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 44–57, Jan. 2002, doi: [10.1109/79.974727](https://doi.org/10.1109/79.974727).
- [3] J. M. Bioucas-Dias et al., "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 354–379, Apr. 2012, doi: [10.1109/JSTARS.2012.2194696](https://doi.org/10.1109/JSTARS.2012.2194696).
- [4] M. Mohammadi and A. Sharifi, "Evaluation of convolutional neural networks for urban mapping using satellite images," *J. Ind. Soc. Remote Sens.*, vol. 49, no. 9, pp. 2125–2131, 2021, doi: [10.1007/s12524-021-01382-x](https://doi.org/10.1007/s12524-021-01382-x).
- [5] S. Jalayer, A. Sharifi, D. Abbasi-Moghadam, A. Tariq, and S. Qin, "Assessment of spatiotemporal characteristic of droughts using in situ and remote sensing-based drought indices," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 1483–1502, 2023, doi: [10.1109/JSTARS.2023.3237380](https://doi.org/10.1109/JSTARS.2023.3237380).
- [6] A. Sharifi and S. Felegari, "Nitrogen dioxide (NO₂) pollution monitoring with sentinel-5P satellite imagery over during the coronavirus pandemic (case study: Tehran)," *Remote Sens. Lett.*, vol. 13, no. 10, pp. 1029–1039, Oct. 2022, doi: [10.1080/2150704X.2022.2120780](https://doi.org/10.1080/2150704X.2022.2120780).
- [7] J. Huang, K. Liu, and X. Li, "Locality constrained low rank representation and automatic dictionary learning for hyperspectral anomaly detection," *Remote Sens.*, vol. 14, no. 6, pp. 1–19, 2022, doi: [10.3390/rs14061327](https://doi.org/10.3390/rs14061327).
- [8] K. Bi, S. Xiao, S. Gao, C. Zhang, N. Huang, and Z. Niu, "Estimating vertical chlorophyll concentrations in maize in different health states using hyperspectral LiDAR," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 11, pp. 8125–8133, Nov. 2020, doi: [10.1109/TGRS.2020.2987436](https://doi.org/10.1109/TGRS.2020.2987436).
- [9] Y. Liu, M. Li, S. Wang, T. Wu, W. Jiang, and Z. Liu, "Identification of heat damage in imported soybeans based on hyperspectral imaging technology," *J. Sci. Food Agriculture*, vol. 100, no. 4, pp. 1775–1786, Mar. 2020, doi: [10.1002/JSFA.10214](https://doi.org/10.1002/JSFA.10214).
- [10] W. Wang, Y. Qian, and Y. Y. Tang, "Hypergraph-regularized sparse NMF for hyperspectral unmixing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 2, pp. 681–694, Feb. 2016, doi: [10.1109/JSTARS.2015.2508448](https://doi.org/10.1109/JSTARS.2015.2508448).
- [11] M. Shimoni, R. Haelterman, and C. Perneel, "Hyperspectral imaging for military and security applications: Combining myriad processing and sensing techniques," *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 2, pp. 101–117, Jun. 2019, doi: [10.1109/MGRS.2019.2902525](https://doi.org/10.1109/MGRS.2019.2902525).
- [12] M. Barberio et al., "Intraoperative guidance using hyperspectral imaging: A review for surgeons," *Diagnostics*, vol. 11, no. 11, pp. 1–20, 2021, doi: [10.3390/diagnostics11112066](https://doi.org/10.3390/diagnostics11112066).
- [13] J. Yang and J. Qian, "Hyperspectral image classification via multiscale joint collaborative representation with locally adaptive dictionary," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 1, pp. 112–116, Jan. 2018, doi: [10.1109/LGRS.2017.2776113](https://doi.org/10.1109/LGRS.2017.2776113).
- [14] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral-spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2018, doi: [10.1109/TGRS.2017.2755542](https://doi.org/10.1109/TGRS.2017.2755542).
- [15] Q. Gao, S. Lim, and X. Jia, "Hyperspectral image classification using joint sparse model and discontinuity preserving relaxation," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 1, pp. 78–82, Jan. 2018, doi: [10.1109/LGRS.2017.2774253](https://doi.org/10.1109/LGRS.2017.2774253).
- [16] P. Hu, X. Liu, Y. Cai, and Z. Cai, "Band selection of hyperspectral images using multiobjective optimization-based sparse self-representation," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 3, pp. 452–456, Mar. 2019, doi: [10.1109/LGRS.2018.2872540](https://doi.org/10.1109/LGRS.2018.2872540).
- [17] B. Tu, X. Zhang, X. Kang, G. Zhang, J. Wang, and J. Wu, "Hyperspectral image classification via fusing correlation coefficient and joint sparse representation," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 3, pp. 340–344, Mar. 2018, doi: [10.1109/LGRS.2017.2787338](https://doi.org/10.1109/LGRS.2017.2787338).
- [18] P. Gao, J. Wang, H. Zhang, and Z. Li, "Boltzmann entropy-based unsupervised band selection for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 3, pp. 462–466, Mar. 2019, doi: [10.1109/LGRS.2018.2872358](https://doi.org/10.1109/LGRS.2018.2872358).
- [19] T. Dundar and T. Ince, "Sparse representation-based hyperspectral image classification using multiscale superpixels and guided filter," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 246–250, Feb. 2019, doi: [10.1109/LGRS.2018.2871273](https://doi.org/10.1109/LGRS.2018.2871273).
- [20] D. Landgrebe, "Hyperspectral image data analysis," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 17–28, Jan. 2002, doi: [10.1109/79.974718](https://doi.org/10.1109/79.974718).
- [21] M. Hamouada, K. S. Ettabaa, and M. S. Bouhlel, "Smart feature extraction and classification of hyperspectral images based on convolutional neural networks," *IET Image Process.*, vol. 14, no. 10, pp. 1999–2005, Aug. 2020, doi: [10.1049/IET-IPR.2019.1282](https://doi.org/10.1049/IET-IPR.2019.1282).
- [22] J. An, X. Zhang, and L. C. Jiao, "Dimensionality reduction based on group-based tensor model for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 10, pp. 1497–1501, Oct. 2016, doi: [10.1109/LGRS.2016.2593789](https://doi.org/10.1109/LGRS.2016.2593789).
- [23] X. Yu, R. Ding, J. Shao, and X. Li, "Hyperspectral remote sensing image feature representation method based on CAE-H with nuclear norm constraint," *Electronics*, vol. 10, no. 21, Oct. 2021, Art. no. 2667, doi: [10.3390/ELECTRONICS10212667](https://doi.org/10.3390/ELECTRONICS10212667).

- [24] M. Esmacili, D. Abbasi-Moghadam, A. Sharifi, A. Tariq, and Q. Li, "Hyperspectral image band selection based on CNN embedded GA (CNNeGA)," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 1927–1950, 2023, doi: [10.1109/JSTARS.2023.3242310](https://doi.org/10.1109/JSTARS.2023.3242310).
- [25] W. Li, C. Chen, M. Zhang, H. Li, and Q. Du, "Data augmentation for hyperspectral image classification with deep CNN," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 4, pp. 593–597, Apr. 2019, doi: [10.1109/LGRS.2018.2878773](https://doi.org/10.1109/LGRS.2018.2878773).
- [26] F. Ullah et al., "Deep hyperspectral shots: Deep snap smooth wavelet convolutional neural network shots ensemble for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 14–34, 2024, doi: [10.1109/JSTARS.2023.3314900](https://doi.org/10.1109/JSTARS.2023.3314900).
- [27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017, doi: [10.1145/3065386](https://doi.org/10.1145/3065386).
- [28] J. Cai and J. Hu, "3D RANs: 3D residual attention networks for action recognition," *Vis. Comput.*, vol. 36, no. 6, pp. 1261–1270, 2020, doi: [10.1007/s00371-019-01733-3](https://doi.org/10.1007/s00371-019-01733-3).
- [29] J. Mahmoodi, H. Nezamabadi-pour, and D. Abbasi-Moghadam, "Violence detection in videos using interest frame extraction and 3D convolutional neural network," *Multimedia Tools Appl.*, vol. 81, no. 15, pp. 20945–20961, 2022, doi: [10.1007/s11042-022-12532-9](https://doi.org/10.1007/s11042-022-12532-9).
- [30] J. Mahmoodi and H. Nezamabadi-pour, "A spatio-temporal model for violence detection based on spatial and temporal attention modules and 2D CNNs," *Pattern Anal. Appl.*, vol. 27, no. 2, 2024, Art. no. 46, doi: [10.1007/s10044-024-01265-0](https://doi.org/10.1007/s10044-024-01265-0).
- [31] S. Mei, J. Ji, Q. Bi, J. Hou, Q. Du, and W. Li, "Integrating spectral and spatial information into deep convolutional Neural networks for hyperspectral classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2016, pp. 5067–5070, doi: [10.1109/IGARSS.2016.7730321](https://doi.org/10.1109/IGARSS.2016.7730321).
- [32] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS J. Photogrammetry Remote Sens.*, vol. 145, pp. 120–147, 2018, doi: [10.1016/j.isprsjprs.2017.11.021](https://doi.org/10.1016/j.isprsjprs.2017.11.021).
- [33] K. Makantasis, K. Karantzos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2015, pp. 4959–4962, doi: [10.1109/IGARSS.2015.7326945](https://doi.org/10.1109/IGARSS.2015.7326945).
- [34] A. Ben Hamida, A. Benoit, P. Lambert, and C. B. Amar, "3-D deep learning approach for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4420–4434, Aug. 2018, doi: [10.1109/TGRS.2018.2818945](https://doi.org/10.1109/TGRS.2018.2818945).
- [35] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5966–5978, Jul. 2021, doi: [10.1109/TGRS.2020.3015157](https://doi.org/10.1109/TGRS.2020.3015157).
- [36] X. Ma, A. Fu, J. Wang, H. Wang, and B. Yin, "Hyperspectral image classification based on deep deconvolution network with skip architecture," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4781–4791, Aug. 2018, doi: [10.1109/TGRS.2018.2837142](https://doi.org/10.1109/TGRS.2018.2837142).
- [37] J. Yang, Y. Q. Zhao, J. C. W. Chan, and L. Xiao, "A multi-scale wavelet 3D-CNN for hyperspectral image super-resolution," *Remote Sens.*, vol. 11, no. 13, Jun. 2019, Art. no. 1557, doi: [10.3390/RS11131557](https://doi.org/10.3390/RS11131557).
- [38] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, "HybridSN: Exploring 3-D-2-D CNN feature hierarchy for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 277–281, Feb. 2020, doi: [10.1109/LGRS.2019.2918719](https://doi.org/10.1109/LGRS.2019.2918719).
- [39] N. Farmonov, D. Abbasi-moghadam, A. Sharifi, and K. Amankulova, "HypsiDNet: 3D-2D CNN model and spatial-spectral morphological attention for crop classification with DESIS and LiDAR data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 11969–11996, 2024, doi: [10.1109/JSTARS.2024.3418854](https://doi.org/10.1109/JSTARS.2024.3418854).
- [40] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral-spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2018, doi: [10.1109/TGRS.2017.2755542](https://doi.org/10.1109/TGRS.2017.2755542).
- [41] T. Li, X. Zhang, S. Zhang, and L. Wang, "Self-supervised learning with a dual-branch ResNet for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 5512905, doi: [10.1109/LGRS.2021.3107321](https://doi.org/10.1109/LGRS.2021.3107321).
- [42] Z. Ge, G. Cao, X. Li, and P. Fu, "Hyperspectral image classification method based on 2D-3D CNN and multibranch feature fusion," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5776–5788, 2020, doi: [10.1109/JSTARS.2020.3024841](https://doi.org/10.1109/JSTARS.2020.3024841).
- [43] W. Li, J. Yin, B. Han, and H. Zhu, "Generative adversarial network with folded spectrum for hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 883–886, doi: [10.1109/IGARSS.2019.8899034](https://doi.org/10.1109/IGARSS.2019.8899034).
- [44] M. Zhang, M. Gong, Y. Mao, J. Li, and Y. Wu, "Unsupervised feature extraction in hyperspectral images based on Wasserstein generative adversarial network," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 5, pp. 2669–2688, May 2019, doi: [10.1109/TGRS.2018.2876123](https://doi.org/10.1109/TGRS.2018.2876123).
- [45] S. Zhang, X. Zhang, T. Li, H. Meng, X. Cao, and L. Wang, "Adversarial representation learning for hyperspectral image classification with small-sized labeled set," *Remote Sens.*, vol. 14, no. 11, May 2022, Art. no. 2612, doi: [10.3390/RS14112612](https://doi.org/10.3390/RS14112612).
- [46] F. Zhou, R. Hang, Q. Liu, and X. Yuan, "Hyperspectral image classification using spectral-spatial LSTMs," *Neurocomputing*, vol. 328, pp. 39–47, 2019, doi: [10.1016/j.neucom.2018.02.105](https://doi.org/10.1016/j.neucom.2018.02.105).
- [47] R. Li, S. Zheng, C. Duan, Y. Yang, and X. Wang, "Classification of hyperspectral image based on double-branch dual-attention mechanism network," *Remote Sens.*, vol. 12, 2020, Art. no. 582.
- [48] M. Ahmed AL-Kubaisi, H. Z. M. Shafri, M. H. Ismail, M. J. M. Yusof, and S. J. Bin Hashim, "Hyperspectral image classification by integrating attention-based LSTM and hybrid spectral networks," *Int. J. Remote Sens.*, vol. 43, no. 9, pp. 3450–3469, 2022, doi: [10.1080/01431161.2022.2093621](https://doi.org/10.1080/01431161.2022.2093621).
- [49] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. 9th Int. Conf. Learn. Represent.*, Oct. 2020. Accessed: Nov. 17, 2023. [Online]. Available: <https://arxiv.org/abs/2010.11929v2>
- [50] A. A. Aleissae et al., "Transformers in remote sensing: A survey," *Remote Sens.*, vol. 15, no. 7, Mar. 2023, Art. no. 1860, doi: [10.3390/RS15071860](https://doi.org/10.3390/RS15071860).
- [51] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM Comput. Surv.*, vol. 54, no. 10, Jan. 2021, Art. no. 200, doi: [10.1145/3505244](https://doi.org/10.1145/3505244).
- [52] L. Sun, G. Zhao, Y. Zheng, and Z. Wu, "Spectral-spatial feature tokenization transformer for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5522214, doi: [10.1109/TGRS.2022.3144158](https://doi.org/10.1109/TGRS.2022.3144158).
- [53] M. Ahmad, U. Ghous, M. Usama, and M. Mazzara, "WaveFormer: Spectral-spatial wavelet transformer for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 21, 2024, Art. no. 5502405, doi: [10.1109/LGRS.2024.3353909](https://doi.org/10.1109/LGRS.2024.3353909).
- [54] Z. Zhao, X. Xu, S. Li, and A. Plaza, "Hyperspectral image classification using groupwise separable convolutional vision transformer network," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5511817, doi: [10.1109/TGRS.2024.3377610](https://doi.org/10.1109/TGRS.2024.3377610).
- [55] X. He, Y. Chen, and Z. Lin, "Spatial-spectral transformer for hyperspectral image classification," *Remote Sens.*, vol. 13, no. 3, pp. 1–22, 2021, doi: [10.3390/rs13030498](https://doi.org/10.3390/rs13030498).
- [56] Y. Ma et al., "A spatial-spectral transformer for hyperspectral image classification based on global dependencies of multi-scale features," *Remote Sens.*, vol. 16, no. 2, 2024, Art. no. 404, doi: [10.3390/rs16020404](https://doi.org/10.3390/rs16020404).
- [57] S. K. Roy, A. Deria, D. Hong, M. Ahmad, A. Plaza, and J. Chanussot, "Hyperspectral and LiDAR data classification using joint CNNs and morphological feature learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5530416, doi: [10.1109/TGRS.2022.3177633](https://doi.org/10.1109/TGRS.2022.3177633).
- [58] S. K. Roy, R. Mondal, M. E. Paoletti, J. M. Haut, and A. Plaza, "Morphological convolutional neural networks for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 8689–8702, 2021, doi: [10.1109/JSTARS.2021.3088228](https://doi.org/10.1109/JSTARS.2021.3088228).
- [59] M. Esmacili, D. Abbasi-Moghadam, A. Sharifi, A. Tariq, and Q. Li, "ResMorCNN model: Hyperspectral images classification using residual-injection morphological features & 3D-CNN layers," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 219–243, 2024, doi: [10.1109/JSTARS.2023.3328389](https://doi.org/10.1109/JSTARS.2023.3328389).
- [60] W. Wang, S. Dou, Z. Jiang, and L. Sun, "A fast dense spectral-spatial convolution network framework for hyperspectral images classification," *Remote Sens.*, vol. 10, no. 7, Jul. 2018, Art. no. 1068, doi: [10.3390/rs10071068](https://doi.org/10.3390/rs10071068).
- [61] H. Gao, Y. Yang, C. Li, X. Zhang, J. Zhao, and D. Yao, "Convolutional neural network for spectral-spatial classification of hyperspectral images," *Neural Comput. Appl.*, vol. 31, no. 12, pp. 8997–9012, 2019, doi: [10.1007/s00521-019-04371-x](https://doi.org/10.1007/s00521-019-04371-x).

- [62] D. Han, J. Kim, and J. Kim, "Deep pyramidal residual networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6307–6315, doi: [10.1109/CVPR.2017.668](https://doi.org/10.1109/CVPR.2017.668).
- [63] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. J. Plaza, and F. Pla, "Deep pyramidal residual networks for spectral–spatial hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 740–754, Feb. 2019, doi: [10.1109/TGRS.2018.2860125](https://doi.org/10.1109/TGRS.2018.2860125).
- [64] L. Dang, P. Pang, and J. Lee, "Depth-wise separable convolution neural network with residual connection for hyperspectral image classification," *Remote Sens.*, vol. 12, no. 20, Oct. 2020, Art. no. 3408, doi: [10.3390/RS12203408](https://doi.org/10.3390/RS12203408).
- [65] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1800–1807, doi: [10.1109/CVPR.2017.195](https://doi.org/10.1109/CVPR.2017.195).
- [66] M. E. Asker, "Hyperspectral image classification method based on squeeze-and-excitation networks, depthwise separable convolution and multibranch feature fusion," *Earth Sci. Inform.*, vol. 16, no. 2, pp. 1427–1448, 2023, doi: [10.1007/s12145-023-00982-0](https://doi.org/10.1007/s12145-023-00982-0).
- [67] X. Ma et al., "A lightweight hybrid convolutional neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5513714, doi: [10.1109/TGRS.2023.3282247](https://doi.org/10.1109/TGRS.2023.3282247).
- [68] S. Ghaderizadeh, D. Abbasi-Moghadam, A. Sharifi, A. Tariq, and S. Qin, "Multiscale dual-branch residual spectral-spatial network with attention for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 5455–5467, 2022, doi: [10.1109/JSTARS.2022.3188732](https://doi.org/10.1109/JSTARS.2022.3188732).
- [69] Y. Zhang, Y. Peng, B. Tu, and Y. Liu, "Local information interaction transformer for hyperspectral and LiDAR data classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 1130–1143, 2023, doi: [10.1109/JSTARS.2022.3232995](https://doi.org/10.1109/JSTARS.2022.3232995).
- [70] H. Firat, M. E. Asker, and D. Hanbay, "Classification of hyperspectral remote sensing images using different dimension reduction methods with 3D/2D CNN," *Remote Sens. Appl. Soc. Environ.*, vol. 25, 2022, Art. no. 100694, doi: [10.1016/j.rsase.2022.100694](https://doi.org/10.1016/j.rsase.2022.100694).
- [71] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Dec. 2016, pp. 770–778, doi: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- [72] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognit.*, Aug. 2017, pp. 2261–2269, doi: [10.1109/CVPR.2017.243](https://doi.org/10.1109/CVPR.2017.243).
- [73] A. Zare and K. C. Ho, "Endmember variability in hyperspectral analysis: Addressing spectral variability during spectral unmixing," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 95–104, Jan. 2014, doi: [10.1109/MSP.2013.2279177](https://doi.org/10.1109/MSP.2013.2279177).
- [74] R. A. Borsoi et al., "Spectral variability in hyperspectral data unmixing: A comprehensive review," *IEEE Geosci. Remote Sens. Mag.*, vol. 9, no. 4, pp. 223–270, Dec. 2021, doi: [10.1109/MGRS.2021.3071158](https://doi.org/10.1109/MGRS.2021.3071158).
- [75] B. Somers, G. P. Asner, L. Tits, and P. Coppin, "Endmember variability in spectral mixture analysis: A review," *Remote Sens. Environ.*, vol. 115, no. 7, pp. 1603–1616, Jul. 2011, doi: [10.1016/J.RSE.2011.03.003](https://doi.org/10.1016/J.RSE.2011.03.003).
- [76] N. Liu, W. Li, and Q. Du, "Unsupervised feature extraction for hyperspectral imagery using collaboration-competition graph," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1491–1503, Dec. 2018, doi: [10.1109/JSTSP.2018.2877474](https://doi.org/10.1109/JSTSP.2018.2877474).
- [77] G. Tejasree and L. Agilandeewari, "An extensive review of hyperspectral image classification and prediction: Techniques and challenges," *Multi-media Tools Appl.*, 2024, doi: [10.1007/s11042-024-18562-9](https://doi.org/10.1007/s11042-024-18562-9).
- [78] X. Ding, X. Zhang, J. Han, and G. Ding, "Scaling up your kernels to 31×31 : A large kernel design in CNNs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11953–11965, doi: [10.1109/CVPR52688.2022.01166](https://doi.org/10.1109/CVPR52688.2022.01166).
- [79] A. Al-Sabaawi, H. M. Ibrahim, Z. M. Arkah, M. Al-Amidie, and L. Alzubaidi, "Amended convolutional neural network with global average pooling for image classification," in *Proc. Intell. Syst. Des. Appl.*, 2021, pp. 171–180, doi: [10.1007/978-3-030-71187-0_16](https://doi.org/10.1007/978-3-030-71187-0_16).
- [80] Y. Guo, Y. Li, L. Wang, and T. Rosing, "Depthwise convolution is all you need for learning multiple visual domains," in *Proc. 33rd AAAI Conf. Artif. Intell., 31st Innov. Appl. Artif. Intell. Conf., 9th AAAI Symp. Educ. Adv. Artif. Intell.*, 2019, pp. 8368–8375, doi: [10.1609/aaai.v33i01.33018368](https://doi.org/10.1609/aaai.v33i01.33018368).
- [81] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. Conf. Track Proc.*, Dec. 2015, pp. 1–15. Accessed: Dec. 2, 2023. [Online]. Available: <https://arxiv.org/abs/1412.6980v9>
- [82] X. Hu, L. Chu, J. Pei, W. Liu, and J. Bian, "Model complexity of deep learning: A survey," *Knowl. Inf. Syst.*, vol. 63, no. 10, pp. 2585–2619, 2021, doi: [10.1007/s10115-021-01605-0](https://doi.org/10.1007/s10115-021-01605-0).
- [83] F. M. Shiri, T. Perumal, N. Mustapha, and R. Mohamed, "A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU," 2023. [Online]. Available: <http://arxiv.org/abs/2305.17473>
- [84] N. Thompson, K. Greenewald, K. Lee, and G. F. Manso, "The computational limits of deep learning," in *Proc. 9th Comput. Within Limits*, 2023, pp. 1–33, doi: [10.21428/bf6fb269.1f033948](https://doi.org/10.21428/bf6fb269.1f033948).



Javad Mahmoodi was born in Kerman, Iran, in 1981. He received the M.Sc. degree in electrical engineering from Semnan University, Kerman, Iran, in 2006, and the Ph.D. degree in communication engineering from Shahid Bahonar University, Kerman, Iran, in 2024.

He is currently an Assistant Professor with the Department of Electrical Engineering, Kerman Branch of Islamic Azad University, Kerman, Iran. His research interests include the areas of image processing, object detection, remote sensing, and video processing.



Dariush Abbasi-Moghadam received the B.S. degree from Shahid Bahonar University, Kerman, Iran, in 1998, and the M.S. and Ph.D. degrees from the Iran University of Science and Technology, Tehran, Iran, in 2001 and 2011, respectively, all in electrical engineering.

He was primarily with the Advanced Electronic Research Center—Iran from 2001 to 2003 and worked on the design and analysis of satellite communication systems. He joined Iranian Telecommunications Company, Tehran, Iran, as a Research Engineer, in 2004. He is currently an Associate Professor with the Department of Electrical Engineering, Shahid Bahonar University of Kerman, Kerman, Iran. His research interests include the areas of wireless communications, satellite communication systems, remote sensing, and signal processing.



Alireza Sharifi was born in Tehran, Iran, in 1981. He received the M.Sc. and Ph.D. degrees in remote sensing engineering from the University of Tehran, Tehran, Iran, in 2008 and 2015, respectively.

He is currently an Associate Professor of remote sensing with the Faculty of Civil, Water and Environmental Engineering, Shahid Beheshti University, Tehran, Iran. His current research activities include remote sensing, time series analysis, and satellite image processing. In particular, he is involved in GeoAI program for food security and environmental monitoring.



Hossein Nezamabadi-Pour received the B.S. degree in electrical engineering from the Shahid Bahonar University of Kerman, Kerman, Iran, in 1998, and the M.Sc. and Ph.D. degrees in electrical engineering from Tarbait Moderres University, Tehran, Iran, in 2000 and 2004, respectively.

In 2004, he joined the Department of Electrical Engineering, Shahid Bahonar University of Kerman, as an Assistant Professor, and was promoted to Associate Professor in 2008. He is the author and coauthor of more than 170 peer-reviewed journal and conference papers. His research interests include image processing, pattern recognition, soft computing, and evolutionary computation.



Mohammad Esmaeili was born in Kerman, Iran. He received the M.Sc. degree in electrical and communication engineering from the Department of Technical and Engineering, Shahid Bahonar University of Iran, Kerman, Iran, in 2012. He is currently working toward the Ph.D. degree in electrical and communication engineering with Kerman Shahid Bahonar University, Kerman, Iran.

He is currently researching deep learning methods for approaches to processing hyperspectral and multispectral images. His research interests include image processing, deep learning, hyperspectral remote sensing, and its applications on hyper and multispectral images.



Alireza Vafaiejad was born in Mashhad, Iran, in 1978. He received the M.Sc. and Ph.D. degrees in GIS engineering from the Khaje Nasir Toosi University of Technology, Tehran, Iran, in 2002 and 2008, respectively.

He is currently an Associate Professor of GIS with the Faculty of Civil, Water and Environmental Engineering, Shahid Beheshti University, Tehran, Iran. His current research activities include GeoAI, artificial intelligence, and applications of GIS in water and environmental issues. In particular, he is involved in the GeoAI program for water and environmental monitoring.